The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Unsupervised Learning of Geometry from Videos with Edge-Aware Depth-Normal Consistency

Zhenheng Yang,[1] Peng Wang,[2] Wei Xu,[2] Liang Zhao,[2] Ramakant Nevatia[1]
zhenheny@usc.edu {wangpeng54,wei.xu,zhaoliang07}@baidu.com nevatia@usc.edu
[1]University of Southern California [2]Baidu Research

Abstract

Learning to reconstruct depths from a single image by watching unlabeled videos via deep convolutional network (DCN) is attracting significant attention in recent years, *e.g.*(Zhou et al. 2017). In this paper, we propose to use surface normal representation for unsupervised depth estimation framework. Our estimated depths are constrained to be compatible with predicted normals, yielding more robust geometry results. Specifically, we formulate an edge-aware depth-normal consistency term, and solve it by constructing a depth-to-normal layer and a normal-to-depth layer inside of the DCN. The depth-to-normal layer takes estimated depths as input, and computes normal directions using cross production based on neighboring pixels. Then given the estimated normals, the normal-to-depth layer outputs a regularized depth map through local planar smoothness. Both layers are computed with awareness of edges inside the image to help address the issue of depth/normal discontinuity and preserve sharp edges. Finally, to train the network, we apply the photometric error and gradient smoothness to supervise both depth and normal predictions. We conducted experiments on both outdoor (KITTI) and indoor (NYUv2) datasets, and showed that our algorithm vastly outperforms state-of-the-art, which demonstrates the benefits of our approach.

1 Introduction

[1] Human beings are highly competent in recovering the 3D geometry of observed natural scenes at a very detailed level in real-time, even from a single image. Being able to do reconstruction for monocular images can be widely applied to large amount of real applications such as augmented reality and robotics.

One group of approaches solve this problem by feature matching and estimating camera and scene geometries, *e.g.*structure from motion (SFM) (Wu 2011) *etc*., or color matching, *e.g.*DTAM (Newcombe, Lovegrove, and Davison 2011). But these techniques are sensitive to correct matching and are ineffective in homogeneous areas. Another way to do 3D reconstruction is a learning based method, where the reconstruction cues can be incrementally discovered by learning from videos. Currently, with the development of pixelwise prediction such as fully convolutional network (FCN) (Long, Shelhamer, and Darrell 2015), supervised learning of depth, *e.g.*(Eigen, Puhrsch, and Fergus 2014), achieved impressive results over public datasets like KITTI (Geiger, Lenz, and Urtasun 2012), NYUv2 (Silberman et al. 2012) and SUN3D (Xiao, Owens, and Torralba 2013). Nevertheless, collecting ground truth depth is almost impossible for random videos. It is hard for the supervisedly learned models to generalize on videos of different scenes.

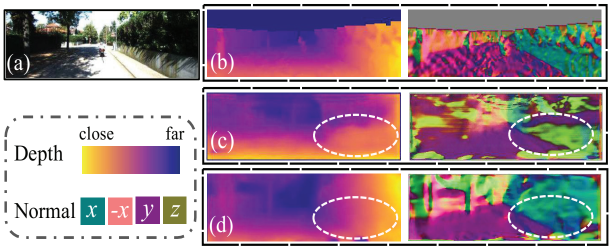

Figure 1: Comparison between (Zhou et al. 2017) and our results with depth-normal consistency. Top to bottom: (b) Ground truth depths (left) and normals (right). (c) Results from (Zhou et al. 2017). (d) Our results. In the circled region, (Zhou et al. 2017) fails to predict scene structure as shown by estimated normals, while ours correctly predict both depths and normals with such consistency.

We can, instead, try to solve this problem in an unsupervised way by imposing 3D scene geometric consistency between video frames. There have been some works in this line, (Zhou et al. 2017) propose a single image depth FCN learning from videos. In their training, rather than using ground truth depth, they warp the target image to other consecutive video frames based on the predicted depths and relative motions, and match the photometry between the warped frames and observed frames (detailed in Sec. 3). Then, the matching errors are used as the supervision of the depth prediction. Similar idea is applied in depth prediction when stereo pairs are available (Garg, G, and Reid 2016; Godard, Mac Aodha, and Brostow 2017).

Altough those methods are able to do single image depth estimation, the results do not well represent the scene structure, especially when visulized with computed normals, as shown in Fig. 1(c). This is mostly due to that photometric matching is ambiguous, *i.e.*a pixel in source frames can

[1] This work is done while Zhengheng Yang is interning at Baidu

be matched to multiple similar pixels in target frames. Researchers usually apply smoothness of depths (Zhou et al. 2017) to reduce the ambiguity, it is often a weak constraint over neighboring pixels, which potentially have similar colors, yielding inconsistent normal results.

Our work falls in the scope of learning based 3D reconstruction of a single image trained on monocular videos, following the work of (Zhou et al. 2017). But we have a step further towards learning a regularized 3D geometry with explicit awareness of normal representation. We are motivated by the fact that human beings are more sensitive in normal directions compared to depth estimation. For instance, one could precisely point out the normal direction of surface at each pixel of a single image while could only roughly know the absolute depth.

Thus, we incorporate an edge-aware depth-normal consistency constraint inside the network which better regularizes the learning of depths (Sec. 4). There are several advantages of having normal estimated. For instance, it gives explicit understanding of normal for learned models. In addition, it provides higher order interaction between estimated depths, which is beyond local neighbor relationships. Lastly, additional operations, *e.g.*Manhattan assumption, over normals could be further integrated. As depth/normal discontinuity often appear at object edges in the image, we incoporate the image edges in this constraint to compensate. As shown in Fig. 1(d), with such a constraint, our recovered geometry is comparably better. We did extensive experiments over the public KITTI and NYUv2 datasets, and show our algorithm can achieve relative 20% improvement over the state-of-the-art method on depth estimation and 10% improvement on predicted normals. More importantly, the training converges around 3× faster. These demonstrate the efficiency and effectiveness of our approach.

2 Related Work

Structure from motion and single view geometry. As discussed in Sec. 1, geometry based methods, such as SFM (Wu 2011), ORB-SLAM (Mur-Artal, Montiel, and Tardos 2015), DTAM (Newcombe, Lovegrove, and Davison 2011), rely on feature matching, which could be effective and efficient in many cases. However, they can fail at where there is low texture, or drastic change of visual perspective *etc*.. More importantly, it cannot be extended to single view reconstruction where humans are good at. Traditionally, specific rules are developed for single view geometry. Methods rely on either computing vanishing point (Hoiem, Efros, and Hebert 2007), following rules of BRDF (Prados and Faugeras 2006), or extracting the scenes with major plane and box representations (Schwing et al. 2013; Srajer et al. 2014) *etc*.. Those methods can only obtain sparse geometry representations, and some of them require certain assumptions (*e.g.* Lambertian, Manhattan world).

Supervised single view geometry via CNN. With the advance of deep neural networks and their strong feature representation, dense geometry, *i.e.*, pixel-wise depth and normal maps, can be readily estimated from a single image (Wang, Fouhey, and Gupta 2015; Eigen and Fergus 2015; Laina et al. 2016). The learned CNN model shows significant improvement compared to other strategies based on hand-crafted features (Karsch, Liu, and Kang 2014; Ladicky, Shi, and Pollefeys 2014; L. Ladicky, Pollefeys, and others 2014). Others tried to improve the estimation further by appending a conditional random field (CRF) (Wang et al. 2015; Liu, Shen, and Lin 2015; Li et al. 2015). However, most works regard depth and normal predictions as independent tasks. (Wang et al. 2016) point out their correlations over large planar regions, and regularize the prediction using a dense CRF (Kong and Black 2015), which improved the results on both depth and normal. However, all those methods require densely labeled ground truths, which are expensive to label in natural environments.

Unsupervised single view geometry. Videos are easy to obtain at the present age, while holding richer 3D information than single images. Thus, it attracts lots of interests if single view geometry can be learned through feature matching from videos. Recently, several deep learning methods have been proposed based on such an intuition. Deep3D (Xie, Girshick, and Farhadi 2016) learns to generate the right view from the given left view by supervision of a stereo pair. In order to do back-propagation to depth values, it quantizes the depth space and learns to select the right one. Concurrently, (Garg, G, and Reid 2016) applies the similar supervision from stereo pairs, while the depth is kept continuous, They apply Taylor expansion to approximate the gradient for depth. (Godard, Mac Aodha, and Brostow 2017) extends Garg's work by including depth smoothness loss and left-right depth consistency. Most recently, (Zhou et al. 2017) induces camera pose estimation into the training pipeline, which makes depth learning possible from monocular videos. And they come up with an explainability mask to relieve the problem of moving object in rigid scenes. At the same time, (Kuznietsov, Stuckler, and Leibe 2017) proposes a network to include modeling rigid object motion. Although vastly developed for depth estimation from video, normal information, which is also highly interesting for geometry prediction, has not been considered inside the pipeline. This paper fills in the missing part, and show that normal can serve as a natural regularization for depth estimation, which significantly improves the state-of-the-art performance. Finally, with our designed loss, we are able to learn the indoor geometry where (Zhou et al. 2017) usually fails to estimate.

3 Preliminaries

In order to make the paper self-contained, we first introduce several preliminaries proposed in the unsupervised learning pipelines (Zhou et al. 2017; Godard, Mac Aodha, and Brostow 2017). The core idea behind, as discussed in Sec. 2, is inverse warping from target view to source view with awareness of 3D geometry, as illustrated in Fig. 3(a), which we will elaborate in the following paragraphs.

Perspective projection between multiple views. Let $D(x_t)$ be the depth value of the target view at image coordinate x_t, and $\mathbf{K}$ be the intrinsic parameter of the camera. Suppose the relative pose from the target view to source view is a rigid transformation $\mathbf{T}_{t\rightarrow s} = [\mathbf{R}|\mathbf{t}] \in \mathcal{SE}(3)$, and $h(x)$ is the homogeneous coordinate given x. The perspective

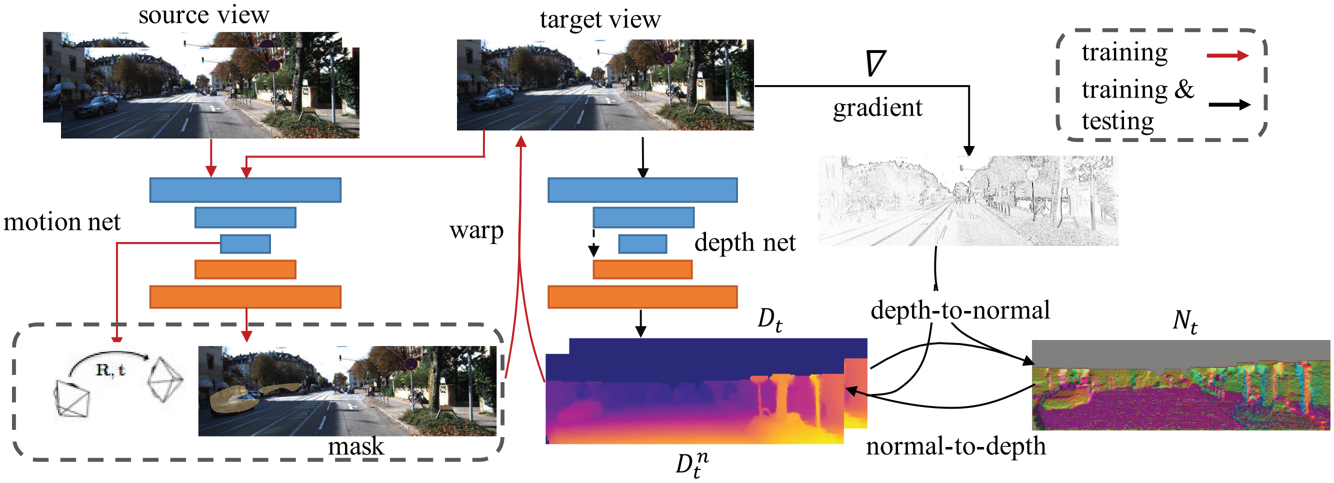

Figure 2: Framework of our approach.

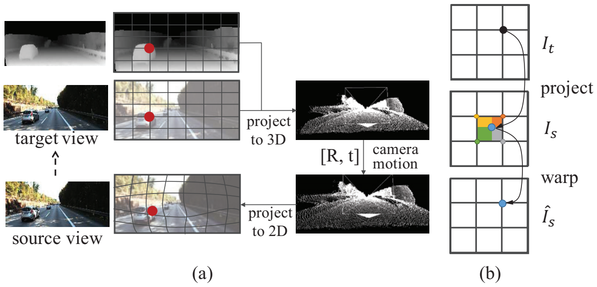

Figure 3: Illustraion of (a) 3D inverse warping and (b) bilinear interpolation.

warping to localize corresponding pixels can be formulated as,

$$D(x_s)h(x_s) = \mathbf{K}\mathbf{T}_{t\rightarrow s}D(x_t)\mathbf{K}^{-1}h(x_t), \qquad (1)$$

and the image coordinate x_s can be obtained by dehomogenisation of $D(x_s)h(x_s)$. Thus, x_s and x_t is a pair of matching coordinates, and we are able to compare the similarity between the two to validate the correctness of structure.

Photometric error from view synthesis. Given pixel matching pairs between target and source views, *i.e.*I_t and I_s, we can synthesize a target view $\hat{I}_s$ from the given source view through bilinear interpolation (Garg, G, and Reid 2016), as illustrated in Fig. 3(b). Then, under the assumption of Lambertian and a static rigid scene, the average photometric error is often used to recover the depth map D for the target view and the relative pose. However, as pointed out by (Zhou et al. 2017), this assumption is not always true, due to the fact of moving objects and occlusion. An explainability mask $\mathbf{M}$ is induced to compensate for this. Formally, the masked photometric error is,

$$\mathcal{L}_{vs}(D,\mathcal{T},\mathcal{M}) = \sum_{s=1}^{S}\sum_{x_t}\mathbf{M}_s(x_t)|I_t(x_t) - \hat{I}_s(x_t)|,$$
$$\text{s.t. } \forall x_t, s \;\; \mathbf{M}_s(x_t) \in [0,1], \;\; D(x_t) > 0 \qquad (2)$$

where $\{\hat{I}_s\}_{s=1}^{S}$ is the set of warped source views, and $\mathcal{T}$ is a set of transformation from target view to each of the source views. $\mathcal{M} = \{\mathbf{M}_s\}$ is a set of explainability masks, and $\mathbf{M}_s(x_t) \in [0,1]$ weights the error at x_t from source view s.

Regularization. As mentioned in Sec. 1, supervision based solely on photometric error is ambiguous. One pixel could match to multiple candidates, especially in low-texture regions. In addition, there is trivial solution for explainability mask by setting all values to zero. Thus, to reduce depth ambiguity and encourage non-zero masks, two regularization terms are applied,

$$\mathcal{L}_s(D,2) = \sum_{x_t}\sum_{d\in x,y}|\nabla_d^2 D(x_t)|e^{-\alpha|\nabla_d I(x_t)|}$$
$$\mathcal{L}_m(\mathcal{M}) = -\sum_{s}\sum_{x_t}\log P(\mathbf{M}_s(x_t) = 1) \qquad (3)$$

$\mathcal{L}_s(D,2)$ is a spatial smoothness term penalizes L1 norm of second-order gradients of depth along both x and y directions, encouraging depth values to align in planar surface when no image gradient appears. Here, the number 2 represents the 2nd order for depth. $\mathcal{L}_m(\mathcal{M})$ is cross-entropy between the masks and maps with value 1.

Finally, a multi-scale strategy is applied to the depth output, and the total loss for depth estimation from videos is a joint function from previous terms,

$$\mathcal{L}_o(\{D_l\},\mathcal{T},\mathcal{M}) = \sum_{l}\{\mathcal{L}_{vs}(D_l,\mathcal{T},\mathcal{M}) + \lambda_s\mathcal{L}_s(D_l)$$
$$+ \lambda_m\mathcal{L}_m(\mathcal{M}_l)\} \qquad (4)$$

D_l and M_l represent the depth and mask under scale l.

Given the objective function, the photometric error can be back-propagated to depth, pose and mask networks by applying the spatial transform operation as proposed by (Jaderberg et al. 2015), which supervises the learning process.

4 Geometry estimation with edge-aware depth-normal consistency

In our scenario, given a target image I, we aim at learning to estimate both depths and normals simultaneously. Formally, let N be the predicted normals from our model, we embed it into the training pipeline and make it a regularization for depths estimation D, which helps to train a more robust model.

4.1 Framework

Fig. 2 illustrates an overview of our approach. For training, we apply supervision from view synthesis following (Zhou et al. 2017). Specifically, the depth network (middle) takes only the target view as input, and outputs a per-pixel depth map D_t, based on which a normal map N_t is generated by the depth-to-normal layer. Then, given the D_t and N_t, a new depth map D_t^n is estimated from the normal-to-depth layer using local orthogonal compatibility between depth and normals. Both of the layers takes in image gradient to avoid non-compatible pixels involving in depth and normal conversion (detailed in Sec. 4.2). Then, the new depth map D_t^n, combined with poses and mask predicted from the motion network (left), are then used to inversely warp the source views to reconstruct the target view, and errors are back propagated through both networks. Here the normal representation naturally serves as a regularization for depth estimation. Finally, for training loss, additional to the usually used photometric reconstruction loss, we also add in smoothness over normals, which induces higher order interaction between pixels (Sec. 4.3)

With the trained model, given a new image, we infer per-pixel depth value and then compute the normal value, yielding consistent results between the two predictions.

4.2 Depth and normal orthogonality.

In reconstruction, depth and normal are two strongly correlated information, which follows locally linear orthogonality. Formally, for each pixel x_i, such a correlation can be written as a quadratic minimization for a set of linear equations,

$$\begin{aligned}&\mathcal{L}_{x_i}(D,N) = \|[\cdots, \omega_{ji}(\phi(x_j)-\phi(x_i)), \cdots]^T N(x_i)\|^2,\\ &\text{where } \phi(x) = D(x)\mathbf{K}^{-1}h(x),\ \|N(x_i)\|_2 = 1,\\ &\quad \omega_{ji} > 0 \text{ if } x_j \in \mathcal{N}(x_i) \end{aligned} \tag{5}$$

where $\mathcal{N}(x_i)$ is a set of predefined neighborhood pixels of x_i, and $N(x_i)$ is a 3×1 vector. $\phi(x)$ is the back projected 3D point from 2D coordinate x. $\phi(x_j) - \phi(x_i)$ is a difference vector in 3D, and ω_{ji} is used to weight the equation for pixel x_j w.r.t. x_i which we will elaborate later.

As discussed in Sec. 2, most previous works try to predict the two information independently without considering such a correlation, while only SURGE (Wang et al. 2016) proposes to apply the consistency by a post CRF processing only over large planar regions. In our case, we enforce the consistency over the full image, and directly apply it to regularize the network learning. Specifically, to model their consistency, we develop two layers by solving Eq. (5), *i.e.*a depth-to-normal layer and a normal-to-depth layer.

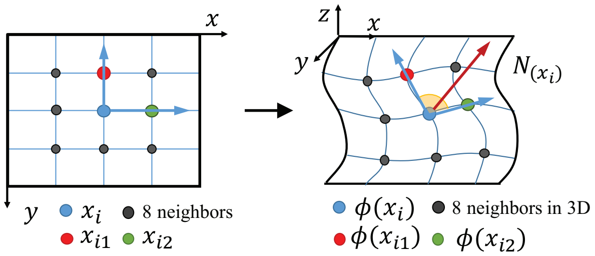

Figure 4: Illustration of computing normal base on a pair of neighboring pixels. x_i, x_{i1}, x_{i2} are 2D points, and $\phi(x_i), \phi(x_{i1}), \phi(x_{i2})$ are corresponding points projected to 3D space. The normal direction $N(x_i)$ is computed with cross product between $\phi(x_{i1}) - \phi(x_i)$ and $\phi(x_{i2}) - \phi(x_i)$.

Infer normals from depths. Given a depth map D, for each point x_i, in order to get $N(x_i)$. From Eq. (5), we need to firstly define neighbors $\mathcal{N}(x_i)$ and weights ω_{ji}, and then solve the set of linear equations. To deal with the first issue, we choose to use the 8-neighbor convention to compute normal directions, which considerably more robust than the 4-neighbor convention. However, it is not reasonable to equally weight all pixels due to depth discontinuity or dramatic normal changes. Thus, for computing ω_{ji}, we weight more for pixels x_j having similar color with x_i, while weight less otherwise. Formally, in our case, it is computed as $\omega_{ji} = \exp\{-\alpha|I(x_j) - I(x_i)|\}$ and $\alpha = 0.1$.

To minimize Eq. (5), one may apply a standard singular value decomposition (SVD) to obtain the solution. However, in our case, we need to embed such an operation in the network for training, and back-propagate the gradient to input depths. SVD is computationally non-efficient for back-propagation. Thus, we choose to use mean cross-product to approximate the minimization (Jia 2006), which is simpler and more efficient. Specifically, from the 8 neighbor pixels around $x_i = [m, n]$, we split them to 4 pairs, where each pair of pixels is perpendicular at 2D coordinate w.r.t. x_i, and in a counter clock-wise order, *i.e.*$\mathcal{P}(x_i) = \{([m-1,n],[m,n+1]), \cdot, ([m+1,n-1],[m-1,n-1])\}$. Then, for each pair, cross product of their difference vector w.r.t. x_i is computed, and the mean direction of the computed vectors is set as the normal direction of x_i. Formally, the solver for normals is written as,

$$\begin{aligned}&\mathbf{n} = \sum_{p\in\mathcal{P}} (\omega_{p_0,x_i}(\phi(p_0) - \phi(x_i)) \times \omega_{p_1,x_i}(\phi(p_1) - \phi(x_i))),\\ &N(x_i) = \mathbf{n}/\|\mathbf{n}\|_2\end{aligned} \tag{6}$$

The process of calculating the normal direction for x_i using one pair of pixels is in Fig. 4.

Compute depths from normals. Due to the fact that we do not have ground truth normals for supervision, it is necessary to recover depths from normals to receive the supervision from photometric error as discussed in Sec. 3. To recover depths, given normal map N, we still need to solve Eq. (5). However, there is no unique solution. Thus, to make it solvable, we provide an initial depth map D_o as input, which might lack normal smoothness, *e.g.*depth map from network output. Then, given $D_o(x_i)$, the depth solution for each neighboring pixel of x_i is unique and can be easily computed. Formally, let $D_e(x_j|x_i) = \psi(D_o(x_i), N(x_i))$ be the solved depth value calculated for a neighbor pixel x_j w.r.t. x_i. However, when computing over the full image, we still need to solve 8 equations jointly for each pixel of the 8 neighbors. Finally, by minimum square estimation (MSE), the solution for depth of x_i is,

$$D_n(x_j) = \sum_{i \in \mathcal{N}} \hat{\omega}_{ij} D_e(x_j|x_i), \;\; \hat{\omega}_{ij} = \omega_{ij} / \sum_i \omega_{ij} \tag{7}$$

4.3 Training losses

Given the consistency, in this section, we describe our training strategy. In order to supervise both the depth and normal predictions, we can directly apply the loss in Eq. (4) by replacing the output from depth network D_o with the output after our normal-to-depth layer D_n to train the model. We show in our experiments (Sec. 5), by doing this, we already outperform the previous state-of-the-art by around 10$\%$ in depth estimation using the same network architecture.

Additionally, with normal representation, we apply smoothness over neighboring normal values, which provides higher order interactive between pixels. Formally, the smoothness for normal has the same form as $\mathcal{L}_s$ in Eq. (3) for depth, while the first order gradient is applied, *i.e.* $\mathcal{L}_s(N, 1)$.

Last but not least, matching corresponding pixels between frames is another central factor to find correct geometry. Addition to the photometric error from matching pixel colors, matching image gradient is more robust to lighting variations, which was frequently applied in computing optical flow (Li 2017). In our case, we compute a gradient map of the target image and synthesized target images, and include the gradient matching error to our loss function. Formally, the loss is represented as,

$$\mathcal{L}_g(D_n, \mathcal{T}, \mathcal{M}) = \sum_{s=1}^{S} \sum_{x_t} \mathbf{M}_s(x_t) \|\nabla I_t(x_t) - \nabla \hat{I}_s(x_t)\|_1,$$

In summary, our final learning objective for multi-scale learning is,

$$\begin{aligned} \mathcal{L}(\mathcal{D}, \mathcal{N}, \mathcal{T}, \mathcal{M}) = \mathcal{L}_o(\{D_{nl}\}, \mathcal{T}, \mathcal{M}) + \\ \sum_l \{\lambda_g \mathcal{L}_g(D_{nl}, \mathcal{T}, \mathcal{M}) + \lambda_n \mathcal{L}_s(N_l, 1)\} \end{aligned} \tag{8}$$

where $\mathcal{D} = \{D_{nl}\}$ and $\mathcal{N} = \{N_l\}$ are the set of depth maps and normal maps for the target view.

Model training. For network architecture, similar to (Zhou et al. 2017) and (Godard, Mac Aodha, and Brostow 2017), we adopt the DispNet (Mayer et al. 2016) architecture with skip connections as in (Zhou et al. 2017). All *conv* layers are followed by a ReLU activation except for the top prediction layer. We train the network from scratch; since too many losses at beginning could be hard to optimize, we choose a two stage training strategy by first train the network with $\mathcal{L}_o$ with 5 epochs and then fine-tune it with the full loss for 1 epoch. We provide ablation study of each term in our experiments.

5 Experiments

In this section, we introduce implementation details, datasets, evaluation metrics. An ablation study of how much each component of the framework contributes and a performance comparison with other supervised or unsupervised methods are also presented.

5.1 Implementation details.

Our framework is implemented with publicly available TensorFlow (Abadi et al. 2016) platform and has 34 million trainable variables in total. During training, Adam optimizer is applied with parameters $\beta_1 = 0.9$, $\beta_2 = 0.000$, $\epsilon = 10^{-8}$. Learning rate and batch size are set to be 2×10^{-3} and 4 respectively. Batch normalization (Ioffe and Szegedy 2015) is not used as we didn't observe a performance improvement with it. Following (Zhou et al. 2017), we use the same loss balancing for λ_s, λ_m, and correct the depth by a scale factor. We set $\lambda_n = 1$ and $\lambda_g = \lambda_s$.

The length of input sequence is fixed to be 3 and the input frames are resized to 128×416. The middle frame is treated as the target image and the other two are source images. Our network starts to show meaningful results after 3 epochs, and converges at the end of the 5th epoch. With a Nvidia Titan X (Pascal), the training process takes around 6 hours. The number of epochs and absolute time needed is much less than (Godard, Mac Aodha, and Brostow 2017) (50 epochs, 25 hours) and (Zhou et al. 2017) (15 epochs).

5.2 Datasets and metrics

Training. Theorectically, our framework can be trained on any frame sequences captured with a monocular camera. To better compare with other methods, we evaluate on the popular KITTI 2015 (Geiger, Lenz, and Urtasun 2012) dataset. It is a large dataset suite for multiple tasks, including optical flow, 3D object detection, tracking, and road segmenations, *etc*. The raw data contains RGB and gray-scale videos, which are captured by stereo cameras from 61 scenes, with a typical image size of 1242×375.

In our experiments, videos captured by both left and right cameras are used for training, but treated independently. We follow the same training sequences as (Zhou et al. 2017; Eigen, Puhrsch, and Fergus 2014), excluding frames from test scenes and static sequences. This results in 40,109 trainig sequences and 4431 validation sequences. Different from (Godard, Mac Aodha, and Brostow 2017), no data augmentation has been performed.

Testing. There are two sets of KITTI 2015 test data: (1) Eigen split contains 697 test images proposed by (Eigen, Puhrsch, and Fergus 2014); (2) KITTI split contains 200 high-quality disparity images provided as part of official KITTI

Table 1: Depth performance of our framework variants on the KITTI split.

Methods	Lower the better				Higher the better		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours (no d-n)	0.208	2.286	7.462	0.297	0.693	0.875	0.948
Ours (smooth no gradient)	0.189	1.627	7.017	0.280	0.713	0.891	0.957
Ours (no img grad for d-n)	0.179	1.566	7.247	0.272	0.720	0.895	0.959
Ours (no normal smooth)	0.172	1.559	6.794	0.252	0.744	0.910	0.969

training set. To better compare with other unsupervised and supervised methods, we present evaluations on both splits.

The depth ground truth of Eigen split is generated by projecting 3D points scanned from Velodyne laser to the camera view. This produces depth values for less than 5% of all pixels in the RGB images. To be consistent when comparing with other methods, the same crop as in (Eigen, Puhrsch, and Fergus 2014) is performed when testing. The depth ground truth of KITTI split contains sparse depth map with CAD models in place of moving cars. It provides better quality depth than projected Velodyne laser scanned points but has ambiguous depth value on object boundaries where the CAD model doesn't align with the images. The predicted depth is capped at 80 meters as in (Godard, Mac Aodha, and Brostow 2017) and (Zhou et al. 2017).

The normal ground truth for two splits is generated by applying our depth-to-normal layer on inpainted depth ground truth, where the same inpainting algorithm as (Silberman et al. 2012) is used. For both depth and normal, following (Eigen, Puhrsch, and Fergus 2014), only the pixels with laser ground truth are used.

Metrics. We apply the same depth evaluation and normal evaluation metrics as in (Eigen and Fergus 2015). For depth evaluation, we use the code provided by (Zhou et al. 2017) and for normal, we implement ourselves and verified the correctness through validating normal results of (Eigen and Fergus 2015) over the NYUv2 dataset.

5.3 Ablation study

To investigate different components proposed in Sec. 4, we perform an ablation study by removing each one from our full model and evaluating on the KITTI split.

Depth-normal consistency. By removing normal-to-depth layer (Eq. (7)), the inverse warping process (Sec. 3) takes an image and directly predicted depth map from the input. We show the performance at the row "Ours (no d-n)" in Tab. 1. It is much worse than our full model on Kitti shown in Tab. 2. Notice that with depth-normal consistency, the network not only performs better but converges faster. In fact, our full model converges after 5 epochs, while the network without such consistency converges at 15th epoch.

Image gradient in smoothness term. To validate image gradient for depth and normal smoothness in Eq. (3), we setting $\alpha = 0$. The results is shown as "Ours (smooth no gradient)" in Tab. 1. It makes less impoact than depth-normal consistency, but still helps the performance.

Image gradient in normal-depth consistency. We set $\omega = 1$ in Eq. (5), thus there is no edge awareness in depth-normal consistency. As show at row "Ours (no img grad for n-d)", the results are again worse than our final results, which demonstrates the effectiveness by only enforcing the consistency between color similar pixels.

Normal smoothness. Finally, by removing normal smoothness $\mathcal{L}_n$ in Eq. (8), we show the results at row "Ours (no normal smooth)" in Tab. 1, where it makes less impact for depth than other components, while still make reasonable contributions. However, it makes relatively more contributions for normal performance as shown in Tab. 3.

5.4 Comparison with other methods

To compare with other state-of-the-arts, we show performances on both KITTI and Eigen split. The depth evaluation results are shown in Tab. 2. Our method outperforms some supervised methods *e.g.*(Eigen, Puhrsch, and Fergus 2014), (Liu et al. 2016) and unsupervised methods (Zhou et al. 2017), (Kuznietsov, Stuckler, and Leibe 2017), while slightly worse than (Godard, Mac Aodha, and Brostow 2017) and (Kuznietsov, Stuckler, and Leibe 2017). It is worth noting that (Kuznietsov, Stuckler, and Leibe 2017) utilizes the depth ground truth and (Godard, Mac Aodha, and Brostow 2017) takes stereo image pairs as input, which implies the camera motion is known. On KITTI test split, our method outperforms (Godard, Mac Aodha, and Brostow 2017) on the "Sq Rel" metric. As "Sq Rel" penalizes large depth error, due to regularization, our results has much less outlier depths. Finally, we show some qualitative results in Fig. 5.

To the best of our knowledge, there is no work reporting normal performance on the KITTI dataset. We thus compare the our normal predictions with that computed from the depth maps predicted by (Zhou et al. 2017). As shown in Tab. 3, our method outperforms the baseline under all metrics. Additionally, to ensure the model is learned reasonably, we set up two naive baselines. "Ground truth normal mean" is that we set a mean normal direction for all pixels using ground truth normals. "Pre-defined scene" is that we separate the image to 4 parts using 4 lines connecting each image corder and image center. We set the bottom part having up-directed normal, left part having right-directed normal, right part having left-directed normal and top part with outward normals. Both of the baselines are significantly worse than our predicted model, demonstrating the correctness of the learned model.

Indoor scene exploration. Besides the outdoor dataset, we also explore applying our framework on the indoor scenes: NYUv2 dataset (Silberman et al. 2012). We use a subset for some preliminary experiments. Specifically, "study room" is picked and split for training and testing. We first try with our baseline method (Zhou et al. 2017), and it fails to predict any reasonable depth maps. One possible explanation is that as there are many planes, with low texture and uniform colors in

Table 2: Single view depth test results on Eigen split (upper part) and KITTI split(lower part). All methods in this table use KITTI dataset for traning and the test result is capped in the range 0-80 meters. Test result on KITTI test split of Zhou et al. 2017 is generated by using their released code to train on KITTI dataset only.

Method	Test data	Supervision		Lower the better				Higher the better		
		Depth	Pose	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train set mean	Eigen split	✓		0.403	5.530	8.709	0.403	0.593	0.776	0.878
(Eigen, Puhrsch, and Fergus 2014) Coarse		✓		0.214	1.605	6.563	0.292	0.673	0.884	0.957
(Eigen, Puhrsch, and Fergus 2014) Fine		✓		0.203	1.548	6.307	0.282	0.702	0.890	0.958
(Kuznietsov, Stuckler, and Leibe 2017) supervised		✓		0.122	0.763	4.815	0.194	0.845	0.957	0.987
(Kuznietsov, Stuckler, and Leibe 2017) unsupervised			✓	0.308	9.367	8.700	0.367	0.752	0.904	0.952
(Godard, Mac Aodha, and Brostow 2017)			✓	0.148	1.344	5.927	0.247	0.803	0.922	0.964
(Zhou et al. 2017)				0.208	1.768	6.856	0.283	0.678	0.885	0.957
Ours				0.182	1.481	6.501	0.267	0.725	0.906	0.963
Train set mean	KITTI split	✓		0.398	5.519	8.632	0.405	0.587	0.764	0.880
(Godard, Mac Aodha, and Brostow 2017)			✓	0.124	1.388	6.125	0.217	0.841	0.936	0.975
(Vijayanarasimhan et al. 2017)				-	-	-	0.340	-	-	-
(Zhou et al. 2017)				0.216	2.255	7.422	0.299	0.686	0.873	0.951
Ours				0.1648	1.360	6.641	0.248	0.750	0.914	0.969

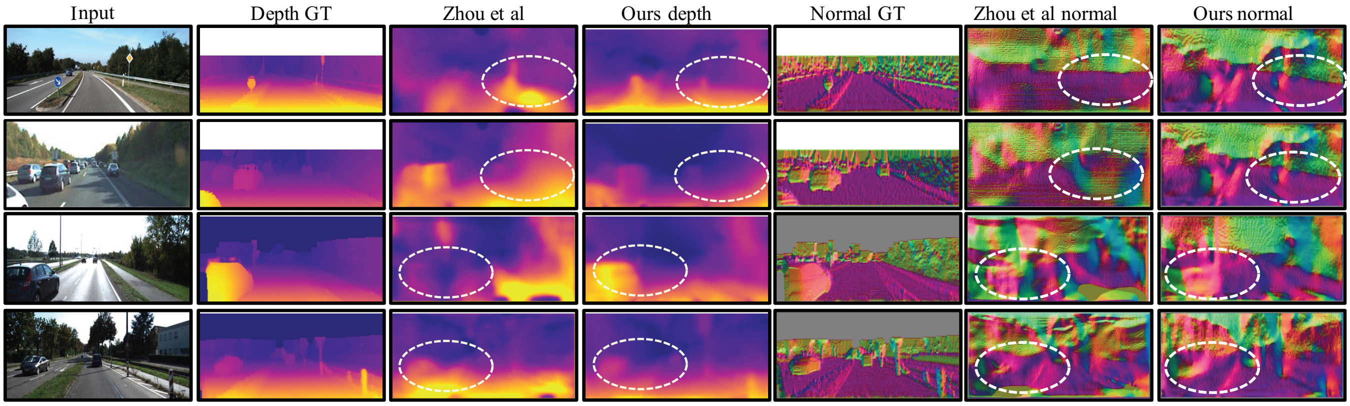

Figure 5: Visual comparison between (Zhou et al. 2017) and ours. We use the interpolated ground truth depths and reshape the image for better visualization. For both depths and normals, our results have less artifacts, reflect the scene layouts much better (as circled in the 1st and 2nd row) and preserve more detailed structures such as cars (as circled in the 3rd and 4th row).

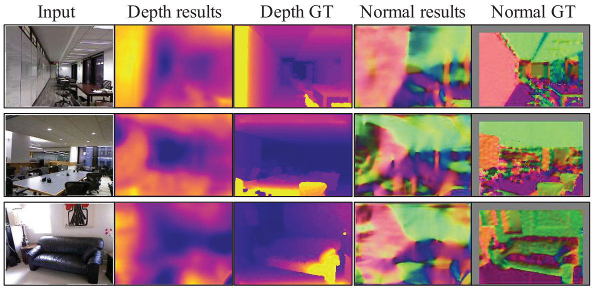

Figure 6: Qualitative results of our framework on a subset of NYU v2 dataset.

Table 3: Normal performances of our method and some baseline methods.

Method	Mean	Median	11.25°	22.5°	30°
Ground truth normal mean	72.39	64.72	0.031	0.134	0.243
Pre-defined scene	63.52	58.93	0.067	0.196	0.302
(Zhou et al. 2017)	50.47	39.16	0.125	0.303	0.425
Ours w/o normal smoothness	49.30	36.83	0.138	0.343	0.436
Ours	47.52	33.98	0.149	0.369	0.473

the indoor scenes, the color matching can fail. Besides color matching, we also add image gradient matching in our loss term, which helps match the plane boundary.

However, as shown in Fig. 6, our framework performs reasonably good on scenes that have multiple intersecting planes. Nevertheless, we still fail on scenes that have only a clutter of object (bottom row of Fig. 6). In the future, we plan to explore more on stronger feature matching rather than just using color matching, which may facilitate the learning under cluttered scenes.

6 Conclusion

In this paper, we propose an unsupervised learning framework for both depth and normal estimation via edge-aware depth-normal consistency. Our novel depth-normal regularization enforces the geometry consistency between different projections of the 3D scene, improving evaluation performances and also the training speed. We present ablation experiments exploring each component of our framework and also on different scenes of images. Our results are even better than some supervised methods, and achieve state-of-the-art performance among methods using monocular videos for training.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Eigen, D., and Fergus, R. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*.

Eigen, D.; Puhrsch, C.; and Fergus, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*.

Garg, R.; G, V. K. B.; and Reid, I. D. 2016. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *ECCV*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*.

Godard, C.; Mac Aodha, O.; and Brostow, G. J. 2017. Unsupervised monocular depth estimation with left-right consistency.

Hoiem, D.; Efros, A. A.; and Hebert, M. 2007. Recovering surface layout from an image. In *ICCV*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2017–2025.

Jia, Z. 2006. Using cross-product matrices to compute the svd. *Numerical Algorithms* 42(1):31–61.

Karsch, K.; Liu, C.; and Kang, S. B. 2014. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence* 36(11):2144–2158.

Kong, N., and Black, M. J. 2015. Intrinsic depth: Improving depth transfer with intrinsic images. In *ICCV*.

Kuznietsov, Y.; Stuckler, J.; and Leibe, B. 2017. Semi-supervised deep learning for monocular depth map prediction.

L. Ladicky, Zeisl, B.; Pollefeys, M.; et al. 2014. Discriminatively trained dense surface normal estimation. In *ECCV*.

Ladicky, L.; Shi, J.; and Pollefeys, M. 2014. Pulling things out of perspective. In *CVPR*.

Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; and Navab, N. 2016. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, 239–248. IEEE.

Li, B.; Shen, C.; Dai, Y.; van den Hengel, A.; and He, M. 2015. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*.

Li, Y. 2017. Pyramidal gradient matching for optical flow estimation. *arXiv preprint arXiv:1704.03217*.

Liu, F.; Shen, C.; Lin, G.; and Reid, I. 2016. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence* 38(10):2024–2039.

Liu, F.; Shen, C.; and Lin, G. 2015. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*.

Mur-Artal, R.; Montiel, J. M. M.; and Tardos, J. D. 2015. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics* 31(5):1147–1163.

Newcombe, R. A.; Lovegrove, S.; and Davison, A. J. 2011. DTAM: dense tracking and mapping in real-time. In *ICCV*.

Prados, E., and Faugeras, O. 2006. Shape from shading. *Handbook of mathematical models in computer vision* 375–388.

Schwing, A. G.; Fidler, S.; Pollefeys, M.; and Urtasun, R. 2013. Box in the box: Joint 3d layout and object reasoning from single images. In *ICCV*.

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from rgbd images. *ECCV*.

Srajer, F.; Schwing, A. G.; Pollefeys, M.; and Pajdla, T. 2014. Match box: Indoor image matching via box-like scene estimation. In *3DV*.

Vijayanarasimhan, S.; Ricco, S.; Schmid, C.; Sukthankar, R.; and Fragkiadaki, K. 2017. Sfm-net: Learning of structure and motion from video. *CoRR* abs/1704.07804.

Wang, P.; Shen, X.; Lin, Z.; Cohen, S.; Price, B. L.; and Yuille, A. L. 2015. Towards unified depth and semantic prediction from a single image. In *CVPR*.

Wang, P.; Shen, X.; Russell, B.; Cohen, S.; Price, B. L.; and Yuille, A. L. 2016. SURGE: surface regularized geometry estimation from a single image. In *NIPS*.

Wang, X.; Fouhey, D.; and Gupta, A. 2015. Designing deep networks for surface normal estimation. In *CVPR*.

Wu, C. 2011. Visualsfm: A visual structure from motion system.

Xiao, J.; Owens, A.; and Torralba, A. 2013. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV*.

Zhou, T.; Brown, M.; Snavely, N.; and Lowe, D. G. 2017. Unsupervised learning of depth and ego-motion from video. In *CVPR*.