Constant Velocity Constraints for Self-Supervised Monocular Depth Estimation

Hang Zhou
University of East Anglia
hang.zhou@uea.ac.uk

David Greenwood
University of East Anglia
david.greenwood@uea.ac.uk

Sarah Taylor
University of East Anglia
s.l.taylor@uea.ac.uk

Han Gong
University of East Anglia
hg299@icloud.com

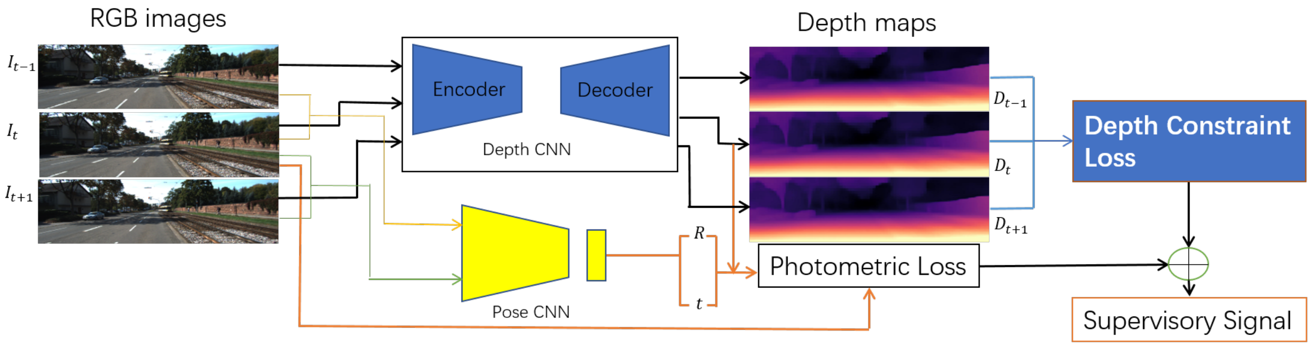

Figure 1: An overview of our method when training. A depth CNN and a pose CNN take a sequence of three consecutive video frames as input I_{t-1}, I_t, I_{t+1}. The depth CNN computes corresponding depth maps D_{t-1}, D_t, D_{t+1} and simultaneously the pose CNN outputs the rotation R and translation t of the camera. D_t, T and R are used to synthesise a new view and a photo-consistency loss is computed with the input image I_t (orange lines). Our main contribution is a velocity constraint loss which is computed over D_{t-1}, D_t, D_{t+1} (blue lines). To mentor training of the networks, a novel supervisory signal is constructed by combining the photo-consistency and depth constraint loss.

ABSTRACT

We present a new method for self-supervised monocular depth estimation. Contemporary monocular depth estimation methods use a triplet of consecutive video frames to estimate the central depth image. We make the assumption that the ego-centric view progresses linearly in the scene, based on the kinematic and physical properties of the camera. During the training phase, we can exploit this assumption to create a depth estimation for each image in the triplet. We then apply a new geometry constraint that supports novel synthetic views, thus providing a strong supervisory signal. Our contribution is simple to implement, requires no additional trainable parameter, and produces competitive results when compared with other state-of-the-art methods on the popular KITTI corpus.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**; **Unsupervised learning**;

KEYWORDS

Deep Learning, Self-supervised Learning, Monocular Depth Estimation

ACM Reference Format:
Hang Zhou, David Greenwood, Sarah Taylor, and Han Gong. 2020. Constant Velocity Constraints for Self-Supervised Monocular Depth Estimation. In *European Conference on Visual Media Production (CVMP '20), December 7–8, 2020, Virtual Event, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3429341.3429355

1 INTRODUCTION

Deriving 3D information from 2D images is a long held goal of the computer vision and machine learning community. Perceptually, humans find it relatively easy to understand the three dimensional properties of a scene. Unfortunately, the loss of a dimension in the 2D image makes the estimation of the true 3D geometry difficult; a so called ill-posed problem. Usually infinitely many different 3D surfaces may produce the same set of images, even though many of

CVMP '20, December 7–8, 2020, Virtual Event, United Kingdom

ACM ISBN 978-1-4503-8198-7/20/12…$15.00
https://doi.org/10.1145/3429341.3429355

those possibilities are implausible. In spite of this difficulty, there is considerable motivation to solve the problem.

Reconstructing 3D geometry from 2D images has many practical applications. These include driverless navigation, robotics, augmented reality (AR), computational photography and 3D modelling. Estimating depth from monocular RGB images is particularly compelling, given the abundant source of real world data, in comparison to expensive LiDAR sensors on driverless cars, or depth cameras in state-of-the-art mobile devices. The relative ubiquity of 2D RGB data is, however, offset by the lack of depth labelling.

Over the past decade, supervised learning using deep Convolutional Neural Networks (CNNs) have addressed many computer vision problems with great success. Depth estimation has yielded impressive results using diverse methods from a number of authors [Eigen et al. 2014; Fu et al. 2018; Laina et al. 2016; Li et al. 2015]. Although supervised learning for estimating depth cues has been shown to be quite possible, the requirement for pixel-wise labelling limits the amount of data available to train these models.

Self-supervised learning of depth can be divided into two main categories: stereo input imagery [Garg et al. 2016; Godard et al. 2017] or monocular input images. Using multi-view input images for self-supervised depth estimation (e.g. [Senoh et al. 2015]) has been explored. The multi-view approach is usually considered as a less popular but general case of stereo vision. Some of these methods aim to reconstruct particular objects of interest in the scene, rather than identify the depth of every pixel.

Structure from Motion (SfM) requires a set of images captured in multiple views to reconstruct 3D scenes by computing the relative positions between each camera. In monocular depth estimation, a sequence of images captured in a time series can be considered as different camera views, if we assume the scene is almost rigid [Zhou et al. 2017]. Our method is a form of SfM, where the monocular camera is moving within an environment to provide multiple views of that scene.

We propose a self-supervised method that exploits a consecutive series of RGB images, without any depth labelling, to produce a series of depth images and camera positions. Our contributions are:

- We introduce the notion of velocity constancy for monocular depth estimation.
- We describe an innovative training framework in which a depth CNN predicts the depth from three consecutive frames of input. We exploit relative depth across these frames and, through a simple motion model, we construct a novel geometry constraint as a supplementary supervisory signal.
- Our method yields state-of-the-art monocular depth estimation results on the KITTI Benchmark.

2 RELATED WORK

Self-supervised learning has been applied to both stereo and monocular depth estimation. In this section, our discussion is thus divided into two parts.

2.1 Stereo Depth Estimation

Stereo image pairs can provide a relative structure of geometry from two camera views [Hartley and Zisserman 2003], which further implies the use of epipolar geometry [Li et al. 2012]. This hardware setup simplifies the model for estimating depth from a static capture.

To achieve self-supervised learning from stereo pairs, neural networks [Garg et al. 2016; Xie et al. 2016] have been trained to predict per-pixel dense disparities between the pair. [Xie et al. 2016] proposed a model using discrete depth to predict novel view synthesis. This is a fully automatic 2D-to-3D conversion algorithm which takes 2D images or video sequences as input and outputs 3D stereo image pairs. This network is trained directly on stereo pairs from a dataset of 3D movies to minimise the pixel-wise photometric reconstruction error of the right view image when given the left view image. The output stereo images can be viewed with 3D glasses or head-mounted Augmented Reality (AR) displays. [Garg et al. 2016] developed a method to predict continuous disparity values. [Godard et al. 2017] further developed a two-view depth consistency constraint which produced results superior to other supervised methods. [Kuznietsov et al. 2018] trained a neural network in a semi-supervised manner, which used the sparse ground truth generated by some LiDAR sensors as supervisory signal.

Some stereo-based approaches have been extended with generative adversarial networks [Pilzer et al. 2018]. [Aleotti et al. 2018] proposed a generator network which learns to infer depth from the reference-view image and generate a warped target-view image. At training time, a discriminator network learns to distinguish between fake images generated by the generator and target frames acquired by a stereo rig. [Ranjan et al. 2019] proposed an additional consistency, which combines monocular depth estimation, optical flow, and segmentation tasks together. In their work, the four fundamental vision problems are solved simultaneously through geometric constraints. [Babu et al. 2018; Li et al. 2018; Zhan et al. 2018] adopted temporal information in optimisation. These methods have all made use of consecutive video frames to train networks (i.e. they utilise the temporal relation cross different frames). This work has shown that it is possible to train a depth predictor from monocular videos explicitly without stereo videos.

2.2 Monocular Depth Estimation

Due to the lack of spatial correspondences in a single frame, there are fewer constraints available to train a monocular depth model. To address this systematic problem, one would have to exploit intra-frame information as the training signal. [Zhou et al. 2017] proposed the first self-supervised monocular depth learning that simultaneously learns both depth and camera pose estimators during training. The camera pose estimator is used to provide rigid feature correspondences between frames. The key idea is to construct a photo-consistency reconstruction loss based on warping nearby views to the target through the depth and ego-motion. The two networks are coupled by the loss during training but can be applied independently when testing. In this pipeline, the combined networks take three consecutive frames as input and only feed the middle one into a depth prediction network to get the corresponding depth map. The other part of the networks – called ego-motion mapping network – takes two frames to estimate a 3-D rotation matrix and a 3-D translation matrix. After obtaining the depth map and the two matrices, the network uses them and nearby view

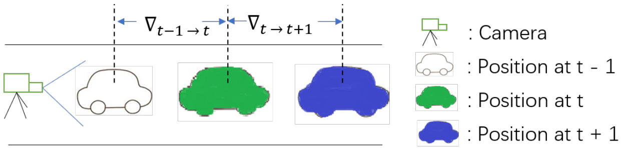

Figure 2: Our main contribution follows an assumption of approximate uniform linear motion:$\nabla_{t-1\rightarrow t}$ and $\nabla_{t\rightarrow t+1}$ denote the relative distance changes in a short sampling period

frames to synthesise a generated frame which should be very similar to the middle frame if the photo-consistency reconstruction quality is high. Training loss is based on the photo-consistency error between the original middle frame and the synthesised frame. This approach is based on the assumption that all objects in the images are rigid. To cope with non-rigid objects (e.g. pedestrians), an additional motion explanation mask generated by the pose CNN (only used in training) was introduced to ignore the regions that violate the rigid scene assumption. However, this additional motion explanation mask was abandoned by some later work [Godard et al. 2017] which obtained better performance.

Most self-supervised approaches (including [Zhou et al. 2017]) are based on a slightly strong assumption of brightness constancy. In practice, common violations of brightness constancy include occlusions, changes of view, moving objects in the scene, and reflective materials. To address this problem, [Klodt and Vedaldi. 2018] proposed a probabilistic learning formulation where the network predicts distributions over variables rather than specific values. This offers the important benefit of extracting as much information as possible from imperfect supervisory signals. It avoids the disruptions by outliers and noise. Apart from this, they proposed traditional SfM methods to generate the depth maps as the supervisory signals. However, their approach is computationally expensive and the generated depth maps are sparse. Inspired by [Byravan and Fox 2017], [Vijayanarasimhan et al. 2017] proposed a motion model by introducing multiple motion masks – but the work was not fully evaluated, making it difficult to understand the real benefits. [Godard et al. 2019] proposed an auto-masking loss to deal with those pixels violating the rigid motion and static scene assumptions. They also developed a minimum re-projection loss to handle occlusions robustly. [Alhashim and Wonka 2018] proposed a simple transfer learning network architecture, which uses features extracted from pre-trained networks. Most methods estimate depth independently for each video frame. [Patil et al. 2020] introduced a network architecture that produces a time series of depth maps. This was achieved by integrating the corresponding networks with a convolutional LSTM. The addition LSTM module exploits the spatio-temporal structures across frames. The latest work [Guizilini et al. 2020], which has similar ideas as ours, proposed a semi-supervised prediction framework. They introduced a velocity signal to mentor networks training and developed a new feature extractor to yield better performance.

3 SELF-SUPERVISED FRAMEWORK

In this section we describe the framework of our model and describe how we provide the supervisory signal during the training of our model. Fundamentally, our method is a form of Structure from Motion (SfM), where the monocular camera is moving within a rigid environment to provide multiple views of that scene. Our framework is built upon Monodepth2 [Godard et al. 2019].

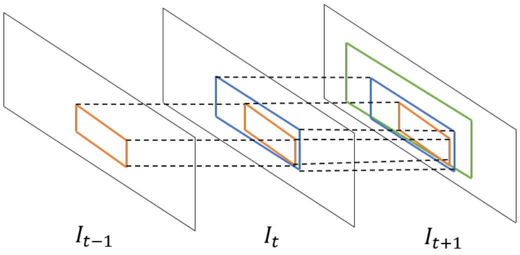

Figure 3: Constructing the depth constraint requires identifying the common pixels belonging to an object in all frames. These three frames denote a consecutive training sample. Red box, blue box, green box represent the same object captured in different views, therefore having corresponding scales. In this case, our proposed depth constraint only takes the area of red box into account.

Let $I_t \in \mathbb{R}^{H\times W\times 3}$, $t \in \{-1, 0, 1\}$ be a frame in a monocular video sequence captured by a moving camera, where t is the frame time index. Similarly, let $D_t \in \mathbb{R}^{H\times W}$ denote the depth map corresponding to image I_t. The camera pose changes from time 0 to time t, $t \in \{-1, 1\}$ is encoded by the 3 × 3 rotation matrix R_t and the translation vector $\mathbf{t}_t$. We obtain the 4 × 4 camera transformation matrix thus:

$$M_t = \begin{bmatrix} R_t & \mathbf{t}_t \\ 0 & 1 \end{bmatrix} \tag{1}$$

Our aim is to train two CNN networks to simultaneously estimate the pose of the camera, and the structure of the scene respectively.

$$M_t = \Theta_{\text{pose}}(I_t) \tag{2}$$

$$D_t = \Theta_{\text{depth}}(I_t) \tag{3}$$

3.1 Novel View Synthesis as Supervision

Self-supervised depth prediction reformulates the learning task as a novel view-synthesis problem. Specifically, during training, we let the coupled network synthesise the photo-consistency appearance of a target frame from another viewpoint of the source frame. We treat the depth map as an intermediate variable to constrain the network to complete the image synthesis task.

Let $(u, v) \in \mathbb{R}^2$ be the calibrated coordinates of a pixel in image I_0. In this case, let the origin $(0, 0)$ be the top-left of the image. In the process of imaging, a 3D point $(X, Y, Z) \in \mathbb{R}^3$ projects onto (u, v) through a perspective projection operator.

Suppose that the transformation matrix M_t encodes the pose change of the camera from time 0 to time t and Equation 4 is the perspective projection operator:

$$\begin{aligned} \pi(X, Y, Z) &= (f_x \frac{X}{Z} + c_x, f_y \frac{Y}{Z} + c_y) \\ &= (u, v) \end{aligned} \tag{4}$$

where(f_x, f_y, c_x, c_y) are the camera intrinsic parameters. Therefore, given a depth map $D(u, v)$, a 2D image point (u, v) backprojects to a 3D point (X, Y, Z) through backprojection operator, Equation 5.

$$\begin{aligned}\pi^{-1}(u, v, D(u, v)) &= D(u, v)(\frac{u - c_x}{f_x}, \frac{v - c_y}{f_y}, 1) \\ &= (X, Y, Z)\end{aligned} \tag{5}$$

then the corresponding pixels in image I_t can be computed as:

$$\begin{aligned}(u', v') &= \pi(M_t \pi^{-1}(u, v, D(u, v))) \\ &= g(u, v | D(u, v), M_t)\end{aligned} \tag{6}$$

We project the pixels of an image to form a novel synthetic view (Equation 6). However, the projected coordinates (u', v') are continuous values. To obtain $I^s(u, v)$ we include a differentiable bilinear sampling mechanism, as proposed in spatial transformer networks [Max et al. 2015]. We can now linearly interpolate the values of the 4-pixel neighbours (top-left, top-right, bottom-left, bottom-right) of $I(u', v')$ to give the RGB intensities as follows:

$$I^s(u, v) = \sum_u \sum_v w^{uv} I(u', v') \tag{7}$$

where w^{uv} is linearly proportional to the spatial proximity between (u, v) and (u', v'), and $\sum_{u,v} w^{uv} = 1$.

Classic depth estimation using SfM relies on a number of assumptions which can fail in the presence of occlusions, fine structures, non-rigid movements, complex geometry, or weak texture. To mitigate these problems our method builds a strong supervisory signal by combining a number of individual loss functions.

3.2 Photo-consistency Losses

For monocular depth estimation, an important supervisory signal to learn geometry from unlabelled video sequences is brightness constancy, which has been adopted as an invariant constraint [Zhou et al. 2017]. The constraint is based on the assumption that pixels in different video frames that correspond to the same scene point must have the same intensity in general. Existing methods have shown that a brightness constancy constraint is sufficient (at least in common cases) to guide the learning of the depth regression network and the camera pose estimation network.

Due to brightness constancy, the RGB intensities of the two corresponding pixels, in two different frames $I_0(u, v)$ and $I_t(u', v')$, should match. Therefore, we can write the fundamental photo-consistency loss as in Equation 8:

$$loss^b = \sum_{t \in [-1,1]} \sum_{(u,v) \in \Omega} |I_t(g(u, v | D_0, M_t)) - I_0(u, v)| \tag{8}$$

where Ω indicates the set of all pixel coordinates in a frame with respect to the defined coordinate origin. Note, we mask the brightness loss with a stationary mask, described in Section 3.3. All quantities in Equation 8 are known except for D_0, M_t which are estimated by the two CNN networks. We can denote the brightness constancy loss function as:

$$\begin{aligned}I^s &= I_t | \Theta_{\text{depth}}, \Theta_{\text{pose}} \\ L^b &= L(I, I^s)\end{aligned} \tag{9}$$

This basic photo-consistency loss only compares pixel intensity values. An additional constraint, Structural Similarity, has been shown to improve robustness for this task [Wang et al. 2004]. Given a pair of images a and b, their Structural Similarity $SSIM(a, b) \in [0, 1]$ is given by:

$$SSIM(a, b) = \frac{(2\mu_a \mu_b)(\sigma_a b + \epsilon)}{({\mu_a}^2 + {\mu_b}^2)({\sigma_a}^2 + {\sigma_b}^2) + \epsilon} \tag{10}$$

where ϵ is a small constant to avoid zero division, $\mu_a = \frac{1}{n} \sum_{i=1}^{n} a_i$ is the mean intensity of image a, ${\sigma_a}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \mu_a)^2$ is its variance, and $\sigma_{ab} = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \mu_a)(b_i - \mu_b)$ is the intensity correlation of the two images. Finally, our combined structural similarity and brightness loss becomes:

$$loss^p = \alpha(1 - SSIM(I, I^s)) + (1 - \alpha) L^b(I, I^s) \tag{11}$$

where the weighting parameter α is set as 0.85 empirically [Godard et al. 2019]. Rather than the mean of the photo-consistency error over all source images, we use the minimum, assuming greater error to be due to stationary violations.

3.3 Stationary Pixel Masking

Important assumptions for training are that the scene is captured by a moving camera, and the scene is static with respect to a world origin point. If any of these conditions is violated, the training performance can be detrimentally affected. Using a simple auto-masking method [Godard et al. 2019], we can filter the pixels that do not change appearance from one frame to the next in the video sequence. This mask allows the depth estimation network to ignore objects which move at the same velocity as the camera and even ignore whole frames in a monocular sequence when the camera is still.

A pixel is defined as moving when the photo-consistency loss between the target view I_t and the synthetic view I_t^s through warping the source view, is lower than the same error between the target view and source view I_0. More formally:

$$mask^s = |I_t - I_t^s| < |I_t - I_0| \tag{12}$$

The mask is binary, and no additional hyperparameter is required, as the mask can be computed in the forward pass of the network training. The pixels with almost unchanged intensities between consecutive frames often indicate no relative camera movement, an object that is relatively static to the camera, or a low texture region such as sky and roads. As such, our training method uses stationary pixel masking to only consider the photo-consistency loss contribution from the "moving" pixels.

3.4 Constant Velocity Depth Constraint

In this section, we describe our main contribution, a novel loss term for training. We allow ourselves the assumption that most training frames have been captured in a short time interval, during which the velocity of the moving camera can be considered as constant. Figure 2 provides an illustration. Maintaining that assumption, in a set of consecutive video frames, the distance from the camera to any rigid object in front of the camera, varies only linearly.

Suppose that we denote D_t as the depth map at some time step, we have the following equation hold for the major areas in the depth maps:

$$|D_{t+1} - D_t| \approx |D_t - D_{t-1}| \tag{13}$$

Individual objects in a scene have a variety of scale. Our idea models those pixel areas that belong to the same object instance in all three

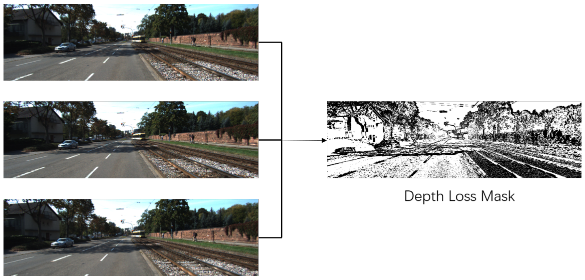

Figure 4: Illustration of Depth Constraint Mask: the brighter regions indicate pixels that are common to objects in all three frames.

frames. We illustrate the concept in Figure 3. We introduce a new mask to constrain the depth loss to ensure we only consider the pixels of an object common to all frames:

$$mask^d = [|I^g_t - I^g_{t+1}| < \beta] \cap [|I^g_t - I^g_{t-1}| < \beta] \qquad (14)$$

where I^g is the mean luminance image and β is a threshold value empirically set as 10 for 8-bit intensity values. A visualisation of an exemplar mask is shown in Figure 4. We apply the mask to form an additional depth loss term as follows:

$$loss^d = \lambda\mu(mask^d \odot (|D_{t+1} - D_t| - |D_t - D_{t-1}|)) \qquad (15)$$

where the weighting parameter λ is set empirically at 0.001 from $\lambda = \{0.1, 0.01, 0.001, 0.0001\}$, μ refers to the function that computes the mean of all matrix elements, and $\odot$ is the Hadamard Product. As a result of this geometry constraint, which models the depth relation of corresponding pixels on different frames, this penalty term makes it possible for the network to estimate depth from frames which contain a lot of moving objects in the scene or even are captured by a static camera and therefore violate the photo-consistency assumptions. Finally, we combine the masked photo-consistency loss and depth constraint loss:

$$loss^{total} = loss^p + loss^d \qquad (16)$$

3.5 Model Topology

Our model trains weights for two discreet networks, a depth estimation network, and a pose network. The depth network takes as input an RGB image, and outputs the corresponding depth estimation map; the pose network takes two RGB images as input to predict the 6–DoF relative pose.

Our depth network follows the well known U-Net architecture [Mayer et al. 2016], It is a symmetric *encoder* and *decoder* with skip connections on every layer but the input and output. The range of spatial resolution allows modelling both deep abstract features and local information. The encoder is ResNet-18 [He et al. 2016] with a total of 11m trainable parameters, initialised with weights trained on ImageNet [Deng et al. 2009]. Pretraining has been shown to improve accuracy compared to training from randomly initialised weights [Godard et al. 2019]. Our depth encoder follows [Godard et al. 2017], with a sigmoid nonlinearality on the output, and ReLU on the internal layers. However, the convolution layers use reflection, rather than zero padding, which gives a better estimate of source image pixel values when sampling from outside the border [Godard et al. 2019].

The pose network follows a similar design as the depth network encoder, however, it requires two frames to infer camera pose. We modify the ResNet-18 design, so the channel dimension of the input layer is doubled. Again, like the depth encoder, we pretrain on ImageNet. The output of the pose network is a 6–DoF relative pose in an axis-angle and translation representation.

3.6 Training

For monocular self-supervised training we use a sequence length of three images. To increase training data, we flip each input image horizontally, and also augment brightness, contrast, saturation and hue ±0.2 randomly. The same augmentation is applied to all three images in the input. We have implemented the networks using PyTorch [Paszke et al. 2019], and they were trained using an NVIDIA Quadro P5000 GPU with 16GB memory. During training all model weights are updated simultaneosly, by minimising the combined loss. The model was trained for 20 epochs, 1105 iterations every epoch using Adam [Kingma and Ba 2015], with a batch size of 12 and an input and output resolution of 640 × 192. We set the initial learning rate as 10^{-4} for the first 15 epochs and then decremented to 10^{-5} for fine-tuning the remainder.

4 EXPERIMENTS

In this section, we describe the dataset, show the evaluation metrics we use from [Eigen et al. 2014] in Table 1, and our evaluation results in comparison with the state-of-the-art methods.

4.1 Dataset

KITTI [Geiger et al. 2013] is a dataset that contains stereo images and corresponding 3-D laser scans of outdoor scenes captured by imaging equipment mounted on a moving vehicle [Kingma and Ba 2015]. The RGB images have a resolution of about 1241 × 376 and the corresponding depth maps are very sparse with a large amount of missing data. For training, we adopted the same dataset split used by [Eigen et al. 2014]. After removing the static frames by a preprocessing step suggested by [Zhou et al. 2017], this results in 39,810 monocular frame triplets for training and 4,424 frame triplets for validation. To simplify the training processing, the camera intrinsic matrix are assumed identical for all the frames in different scenes. To obtain this "universal" intrinsic matrix, we offset the principal point of the camera to the image centre and reset the focal length as the average of all the focal lengths in KITTI. This assumption is only valid when the capturing cameras are similar. Indeed, a more precise solution would be required to also estimate the individual intrinsic matrices for different videos sequences.

4.2 Results

In this section, we perform a quantitative evaluation to compare our proposed method with the other representative algorithms by using the common metrics discussed above. Table 2 shows that our method outperforms all other methods on the KITTI

Table 1: Definitions of Evaluation Metrics. y_p is a pixel in the ground-truth depth map y, y'_p is a pixel in the estimated depth map y', and n is the total number of pixels for each depth image.

Mean Relative Error (Abs Rel)	$\frac{1}{n}\sum_p^n \frac{\lvert y_p - y'_p \rvert}{y_p}$
Mean Relative Squared Error (Sq Rel)	$\frac{1}{n}\sum_p^n \frac{(y_p - y'_p)^2}{y_p}$
Root Mean Squared Error (RMSE)	$\sqrt{\frac{1}{n}\sum_p^n (y_p - y'_p)^2}$
Root Mean Squared Log Error (RMSE log)	$\sqrt{\frac{1}{n}\sum_p^n (log(y_p) - log(y'_p))^2}$
Threshold Accuracy (δ_i)	% of y_p, s.t. $max(\frac{y_p}{y'_p}, \frac{y'_p}{y_p}) = \delta_i < threshold_i, threshold_i = 1.25^i, i \in 1, 2, 3$

Table 2: Quantitative results on KITTI Benchmark using the Eigen split: ↑ represents the higher the better, and ↓, lower is better. The best scores in the table are underlined.

Method	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta_1 < 1.25$ ↑	$\delta_2 < 1.25^2$ ↑	$\delta_3 < 1.25^3$ ↑
SfMlearner [Zhou et al. 2017]	0.183	1.595	6.709	0.27	0.734	0.902	0.959
Yang [Yang et al. 2017]	0.182	1.481	6.501	0.267	0.725	0.906	0.963
GeoNet [Yin and Shi 2018]	0.149	1.060	5.567	0.226	0.796	0.935	0.975
Wang [Wang et al. 2018]	0.151	1.257	5.583	0.228	0.81	0.936	0.974
DF-Net [Zou et al. 2018]	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO [Yang et al. 2018]	0.162	1.352	6.276	0.252	-	-	-
EPC++ [Newcombe et al. 2011]	0.141	1.029	5.35	0.216	0.816	0.941	0.976
Struct2depth [Casser et al. 2019]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth2 [Godard et al. 2019]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [Guizilini et al. 2020]	$\underline{0.111}$	$\underline{0.785}$	$\underline{4.601}$	$\underline{0.189}$	0.878	0.960	0.982
Our method	0.112	0.816	4.715	0.190	$\underline{0.880}$	$\underline{0.960}$	$\underline{0.982}$

Table 3: Ablation. The first row represents the baseline, and * denotes an implementation option. ↑ represents the higher the better, and ↓ means the lower the better. The best scores in the table are underlined.

Stationary Mask	Depth Constraint	Pretrained	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta_1 < 1.25$ ↑	$\delta_2 < 1.25^2$ ↑	$\delta_3 < 1.25^3$ ↑
			0.16	1.44	5.72	0.262	0.77	0.921	0.969
		*	0.14	1.61	5.512	0.223	0.852	0.946	0.973
*			0.135	1.043	5.128	0.211	0.839	0.947	0.977
	*	*	0.132	1.044	5.142	0.210	0.845	0.948	0.977
*		*	0.124	0.936	5.010	0.203	0.865	0.952	0.977
*	*	*	$\underline{0.112}$	$\underline{0.816}$	$\underline{4.715}$	$\underline{0.190}$	$\underline{0.880}$	$\underline{0.960}$	$\underline{0.982}$

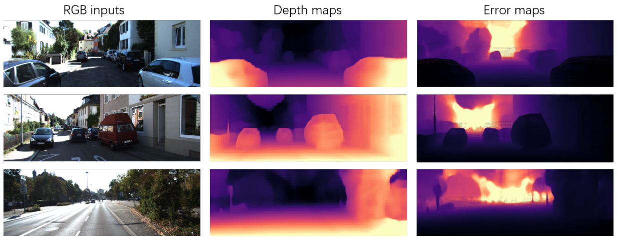

Figure 5: Visualisation of depth error maps. Here we show the error from our predicted depth maps compared to the ground truth depth maps from the KITTI test set. Note the error is largely at regions at infinity. The first column contains the input images, the middle column shows the depth estimation and the right column show the per-pixel depth error. Hotter colours indicate greater error.

2015 dataset [Geiger et al. 2013]. The exception to this is PackNet-SfM [Guizilini et al. 2020] which achieves marginally better performance on relative and RMSE errors, and equal or worse performance on threshold accuracy.

One of the reasons that our method produces more robust results given the same training data is that it uses a triplet of frames to supervise the training process while other approaches, such as Struct2Depth [Casser et al. 2019], rely on a pair of source and target images. Of course, this could also mean that the computational cost of training using our method would also be increased.

Another reason is that in the KITTI dataset [Geiger et al. 2013], there are many frames that are captured by a static camera that contain moving objects. These problematic frames are filtered out by the other existing methods as their training methods cannot make use of these frames. However, with our novel depth constraint loss, those frames are made useful for training.

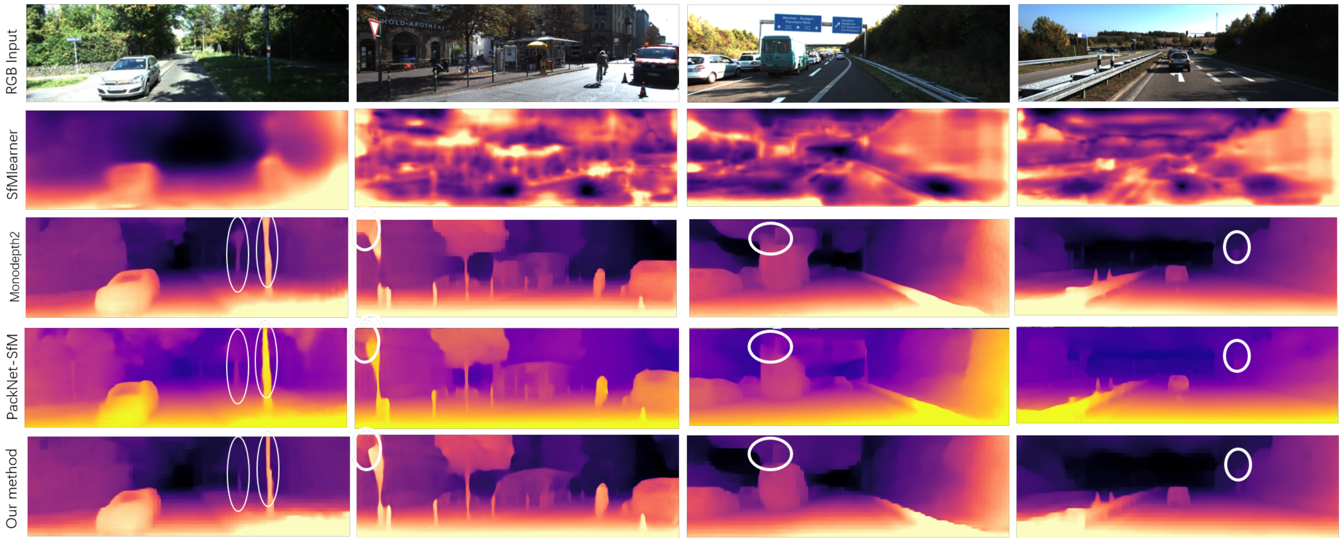

Figure 6: Visualisation of depth estimation results. The top row contains the input images. The remaining rows show the depth estimation results from contemporary methods, visualised by false colours. Hotter colours indicate closer objects.

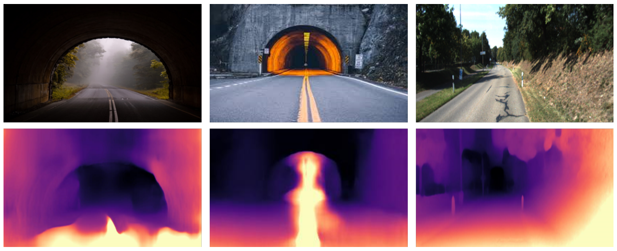

Figure 7: Common failure cases. Road marks have been incorrectly recognised as closer objects in the left and middle figures. The tunnel structure has been recognised as infinity (i.e. similar to Sky) in the middle figure. The sky in the right figure has been recognised as an object not at infinity. These failures exist in all contemporary methods, and motivate future work that can handle these difficult examples.

It should be noted that our model architecture is the same as that in Monodepth2 [Godard et al. 2019]. However, training with our proposed depth constraints has resulted in increased performance over all evaluation metrics — a clear indication that our constant velocity assumptions are valid.

Figure 6 shows the depth maps generated by SfMlearner [Zhou et al. 2017], Monodepth2 [Godard et al. 2019], PackNet [Guizilini et al. 2020] and our method for some target frames. We observe that our method predicts fewer artefacts affected by the shadows in the scene, and more robustly identifies the contours of objects. For example, in the first column our method more accurately segments the post in the foreground, and correctly identifies that the furthest post is obscured by a tree. In the third column it is clear that our method better captures depth details around the vehicle's contour.

To better understand the behaviour of our system, we visualized the per-pixel errors of the depth map, as shown in Figure 5. We observe that objects that are far from the camera have lower accuracy than those that are closer. Therefore our approach is very well suited to applications that require precise near-field depth information.

As common with all contemporary works, our method suffers occasional failures in difficult scenes. Figure 7 provides some examples. We remain highly motivated to tackle these problematic areas in future work.

4.3 Ablation Study

To understand how the components of our CNN network contribute to the overall performance in monocular depth learning, we perform an ablation study by changing variables of our components as shown in Table 3. We observe that the baseline model (top row) performs the worst. A significant improvement was made by pretraining the depth CNN on ImageNet [Deng et al. 2009] (second row). The fifth row, verifies that the stationary pixel mask (described in Section 3.3) improved the result compared with no masking. In the final row, we show the improvement by introducing our novel depth constraint loss (described in Section 3.4).

5 CONCLUSIONS

In this work, we have presented a novel framework for monocular depth estimation and achieved state-of-the-art results on a popular benchmark. As far as we know, no work before exploits the relationship between depth maps from consecutive video frames. From a simple real world conception, we introduce and develop an additional loss item as a supplementary supervisory signal to photo-consistency loss. Our novel depth loss is based on the assumption that the velocity of the camera moving through the scene in consecutive video frames is constant. We validate this assumption by comparing against similar approaches objectively and show depth visualisations of the competing methods. Our idea is simple to understand and implement and introduces no additional learn-able parameters.

REFERENCES

Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. 2018. Generative adversarial networks for unsupervised monocular depth prediction. In *European Conference on Computer Vision Workshops*. Springer, Munich, Germany, 337–354.

Ibraheem Alhashim and Peter Wonka. 2018. High Quality Monocular Depth Estimation via Transfer Learning. arXiv:1812.11941 [cs.CV]

V Madhu Babu, Kaushik Das, Anima Majumdar, and Swagat Kuma. 2018. Undemon: Unsupervised deep network for depth and ego-motion estimation. In *International Conference on Intelligent Robots and Systems*. IEEE, Madrid, Spain, 1082–1088.

Arunkumar Byravan and Dieter Fox. 2017. Se3-nets: Learning rigid body motion using deep neural networks. In *International Conference on Robotics and Automation*. IEEE, Marina Bay Sands, Singapore, 173–180.

Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. 2019. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 8001–8008.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Miami, FL, 248–255.

David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth map prediction from a single image using a multi-scale deep network. In *Conference on Neural Information Processing Systems*. NeurIPS Foundation, Montréal, Canada, 1–9.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 2002–2011.

Ravi Garg, Vijay Kumar BG, and Ian Reid. 2016. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*. Springer, Amsterdam, The Netherlands, 740–756.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research* 2, 11 (2013), 1231–1237.

Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. 2017. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Honolulu, Hawaii, 6602–6611.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. 2019. Digging Into Self-Supervised Monocular Depth Estimation. In *International Conference on Computer Vision*. IEEE, Seoul, Korea, 3827–3837.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 2020. 3D Packing for Self-Supervised Monocular Depth Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Virtual, 2485–2494.

Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, Nevada, 770–778.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. IEEE, San Diego, California, 1–15.

Maria Klodt and Andrea Vedaldi. 2018. Supervising the new with the old: learning SFM from SFM. In *European Conference on Computer Vision*. Springer, Munich, Germany, 698–713.

Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. 2018. Semi-supervised deep learning for monocular depth map prediction. In *European Conference on Computer Vision*. Springer, Munich, Germany, 2215–2223.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. 2016. Deeper Depth Prediction with Fully Convolutional Residual Networks. In *Fourth International Conference on 3D Vision*. IEEE, Stanford University, California, 239–248.

Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. 2015. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Boston, Massachusetts, 1119–1127.

Qijie Li, Tianqing Chang, and Xuejun Jiao. 2012. A new targets matching method based on epipolar geometry. In *International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems*. IEEE, Tianjin, China, 135–139.

Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. 2018. UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning. In *International Conference on Robotics and Automation*. IEEE, Brisbane, Australia, 7286–7291.

Jaderberg Max, Simonyan Karen, Zisserman Andrew, et al. 2015. Spatial transformer networks. In *Advances in neural information processing systems*. NeurIPS Foundation, Montréal, Canada, 2017–2025.

Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, Nevada, 4040–4048.

Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. 2011. DTAM: Dense tracking and mapping in real-time. In *International Conference on Computer Vision*. IEEE, Barcelona, Spain, 2320–2327.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. 2020. Don't Forget The Past: Recurrent Depth Estimation from Monocular Video. arXiv:2001.02613 [cs.CV]

Andrea Pilzer, Dan Xu, Mihai Marian Puscas, Elisa Ricci, and Nicu Sebe. 2018. Unsupervised adversarial depth estimation using cycled generative networks. In *International Conference on 3D Vision*. IEEE, Verona, Italy, 587–595.

Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. 2019. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Long Beach, California, 12232–12241.

Takanori Senoh, Koki Wakunami, Hisayuki Sasaki, Ryutaro Oi, and Kenji Yamamoto. 2015. Fast depth estimation using non-iterative local optimization for super multi-view images. In *Global Conference on Signal and Information Processing*. IEEE, Orlando, Florida, 1042–1046.

Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. 2017. SfM-Net: Learning of Structure and Motion from Video. arXiv:1704.07804 [cs.CV]

Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. 2018. Learning Depth from Monocular Videos Using Direct Methods. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 2022–2030.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*. Springer, Amsterdam, The Netherlands, 842–857.

Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. 2018. LEGO: Learning Edge with Geometry all at Once by Watching Videos. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 225–234.

Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. 2017. Unsupervised Learning of Geometry with Edge-aware Depth-Normal Consistency. arXiv:1711.03665 [cs.CV]

Zhichao Yin and Jianping Shi. 2018. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 1983–1992.

Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian M. Reid. 2018. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 340–349.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. 2017. Unsupervised Learning of Depth and Ego-Motion from Video. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Honolulu, Hawaii, 6612–6619.

Yuliang Zou, Zelun Luo, and Jia-Bin Huang. 2018. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*. Springer, Munich, Germany, 1–18.