Self-Supervised Monocular Depth Hints

Jamie Watson[1] Michael Firman[1] Gabriel J. Brostow[1,2] Daniyar Turmukhambetov[1]
[1]Niantic [2]UCL

www.github.com/nianticlabs/depth-hints

自监督：不用gt，预测不好，光度重投影损失极小值太多难优化，细杆状结构边缘不好

Abstract

Monocular depth estimators can be trained with various forms of self-supervision from binocular-stereo data to circumvent the need for high-quality laser scans or other ground-truth data. The disadvantage, however, is that the photometric reprojection losses used with self-supervised learning typically have multiple local minima. These plausible-looking alternatives to ground truth can restrict what a regression network learns, causing it to predict depth maps of limited quality. As one prominent example, depth discontinuities around thin structures are often incorrectly estimated by current state-of-the-art methods.

Here, we study the problem of ambiguous reprojections in depth prediction from stereo-based self-supervision, and introduce Depth Hints to alleviate their effects. Depth Hints are complementary depth suggestions obtained from simple off-the-shelf stereo algorithms. These hints enhance an existing photometric loss function, and are used to guide a network to learn better weights. They require no additional data, and are assumed to be right only sometimes. We show that using our Depth Hints gives a substantial boost when training several leading self-supervised-from-stereo models, not just our own. Further, combined with other good practices, we produce state-of-the-art depth predictions on the KITTI benchmark.

引入Depth Hints避免这些问题，改善光度损失

1. Introduction

As the accuracy of depth-from-color algorithms improves, new opportunities are unlocked in augmented reality, robotics, and autonomous driving. Per-pixel, ground truth depth supervision is difficult to acquire, requiring cumbersome and expensive depth-sensing devices [8]. As an alternative, there is an active search for *self-supervised* depth-estimation models, where a training signal is derived from data captured using commodity color cameras. In such self-supervised settings, training involves adjusting a network's depth predictions to minimize a photometric loss. This loss is usually the distance between a reference image and the depth-guided reprojection of other views into that reference viewpoint. Depth regression is optimized and relative poses come from stereo camera calibration in a training-from-stereo setting [9, 7, 25, 27, 26], while depth values and camera poses can be optimized jointly when training on videos [42, 20, 37, 22, 38, 32, 44, 36, 28].

The photometric distance between the reference and depth-reprojected images could be measured with L_1 or L_2 distance, more complicated structural dissimilarity distances (DSSIM [34]), or a combination of DSSIM+L_1 distances [41, 9] used in state-of-the-art methods. A drawback of self-supervision is that finding the optimal depth value is normally difficult, especially where the photometric loss can be low for *multiple* depth values (*e.g.* due to repeating structures and uniformly textures areas). Consequently, training is harder, which leads to lower accuracy predictions.

When training depth-from-color models, our Depth Hints offer a specific alternative to the model's current depth predictions. Where the alternative's reprojection is better, the training proceeds in following the "hint." Surprisingly, simply using our Depth Hints as labels for direct supervision already gives a nearly state of the art baseline. Overall, our contributions are:

1. We show that existing self-supervised regression methods can struggle during training to find the global optimum when minimizing photometric reprojection loss.

2. We demonstrate that our selective training using Depth Hints is a general enhancement that can improve multiple leading self-supervised training algorithms, allowing our implementations to reach better minima. The Depth Hints can come from the same stereo image data, via, *e.g.* OpenCV's stereo estimates [13, 14].

3. We show that our selective training with Depth Hints, coupled with sensible network design choices, leads us to outperform most other algorithms. We achieve state-of-the-art results on the KITTI dataset [8], outperforming both our baseline model and previously published results.

2. Related Work

A neural network that predicts depth from a single image could be trained with supervised depth data, or using self-supervision by exploiting photometric consistency. The many flavors of self-supervision differ by design, opting for pre-training, cropping *vs*. scaling, use of synthetic data, online *vs*. batch pose-estimation, *etc*. Here we discuss the current leading methods, and where we expect Depth Hints are and are not applicable.

2.1. Self-supervised depth prediction

Self-supervised approaches can exploit photometric consistency in binocular stereo pairs, in consecutive video frames, or in consecutive frames of a stereo video.

Stereo training: Garg *et al*. [7] formulated the self-supervised training of monocular depth estimation with photometric consistency loss between stereo pairs. They chose an L_2 loss, which tends to generate blurry results. Godard *et al*. [9] (Monodepth) used a weighted sum of DSSIM [34] and L_1 measures between correspondences. They regularized network predictions with left-right consistency between left and right disparity maps and introduced a post-processing technique that boosts depth quality, where the final depth map is a weighted average of network predictions generated from the original and horizontally flipped images. The left-right consistency was extended to a trinocular assumption by [27] for improved results.

Computing reprojection loss at a higher resolutions has been shown to improve depth map quality [10, 26, 20]. Pillai *et al*. [26] also introduced differentiable flip augmentation and subpixel convolutions for increased fidelity of depth maps. Depth Hints are computed from binocular stereo data, so should be able to enhance training for any of these stereo-derived models that use the very effective DSSIM+L_1 photometric loss.

Monocular training: SfMLearner by Zhou *et al*. [42] was the first method to train a depth prediction network from monocular video only. Their network jointly predicts depth and relative camera pose changes from a frame at time t to frame $t-1$, and from frame t to $t+1$. Using these predictions, both the future and past frames are reprojected into the current frame, and an L_1 loss is applied. Additionally, this per-pixel loss is multiplied by a predicted mask to enable occluded pixels to be ignored.

Godard *et al*. [10] build upon this, proposing that instead of averaging the loss from the reprojected future and past frames, the minimum of reprojection losses should be minimized. They also propose during training to detect and ignore pixels that appear to be stationary with respect to ego-motion. Multiple works propose additional regularization of predicted depths, such as surface normal consistency [37], edge consistency [36] and 3D pointcloud consistency [22]. Recently, multiple works [20, 28, 38, 44] have proposed to model the relationship of pixels in the consecutive frames of a video with joint estimation of optical flow, depth and camera poses with loss terms that supervise the different estimates to be consistent. Depth Hints are not naturally compatible with monocular-video only data; extensions are left as future work.

It is also possible to train from both monocular video (forward and backward in time) and stereo pairs for improved pose and depth estimation [10, 40].

2.2. Additional supervision

Following the work of Eigen *et al*. [5], many others have trained using forms of per-pixel ground-truth depth labels. Training with ground truth is almost always a good idea when it is available, and we strive to push self-supervised performance closer to this ceiling.

With LiDAR Depth: Kuznietsov *et al*. [18] optimize a fused loss, which sums a supervised loss based on sparse LiDAR pointclouds and a self-supervised loss from stereo images. They follow Godard *et al*. [9] by using DSSIM+L_1 as the photometric reprojection loss, and they follow Laina *et al*. [19] by using berHu loss (inverse Huber) [19] on the LiDAR pointcloud.

Fu *et al*. [6] showed that framing the regression of depths as ordinal classification can bring significant improvements to supervised prediction, though this concept is difficult to adopt for self-supervised training.

With Synthetic Depth: Synthetic data is an interesting source of ground-truth depths and/or stereo pairs. Instead of the usual photometric loss, domain adaptation is possible using generative adversarial networks [25], or by leveraging the ability of stereo matching networks to better generalize to real world data [11]. Luo *et al*. [21] demonstrate how synthetic data can be incorporated into single-image depth estimation with a two stage process. First, a network synthesizes a right view from the left view. Then, a second network performs stereo matching to recover depth from the half-synthetic stereo pair. Both networks can be trained on stereo+synthetic data, and optionally fine-tuned with ground truth.

With SLAM Depth: Yang *et al*. [35] train a monocular depth estimation network with both self-supervision from stereo pairs, and supervision from sparse depths estimated in batch by the Stereo DSO [33] algorithm. They demonstrate that a depth estimator network can improve visual odometry for monocular videos, resolving some scale ambiguity.

Klodt and Vedaldi [17] use sparse depths and poses from a traditional SLAM system as a supervisory signal to train depth and pose prediction networks. They train from monocular videos (in contrast to [35]), which requires special consideration of scale, and modeling of uncertainty in the depth and poses.

With Semantic Labels: Ramirez *et al*. [39] show that a depth estimation network can be improved by jointly predicting depth and semantic labels. They propose a novel cross domain discontinuity loss to help align depth discontinuities with semantic boundaries.

With Estimated Depth: The concurrent work monoResMatch by Tosi *et al*. [30] also exploits proxy ground truth labels generated with a traditional stereo matching method [13]. The inclusion of the proxy supervision is shown to greatly improve accuracy over using a standard self-supervised loss. Our proposed loss is different from theirs.

3. Background

In monocular depth estimation, the task is to train a neural network to predict a depth map d from a single input image I. In the self-supervised setting, the training data consists of pairs of images I and $I^\dagger$ with known camera intrinsics K and $K^\dagger$, and relative camera pose (R, t). The network is trained to reconstruct the reference image I by reprojecting the other image into the reference view, so

$$\tilde{I} = \pi(I^\dagger, K^\dagger, R, t, K, d). \quad (1)$$

I+根据深度和位姿重投影到I

Hence, pixel i at predicted depth d_i gets a color value $\tilde{I}_i$. Under idealized training conditions, the predicted color $\tilde{I}_i$ would perfectly match I_i for all i.

只需要估计深度，不再追求位姿线索

When training from stereo, the only unknown parameter in $\pi()$ is the estimated depth d. For monocular or stereo video, in addition to d, the network also needs to predict the camera pose (R, t). Presently, we do not pursue hints for pose, though this is a natural extension of our method.

光度损失衡量形变图和参考图的逼近程度

Many leading algorithms now use a differentiable photometric consistency loss to measure how well the warped image approximates the reference image. We focus on the DSSIM+L_1 loss, a photometric consistency loss used in many self-supervised monocular depth estimation methods [9, 26, 35, 20]. This loss is computed per pixel as

$$l_r(d_i) = \alpha \frac{1 - \text{SSIM}(I_i, \tilde{I}_i)}{2} + (1 - \alpha)|I_i - \tilde{I}_i|, \quad (2)$$

where SSIM() is computed over a 3x3 pixel window, with α set to 0.85.

If we were training *with* supervision, we would minimize the distance between continuous depth d_i predicted by the network at pixel i, and depth d'_i procured by a LiDAR system, Kinect sensor, a stereo algorithm, or a SLAM system, depending on the training context. Note that the last two contexts could count as a form of self-supervision, in that the labels d'_i are inferred, and not ground-truth measurements. There are several supervised losses l_s used and compared in the literature *e.g*. [19, 5, 15], such as L_1, L_2 and (names in superscripts):

$$l_s^{\log L_1}(d_i, d'_i) = \log(1 + |d_i - d'_i|); \quad (3)$$

log L1损失

$$l_s^{\text{berHu}}(d_i, d'_i) = \begin{cases} |d_i - d'_i|, & \text{if } |d_i - d'_i| \le \delta, \\ \frac{(d_i - d'_i)^2 + \delta^2}{2\delta}, & \text{otherwise.} \end{cases} \quad (4)$$

Typically $\delta = 0.2 \max_{i=0..N}(|d_i - d'_i|)$. Similarly, the same losses are often applied on inverse depth (*i.e*. disparity). We found that $l_s^{\log L_1}$ works well with estimated depths (and [15] favors it for Kinect data), while l_s^{berHu} is an established choice for accurate LiDAR and SLAM depths [18, 19] and disparities [35].

4. The Need for Depth Hints

Figure 1 (top) shows an input image from the training set, and the corresponding depth map produced by Godard *et al*. [10]'s network, trained on stereo data with DSSIM+L_1 loss. We can see that the network failed to converge to the correct solution, with many thin structures missing in the predicted depth map.

How do these mistakes come about? It is not failure to generalize or the result of overfitting, as this is an image from the training set. Another explanation could be that the depth map's artifacts are due to a poor choice of photometric reprojection loss, where failures on thin structures are not penalized enough. However, Figure 1 (bottom) shows DSSIM+L_1 loss for a pixel on a thin object, and we can see that the loss is lower still for more appropriate depth values.

We hypothesize that, in the absence of a ground-truth depth label, the network becomes stuck, learning to regress depth for a local minimum of the reprojection loss and failing to seek the global minimum. To escape such bad minima, we propose to consult an alternative depth value in case it can offer a more plausible reprojection, and if so, incorporate it into the objective function. We refer to these alternative depth values as Depth Hints. Depth Hints, born from noisy estimates, can be more or less accurate than our current network prediction, and therefore we expect the iterative training of a CNN to gradually change its uptake of these hints as it converges. In contrast to supervised depth prediction, though, our main focus is to converge to the best minimum using a standard self-supervised reprojection loss. Depth Hints are only used, when needed, to guide the network out of local minima.

5. Method

We assume that stereo data is being used to train a CNN to regress a depth map from a color image. We start from an existing loss function, designed for self-supervised training from such stereo images, that uses a photometric reprojection measure like DSSIM+L_1. We propose to adaptively modify the existing training process only where the currently estimated depth map is worse than the Depth Hint. A

自适应修改现有方法只处理深度估计比Depth Hint差的地方
DH由第三方立体方法得到的深度图

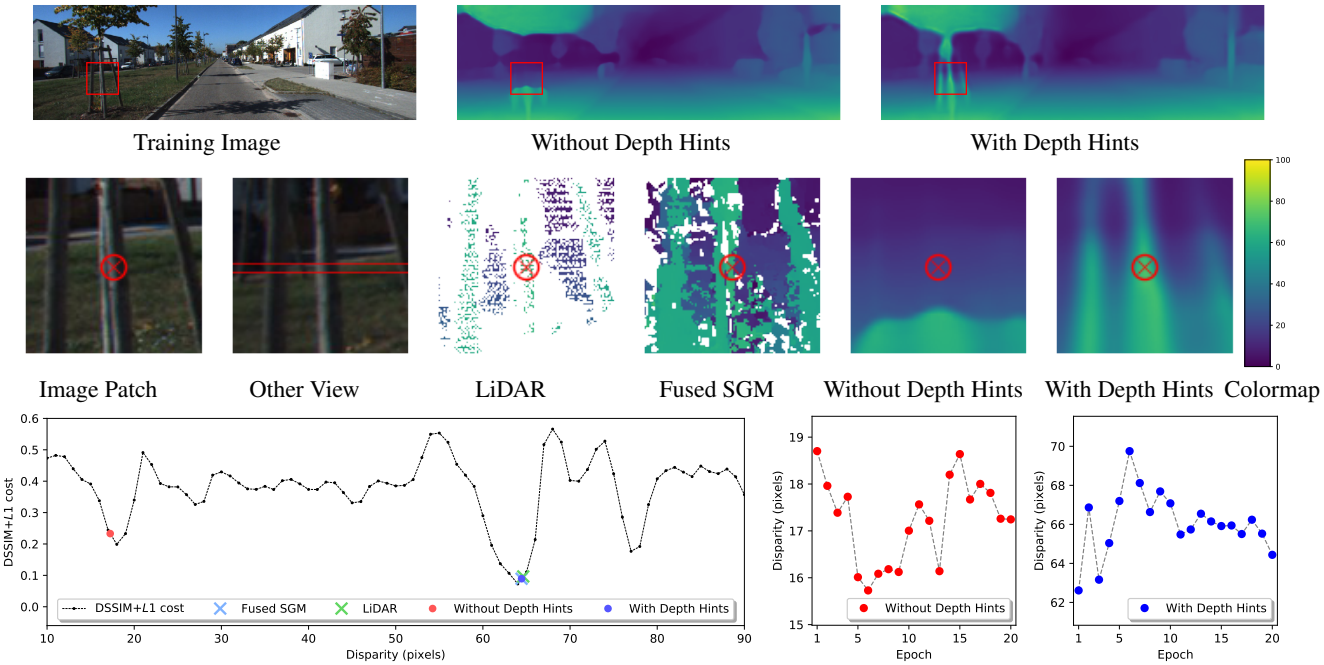

Figure 1. **Top row:** Image from the training set and corresponding depth maps produced by neural networks trained without and with Depth Hints (Godard *et al*. [10] (Monodepth2) architecture and loss). **Middle row, left to right:** Crop of the image centered around a thin structure with the center pixel circled, the scanline in the other image for the circled pixel, LiDAR pointcloud, fused depth map from SGM, crop of the depth map produced by a network trained without Depth Hints, our result, and the color coding illustrating pixel disparities. **Bottom row:** On the left is the plot of DSSIM+L_1 cost of the pixel on the thin structure for every pixel disparity. Plots on the right show the predictions made by the network after q epochs when trained with and without Depth Hints. The network trained without Depth Hints gets stuck in a local minimum and does not escape even after 20 epochs. On the other hand, the network trained with Depth Hints is in the vicinity of the correct solution (disparity of 64.63 according to LiDAR) after the first epoch. We visualize depths as disparities in pixel space for clarity. (Best viewed in color.)

Depth Hint is essentially a depth map estimated by a third-party binocular stereo algorithm.

5.1. Training from stereo pairs

只想从DH提供监督信号得到比网络预测更好的重投影图l

During training, we provide our network with a per-pixel Depth Hint, *i.e*. a potential alternative hypothesis to the network's own depth estimate. Our key idea is that we only want to provide a supervisory signal from the Depth Hints in places where they make for a superior reprojected image $\tilde{I}$, compared to using the network prediction. Else the hint is ignored. To be clear, the proposed objective is not learning to regress a map of hinted depth values. That would be a supervised loss, and is indeed one of our baselines. Interestingly, [7] explored that baseline and found it disappointing, because L_2 was in favor at the time. Rather, our objective remains to optimize a given algorithm's existing loss, and to consult a pixel's Depth Hint only when the reprojection loss can be improved upon.

In light of this, we reformulate our loss for pixel i as:

$$l_{ours}(d_i) = \begin{cases} l_r(d_i) + l_s^{\log L_1}(d_i, h_i) & \text{if } l_r(h_i) < l_r(d_i) \\ l_r(d_i) & \text{otherwise,} \end{cases} \quad (5)$$

r代表光度重投影损失，h代表DH，相当于简单地方用自监督，难的地方用gt做有监督的L1损失

for an inferred network depth d_i and a depth hint h_i, with an associated self-supervised loss function l_r from (2) judging the photometric quality of the depth estimate.

Computing Depth Hints: We propose to generate Depth Hints using stereo pairs. Depth Hints with perfect accuracy are unattainable, and it would be extremely expensive to sweep discrete per-pixel depth values to find those that generate the optimal DSSIM+L_1 reprojection. Instead, we use a standard heuristically-designed stereo method to compute depth. It is tempting to use a state-of-the-art stereo algorithm instead, *e.g*. [3, 2, 29], however most modern stereo algorithms are supervised using the LiDAR ground truth from the KITTI dataset. Using one of these would cause us to be implicitly learning from laser-scanned ground-truth data. Further, generating multiple depth maps is not trivial with most stereo methods.

Semi-Global Matching (SGM) [13, 14] is an off-the-shelf stereo matching algorithm available in OpenCV. SGM allows generation of different depth maps, depending on the hyperparameters used. For example, one can specify the size of the block to match between images, and the number of discrete disparities to evaluate. Hence, at training time, we can randomly choose hyperparameters for SGM to gen-

erate Depth Hints on the fly. We refer to such Depth Hints as "Random SGM." Alternatively, for each training image pair, we can generate a collection of depth maps by running SGM with every possible hyperparameter choice. We discretize this space into 12 parameter choices, formed of combinations of three block sizes with four resolutions of disparities. We call this version of Depth Hints "Fused SGM," because it checks that collection of depth maps and chooses the depth value at each pixel based on the DSSIM+L_1 score. Fused SGM Depth Hints are pre-computed just once for the training corpus. Unless specified, we use Fused SGM depths as hints in our models.

Finally, SGM's depth maps can contain holes where the matching cost is ambiguous. All losses associated with SGM's depth maps are set to infinity for such pixels.

5.2. Training from stereo video

网络预测2个位姿对应前后帧输入只有当前帧，位姿有3帧

We can also apply this same method in the stereo *video* self-supervised task, where training data is a video of binocular pairs. In addition to the depth prediction for the current frame at time t, the network also produces two camera poses for the forward $t+1$ and backward $t-1$ frames. The input to the depth prediction network is just the current frame t, while the pose prediction network is given 3 frames at times t, $t-1$ and $t+1$. Similarly to Godard *et al*. [10], we warp all three other views (other image of the stereo pair, forward frame and backward frame) into the reference viewpoint, and select the photometric reprojection loss as the minimum of the 3 associated losses at each pixel.

5.3. Implementation Details

Our network architecture and training regime closely follow Godard *et al*. [10], and can be viewed in the supplementary materials. Unless otherwise specified, we use Resnet-18 [12] as the encoder, pretrained on ImageNet [4], also following [10]. We specify the resolution of the input images explicitly, as it was shown to impact accuracy [10, 26].

Depth map post-processing [9] improves the quality of the final depth maps, so, for the quantitative results in Tables 1, 2 and 3, we add a "PP" column to indicate if post-processing was applied.

Due to GPU memory restrictions, some methods train the network with a random crop of the full resolution image *e.g*. as in DORN [6]. At test time, the full resolution image is tiled into suitable crops, then each crop is processed by the network and the depth maps are averaged to produce the full resolution output. Training on crops has the potential to improve most models, because the network processes more data and is able to 'see' finer details. We specify if a network was trained with crops instead of downsampling.

6. Experiments

Our validation consists of four sets of experiments, all exploring the task of training a CNN to predict depth from a single color image, using binocular stereo data instead of ground-truth labels. Depending on the experiment, we compare against known leading baselines that supplement, and pre- and post-process the input stereo pairs and output depths to various degrees. The four experiments are:

1. Section 6.1 illustrates that local minima exist when photometric reconstruction loss is used for self-supervision, and that Depth Hints can help.
2. Section 6.2 reports ablation-type experiments on Depth Hints, showing the negative impact of using the same SGM-computed stereo depths in more traditional loss functions.
3. Section 6.3 shows how Depth Hints usually help other modern self-supervised models.
4. Section 7 pits Depth Hints against other state of the art algorithms, grouped by preconditions.

We run experiments on the KITTI dataset [8] which consists of calibrated stereo video registered to LiDAR measurements of a city, captured from a moving car. The depth evaluation is done on the LiDAR pointcloud, and we report all seven of the standard metrics. See [5] for evaluation details, but broadly, lower numbers are better in red columns, while higher numbers are better in blue columns. To enable direct comparison with recent works, we use the Eigen split of KITTI [5] and evaluate with Garg's crop [7], using the standard cap of 80m [9]. We note that there are potential evaluation issues with the KITTI ground-truth data due to a translational offset between the color camera used to record images and the LiDAR scanner. In the supplementary material we also present some evaluations on the updated KITTI ground truth data provided by [31].

6.1. Solution with Depth Hints

The experiment described in Figure 1 is typical, showing that recent self-supervision approaches can get by without ground truth depths for most pixels, because a DSSIM+L_1 loss trains the CNN to regress reasonable depths. However, even seemingly distinct structures like a tree induce local minima that are plausible, and hard for the training process to escape. Supervised training with LiDAR data would yield an excellent photometric match, but in its absence, a Depth Hint can provide an alternative that our loss function (5) incorporates in a gradual way: the hint isn't trusted explicitly, and as training progresses, the hint may be ignored.

In experiments, the network initially makes use of Depth Hints for 85% of available pixels, dropping to 50% at the end of training.

Method	PP	H × W	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
l_{ps} Random SGM	✓	192 × 640	0.110	0.901	4.816	**0.193**	0.871	**0.958**	**0.981**
l_{ps} Random SGM LR	✓	192 × 640	0.109	0.877	4.800	**0.193**	0.870	**0.958**	**0.981**
l_{ps} Fused SGM	✓	192 × 640	0.109	0.850	4.741	**0.193**	0.873	0.956	0.980
l_{sum} Fused SGM	✓	192 × 640	0.108	0.841	4.754	0.194	0.871	0.957	0.980
l_{ps} Fused SGM → l_r	✓	192 × 640	0.109	0.916	4.910	0.203	0.866	0.952	0.977
Klodt [17] uncertainty	✓	192 × 640	0.108	0.905	4.815	0.196	0.871	0.955	0.979
Ours	✓	192 × 640	**0.106**	**0.780**	**4.695**	**0.193**	**0.875**	**0.958**	0.980

Table 1. **Ours *vs*. Baselines.** Comparison of baselines evaluated on KITTI 2015 [8] using the Eigen split. All methods here were trained on stereo pairs only.

6.2. Baseline Loss Functions

Besides our proposed loss in (5), there are various alternative strategies for incorporating Depth Hints in the objective function. Here we discuss such alternatives and compare them experimentally in Table 1.

First, we start with a simple baseline, where a neural network is trained to predict depth labels produced by an off-the-shelf stereo algorithm. This baseline is trained with loss

$$l_{ps}(d_i) = l_s^{\log L_1}(d_i, h_i), \tag{6}$$

where "ps" indicates proxy-supervised losses. Here h_i is estimated by the SGM algorithm. We train three baselines with this loss. The first uses depth maps generated on the fly with a random selection of hyperparameters (Random SGM) to avoid the influence of DSSIM+L_1 loss. The second baseline uses the same method, but with a left-right consistency check to reduce noise by invalidating pixels which have disagreeing depth values in the two views (Random SGM LR). The last baseline uses the single Fused SGM depth maps from Section 5.1 that give an indirect signal from the DSSIM+L_1 loss.

Another approach is to optimize the sum of self-supervised and supervised losses, so

$$l_{\text{sum}}(d_i) = l_r(d_i) + l_s^{\log L_1}(d_i, h_i). \tag{7}$$

This baseline is similar to the additional supervision from SLAM found in [17, 35]. Similarly, Zhu *et al*. [43] add a supervised loss [1] to solve for optical flow and Kuznietsov *et al*. [18] add a supervised loss for depth estimation from LiDAR. Concurrently proposed monoResMatch [30] uses this method to incorporate a proxy-supervised signal, albeit using a reverse Huber loss [19] as opposed to $\log L_1$. The addition of supervised losses change the objective function that is being minimized; one could view the additional term as a form of regularization, constraining the network prediction to adhere to the proposed depth values. However, this strategy can struggle to contend with noise in the depths estimated by stereo algorithms.

A different way of incorporating Depth Hints is to pretrain a network using l_{ps} on the fused Depth Hints and finetune using l_r. In Table 1, this method is denoted as "l_{ps} Fused SGM → l_r". We train l_{ps} for 10 epochs followed by l_r for another 10 epochs with the original learning rate.

Since the fused SGM depths may be a noisy estimate of depth, we could enable our model to train from them more robustly by explicitly modeling uncertainty [16, 17]. In these prior works, imperfections in the supervisory signal are modelled as part of the training loss; in addition to disparity, the network predicts a per-pixel data-dependent estimate of the residual error of the supervised loss. For pixels where the network expects that it will not be able to accurately satisfy the main training loss, it can pay a 'penalty' by predicting a higher residual error. This method was exploited by Klodt and Vedaldi [17] to make learning from potentially noisy SLAM depths and poses more robust.

Referring to Table 1, we note the clear benefit of treating the Depth Hints as noisy and only incorporating their estimates when they are superior to the network prediction. Surprisingly, our various baselines are competitive when compared to state of the art methods in Table 3. For example, even "l_{ps} Fused SGM" scores better than 3Net [27] and SuperDepth [26], and is highly competitive with Monodepth2 (S and MS) [10] on all metrics, albeit with pretraining.

6.3. Depth Hints for Existing Methods

Here we demonstrate the benefits of using Depth Hints to improve existing methods. As most existing methods do not provide training code, we have implemented a selection of them that are trained with self-supervised loss. Hence, we modify our loss functions to closely match the selected methods, while keeping our network architecture, image resolution, optimization parameters, and number of epochs consistent across experiments.

Table 2 shows quantitative results of existing methods that were augmented with Depth Hints. We see noticeable improvements in all methods which are trained using stereo (S) and stereo video (MS), demonstrating the effectiveness of incorporating Depth Hints. Additionally, we do not observe an improvement for the semi-supervised case [18], nor do the comparatively noisy Depth Hints hurt its results. Please see supplementary material for additional information regarding these implementations.

Finally, Depth Hints show substantial improvements when trained and evaluated on synthetic FlyingThings3D Sceneflow dataset [24]. The improvements are significant due to many objects with thin structures present in the

Cit.	Method	PP	Data	Dataset	H × W	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
[18]	Kuznietsov	✗	DS	K	192 × 640	0.109	**0.693**	**4.305**	**0.176**	0.878	**0.965**	**0.987**
[18]	Kuznietsov	✗	DS	K	192 × 640	**0.108**	**0.693**	4.312	**0.176**	**0.879**	**0.965**	0.986
[10]	Monodepth2 no pt	✗	S	K	192 × 640	0.129	1.102	5.440	0.232	0.829	0.933	0.969
[10]	Monodepth2 no pt	✗	S	K	192 × 640	**0.127**	**1.039**	**5.239**	**0.219**	**0.835**	**0.942**	**0.974**
[10]	Monodepth2	✗	S	K	192 × 640	0.110	0.896	4.986	0.208	0.866	0.948	0.975
[10]	Monodepth2	✗	S	K	192 × 640	**0.109**	**0.845**	**4.800**	**0.196**	**0.870**	**0.956**	**0.980**
[27]	3Net (Resnet18)	✗	S	K	192 × 640	**0.112**	0.953	5.007	0.207	0.862	0.949	**0.976**
[27]	3Net (Resnet18)	✗	S	K	192 × 640	**0.112**	**0.929**	**4.960**	**0.204**	**0.867**	**0.951**	**0.976**
[9]	Monodepth	✗	S	K	192 × 640	0.111	0.912	4.977	0.205	0.863	0.950	**0.977**
[9]	Monodepth	✗	S	K	192 × 640	**0.109**	**0.862**	**4.862**	**0.201**	**0.868**	**0.952**	**0.977**
[10]	Monodepth2	✗	MS	K	320 × 1024	0.106	0.806	4.630	0.193	0.876	0.958	0.980
[10]	Monodepth2	✗	MS	K	320 × 1024	**0.100**	**0.728**	**4.469**	**0.185**	**0.885**	**0.962**	**0.982**
[10]	Monodepth2	✗	S	SF	352 × 640	0.340	6.176	5.938	0.449	0.639	0.852	0.923
[10]	Monodepth2	✗	S	SF	352 × 640	**0.219**	**1.157**	**3.889**	**0.344**	**0.706**	**0.900**	**0.953**

Table 2. **Depth Hints with Existing Methods.** Comparison of our implementations of existing methods with and without Depth Hints. The data used to train/test is defined in the *Dataset* column, whereby 'K' is for KITTI 2015 [8] using the Eigen split, and 'SF' is for the FlyingThings3D Sceneflow dataset [23]. Highlighted methods are augmented with Depth Hints, and score better than their regular counterparts almost universally. [18] is an exception, possibly because it already uses LiDAR data. We also show results for [10] without ImageNet [4] pretraining, denoted as 'Monodepth2 no pt'. *Data column* (data source used for training): D refers to methods that use depth supervision at training time, S is for self-supervised training on stereo images, and MS is for models trained with stereo video.

dataset. These results demonstrate that Depth Hints can improve monocular depth estimation in various domains.

7. Depth From Color Tournament

Although it only represents one application domain, the KITTI dataset has been established as the dominant benchmark for measuring the accuracy of depth inferred from color. Broadly, our Depth Hints approach produces better looking results (see Figure 2) and scores indicating that we are the new state of the art across three major competition "categories." Please see Table 3. Of course there are more or less flattering ways to cluster the competition, so we present "Our" method in multiple forms, for better compatibility within each category. In doing so, we show that Depth Hints are useful across multiple settings (stereo *vs*. mono+stereo, low *vs*. high resolution, with/without pretraining), making the difference between first and second place.

Rows in Table 3 are color-coded by category, with the winning score for each of seven measures marked in **bold**.

Low-res Stereo is the classic category, with the longest history of competitors (we show the highest scorers). Our full method ("Ours Resnet50") wins decisively on every metric. One could argue about two "outside" advantages: we pre-train on Imagenet and our SGM step gets the benefit of a time-tested heuristic. Our ablation experiments in Sec 6.2 show the difference between using SGM naively and incorporating its output as a Depth Hint. For completeness, we present results for our method with no pretraining ("Ours Resnet50 w/o pretraining"). When we compare this to the highest scoring non pretrained network 3Net [27], we show better scores in all seven metrics.

High-res allows for processing of larger inputs. Again our method ("Ours HR Resnet50") shows a considerable improvement over existing methods in all metrics. Similar to before, we also show results for our method without pretraining ("Ours HR Resnet50 w/o pretraining"). Our non-pretrained model beats SuperDepth [26] in six out of seven metrics (tied in one), and compares favourably to the concurrent work monoResMatch [30], which makes use of a significantly more complex network compared to our encoder-decoder architecture.

Stereo Video MS could theoretically be the category with the strongest scores, because each self-supervised algorithm has access to time series movies (M) in stereo (S), with the opportunity to match occluded regions by searching elsewhere in time. Interestingly, in this category we see smaller improvements by using our approach over [10] for lower resolution ("Ours"), but observe a substantial boost in the high resolution case ("Ours HR").

Overall, we note that error metrics like SqRel and RMSE, which penalize large errors in a few pixels, benefit most from Depth Hints. Depth Hints help to recover thin structures and to more accurately delineate object boundaries (Figure 2). The AbsRel metric has smaller gains, since only a minority of pixels in each image are improved.

The **Depth Supervised** category is one we cannot compete in. The clear winner here is DORN [6], who avoid self-supervision entirely, training directly from LiDAR data. SVSM [21] uses outside synthetic data, and LiDAR data for finetuning. DVSO [35] obtains depth supervision through an excellent SLAM system, yielding LiDAR-like point-clouds, and combines them with self-supervision to achieve scores similar to ours in their "SimpleNet" model. However, their paper introduces an important enhancement that we lack, namely a depth refinement network.

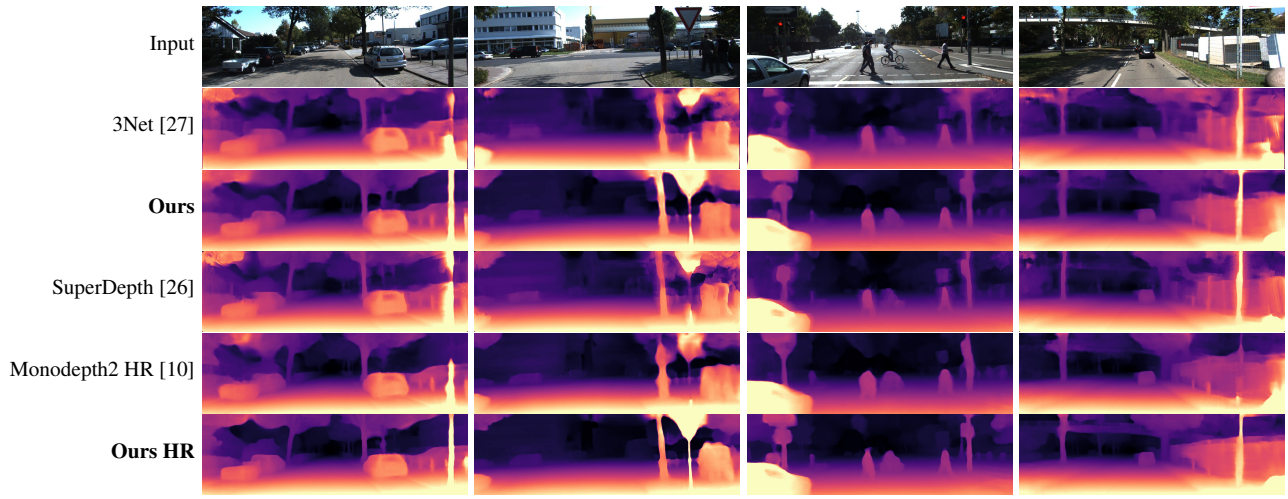

Figure 2. **Qualitative comparison with existing methods.** Top row: Four test set images. Each subsequent row: Depth maps generated by a stereo-only method. Notice how Ours and Ours HR capture thin structures such as traffic lights, traffic signs, lampposts, etc.

Cit.	Method	PP	Data	H × W	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
[18]	Kuznietsov		DS	187 × 621	0.113	0.741	4.621	0.189	0.862	0.960	0.986
[6]	DORN		D	385 × 513 crop	**0.072**	**0.307**	**2.727**	**0.120**	**0.932**	**0.984**	**0.994**
[35]	DVSO SimpleNet	✓	D†S	256 × 512	0.107	0.852	4.785	0.199	0.866	0.950	0.978
[35]	DVSO	✓	D†S	256 × 512	0.097	0.734	4.442	0.187	0.888	0.958	0.980
[11]	Guo StereoUnsupFT → Mono pt		D*S	256 × 512	0.099	0.745	4.424	0.182	0.884	0.963	0.983
[21]	SVSM w/o finetuning		D*S	192 × 640 crop	0.102	0.700	4.681	0.200	0.872	0.954	0.978
[11]	Guo StereoSupFTAll → Mono pt		D*DS	256 × 512	0.097	0.653	**4.170**	**0.170**	0.889	**0.967**	**0.986**
[21]	SVSM finetuned		D*DS	192 × 640 crop	**0.094**	**0.626**	4.252	0.177	**0.891**	0.965	0.984
[9]	Monodepth	✓	S	256 × 512	0.138	1.186	5.650	0.234	0.813	0.930	0.969
[25]	StrAT		S	256 × 512	0.128	1.019	5.403	0.227	0.827	0.935	0.971
[10]	Monodepth2 (w/o pretraining)	✓	S	192 × 640	0.128	1.089	5.385	0.229	0.832	0.934	0.969
[27]	3Net (Resnet50)	✓	S	256 × 512	0.126	0.961	5.205	0.220	0.835	0.941	0.974
	Ours Resnet50 w/o pretraining	✓	S	192 × 640	0.118	0.941	5.055	0.210	0.850	0.948	0.976
[10]	Monodepth2	✓	S	192 × 640	0.108	0.842	4.891	0.207	0.866	0.949	0.976
	Ours	✓	S	192 × 640	0.106	0.780	4.695	0.193	0.875	0.958	0.980
	Ours Resnet50	✓	S	192 × 640	**0.102**	**0.762**	**4.602**	**0.189**	**0.880**	**0.960**	**0.981**
[26]	SuperDepth	✓	S	384 × 1024	0.112	0.875	4.958	0.207	0.852	0.947	0.977
	Ours HR Resnet50 w/o pretraining	✓	S	320 × 1024	0.112	0.857	4.807	0.203	0.861	0.952	0.978
[30]	monoResMatch	✓	S	256 × 512 crop	0.111	0.867	4.714	0.199	0.864	0.954	0.979
[10]	Monodepth2	✓	S	320 × 1024	0.105	0.822	4.692	0.199	0.876	0.954	0.977
	Ours HR	✓	S	320 × 1024	0.099	0.723	4.445	0.187	0.886	**0.962**	**0.981**
	Ours HR Resnet50	✓	S	320 × 1024	**0.096**	**0.710**	**4.393**	**0.185**	**0.890**	**0.962**	**0.981**
[40]	Zhan	✗	MS	160 × 608	0.135	1.132	5.585	0.229	0.820	0.933	0.971
[20]	EPC++		MS	256 × 832	0.128	0.935	5.011	0.209	0.831	0.945	0.979
[10]	Monodepth2	✓	MS	192 × 640	**0.104**	0.786	4.687	0.194	**0.876**	0.958	0.980
	Ours	✓	MS	192 × 640	0.105	**0.769**	**4.627**	**0.189**	0.875	**0.959**	**0.982**
[10]	Monodepth2	✓	MS	320 × 1024	0.104	0.775	4.562	0.191	0.878	0.959	0.981
	Ours HR	✓	MS	320 × 1024	**0.098**	**0.702**	**4.398**	**0.183**	**0.887**	**0.963**	**0.983**

Table 3. **Quantitative results.** Adjusting our model slightly, we compare it to the top performers in three different categories on KITTI 2015 [8], using the Eigen split. *Data column* (data source used for training): D refers to methods that use KITTI depth supervision at training time, D* use auxiliary depth supervision from synthetic data, D† use auxiliary depth supervision from SLAM, S is for self-supervised training on stereo images, MS is for models trained with both M (forward and backward frames) and S data.

8. Conclusion

We investigated current issues with reprojection losses in the self-supervised monocular depth estimation setting. Based on these observations, we introduced Depth Hints as a practical approach to help escape from local minima, and to guide the network toward a better overall solution. The depth proposals make for a strong baseline themselves, but our training mechanism reverts to the default reprojection loss when the proposals are unhelpful. Qualitatively, Depth Hints seem to help most with thin structures and sharp boundaries. Extensive experimentation supports this. Further, Depth Hints provide a boost when applied to existing self-supervision schemes. Combined with a common network architecture, without but preferably with pre-training, our Depth Hints model achieves the top-scores on the self-supervised KITTI Eigen benchmark by a significant margin.

Acknowledgements

We would like to thank Aron Monszpart and Galen Han for helping to run our experiments, and our anonymous reviewers for their positive comments and helpful suggestions.

References

[1] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015.

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018.

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *arXiv:1810.02695*, 2018.

[4] Jia Deng, Wei Dong, Richard Socher, Li Li-Jia, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[6] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[7] Ravi Garg, Vijay Kumar BG, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.

[11] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, 2005.

[14] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 2008.

[15] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2018.

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[17] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning SFM from SFM. In *ECCV*, 2018.

[18] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, 2017.

[19] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[20] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3D holistic understanding. *arXiv:1810.06125*, 2018.

[21] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *CVPR*, 2018.

[22] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *CVPR*, 2018.

[23] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[24] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[25] Ishit Mehta, Parikshit Sakurikar, and P.J. Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *3DV*, 2018.

[26] Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *ICRA*, 2018.

[27] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, 2018.

[28] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019.

[29] Xiao Song, Xu Zhao, Liangji Fang, and Hanwen Hu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *arXiv:1903.01700*, 2019.

[30] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *CVPR*, 2019.

[31] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant CNNs. In *3DV*, 2017.

[32] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[33] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *ICCV*, 2017.

[34] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Transactions on Image Processing*, 2004.

[35] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *ECCV*, 2018.

[36] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. LEGO: Learning edge with geometry all at once by watching videos. In *CVPR*, 2018.

[37] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *AAAI*, 2018.

[38] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

[39] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Geometry meets semantic for semi-supervised monocular depth estimation. In *ACCV*, 2018.

[40] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, 2018.

[41] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *Transactions on Computational Imaging*, 2017.

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

[43] Yi Zhu, Zhen-Zhong Lan, Shawn D. Newsam, and Alexander G. Hauptmann. Guided optical flow learning. In *CVPR Workshops*, 2017.

[44] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018.