Supervising the new with the old: learning SFM from SFM

Maria Klodt[0000−0003−3015−9584] and Andrea Vedaldi[0000−0003−1374−2858]

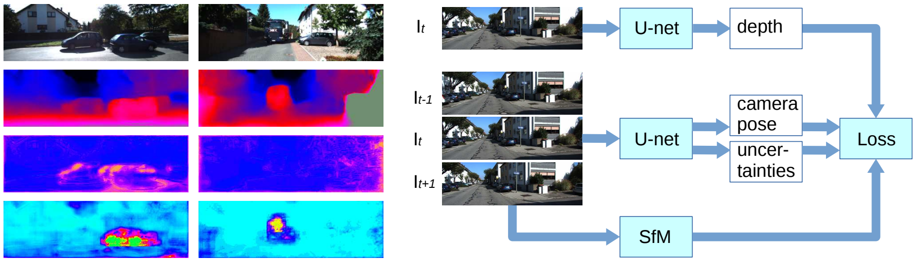
Visual Geometry Group, University of Oxford
{klodt,vedaldi}@robots.ox.ac.uk

Abstract. Recent work has demonstrated that it is possible to learn deep neural networks for monocular depth and ego-motion estimation from unlabelled video sequences, an interesting theoretical development with numerous advantages in applications. In this paper, we propose a number of improvements to these approaches. First, since such self-supervised approaches are based on the brightness constancy assumption, which is valid only for a subset of pixels, we propose a probabilistic learning formulation where the network predicts distributions over variables rather than specific values. As these distributions are conditioned on the observed image, the network can learn which scene and object types are likely to violate the model assumptions, resulting in more robust learning. We also propose to build on dozens of years of experience in developing handcrafted structure-from-motion (SFM) algorithms. We do so by using an off-the-shelf SFM system to generate a supervisory signal for the deep neural network. While this signal is also noisy, we show that our probabilistic formulation can learn and account for the defects of SFM, helping to integrate different sources of information and boosting the overall performance of the network.

1 Introduction

Visual geometry is one of the few areas of computer vision where traditional approaches have partially resisted the advent of deep learning. However, the community has now developed several deep networks that are very competitive in problems such as ego-motion estimation, depth regression, 3D reconstruction, and mapping. While traditional approaches may still have better absolute accuracy in some cases, these networks have very interesting properties in terms of speed and robustness. Furthermore, they are applicable to cases such as monocular reconstruction where traditional methods cannot be used.

A particularly interesting aspect of the structure-from-motion problem is that it can be used for bootstrapping deep neural networks without the use of manual supervision. Several recent papers have shown in fact that it is possible to learn networks for ego-motion and monocular depth estimation only by watching videos from a moving camera (SfMLearner [1]) or a stereo camera pair (MonoDepth [2]). These methods rely mainly on low-level cues such as brightness constancy and only mild assumptions on the camera motion. This is particularly



(a) RGB input image and predictions: depth, photometric uncertainty, and depth uncertainty.

(b) proposed network architecture: the depth and pose-uncertainty networks are supervised by traditional SfM.

Fig. 1. (a) Depth and uncertainty prediction on the KITTI dataset: In addition to monocular depth prediction, we propose to predict photometric and depth uncertainty maps in order to facilitate training from monocular image sequences. (b) Overview of the training data flow: Two convolutional neural networks are trained under the supervision of a traditional SfM method, and are combined via a joint loss including photo-consistency terms.

appealing as it allows to learn models very cheaply, without requiring specialized hardware or setups. This can be used to deploy cheaper and/or more robust sensors, as well as to develop sensors that can automatically learn to operate in new application domains.

In this paper, we build on the SfMLearner approach and consider the problem of learning from scratch a neural network for ego-motion and monocular depth regression using only unlabelled video data from a single, moving camera. Compared to SfMLearner and similar approaches, we contribute three significant improvements to the learning formulation that allows the method to learn better models.

Our first and simplest improvement is to strengthen the brightness constancy loss, importing the structural similarity loss used in MonoDepth in the SfM-Learner setup. Despite its simplicity, this change does improve results.

Our second improvement is to incorporate an explicit model of confidence in the neural network. SfMLearner predicts an “explainability map” whose goal is to identify regions in an image where the brightness constancy constraint is likely to be well satisfied. However, the original formulation is heuristic. For example, the explainability maps must be regularized ad-hoc to avoid becoming degenerate. We show that much better results can be obtained by turning explainability into a proper probabilistic model, yielding a self-consistent formulation which measures the likelihood of the observed data. In order to do so, we predict for each pixel a distribution over possible brightnesses, which allows the model to express a degree of confidence on how accurately brightness constancy will be satisfied at a certain image location. For example, this model can learn to expect slight misalignments on objects such as tree branches and cars that could move independently of the camera.

Our third improvement is to integrate another form of cheap supervision in the process. We note that the computer vision community has developed in the past 20 years a treasure trove of high-quality handcrafted structure-from-motion methods (SFM). Thus, it is natural to ask whether these algorithms can be used to teach better deep neural networks. In order to do so, during training we propose to run, in parallel with the forward pass of the network, a standard SFM method. We then require the network to optimize the brightness constancy equation as before *and* to match motion and depth estimates from the SFM algorithm, in a multi-task setting.

Ideally, we would like the network to ultimately perform *better* than traditional SFM methods. The question, then, is how can such an approach train a model that outperforms the teacher. There is clearly an opportunity to do so because, while SFM can provide very high-quality supervision when it works, it can also fail badly. For example, feature triangulation may be off in correspondence of reflections, resulting in inconsistent depth values for certain pixels. Thus, we adopt a probabilistic formulation for the SFM supervisory signal as well. This has the important effect of allowing the model to *learn when and to which extent it can trust the SFM supervision.* In this manner, the deep network can learn failure modalities of traditional SFM, and discount them appropriately while learning.

While we present such improvements in the specific context of 3D reconstruction, we note that the idea of using probabilistic predictions to integrate information from a collection of imperfect supervisory signals is likely to be broadly applicable.

We test our method against SfMLearner, the state of the art in this setting, and show convincing improvements due to our three modifications. The end result is a system that can learn an excellent monocular depth and ego-motion predictor, all without any manual supervision.

2 Related Work

Structure from motion is a well-studied problem in Computer Vision. Traditional approaches such as ORB-SLAM2 [3,4] are based on a pipeline of matching feature points, selecting a set of inlier points, and optimizing with respect to 3D points and camera positions on these points. Typically, the crucial part of these methods is a careful selection of feature points [5–8].

More recently, deep learning methods have been developed for learning 3D structure and/or camera motion from image sequences. In [9] a supervised learning method for estimating depth from a single image has been proposed. For supervision, additional information is necessary, either in form of manual input or as in [9], laser scanner measurements. Supervised approaches for learning camera poses include [10–12].

Unsupervised learning avoids the necessity of additional input by learning from RGB image sequences only. The training is guided by geometric and photometric consistency constraints between multiple images of the same scene. It has been shown that dense depth maps can be robustly estimated from a single image by unsupervised learning [2, 13], and furthermore, depth and camera poses [14]. While these methods perform single image depth estimation, they use stereo image pairs for training. This facilitates training, due to a fixed relative geometry between the two stereo cameras and simultaneous image acquisition yielding a static scene.

A more difficult problem is learning structure from motion from monocular image sequences. Here, depth and camera position have to be estimated simultaneously, and moving objects in the scene can corrupt the overall consistency with respect to the world coordinate system. A method for estimating and learning structure from motion from monocular image sequences has been proposed in SfMLearner [1]. Unsupervised learning can be enhanced by supervision in cases where ground truth is partially available in the training data, as has been shown in [15]. Results from traditional SfM methods can be used to guide other methods like 3D localization [16] and prediction of occlusion models [17].

Uncertainty learning for depth and camera pose estimation have been investigated in [18, 19] where different types of uncertainties have been investigated for depth map estimation, and in [20] where uncertainties for partially reliable ground truths have been learned.

3 Method

Let $\mathbf{x}_t \in \mathbb{R}^{H \times W \times 3}$, $t \in \mathbb{Z}$ be a video sequence consisting of RGB images captured from a moving camera. Our goal is to train two neural networks. The first $\mathbf{d} = \Phi_{\text{depth}}(\mathbf{x}_t)$ is a *monocular depth estimation network* producing as output a depth map $\mathbf{d} \in \mathbb{R}^{H \times D}$ from a single input frame. The second $(R_t, T_t : t \in \mathcal{T}) = \Phi_{\text{ego}}(\mathbf{x}_t : t \in \mathcal{T})$ is an *ego-motion and uncertainty estimation network*. It takes as input a short time sequence $\mathcal{T} = (-T, \ldots, 0, \ldots, T)$ and estimates 3D camera rotations and translations (R_t, T_t), $t \in \mathcal{T}$ for each of the images $\mathbf{x}_t$ in the sequence. Additionally, it predicts the pose uncertainty, as well as photometric and depth uncertainty maps which help the overall network to learn about outliers and noise caused by occlusions, specularities and other modalities that are hard to handle.

Learning the neural networks Φ_{depth} and Φ_{ego} from a video sequence without any other form of supervision is a challenging task. However, methods such as SfMLearner [1] have shown that this task can be solved successfully using the brightness constancy constraint as a learning cue. We improve over the state of the art in three ways: by improving the photometric loss that captures brightness constancy (section 3.1), by introducing a more robust probabilistic formulation for the observations (section 3.2) and by using the latter to integrate cues from off-the-shelf SFM methods for supervision (section 3.3).

3.1 Photometric losses

The most fundamental supervisory signal to learn geometry from unlabelled video sequences is the *brightness constancy constraint.* This constraint simply states that pixels in different video frames that correspond to the same scene point must have the same color. While this is only true under certain conditions (Lambertian surfaces, constant illumination, no occlusions, etc.), SfMLearner and other methods have shown it to be sufficient to learn the ego-motion and depth reconstruction networks Φ_{ego} and Φ_{depth}. In fact, the output of these networks can be used to put pixels in different video frames in correspondence and test whether their color match. This intuition can be easily captured in a loss, as discussed below.

Basic photometric loss. Let $\mathbf{d}_0$ be the depth map corresponding to image $\mathbf{x}_0$. Let $(u,v) \in \mathbb{R}^2$ be the calibrated coordinate of a pixel in image $\mathbf{x}_0$ (so that $(0,0)$ is the optical centre and the focal length is unit). Then the coordinates of the 3D point that projects onto (u,v) are given by $\mathbf{d}(u,v)\cdot(u,v,1)$. If the roto-translation (R_t,T_t) is the motion of the camera from time 0 to time t and $\pi(q_1,q_2,q_3) = (q_1/q_3, q_2/q_3)$ is the perspective projection operator, then the corresponding pixel in image $\mathbf{x}_t$ is given by $(u',v') = g(u,v|\mathbf{d},R_t,T_t) = \pi(R_t\mathbf{d}(u,v)(u,v,1)^\top + T_t)$. Due to brightness constancy, the colors $\mathbf{x}_0(u,v) = \mathbf{x}_t(g(u,v|\mathbf{d},R_t,T_t))$ of the two pixels should match. We then obtain the photometric loss:

$$\mathcal{L} = \sum_{t\in\mathcal{T}-\{0\}} \sum_{(u,v)\in\Omega} |\mathbf{x}_t(g(u,v|\mathbf{d},R_t,T_t)) - \mathbf{x}_0(u,v)| \tag{1}$$

where Ω is a discrete set of image locations (corresponding to the calibrated pixel centres). The absolute value is used for robustness to outliers.

All quantities in eq. (1) are known except depth and camera motion, which are estimated by the two neural networks. This means that we can write the loss as a function:

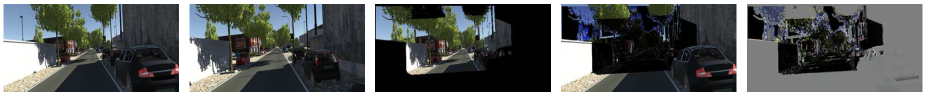
$$\mathcal{L}(\mathbf{x}_t : t \in \mathcal{T}|\Phi_{\text{depth}}, \Phi_{\text{ego}})$$

This expression can then be minimized w.r.t. Φ_{depth} and Φ_{ego} to learn the neural networks.

Structural-similarity loss. Comparing pixel values directly may be too fragile. Thus, we complement the simple photometric loss (1) with the more advanced image matching term used in [2] for the case of stereo camera pairs. Given a pair of image patches $\mathbf{a}$ and $\mathbf{b}$, their *structural similarity* [21] $\text{SSIM}(\mathbf{a},\mathbf{b}) \in [0,1]$ is given by:

$$\text{SSIM}(\mathbf{a},\mathbf{b}) = \frac{(2\mu_{\mathbf{a}}\mu_{\mathbf{b}})(\sigma_{\mathbf{ab}}+\epsilon)}{(\mu_{\mathbf{a}}^2+\mu_{\mathbf{b}}^2)(\sigma_{\mathbf{a}}^2+\sigma_{\mathbf{b}}^2+\epsilon)}$$

where ϵ is a small constant to avoid division by zero for constant patches, $\mu_{\mathbf{a}} = \frac{1}{n}\sum_{i=1}^n a_i$ is the mean of patch $\mathbf{a}$, $\sigma_{\mathbf{a}}^2 = \frac{1}{n-1}\sum_{i=1}^n (a_i - \mu_{\mathbf{a}})^2$ is its variance, and $\sigma_{\mathbf{ab}} = \frac{1}{n-1}\sum_{i=1}^n (a_i-\mu_{\mathbf{a}})(b_i-\mu_{\mathbf{b}})$ is the correlation of the two patches.



(a) I_t (target img) (b) I_{t+1} (source img) (c) I^w_{t+1} (warped) (d) ℓ_1 matching (e) SSIM matching

Fig. 2. Image matching: The photometric loss terms penalize high values in the ℓ_1 difference (d) and SSIM image matching (e) of the target image (a) and the warped source image (c).

This means that the combined structural similarity and photometric loss can be written as $\mathcal{L} = \sum_{(u,v)\in\Omega} \ell(u,v|\mathbf{x},\mathbf{x}')$ where

$$\ell(u,v|\mathbf{x},\mathbf{x}') = \alpha\frac{1-\mathrm{SSIM}(\mathbf{x}|_{\Theta(u,v)},\mathbf{x}'|_{\Theta(u,v)})}{2} + (1-\alpha)|\mathbf{x}(u,v)-\mathbf{x}'(u,v)|. \quad (2)$$

The weighting parameter α is set to 0.85.

Multi-scale loss and regularization. Figure 2 shows an example for ℓ_1 and SSIM image matching, computed from ground truth depth and poses for two example images of the Virtual KITTI data set [22]. Even with ground truth depth and camera poses, a perfect image matching cannot be guaranteed.

Hence, for added robustness, eq. (2) is computed at multiple scales. Further robustness is achieved by a suitable smoothness term for regularizing the depth map which is added to the loss function, as in [2].

3.2 Probabilistic outputs

The brightness constancy constraint fails whenever one of its several assumptions is violated. In practice, common failure cases include occlusions, changes in the field of view, moving objects in the scene, and reflective materials. The key idea to handle such issues is to allow the neural network to *learn to predict such failure modalities.* If done properly, this has the important benefit of extracting as much information as possible from the imperfect supervisory signal while avoiding being disrupted by outliers and noise.

General approach. Consider at first a simple case in which a predictor estimates a quantity $\hat{y} = \Phi(x)$, where x is a data point and y its corresponding "ground-truth" label. In a standard learning formulation, the predictor Φ would be optimized to minimize a loss such as $\ell = |\hat{y} - y|$. However, if we knew that for this particular example the ground truth is not reliable, we could down-weight the loss as ℓ/σ by dividing it by a suitable coefficient σ. In this manner, the model would be less affected by such noise.

The problem with this idea is how to set the coefficient σ. For example, optimizing it to minimize the loss does not make sense as this has the degenerate solution $\sigma = +\infty$.

An approach is to make σ one of the quantities *predicted by the model* and use it in a probabilistic output formulation. To this end, let the neural network output the parameters $(\hat{y}, \sigma) = \varPhi(x)$ of a posterior probability distribution $p(y|\hat{y}, \sigma)$ over possible "ground-truth" labels y. For example, using Laplace's distribution:

$$p(y|\hat{y}, \sigma) = \frac{1}{2\sigma} \exp \frac{-|y - \hat{y}|}{\sigma}.$$

The learning objective is then the negative log-likelihood arising from this distribution:

$$-\log p(y|\hat{y}, \sigma) = \frac{|y - \hat{y}|}{\sigma} + \log \sigma + \text{const.}$$

A predictor that minimises this quantity will try to guess $\hat{y}$ as close as possible to y. At the same time, it will try to set σ to the *fitting error it expects.* In fact, it is easy to see that, for a fixed $\hat{y}$, the loss is minimised when $\sigma = |y - \hat{y}|$, resulting in a log-likelihood value of

$$-\log p(y|\hat{y}, |y - \hat{y}|) = \log |y - \hat{y}| + \text{const.}$$

Note that the model is incentivized to learn σ to reflect as accurately as possible the prediction error. Note also that σ may resemble the threshold in a robust loss such as Huber's. However, there is a very important difference: it is the predictor itself that, after having observed the data point x, estimates on the fly an optimal data-dependent "threshold" σ. This allows the model to perform introspection, thus potentially discounting cases that are too difficult to fit. It also allows the model to learn, and compensate for, cases where the supervisory signal y itself may be unreliable. Furthermore this probabilistic formulation does not have any tunable parameter.

Implementation for the photometric loss. For the photometric loss (2), the model above is applied by considering an additional output $(\sigma_t)_{t \in \mathcal{T} - \{0\}}$ to the network $\varPhi_{\text{ego}}$, to predict, along with the depth map $\mathbf{d}$ and poses (R_t, T_t), an uncertainty map σ_t for photometric matching at each pixel. Then the loss is given by

$$\sum_{t \in \mathcal{T} - \{0\}} \sum_{(u,v) \in \varOmega} \frac{\ell(u, v|\mathbf{x}_0, \mathbf{x}_t \circ g_t)}{\sigma_t(u, v)} + \log \sigma_t(u, v),$$

where ℓ is given by eq. (2) and $g_t(u, v) = g(u, v|\mathbf{d}, R_t, T_t)$ is the warp induced by the estimated depth and camera pose.

3.3 Learning SFM from SFM

In this section, we describe our third contribution: learning a deep neural network that distills as much information as possible from a classical (handcrafted) method for SFM. To this end, for each training subsequence $(\mathbf{x}_t : t \in \mathcal{T})$ a standard high-quality SFM pipeline such as ORB-SLAM2 is used to estimate a depth map $\bar{\mathbf{d}}$ and camera motions $(\bar{R}_t, \bar{T}_t)$. This information can be easily used to supervise the deep neural network by adding suitable losses:

$$\mathcal{L}_{\text{SFM}} = \|\bar{\mathbf{d}} - \mathbf{d}\|_1 + \|\ln \bar{R}_t R_t^\top\|_F + \|\bar{T}_t - T_t\|_2 \tag{3}$$

Here ln denotes the principal matrix logarithm, which maps the residual rotation to its Lie group coordinates which provides a natural metric for small rotations.

While standard SFM algorithms are usually reliable, they are far from perfect. This is particularly true for the depth map $\bar{\mathbf{d}}$. First, since SFM is based on matching discrete features, $\bar{\mathbf{d}}$ will not contain depth information for all image pixels. While missing information can be easily handled in the loss, a more challenging issue is that triangulation will sometimes result in incorrect depth estimates due for example to highlights, objects moving in the scene, occlusion, and other challenging visual effects.

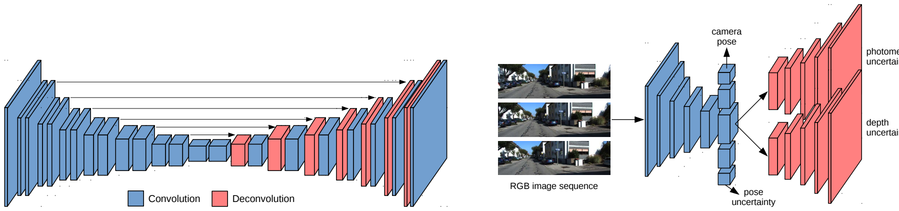
In order to address these issues, as well as to automatically balance the losses in a multi-task setting [19], we propose once more to adopt the probabilistic formulation of section 3.2. Thus loss (3) is replaced with

$$\begin{aligned}\mathcal{L}_{\text{SFM}}^p = \chi_{\text{SFM}} \Bigg[&\sum_{t \in \mathcal{T} - \{t\}} \left[\frac{\|\ln \bar{R}_t R_t^\top\|_F}{\sigma_{\text{SFM}}^{R_t}} + \log \sigma_{\text{SFM}}^{R_t} + \frac{\|\lambda_T \bar{T}_t - T_t\|_2}{\sigma_{\text{SFM}}^{T_t}} + \log \sigma_{\text{SFM}}^{T_t} \right] \\ &+ \sum_{(u,v) \in S} \left[\frac{|(\lambda_{\mathbf{d}} \bar{\mathbf{d}}(u,v))^{-1} - (\mathbf{d}(u,v))^{-1}|}{\sigma_{\text{SFM}}^{\mathbf{d}}(u,v)} + \log \sigma_{\text{SFM}}^{\mathbf{d}}(u,v) \right] \Bigg]\end{aligned} \tag{4}$$

where pose uncertainties $\sigma_{\text{SFM}}^R, \sigma_{\text{SFM}}^T$ and pixel-wise depth uncertainty maps $\sigma_{\text{SFM}}^{\mathbf{d}}$ are also estimated as output of the neural network Φ_{ego} from the video sequence. $S \in \Omega$ is a sparse subset of pixels where depth supervision is available.

The translation and depth values from SFM are multiplied by scalars $\lambda_T = \sum_t \|T_t\| / \sum_t \|\bar{T}_t\|$ and $\lambda_{\mathbf{d}} = \text{median}(\mathbf{d})/\text{median}(\bar{\mathbf{d}})$, respectively, because of the scale ambiguity which is inherent in monocular SFM. Furthermore, the binary variable χ_{SFM} denotes whether a corresponding reconstruction from SFM is available. This allows to include training examples where traditional SFM fails to reconstruct pose and depths. Note that we measure the depth error using inverse depth, in order to get a suitable domain of error values. Thus, small depth values, which correspond to points that are close to the camera, get higher importance in the loss function, and far away points, which are often more unreliable, are down-weighted.

Just as for supervision by the brightness constancy, this allows the neural network to learn about systematic failure modes of the SFM algorithm. Supervision can then avoid to be overly confident about this supervisory signal, resulting in a system which is better able to distill the useful information while discarding noise.



(a) Depth network layers (b) Pose and uncertainty network layers

Fig. 3. Network architecture: (a) Depth network: The network takes a single RGB image as input and estimates pixel-wise depth through 29 layers of convolution and deconvolution. Skip connections between encoder and decoder allow to recover fine-scale details. (b) Pose and uncertainty network: Input to the network is a short image sequence of variable length. The fourfold output shares a common encoder and splits to pose estimation, pose uncertainty and the two uncertainty maps afterwards. While photometric uncertainty estimates confidence in the photometric image matching, depth uncertainty estimates confidence in depth supervision from SfM.

4 Architecture learning and details

Section 3 discussed two neural networks, one for depth estimation (Φ_{depth}) and one for ego-motion and prediction confidence estimation (Φ_{ego}). This section provides the details of these networks. An overview of the network architecture and training data flow with combined pose and uncertainty networks is shown in fig. 1 (b). First, we note that, while two different networks are learned, in practice the pose and uncertainty nets share the majority of their parameters. As a trunk, we consider a U-net [23] architecture similar to the ones used in Monodepth [2] and SfMLearner [1].

Fig. 3 (a) shows details of the layers of the deep network. The network consists of an encoder and a decoder. The input is a single RGB image, and the output is a map of depth values for each pixel. The encoder is a concatenation of convolutional layers followed by ReLU activations where layers' resolution progressively decreases and the number of feature channels progressively increases. The decoder consists of concatenated deconvolution and convolution layers, with increasing resolution. Skip connections link encoder layers to decoder layers of corresponding size, in order to be able to represent high-resolution details. The last four convolution layers further have a connection to the output layers of the network, with sigmoid activations.

Fig. 3 (b) shows details of the pose and uncertainty network layers. The input of the network is an image sequence consisting of the target image I_t, which is also the input of the depth network, and n neighboring views before and after I_t in the sequence $\{I_{t-n}, \dots, I_{t-1}\}$ and $\{I_{t+1}, \dots, I_{t+n}\}$, respectively. The output of the network is the relative camera pose for each neighboring view with respect to the target view, two uncertainty values for the rotation and translation, respectively, and pixel-wise uncertainties for photo-consistency and depth.

	error measures			accuracy		
	abs. rel.	sq. rel.	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
SfMLearner (paper)	0.208	1.768	6.856	0.678	0.885	0.957
SfMLearner (website)	0.183	1.595	6.709	0.734	0.902	0.959
SfMLearner (reproduced)	0.198	2.423	6.950	0.732	0.903	0.957
+ image matching	0.181	2.054	6.771	0.763	0.913	0.963
+ photometric uncertainty	0.180	1.970	6.855	0.765	0.913	0.962
+ pose from SFM	0.171	1.891	6.588	0.776	**0.919**	0.963
+ pose and depth from SFM	**0.166**	**1.490**	**5.998**	**0.778**	**0.919**	**0.966**
ours, trained on VK	0.270	2.343	7.921	0.546	0.810	0.926
ours, trained on CS	0.254	2.579	7.652	0.611	0.857	0.942
ours, trained on CS+K	0.165	1.340	5.764	0.784	0.927	0.970

Table 1. Depth evaluation in comparison to SfMLearner: We evaluate the three contributions image matching, photometric uncertainty, and depth and pose from SfM. Each of these show an improvement to the current state of the art. Training datasets are KITTI (K), Virtual KITTI (VK) and Cityscapes (CS). Rows 1–7 trained on KITTI.

The different outputs share a common encoder, which consists of convolution layers, each followed by a ReLU activation. The pose output is of size $2n \times 6$, representing a 6 DoF relative pose for each source view, each consisting of a 3D translation vector and 3 Euler angles representing the camera rotation matrix, as in [1]. The uncertainty output is threefold, consisting of pose, photometric, and depth uncertainty. The pose uncertainty shares weights with the pose estimation, and yields a $2n \times 2$ output representing translational and rotational uncertainty for each source view. The pixel-wise photometric and depth uncertainties each consist of a concatenation of deconvolution layers of increasing width. All uncertainties are activated by a sigmoid activation function.

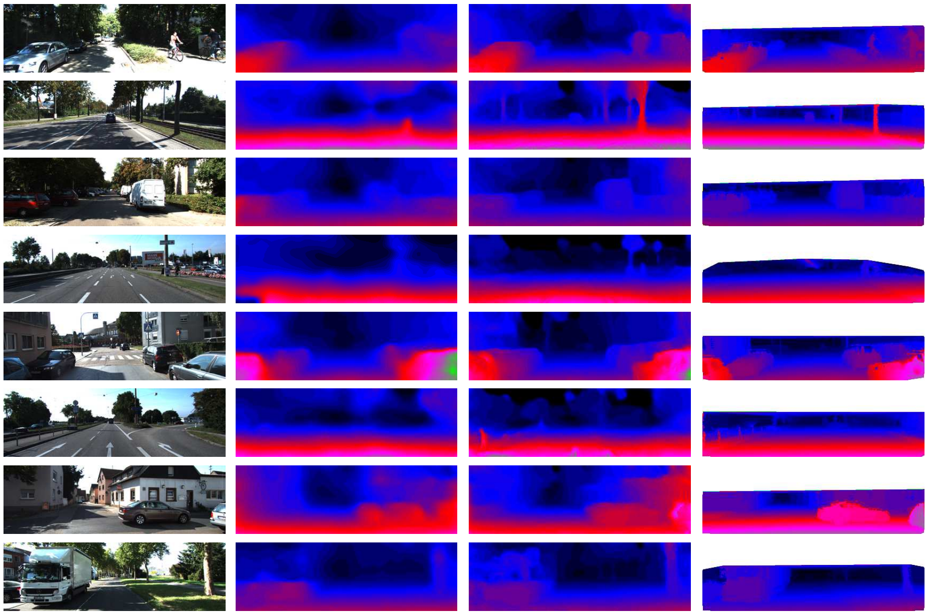
A complete description of the network architecture is provided in the supplementary material.

5 Experiments

We compare results of the proposed method to SfMLearner [1] which is the only method to our knowledge which estimates monocular depth and relative camera poses from monocular training data only. The experiments show that our method achieves better results that SfMLearner.

5.1 Monocular depth estimation

For training and testing monocular depth we use the Eigen split of the KITTI raw dataset [24] as proposed by [9]. This yields a split of 39835 training images, 4387 for validation, and 697 test images. We only use monocular sequences for training. Training is performed on sequences of three images, where depth is estimated for the centre image.



(a) test image (b) SfMLearner (c) proposed method (d) ground truth

Fig. 4. Comparison to SfMLearner and ground truth on test images from KITTI.

The state of the art in learning depth maps from a single image using monocular sequences for training only, is SfMLearner [1]. Therefore we compare to this method in our experiments. The laser scanner measurements are used as ground truth for testing only. The predicted depth maps are multiplied by a scalar $s = \text{median}(\mathbf{d}^*)/\text{median}(\mathbf{d})$ before evaluation. This is done in the same way as in [1], in order to resolve scale ambiguity which is inherent to monocular SfM.

Table 1 shows a quantitative comparison of SfMLearner with the different contributions of the proposed method. We compute the error measures used in [9] to compare predicted depth $\mathbf{d}$ with ground truth depth $\mathbf{d}^*$:

- Absolute relative difference (abs. rel.): $\frac{1}{N}\sum_{i=1}^{N} |\mathbf{d}_i - \mathbf{d}_i^*|/\mathbf{d}_i^*$
- Squared relative difference (sq. rel.): $\frac{1}{N}\sum_{i=1}^{N} |\mathbf{d}_i - \mathbf{d}_i^*|^2/\mathbf{d}_i^*$
- Root mean square error (RMSE): $\left(\frac{1}{N}\sum_{i=1}^{N} |\mathbf{d}_i - \mathbf{d}_i^*|^2\right)^{1/2}$

The accuracy measures are giving the percentage of $\mathbf{d}_i$ s.t. $\max\left(\mathbf{d}_i/\mathbf{d}_i^*, \mathbf{d}_i^*/\mathbf{d}_i\right) = \delta$ is less than a threshold, where we use the same thresholds as in [9].

We compare to the error measures given in [1], as well as to a newer version of SfMLearner provided on the website[1]. We also compare to running the code downloaded from this website, as we got slightly different results. We use this

[1] https://github.com/tinghuiz/SfMLearner

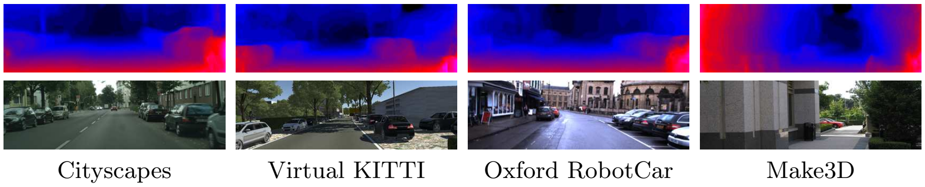


Fig. 5. Training on KITTI and testing on different datasets yields visually reasonable results.

as baseline for our method. These evaluation results are shown in rows 1–3 of table 1. Rows 4–7 refer to our implementation as described in section 3, while changes referred to in each row add to the previous row. The results show that structural similarity based image matching gives an improvement to the brightness constancy loss as used in SfMLearner. The photometric uncertainty is able to improve accuracy while giving slightly worse results on the RMSE, as the method is able to allow for higher errors in parts of the image domain. A more substantial improvement is obtained by adding pose and depth supervision from SFM. In these experiments we used in particular predictions from ORB-SLAM2 [4]. Numbers in bold indicate best performance for training on KITTI. The last three rows show results on the same test set (KITTI eigen split), for the final model with pose and depth from SfM, trained on Virtual KITTI (VK) [22], Cityscapes (CS) [25], and pre-training on Cityscapes with fine-tuning on KITTI (CS+K).

Figure 4 shows a qualitative comparison of depth predicted by SfMlearner against ground truth measurements from a laser scanner. Since the laser scanner measurements are sparse, we densify them for better visualization. While SfMLearner robustly estimates depth, our proposed approach is able to recover many more small-scale details from the images. The last row shows a typical failure case, where the estimated depth is less accurate on regions like car windows. Figure 5 shows a qualitative evaluation of depth prediction for different datasets. The model trained on KITTI was tested on images from Cityscapes [25], Virtual KITTI [22], Oxford RobotCar [26] and Make3D [27], respectively. Test images were cropped to match the ratio of width and height of the KITTI training data. These results show that the method is able to generalize to unknown scenarios and camera settings.

5.2 Uncertainty estimation

Figure 6 shows example visualizations of the photometric and depth uncertainty maps for some of the images from the KITTI dataset. The color bar indicates high uncertainty at the top and low uncertainty at the bottom. We observe that high photometric uncertainty typically occurs in regions with vegetation, where matching is hard due to repetitive structures, and in regions with specularities which corrupt the brightness constancy assumption, for example car windows

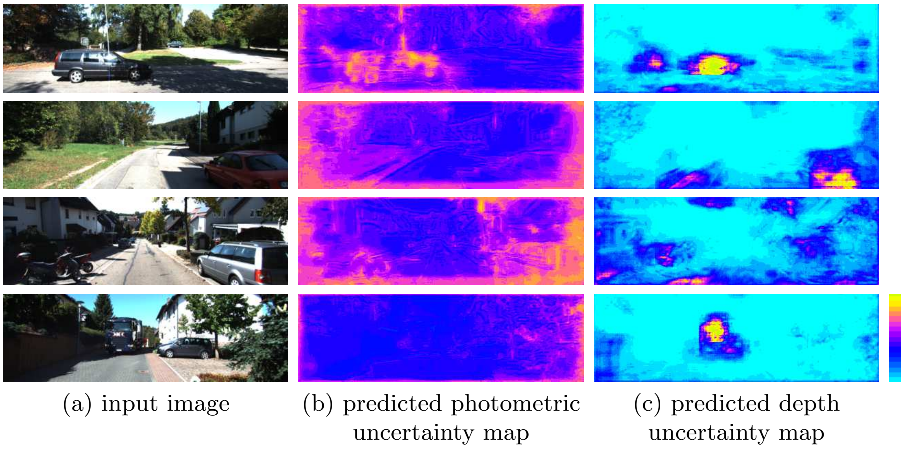

(a) input image
(b) predicted photometric uncertainty map
(c) predicted depth uncertainty map

Fig. 6. Prediction of uncertainty maps: the pixel-wise estimated uncertainty maps allow for higher errors in the image matching at regions with high uncertainty, leading to improved overall network performance. We observe that the photometric uncertainty maps (b) tend to predict high uncertainty for reflective surfaces, lens flares, vegetation, and at the image borders, as these induce high photometric errors when matching subsequent frames. The depth uncertainty maps (c) tend to predict high uncertainties for potentially moving objects, and the sky, where depth values are less reliable. The network seems to be able to discern between moving and stationary cars.

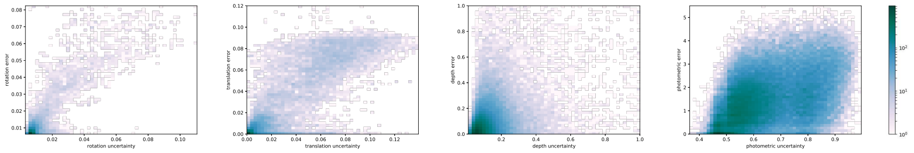
or lens flares. High depth uncertainty occurs typically on moving object as for example cars. We further observe that the network often seems to be able to discern between moving and stationary cars.

Figure 7 shows rotational, translational, depth and photometric uncertainty versus their respective error. The plots show that uncertainties tend to be lower in regions with good matching, and worse in regions with less good matching.

5.3 Camera pose estimation

We trained and tested the proposed method on the KITTI odometry dataset [28], using the same split of training and test sequences as in [1]: sequences 00–08 for training and sequences 09–10 for testing, using the left camera images of all sequences only. This gives a split of 20409 training images and 2792 test images. The ground truth odometry provided in the KITTI dataset is used for evaluation purposes only. Again, depth and pose from SFM are obtained from ORB-SLAM2 [4].

Table 2 shows a comparison to SfMLearner with numbers as given in the paper and on the website for the two test sequences 09 and 10. For odometry evaluation, a sequence length of 5 images has been used for training and testing. The error measure is the Absolute Trajectory Error (ATE) [29] on the 5-frame snippets, which are averaged on the whole sequence. The same error measure was used in [1]. We compare results from SfMLearner as stated in the paper and



(a) rotation uncertainty (b) translation uncertainty (c) depth uncertainty (d) photometric uncertainty

Fig. 7. Uncertainty of rotation, translation, depth, and photo-consistency versus the respective error term. The plots show a correspondence between uncertainty and error.

	Seq. 09	Seq. 10
ORB-SLAM (full)	0.014 ± 0.008	**0.012 ± 0.011**
ORB-SLAM (short)	0.064 ± 0.141	0.064 ± 0.130
DSO (full)	0.065 ± 0.059	0.047 ± 0.043
SfMLearner (paper)	0.021 ± 0.017	0.020 ± 0.015
SfMLearner (website)	0.016 ± 0.009	0.013 ± 0.009
proposed method	**0.014 ± 0.007**	0.013 ± 0.009

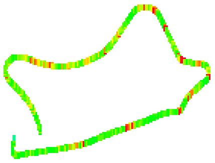


Table 2. Left: Odometry evaluation in comparison to SfMLearner for the two test sequences 09 and 10. The proposed threefold contributions yield an improvement to the state of the art in Seq. 09 and comparable results in Seq. 10. Right: Concatenated poses with color coded pose uncertainty (green=certain, red=uncertain) for Seq. 09.

on the website, to the proposed method with uncertainties and depth and pose supervision from SfM. Furthermore we compare to traditional methods ORB-SLAM (results as provided in [1]), and DSO [30]. “Full” refers to reconstruction from all images, and “short” refers to reconstruction from snippets of 5-frames. For DSO we were not able to get results for short sequences, as initialization is based on 5-10 keyframes.

6 Conclusions

In this paper we have presented a new method for simultaneously estimating depth maps and camera positions from monocular image sequences. This method is based on SfMLearning and uses only monocular RGB image sequences for training.

We have improved this baseline in three ways: by improving the image matching loss, by incorporating a probabilistic model of observation confidence and, extending the latter, by leveraging a standard SFM method to help supervising the deep network. Experiments show that our contributions lead to substantial improvements over the current state of the art both for the estimation of depth maps and odometry from monocular image sequences.

Acknowledgements. We are very grateful to Continental Corporation for sponsoring this research.

References

1. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR. (2017)
2. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR. (2017)
3. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**(5) (2015) 1147–1163
4. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5) (Oct 2017) 1255–1262
5. Buczko, M., Willert, V.: Monocular outlier detection for visual odometry. In: IEEE Intelligent Vehicles Symposium (IV). (2017)
6. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3d reconstruction in real-time. In: Intelligent Vehicles Symposium (IV), 2011 IEEE, Ieee (2011) 963–968
7. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on, IEEE (2007) 225–234
8. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 3248–3255
9. 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. (2014) 2366–2374
10. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. arXiv preprint arXiv:1704.00390 (2017)
11. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE international conference on computer vision. (2015) 2938–2946
12. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. arXiv preprint arXiv:1612.02401 (2016)
13. Garg, R., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision, Springer (2016) 740–756
14. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. International Conference on Robotics and Automation (2017)
15. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: Sfm-net: Learning of structure and motion from video. arXiv preprint arXiv:1704.07804 (2017)
16. Song, S., Chandraker, M.: Joint sfm and detection cues for monocular 3d localization in road scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3734–3742
17. Dhiman, V., Tran, Q.H., Corso, J.J., Chandraker, M.: A continuous occlusion model for road scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4331–4339
18. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems (NIPS). (2017)

19. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
20. Novotny, D., Larlus, D., Vedaldi, A.: Learning 3d object categories by looking around them. In: IEEE International Conference on Computer Vision. (2017)
21. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. on Image Processing **13**(4) (2004)
22. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: CVPR. (2016)
23. Ronneberger, O., P.Fischer, Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). Volume 9351 of LNCS., Springer (2015) 234–241
24. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
25. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
26. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000km: The Oxford RobotCar Dataset. The International Journal of Robotics Research (IJRR) **36**(1) (2017) 3–15
27. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence **31**(5) (2009) 824–840
28. 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)
29. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS). (Oct. 2012)
30. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence **4** (2017)