



Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry

Nan Yang^{1,2}(✉) , Rui Wang^{1,2} , Jörg Stückler¹ , and Daniel Cremers^{1,2}

¹ Technical University of Munich, Munich, Germany

{yangn,wangr,stueckle,cremers}@in.tum.de

² Artisense, Munich, Germany

Abstract. Monocular visual odometry approaches that purely rely on geometric cues are prone to scale drift and require sufficient motion parallax in successive frames for motion estimation and 3D reconstruction. In this paper, we propose to leverage deep monocular depth prediction to overcome limitations of geometry-based monocular visual odometry. To this end, we incorporate deep depth predictions into Direct Sparse Odometry (DSO) as direct virtual stereo measurements. For depth prediction, we design a novel deep network that refines predicted depth from a single image in a two-stage process. We train our network in a semi-supervised way on photoconsistency in stereo images and on consistency with accurate sparse depth reconstructions from Stereo DSO. Our deep predictions excel state-of-the-art approaches for monocular depth on the KITTI benchmark. Moreover, our Deep Virtual Stereo Odometry clearly exceeds previous monocular and deep-learning based methods in accuracy. It even achieves comparable performance to the state-of-the-art stereo methods, while only relying on a single camera.

Keywords: Monocular depth estimation
Monocular visual odometry · Semi-supervised learning

1 Introduction

Visual odometry (VO) is a highly active field of research in computer vision with a plethora of applications in domains such as autonomous driving, robotics, and augmented reality. VO with a single camera using traditional geometric approaches inherently suffers from the fact that camera trajectory and map can only be estimated up to an unknown scale which also leads to scale drift. Moreover, sufficient motion parallax is required to estimate motion and structure

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01237-3_50) contains supplementary material, which is available to authorized users.

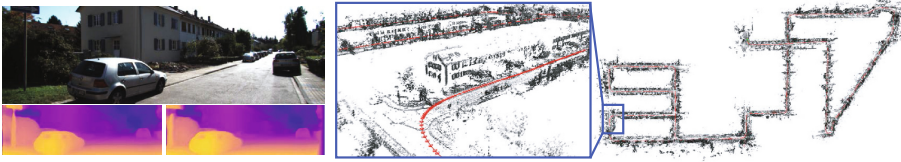


Fig. 1. DVSOD achieves monocular visual odometry on KITTI on par with state-of-the-art stereo methods. It uses deep-learning based left-right disparity predictions (lower left) for initialization and virtual stereo constraints in an optimization-based direct visual odometry pipeline. This allows for recovering accurate metric estimates.

from successive frames. To avoid these issues, typically more complex sensors such as active depth cameras or stereo rigs are employed. However, these sensors require larger efforts in calibration and increase the costs of the vision system.

Metric depth can also be recovered from a single image if a-priori knowledge about the typical sizes or appearances of objects is used. Deep learning based approaches tackle this by training deep neural networks on large amounts of data. In this paper, we propose a novel approach to monocular visual odometry, Deep Virtual Stereo Odometry (DVSOD), which incorporates deep depth predictions into a geometric monocular odometry pipeline. We use deep stereo disparity for virtual direct image alignment constraints within a framework for windowed direct bundle adjustment (e.g. Direct Sparse Odometry [8]). DVSOD achieves comparable performance to the state-of-the-art stereo visual odometry systems on the KITTI odometry benchmark. It can even outperform the state-of-the-art geometric VO methods when tuning scale-dependent parameters such as the virtual stereo baseline.

As an additional contribution, we propose a novel stacked residual network architecture that refines disparity estimates in two stages and is trained in a semi-supervised way. In typical supervised learning approaches [6, 24, 25], depth ground truth needs to be acquired for training with active sensors like RGB-D cameras and 3D laser scanners which are costly to obtain. Requiring a large amount of such labeled data is an additional burden that limits generalization to new environments. Self-supervised [11, 14] and unsupervised learning approaches [49], on the other hand, overcome this limitation and do not require additional active sensors. Commonly, they train the networks on photometric consistency, for example in stereo imagery [11, 14], which reduces the effort for collecting training data. Still, the current self-supervised approaches are not as accurate as supervised methods [23]. We combine self-supervised and supervised training, but avoid the costly collection of LiDAR data in our approach. Instead, we make use of Stereo Direct Sparse Odometry (Stereo DSO [40]) to provide accurate sparse 3D reconstructions on the training set. Our deep depth prediction network outperforms the current state-of-the-art methods on KITTI.

A video demonstrating our methods as well as the results is available at <https://youtu.be/sLZOeC9z.tw>.

1.1 Related Work

Deep Learning for Monocular Depth Estimation. Deep learning based approaches have recently achieved great advances in monocular depth estimation. Employing deep neural network avoids the hand-crafted features used in previous methods [19, 36]. Supervised deep learning [6, 24, 25] has recently shown great success for monocular depth estimation. Eigen et al. [5, 6] propose a two scale CNN architecture which directly predicts the depth map from a single image. Laina et al. [24] propose a residual network [17] based fully convolutional encoder-decoder architecture [27] with a robust regression loss function. The aforementioned supervised learning approaches need large amounts of ground-truth depth data for training. Self-supervised approaches [11, 14, 44] overcome this limitation by exploiting photoconsistency and geometric constraints to define loss functions, for example, in a stereo camera setup. This way, only stereo images are needed for training which are typically easier to obtain than accurate depth measurements from active sensors such as 3D lasers or RGB-D cameras. Godard et al. [14] achieve the state-of-the-art depth estimation accuracy for a fully self-supervised approach. The semi-supervised scheme proposed by Kuznetsov et al. [23] combines the self-supervised loss with supervision with sparse LiDAR ground truth. They do not need multi-scale depth supervision or left-right consistency in their loss, and achieve better performance than the self-supervised approach in [14]. The limitation of this semi-supervised approach is the requirement for LiDAR data which are costly to collect. In our approach we use Stereo Direct Sparse Odometry to obtain sparse depth ground-truth for semi-supervised training. Since the extracted depth maps are even sparser than LiDAR data, we also employ multi-scale self-supervised training and left-right consistency as in Godard et al. [14]. Inspired by [20, 34], we design a stacked network architecture leveraging the concept of residual learning [17].

Deep Learning for VO/SLAM. In recent years, large progress has been achieved in the development of monocular VO and SLAM methods [8, 9, 31, 32]. Due to projective geometry, metric scale cannot be observed with a single camera [37] which introduces scale drift. A popular approach is hence to use stereo cameras for VO [8, 10, 31] which avoid scale ambiguity and leverage stereo matching with a fixed baseline for estimating 3D structure. While stereo VO delivers more reliable depth estimation, it requires self-calibration for long-term operation [4, 46]. The integration of a second camera also introduces additional costs. Some recent monocular VO approaches have integrated monocular depth estimation [39, 46] to recover the metric scale by scale-matching. CNN-SLAM [39] extends LSD-SLAM [9] by predicting depth with a CNN and refining the depth maps using Bayesian filtering [7, 9]. Their method shows superior performance over monocular SLAM [9, 30, 35, 45] on indoor datasets [15, 38]. Yin et al. [46] propose to use convolutional neural fields and consecutive frames to improve the monocular depth estimation from a CNN. Camera motion is estimated using the refined depth. CodeSLAM [2] focuses on the challenge of dense 3D reconstruction. It jointly optimizes a learned compact representation of the dense

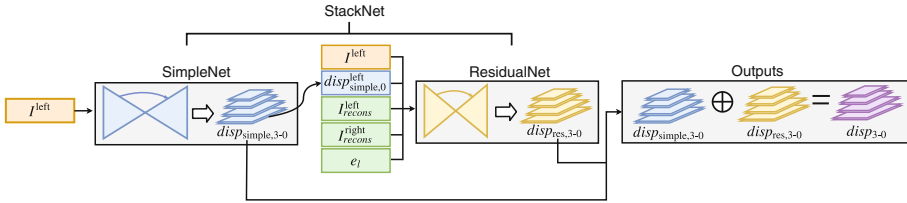


Fig. 2. Overview of StackNet architecture.

geometry with camera poses. Our work tackles the problem of odometry with monocular cameras and integrates deep depth prediction with multi-view stereo to improve camera pose estimation. Another line of research trains networks to directly predict the ego-motion end-to-end using supervised [41] or unsupervised learning [26, 49]. However, the estimated ego-motion of these methods is still by far inferior to geometric visual odometry approaches. In our approach, we phrase visual odometry as a geometric optimization problem but incorporate photoconsistency constraints with state-of-the-art deep monocular depth predictions into the optimization. This way, we obtain a highly accurate monocular visual odometry that is not prone to scale drift and achieves comparable results to traditional stereo VO methods.

2 Semi-Supervised Deep Monocular Depth Estimation

In this section, we will introduce our semi-supervised approach to deep monocular depth estimation. It builds on three key ingredients: self-supervised learning from photoconsistency in a stereo setup similar to [14], supervised learning based on accurate sparse depth reconstruction by Stereo DSO, and two-stage refinement of the network predictions in a stacked encoder-decoder architecture.

2.1 Network Architecture

We coin our architecture StackNet since it stacks two sub-networks, SimpleNet and ResidualNet, as depicted in Fig. 2. Both sub-networks are fully convolutional deep neural network adopted from DispNet [28] with an encoder-decoder scheme. ResidualNet has fewer layers and takes the outputs of SimpleNet as inputs. Its purpose is to refine the disparity maps predicted by SimpleNet by learning an additive residual signal. Similar residual learning architectures have been successfully applied to related deep learning tasks [20, 34]. The detailed network architecture is illustrated in the [supplementary material](#).

SimpleNet. SimpleNet is an encoder-decoder architecture with a ResNet-50 based encoder and skip connections between corresponding encoder and decoder layers. The decoder upprojects the feature maps to the original resolution and generates 4 pairs of disparity maps $disp_{\text{simple},s}^{\text{left}}$ and $disp_{\text{simple},s}^{\text{right}}$ in different resolutions $s \in [0, 3]$. The upprojection is implemented by resize-convolution [33],

i.e. a nearest-neighbor upsampling layer by a factor of two followed by a convolutional layer. The usage of skip connections enables the decoder to recover high-resolution results with fine-grained details.

ResidualNet. The purpose of ResidualNet is to further refine the disparity maps predicted by SimpleNet. ResidualNet learns the residual signals $disp_{res,s}$ to the disparity maps $disp_{simple,s}$ (both left and right and for all resolutions). Inspired by FlowNet 2.0 [20], the inputs to ResidualNet contain various information on the prediction and the errors made by SimpleNet: we input I^{left} , $disp_{simple,0}^{left}$, I_{recons}^{right} , I_{recons}^{left} and e_l , where

- I_{recons}^{right} is the reconstructed right image by warping I^{left} using $disp_{simple,0}^{right}$.
- I_{recons}^{left} is the generated left image by back-warping I_{recons}^{right} using $disp_{simple,0}^{left}$.
- e_l is the ℓ_1 reconstruction error between I^{left} and I_{recons}^{left}

For the warping, rectified stereo images are required while stereo camera intrinsics and extrinsics are not needed as our network directly outputs disparity.

The final refined outputs $disp_s$ are $disp_s = disp_{simple,s} \oplus disp_{res,s}$, $s \in [0, 3]$, where \oplus is element-wise summation. The encoder of ResidualNet contains 12 residual blocks in total and predicts 4 scales of residual disparity maps as SimpleNet. Adding more layers does not further improve performance in our experiments. Notably, only the left image is used as an input to either SimpleNet and ResidualNet, while the right image is not required. However, the network outputs a refined disparity map for the left and right stereo image. Both facts will be important for our monocular visual odometry approach.

2.2 Loss Function

We define a loss \mathcal{L}_s at each output scale s , resulting in the total loss $\mathcal{L} = \sum_{s=0}^3 \mathcal{L}_s$. The loss at each scale \mathcal{L}_s is a linear combination of five terms which are symmetric in left and right images,

$$\begin{aligned} \mathcal{L}_s = & \alpha_U \left(\mathcal{L}_U^{left} + \mathcal{L}_U^{right} \right) + \alpha_S \left(\mathcal{L}_S^{left} + \mathcal{L}_S^{right} \right) + \alpha_{lr} \left(\mathcal{L}_{lr}^{left} + \mathcal{L}_{lr}^{right} \right) \\ & + \alpha_{smooth} \left(\mathcal{L}_{smooth}^{left} + \mathcal{L}_{smooth}^{right} \right) + \alpha_{occ} \left(\mathcal{L}_{occ}^{left} + \mathcal{L}_{occ}^{right} \right), \end{aligned} \quad (1)$$

where \mathcal{L}_U is a self-supervised loss, \mathcal{L}_S is a supervised loss, \mathcal{L}_{lr} is a left-right consistency loss, \mathcal{L}_{smooth} is a smoothness term encouraging the predicted disparities to be locally smooth and \mathcal{L}_{occ} is an occlusion regularization term. In the following, we detail the left components \mathcal{L}^{left} of the loss function at each scale. The right components \mathcal{L}^{right} are defined symmetrically.

Self-supervised Loss. The self-supervised loss measures the quality of the reconstructed images. The reconstructed image is generated by warping the input image into the view of the other rectified stereo image. This procedure is fully (sub-)differentiable for bilinear sampling [21]. Inspired by [14, 47], the quality of

the reconstructed image is measured with the combination of the ℓ_1 loss and single scale structural similarity (SSIM) [42]:

$$\begin{aligned} \mathcal{L}_U^{left} = & \frac{1}{N} \sum_{x,y} \alpha \frac{1 - \text{SSIM}(I^{left}(x,y), I_{recons}^{left}(x,y))}{2} \\ & + (1 - \alpha) \|I^{left}(x,y) - I_{recons}^{left}(x,y)\|_1, \end{aligned} \tag{2}$$

with a 3×3 box filter for SSIM and α set to 0.84.

Supervised Loss. The supervised loss measures the deviation of the predicted disparity map from the disparities estimated by Stereo DSO at a sparse set of pixels:

$$\mathcal{L}_s^{left} = \frac{1}{N} \sum_{(x,y) \in \Omega_{DSO, left}} \beta_\epsilon \left(disp^{left}(x,y) - disp_{DSO}^{left}(x,y) \right) \tag{3}$$

where $\Omega_{DSO, left}$ is the set of pixels with disparities estimated by DSO and $\beta_\epsilon(x)$ is the reverse Huber (berHu) norm introduced in [24] which lets the training focus more on larger residuals. The threshold ϵ is adaptively set as a batch-dependent value $\epsilon = 0.2 \max_{(x,y) \in \Omega_{DSO, left}} |disp^{left}(x,y) - disp_{DSO}^{left}(x,y)|$.

Left-Right Disparity Consistency Loss. Given only the left image as input, the network predicts the disparity map of the left as well as the right image as in [14]. As proposed in [14, 47], consistency between the left and right disparity image is improved by

$$\mathcal{L}_{lr}^{left} = \frac{1}{N} \sum_{x,y} \left| disp^{left}(x,y) - disp^{right}(x - disp^{left}(x,y), y) \right|. \tag{4}$$

Disparity Smoothness Regularization. Depth reconstruction based on stereo image matching is an ill-posed problem on its own: the depth of homogeneously textured areas and occluded areas cannot be determined. For these areas, we apply the regularization term

$$\mathcal{L}_{smooth}^{left} = \frac{1}{N} \sum_{x,y} \left| \nabla_x^2 disp^{left}(x,y) \right| e^{-\left\| \nabla_x^2 I^{left}(x,y) \right\|} + \left| \nabla_y^2 disp^{left}(x,y) \right| e^{-\left\| \nabla_y^2 I^{left}(x,y) \right\|} \tag{5}$$

that assumes that the predicted disparity map should be locally smooth. We use a second-order smoothness prior [43] and downweight it when the image gradient is high [18].

Occlusion Regularization. $\mathcal{L}_{smooth}^{left}$ itself tends to generate a shadow area where values gradually change from foreground to background due to stereo occlusion. To favor background depths and hard transitions at occlusions [48], we impose \mathcal{L}_{occ}^{left} which penalizes the total sum of absolute disparities. The combination of smoothness- and occlusion regularizer prefers to directly take the

(smaller) closeby background disparity which better corresponds to the assumption that the background part is uncovered

$$\mathcal{L}_{occ}^{left} = \frac{1}{N} \sum_{x,y} \left| disp^{left}(x,y) \right|. \quad (6)$$

3 Deep Virtual Stereo Odometry

Deep Virtual Stereo Odometry (DVSO) builds on the windowed sparse direct bundle adjustment formulation of monocular DSO. We use our disparity predictions for DSO in two key ways: Firstly, we initialize depth maps of new keyframes from the disparities. Beyond this rather straight-forward approach, we also incorporate virtual direct image alignment constraints into the windowed direct bundle adjustment of DSO. We obtain these constraints by warping images with the estimated depth by bundle adjustment and the predicted right disparities by our network assuming a virtual stereo setup. As shown in Fig. 3, DVSO integrates both the predicted left disparities and right disparities for the left image. The right image of the stereo setup is not used for our VO method at any stage, making it a monocular VO method.

In the following, we use D^L and D^R as shorthands to represent the predicted left ($disp_0^{left}$) and right disparity map ($disp_0^{right}$) at scale $s = 0$, respectively. When using purely geometric cues, scale drift is one of the main sources of error of monocular VO due to scale unobservability [37]. In DVSO we use the left disparity map D^L predicted by StackNet for initialization instead of randomly initializing the depth like in monocular DSO [8]. The disparity value of an image point with coordinate \mathbf{p} is converted to the inverse depth $d_{\mathbf{p}}$ using the rectified camera intrinsics and stereo baseline of the training set of StackNet [16], $d_{\mathbf{p}} = \frac{D^L(\mathbf{p})}{f_x b}$. In this way, the initialization of DVSO becomes more stable than monocular DSO and the depths are initialized with a consistent metric scale.

The point selection strategy of DVSO is similar to monocular DSO [8], while we also introduce a left-right consistency check (similar to Equation (4)) to filter out the pixels which likely lie in the occluded area

$$e_{lr} = \left| D^L(\mathbf{p}) - D^R(\mathbf{p}') \right| \quad \text{with} \quad \mathbf{p}' = \mathbf{p} - [D^L(\mathbf{p}) \ 0]^\top. \quad (7)$$

The pixels with $e_{lr} > 1$ are not selected.

Every new frame is firstly tracked with respect to the reference keyframe using direct image alignment in a coarse-to-fine manner [8]. Afterwards DVSO decides if a new keyframe has to be created for the new frame following the criteria proposed by [8]. When a new keyframe is created, the temporal multi-view energy function $E_{photo} := \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{j \in \text{obs}(\mathbf{p})} E_{ij}^{\mathbf{p}}$ needs to be optimized, where \mathcal{F} is a fixed-sized window containing the active keyframes, \mathcal{P}_i is the set of points selected from its host keyframe with index i and $j \in \text{obs}(\mathbf{p})$ is the index of the keyframe which observes \mathbf{p} . $E_{ij}^{\mathbf{p}}$ is the photometric error of the point \mathbf{p} when projected from the host keyframe I_i onto the other keyframe I_j :

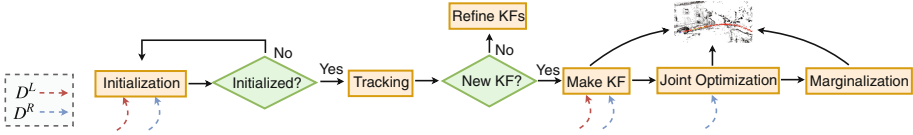


Fig. 3. System overview of DVSO. Every new frame is used for visual odometry and fed into the proposed StackNet to predict left and right disparity. The predicted left and right disparities are used for depth initialization, while the right disparity is used to form the virtual stereo term in direct sparse bundle adjustment.

$$E_{ij}^{\mathbf{P}} := \omega_{\mathbf{P}} \left\| (I_j[\tilde{\mathbf{p}}] - b_j) - \frac{e^{a_j}}{e^{a_i}} (I_i[\mathbf{p}] - b_i) \right\|_{\gamma}, \quad (8)$$

where $\tilde{\mathbf{p}}$ is the projected image coordinate using the relative rotation matrix $\mathbf{R} \in SO(3)$ and translation vector $\mathbf{t} \in \mathbb{R}^3$ [16], $\tilde{\mathbf{p}} = \Pi_c(\mathbf{R}\Pi_c^{-1}(\mathbf{p}, d_{\mathbf{p}}) + \mathbf{t})$, where Π_c and Π_c^{-1} are the camera projection and back-projection functions. The parameters a_i, a_j, b_i and b_j are used for modeling the affine brightness transformation [8]. The weight $\omega_{\mathbf{P}}$ penalizes the points with high image gradient [8] with the intuition that the error originating from bilinear interpolation of the discrete image values is larger. $\|\cdot\|_{\gamma}$ is the Huber norm with the threshold γ . For the detailed explanation of the energy function, please refer to [8].

To further improve the accuracy of DVSO, inspired by Stereo DSO [40] which couples the static stereo term with the temporal multi-view energy function, we introduce a novel *virtual stereo* term $E^{\dagger\mathbf{P}}$ for each point \mathbf{p}

$$E_i^{\dagger\mathbf{P}} = \omega_{\mathbf{P}} \left\| I_i^{\dagger}[\mathbf{p}^{\dagger}] - I_i[\mathbf{p}] \right\|_{\gamma} \quad \text{with} \quad I_i^{\dagger}[\mathbf{p}^{\dagger}] = I_i[\mathbf{p}^{\dagger} - [D^R(\mathbf{p}^{\dagger}) \ 0]^{\top}], \quad (9)$$

where $\mathbf{p}^{\dagger} = \Pi_c(\Pi_c^{-1}(\mathbf{p}, d_{\mathbf{p}}) + \mathbf{t}_b)$ is the virtual projected coordinate of \mathbf{p} using the vector \mathbf{t}_b denoting the virtual stereo baseline which is known during the training of StackNet. The intuition behind this term is to optimize the estimated depth of the visual odometry to become consistent with the disparity prediction of StackNet. Instead of imposing the consistency directly on the estimated and predicted disparities, we formulate the residuals in photoconsistency which better reflects the uncertainties of the prediction of StackNet and also keeps the unit of the residuals consistent with the temporal direct image alignment terms.

We then optimize the total energy

$$E_{photo} := \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \left(\lambda E_i^{\dagger\mathbf{P}} + \sum_{j \in \text{obs}(\mathbf{p})} E_{ij}^{\mathbf{P}} \right), \quad (10)$$

where the coupling factor λ balances the temporal and the virtual stereo term. All the parameters of the total energy are jointly optimized using the Gauss Newton method [8]. In order to keep a fixed size of the active window ($N = 7$ keyframes in our experiments), old keyframes are removed from the system by marginalization using the Schur complement [8]. Unlike sliding window bundle

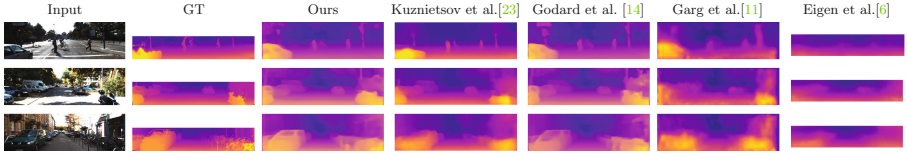


Fig. 4. Qualitative comparison with state-of-the-art methods. The ground truth is interpolated for better visualization. Our approach shows better prediction on thin structures than the self-supervised approach [14], and delivers more detailed disparity maps than the semi-supervised approach using LiDAR data [23].

adjustment, the parameter estimates outside the optimization window including camera poses and depths in a marginalization prior are also incorporated into the optimization. In contrast to the MSCKF [29], the depths of pixels are explicitly maintained in the state and optimized for. In our optimization framework we trade off predicted depth and triangulated depth using robust norms.

4 Experiments

We quantitatively evaluate our StackNet with other state-of-the-art monocular depth prediction methods on the publicly available KITTI dataset [12]. In the supplementary materials, we demonstrate results on the Cityscapes dataset [3] and the Make3D dataset [36] to show the generalization ability. For DVSO, we evaluate its tracking accuracy on the KITTI odometry benchmark with other state-of-the-art monocular as well as stereo visual odometry systems. In the supplementary material, we also demonstrate its results on the *Frankfurt* sequence of the Cityscapes dataset to show the generalization of DVSO.

4.1 Monocular Depth Estimation

Dataset. We train StackNet using the train/test split (\mathbf{K}) of Eigen et al. [6]. The training set contains 23488 images from 28 scenes belonging to the categories “city”, “residential” and “road”. We used 22600 images of them for training and the remaining ones for validation. We further split \mathbf{K} into 2 subsets \mathbf{K}_o and \mathbf{K}_r . \mathbf{K}_o contains the images of the sequences which appear in the training set (but not the test set) of the KITTI odometry benchmark on which we use Stereo DSO [40] to extract sparse ground-truth depth data. \mathbf{K}_r contains the remaining images in \mathbf{K} . Specifically, \mathbf{K}_o contains the images of sequences 01, 02, 06, 08, 09 and 10 of the KITTI odometry benchmark.

Implementation Details. StackNet is implemented in TensorFlow [1] and trained from scratch on a single Titan X Pascal GPU. We resize the images to 512×256 for training and it takes less than 40 ms for inference including the I/O overhead. The weights are set to $\alpha_u = 1$, $\alpha_s = 10$, $\alpha_{lr} = 1$, $\alpha_{smooth} = 0.1/2^s$

and $\alpha_{occ} = 0.01$, where s is the output scale. As suggested by [14], we use exponential linear units (ELUs) for SimpleNet, while we use leaky rectified linear units (Leaky ReLUs) for ResidualNet. We first train SimpleNet on \mathbf{K}_o in the semi-supervised way for 80 epochs with a batch size of 8 using the Adam optimizer [22]. The learning rate is initially set to $\lambda = 10^{-4}$ for the first 50 epochs and halved every 15 epochs afterwards until the end. Then we train SimpleNet with $\lambda = 5 \times 10^{-5}$ on \mathbf{K}_r for 40 epochs in the self-supervised way without \mathcal{L}_S . In the end, we train again on \mathbf{K}_o without \mathcal{L}_U using $\lambda = 10^{-5}$ for 5 epochs. We explain the dataset schedule as well as the parameter tuning in detail in the [supplementary material](#).

After training SimpleNet, we freeze its weights and train StackNet by cascading ResidualNet. StackNet is trained with $\lambda = 5 \times 10^{-5}$ in the same dataset schedules but with less epochs, i.e. 30, 15, 3 epochs, respectively. We apply random gamma, brightness and color augmentations [14]. We also employ the post-processing for left disparities proposed by Godard et al. [14] to reduce the effect of stereo disocclusions. In the [supplementary material](#) we also provide an ablation study on the various loss terms.

KITTI. Table 1 shows the evaluation results with the error metrics used in [6]. We crop the images as applied by Eigen et al. [6] to compare with [14], [23] within different depth ranges. The best performance of our network is achieved with the dataset schedule $\mathbf{K}_o \rightarrow \mathbf{K}_r \rightarrow \mathbf{K}_o$ as we described above. We outperform the state-of-the-art self-supervised approach proposed by Godard et al. [14] by a large margin. Our method also outperforms the state-of-the-art semi-supervised method using the LiDAR ground truth proposed by Kuznetsov et al. [23] on all the metrics except for the less restrictive $\delta < 1.25^2$ and $\delta < 1.25^3$.

Figure 4 shows a qualitative comparison with other state-of-the-art methods. Compared to the semi-supervised approach, our results contain more details and deliver comparable prediction on thin structures like the poles. Although the results of Godard et al. [14] appear more detailed on some parts, they are not actually accurate, which can be inferred by the quantitative evaluation. In general, the predictions of Godard et al. [14] on thin objects are not as accurate as our method. In the [supplementary material](#), we show the error maps for the predicted depth maps. Figure 5 further show the advantages of our method compared to the state-of-the-art self-supervised and semi-supervised approaches. The results of Godard et al. [14] are predicted by the network trained with both the Cityscapes dataset and the KITTI dataset. On the wall of the far building in the left figure, our network can better predict consistent depth on the surface, while the prediction of the self-supervised network shows strong checkerboard artifact, which is apparently inaccurate. The semi-supervised approach also shows checkerboard artifact (but much slighter). The right side of the figure shows shadow artifacts for the approach of Godard et al. [14] around the boundaries of the traffic sign, while the result of Kuznetsov et al. [23] fails to predict the structure. Please refer to our [supplementary material](#) for further results. We also demonstrate how our trained depth prediction network generalizes to other datasets in the [supplementary material](#).

Table 1. Evaluation results on the KITTI [13] Raw test split of Eigen et al. [6]. **CS** refers to the Cityscapes dataset [3]. Upper part: depth range 0-80m, lower part: 1-50m. All results are obtained using the crop from [6]. Our SimpleNet trained on \mathbf{K}_o outperforms [14] (self-supervised) trained on **CS** and **K**. StackNet also outperforms semi-supervision with LiDAR [23] on most metrics.

Approach	Dataset	RMSE	RMSE (log)	ARD	SRD	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		Lower is better				Higher is better		
Godard et al. [14], ResNet	CS → K	4.935	0.206	0.114	0.898	0.861	0.949	0.976
Kuznetsov et al. [23]	K	<i>4.621</i>	<i>0.189</i>	0.113	<i>0.741</i>	0.862	0.960	0.986
Ours, SimpleNet	\mathbf{K}_o	4.886	0.209	0.112	0.888	0.862	0.950	0.976
Ours, SimpleNet	$\mathbf{K}_o \rightarrow \mathbf{K}_r$	4.817	0.202	0.108	0.862	<i>0.867</i>	0.950	0.977
Ours, SimpleNet	$\mathbf{K}_r \rightarrow \mathbf{K}_o$	4.890	0.208	0.115	0.870	0.863	0.950	0.977
Ours, SimpleNet	$\mathbf{K}_o \rightarrow \mathbf{K}_r \rightarrow \mathbf{K}_o$	4.785	0.199	<i>0.107</i>	0.852	0.866	0.950	0.978
Ours, StackNet	$\mathbf{K}_o \rightarrow \mathbf{K}_r \rightarrow \mathbf{K}_o$	4.442	0.187	0.097	0.734	0.888	<i>0.958</i>	<i>0.980</i>
Garg et al. [11] L12 Aug 8x	K	5.104	0.273	0.169	1.080	0.740	0.904	0.962
Godard et al. [14], ResNet	CS → K	3.729	0.194	0.108	0.657	0.873	0.954	0.979
Kuznetsov et al. [23]	K	<i>3.518</i>	<i>0.179</i>	<i>0.108</i>	<i>0.595</i>	<i>0.875</i>	0.964	0.988
Ours, StackNet	$\mathbf{K}_o \rightarrow \mathbf{K}_r \rightarrow \mathbf{K}_o$	3.390	0.177	0.092	0.547	0.898	<i>0.962</i>	<i>0.982</i>

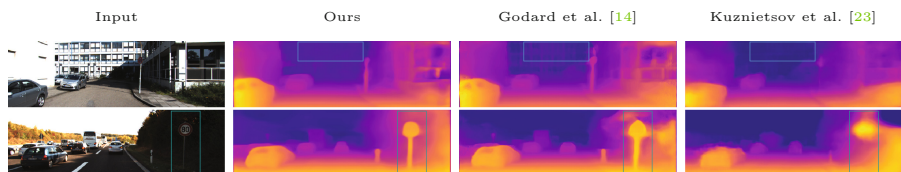


Fig. 5. Qualitative results on Eigen et al.’s KITTI Raw test split. The result of Godard et al. [14] shows a strong shadow effect around object contours, while our result does not. The result of Kuznetsov [23] shows failure on predicting the traffic sign. Both other methods [14, 23] predict checkerboard artifacts on the far building, while our approach predicts such artifacts less.

4.2 Monocular Visual Odometry

KITTI Odometry Benchmark. The KITTI odometry benchmark contains 11 (0–10) training sequences and 11 (11–21) test sequences. Ground-truth 6D poses are provided for the training sequences, whereas for the test sequences evaluation results are obtained by submitting to the KITTI website. We use the error metrics proposed in [12].

We firstly provide an ablation study for DVSO to show the effectiveness of the design choices in our approach. In Table 2 we give results for DVSO in different variants with the following components: initializing the depth with the left disparity prediction (*in*), using the right disparity for the virtual stereo term

in windowed bundle adjustment (*vs*), checking left-right disparity consistency for point selection (*lr*), and tuning the virtual stereo baseline *tb*. The intuition behind the virtual stereo baseline is that StackNet is trained over various camera parameters and hence provides a depth scale for an average baseline. For *tb*, we therefore tune the scale factors of different sequences with different camera intrinsics, to better align the estimated scale with the ground truth. Baselines are tuned for each of the 3 different camera parameter sets in the training set individually using grid search on one training sequence. Specifically, we tuned the baselines on sequences 00, 03 and 05 which correspond to 3 different camera parameter sets. The test set contains the same camera parameter sets as the training set and we map the virtual baselines for *tb* correspondingly. Monocular DSO (after Sim(3) alignment) is also shown as the baseline. The results show that our full approach achieves the best average performance. Our StackNet also adds significantly to the performance of DVSO compared with using depth predictions from [14].

Table 2. Ablation study for DVSO. * and † indicate the sequences used and not used for training StackNet, respectively. $t_{rel}(\%)$ and $r_{rel}(^\circ)$ are translational- and rotational RMSE, respectively. Both t_{rel} and r_{rel} are averaged over 100 to 800 m intervals. *in*: D^L is used for depth initialization. *vs*: virtual stereo term is used with D^R . *lr*: left-right disparity consistency is checked using predictions. *tb*: tuned virtual baseline is used. DVSO’([14]): full (*in, vs, lr, tb*) with depth from [14]. DVSO: full with depth from StackNet. Best results are shown as bold, second best italic. DVSO clearly outperforms the other variants.

	Mono DSO		<i>in</i>		<i>in, vs</i>		<i>in, vs, lr</i>		<i>in, vs, tb</i>		DVSO’([14])		DVSO	
Seq.	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00†	188	<i>0.25</i>	13.1	0.30	0.95	0.24	0.93	0.24	<i>0.73</i>	<i>0.25</i>	1.02	0.28	0.71	0.24
03†	17.7	0.17	9.10	0.29	2.56	0.19	2.56	<i>0.18</i>	0.78	0.19	4.78	0.18	<i>0.79</i>	<i>0.18</i>
04†	0.82	0.16	0.83	0.29	0.69	0.06	0.67	<i>0.07</i>	<i>0.36</i>	0.06	2.03	0.14	0.35	0.06
05†	72.6	<i>0.23</i>	12.7	<i>0.23</i>	0.67	<i>0.23</i>	0.64	<i>0.23</i>	<i>0.61</i>	0.23	2.11	0.21	0.58	0.22
07†	48.4	0.32	18.5	0.91	0.85	0.41	<i>0.80</i>	0.38	0.81	0.40	1.09	0.39	0.73	<i>0.35</i>
01*	9.17	<i>0.12</i>	4.30	0.41	1.50	0.11	1.52	<i>0.12</i>	1.15	0.11	1.23	0.11	<i>1.18</i>	0.11
02*	114	0.22	9.58	0.26	1.08	<i>0.23</i>	1.05	<i>0.23</i>	<i>0.86</i>	<i>0.23</i>	0.87	<i>0.23</i>	0.84	0.22
06*	42.2	0.20	11.2	0.30	0.84	<i>0.23</i>	0.80	0.24	<i>0.73</i>	<i>0.23</i>	0.87	0.24	0.71	0.20
08*	177	<i>0.26</i>	14.9	0.28	1.11	<i>0.26</i>	1.10	<i>0.26</i>	<i>1.05</i>	<i>0.26</i>	<i>1.05</i>	<i>0.26</i>	1.03	0.25
09*	28.1	0.21	14.2	<i>0.23</i>	1.03	0.21	0.95	0.21	0.88	0.21	<i>0.87</i>	0.21	0.83	0.21
10*	24.0	<i>0.22</i>	9.93	0.27	0.58	0.23	<i>0.59</i>	<i>0.22</i>	0.74	<i>0.22</i>	0.68	0.21	0.74	0.21
Mean†	65.50	0.23	10.85	0.40	1.14	0.23	1.12	<i>0.22</i>	<i>0.66</i>	0.23	2.21	0.24	0.63	0.21
Mean*	65.75	<i>0.21</i>	10.69	0.29	1.02	<i>0.21</i>	1.00	<i>0.21</i>	<i>0.90</i>	<i>0.21</i>	0.93	<i>0.21</i>	0.89	0.20
Overall mean	65.64	<i>0.21</i>	10.76	0.34	1.08	0.22	1.06	0.22	<i>0.79</i>	0.22	1.51	0.22	0.77	0.20

We also compare DVSO with other state-of-the-art *stereo* visual odometry systems on the sequences 00–10. The sequences with marker * are used for training StackNet and the sequences with marker † are not used for training the network. In Table 3 and the following tables, DVSO means our full approach with baseline tuning (*in, vs, lr, tb*). The average RMSE of DVSO without baseline

Table 3. Comparison with state-of-the-art stereo visual odometry. DVSO: our full approach (*in, vs, lr, tb*). Global optimization and loop-closure are turned off for stereo ORB-SLAM2 and Stereo LSD-SLAM. DVSO (monocular) achieves comparable performance to these stereo methods.

Seq.	St. LSD-VO [10]		ORB-SLAM2 [31]		St. DSO [40]		<i>in, vs, lr</i>		DVSO	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00 [†]	1.09	0.42	0.83	0.29	0.84	0.26	0.93	0.24	0.71	0.24
03 [†]	1.16	0.32	0.71	0.17	0.92	0.16	2.56	0.18	0.77	0.18
04 [†]	0.42	0.34	0.45	0.18	0.65	0.15	0.67	0.07	0.35	0.06
05 [†]	0.90	0.34	0.64	0.26	0.68	0.19	0.64	0.23	0.58	0.22
07 [†]	1.25	0.79	0.78	0.42	0.83	0.36	0.80	0.38	0.73	0.35
01*	2.13	0.37	1.38	0.20	1.43	0.09	1.52	0.12	1.18	0.11
02*	1.09	0.37	0.81	0.28	0.78	0.21	1.05	0.23	0.84	0.22
06*	1.28	0.43	0.82	0.25	0.67	0.20	0.80	0.24	0.71	0.20
08*	1.24	0.38	1.07	0.31	0.98	0.25	1.10	0.26	1.03	0.25
09*	1.22	0.28	0.82	0.25	0.98	0.18	0.95	0.21	0.83	0.21
10*	0.75	0.34	0.58	0.28	0.49	0.18	0.59	0.22	0.74	0.21
mean [†]	0.96	0.44	0.68	0.26	0.78	0.22	1.12	0.22	0.63	0.21
mean*	1.29	0.36	0.91	0.26	0.89	0.19	1.00	0.21	0.89	0.20
overall mean	1.14	0.40	0.81	0.26	0.84	0.20	1.06	0.22	0.77	0.20

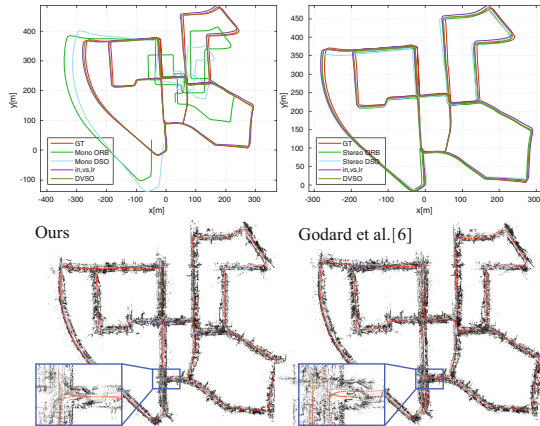


Fig. 6. Results on KITTI odometry seq. 00. Top: comparisons with monocular methods (Sim(3)-aligned) and stereo methods. DVSO provides significantly more consistent trajectories than other monocular methods and compares well to stereo approaches. Bottom: DVSO with StackNet produces more accurate trajectory and map than with [14].

tuning is better than Stereo LSD-VO, but not as good as Stereo DSO [40] or ORB-SLAM2 [31] (stereo, without global optimization and loop closure). Importantly, DVSO uses only monocular images. With the baseline tuning, DVSO achieves even better average performance than all other stereo systems on both rotational and translational errors. Figure 6 shows the estimated trajectory on sequence 00. Both monocular ORB-SLAM2 and DSO suffer from strong scale drift, while DVSO achieves superior performance on eliminating the scale drift. We also show the estimated trajectory on 00 by running DVSO using the depth map predicted by Godard et al. [14] with the model trained on the Cityscapes and the KITTI dataset. For the results in Fig. 6 our depth predictions are more

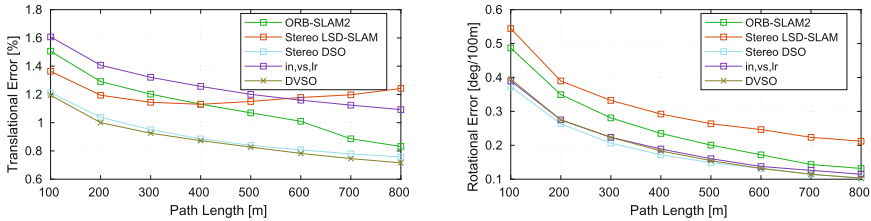


Fig. 7. Evaluation results on the KITTI odometry test set. We show translational and rotational errors with respect to path length intervals. For translational errors, DVSO achieves comparable performance to Stereo LSD-SLAM, while for rotational errors, DVSO achieves comparable results to Stereo DSO and better results than all other methods. Note that with virtual baseline tuning, DVSO achieves the best performance among all the methods evaluated.

Table 4. Comparison with deep learning approaches. Note that Deep VO [41] is trained on sequences 00, 02, 08 and 09 of the KITTI Odometry Benchmark. UnDeepVO [26] and SfMLearner [49] are trained unsupervised on seqs 00-08 end-to-end. Results of DeepVO and UnDeepVO taken from [41] and [26] while for SfMLearner we ran their pre-trained model. Our DVSO clearly outperforms state-of-the-art deep learning based VO methods.

Seq.	DeepVO [41]		UnDeepVO [26]		Yin et al. [46]		SfMLearner [49]		<i>in, vs, lr</i>		DVSO	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00 [†]	—	—	4.41	1.92	—	—	66.35	6.13	0.93	0.24	0.71	0.24
03 [†]	8.49	6.89	5.00	6.17	—	—	10.78	3.92	2.56	0.18	0.77	0.18
04 [†]	7.19	6.97	4.49	2.13	—	—	4.49	5.24	0.67	0.07	0.35	0.06
05 [†]	2.62	3.61	3.40	1.50	—	—	18.67	4.10	0.64	0.23	0.58	0.22
07 [†]	3.91	4.60	3.15	2.48	—	—	21.33	6.65	0.80	0.38	0.73	0.35
01*	—	—	69.07	1.60	—	—	35.17	2.74	1.52	0.12	1.18	0.11
02*	—	—	5.58	2.44	—	—	58.75	3.58	1.05	0.23	0.84	0.22
06*	5.42	5.82	6.20	1.98	—	—	25.88	4.80	0.80	0.24	0.71	0.20
08*	—	—	4.08	1.79	2.22	0.10	21.90	2.91	1.10	0.26	1.03	0.25
09*	—	—	7.01	3.61	4.14	0.11	18.77	3.21	0.95	0.21	0.83	0.21
10*	8.11	8.83	10.63	4.65	1.70	0.17	14.33	3.30	0.59	0.22	0.74	0.21

accurate. Figure 7 shows the evaluation result of the sequences 11–21 by submitting results of DVSO with and without baseline tuning to the KITTI odometry benchmark. Note that in Fig. 7, Stereo LSD-SLAM and ORB-SLAM2 are both full stereo SLAM approaches with global optimization and loop closure. For qualitative comparisons of further estimated trajectories, please refer to our [supplementary material](#).

We also compare DVSO with DeepVO [41], UnDeepVO [26] and SfM-Learner [49] which are deep learning based visual odometry systems trained end-to-end on KITTI. As shown in Table 4, on all available sequences, DVSO achieves better performance than the other two end-to-end approaches. Table 4 also shows the comparison with the deep learning based scale recovery methods for monocular VO proposed by Yin et al. [46]. DVSO also outperforms their method. In the [supplementary material](#), we also show the estimated trajectory on the Cityscapes *Frankfurt* sequence to demonstrate generalization capabilities.

5 Conclusion

We presented a novel monocular visual odometry system, DVSO, which recovers metric scale and reduces scale drift in geometric monocular VO. A deep learning approach predicts monocular depth maps for the input images which are used to initialize sparse depths in DSO to a consistent metric scale. Odometry is further improved by a novel virtual stereo term that couples estimated depth in windowed bundle adjustment with the monocular depth predictions. For monocular depth prediction we have presented a semi-supervised deep learning approach, which utilizes a self-supervised image reconstruction loss and sparse depth predictions from Stereo DSO as ground truth depths for supervision. A stacked network architecture predicts state-of-the-art refined disparity estimates.

Our evaluation conducted on the KITTI odometry benchmark demonstrates that DVSO outperforms the state-of-the-art monocular methods by a large margin and achieves comparable results to stereo VO methods. With virtual baseline tuning, DVSO can even outperform state-of-the-art stereo VO methods, i.e., Stereo LSD-VO, ORB-SLAM2 without global optimization and loop closure, and Stereo DSO, while using only monocular images.

The key practical benefit of the proposed method is that it allows us to recover accurate and scale-consistent odometry with only a single camera. Future work could comprise fine-tuning of the network inside the odometry pipeline end-to-end. This could enable the system to adapt online to new scenes and camera setups. Given that the deep net was trained on driving sequences, in future work we also plan to investigate how much the proposed approach can generalize to other camera trajectories and environments.

Acknowledgements. We would like to thank Clément Godard, Yevhen Kuznetsov, Ruihao Li and Tinghui Zhou for providing the data and code. We also would like to thank Martin Schwörer for the fruitful discussion.

References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. *OSDI* **16**, 265–283 (2016)
2. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: Codeslam-learning a compact, optimisable representation for dense visual slam. arXiv preprint [arXiv:1804.00874](https://arxiv.org/abs/1804.00874) (2018)
3. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
4. Dang, T., Hoffmann, C., Stiller, C.: Continuous stereo self-calibration by camera parameter tracking. *IEEE Trans. Image Process.* **18**(7), 1536–1550 (2009)
5. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658 (2015)
6. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: *Advances in Neural Information Processing Systems*, pp. 2366–2374 (2014)
7. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: *IEEE International Conference on Computer Vision (ICCV)* (2013)
8. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(3), 611–625 (2017)
9. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
10. Engel, J., Stückler, J., Cremers, D.: Large-scale direct slam with stereo cameras. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1935–1942. IEEE (2015)
11. Garg, R., B.G., V.K., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: geometry to the rescue. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9912, pp. 740–756. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_45
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res. (IJRR)* **32**(11), 1231–1237 (2013)
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2012)
14. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. arXiv preprint [arXiv:1609.03677](https://arxiv.org/abs/1609.03677) (2016)
15. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: *IEEE international conference on Robotics and automation (ICRA)*, pp. 1524–1531. IEEE (2014)
16. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2003)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
18. Heise, P., Klose, S., Jensen, B., Knoll, A.: Pm-huber: Patchmatch with huber regularization for stereo matching. In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2360–2367. IEEE (2013)

19. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. In: ACM Transactions on Graphics (TOG), vol. 24, pp. 577–584. ACM (2005)
20. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, p. 6 (2017)
21. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, pp. 2017–2025 (2015)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
23. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. arXiv preprint [arXiv:1702.02706](https://arxiv.org/abs/1702.02706) (2017)
24. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: Fourth International Conference on 3D Vision (3DV), pp. 239–248. IEEE (2016)
25. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFS. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1119–1127 (2015)
26. Li, R., Wang, S., Long, Z., Gu, D.: UnDeepVO: Monocular visual odometry through unsupervised deep learning. arXiv preprint [arXiv:1709.06841](https://arxiv.org/abs/1709.06841) (2017)
27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
28. Mayer, N., et al.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4040–4048 (2016)
29. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint kalman filter for vision-aided inertial navigation. In: IEEE International Conference on Robotics and Automation, pp. 3565–3572. IEEE (2007)
30. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
31. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2017)
32. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: IEEE International Conference on Computer Vision (ICCV), pp. 2320–2327. IEEE (2011)
33. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). <https://doi.org/10.23915/distill.00003>, <http://distill.pub/2016/deconv-checkerboard>
34. Pang, J., Sun, W., Ren, J., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: International Conference on Computer Vision-Workshop on Geometry Meets Deep Learning (ICCVW 2017), vol. 3 (2017)
35. Pizzoli, M., Forster, C., Scaramuzza, D.: Remode: Probabilistic, monocular dense reconstruction in real time. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2609–2616. IEEE (2014)
36. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Advances in Neural Information Processing Systems, pp. 1161–1168 (2006)
37. Strasdat, H., Montiel, J., Davison, A.J.: Scale drift-aware large scale monocular SLAM. Robotics: Science and Systems VI 2 (2010)

38. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D slam systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 573–580. IEEE (2012)
39. Tateno, K., Tombari, F., Laina, I., Navab, N.: Cnn-slam: Real-time dense monocular slam with learned depth prediction. arXiv preprint [arXiv:1704.03489](https://arxiv.org/abs/1704.03489) (2017)
40. Wang, R., Schwörer, M., Cremers, D.: Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In: International Conference on Computer Vision (ICCV). Venice, Italy, October 2017
41. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2043–2050. IEEE (2017)
42. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
43. Woodford, O., Torr, P., Reid, I., Fitzgibbon, A.: Global stereo reconstruction under second-order smoothness priors. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(12), 2115–2128 (2009)
44. Xie, J., Girshick, R., Farhadi, A.: Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 842–857. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_51
45. Yang, N., Wang, R., Gao, X., Cremers, D.: Challenges in monocular visual odometry: photometric calibration, motion bias and rolling shutter effect. *IEEE Robot. Autom. Lett. (RA-L)* **3**, 2878–2885 (2018). <https://doi.org/10.1109/LRA.2018.2846813>
46. Yin, X., Wang, X., Du, X., Chen, Q.: Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5870–5878 (2017)
47. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Is l2 a good loss function for neural networks for image processing?. *ArXiv e-prints* **1511** (2015)
48. Zhong, Y., Dai, Y., Li, H.: Self-supervised learning for stereo matching with self-improving ability. arXiv preprint [arXiv:1709.00930](https://arxiv.org/abs/1709.00930) (2017)
49. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR, vol. 2, p. 7 (2017)