

Unsupervised Learning of Geometry with Edge-aware Depth-Normal Consistency

Paper ID: 1138

Abstract

Learning to reconstruct 2.5D geometry from a single image in an unsupervised manner with deep convolutional network (DCN) is attracting significant attention in recent years, due to that it can be widely applied to unlabeled videos online for applications such as augmented reality etc. In this paper, to better recover the geometry inside an image, we propose to jointly estimate depth and normal in the unsupervised learning pipeline. Our estimated depth is guaranteed to be consistent with predicted normals in 3D space, yielding much more robust results for both depth and normals. Specifically, we introduce two layers, i.e. a depth to normal layer and a normal to depth layer. The depth to normal layer takes estimated depth for each pixel as input, and compute normal directions with inner product. Then given the normal and depth, the normal to depth layer outputs a regularized depth through local planar smoothness. Finally, to train the network we apply the photometric loss, and further require gradient smoothness for both depth and normals predictions. We conducted experiments on both outdoor (KITTI) and indoor (NYUv2) videos, and show that our algorithm vastly outperform the state-of-arts, reducing both depth and normal error over 20% relatively, which demonstrates the benefits from the normal representation.

1 Introduction

Human beings are professional in recovering the 3D geometry of observed natural scenes at very detailed level in real time, even from a single image. More impressively, we leaned to achieve this ability only through visual perception of consecutive changing of outside world and ego-motion. Practically, being able to do reconstruction through unlabeled videos can be widely applied to large amount of real applications like augmented reality, robotics etc..

Therefore, letting computer manage dense 3D reconstruction ability by watching videos is the central problem of computer vision. One group of approaches is geometric based method depend on feature matching, e.g.structure from motion (SFM) (?) etc., or color matching, e.g.DTAM (?), etc.. Since our world exists in 3D space and the images projection follows camera geometry, as long as the matching from corresponding frames are correct, we can exactly

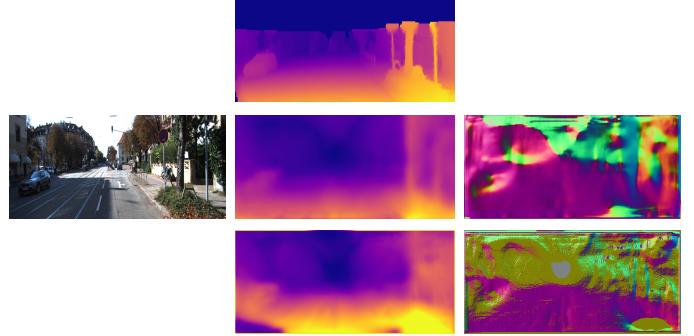


Figure 1: Comparison of visual outputs of our model with and without depth-normal consistency module. Top to bottom: ground truth depth and normal, our results with depth-normal consistency, our results without depth-normal consistency.

solve the 3D structure. However, theoretically, those methods does not explain why human can also do reconstruction from a single image by observing videos. In addition, they can easily fail once the feature matching is wrong, e.g.when SIFT (?) feature meets a wall with white color. In summary, geometric based methods do not have the ability to discover new reconstruction cues using videos, and ignore the information from monocular images.

Another way to do 3D reconstruction is learning based method, where the reconstruction cues can be incrementally discovered and applied by keeping feeding in unobserved videos. Currently, with the development of pixel-wise prediction via deep learning such as fully convolutional network (FCN) (?), supervised learning of depth, e.g.(?; ?), achieved impressive results over public datasets like KITTI (?), NYUv2 (?) and SUN3D (?). Nevertheless, collecting ground truth depth is almost impossible for most online videos, where the learned models are hard to generalize. Thus, learning depth with unlabeled videos attracts lots of attention in recent years. Garg *et al.*(?) try to use FCN to predict depth from a single image, by using photometric matching from stereo pairs, and back-propagate the matching error to the network. Later works (?; ?) extends to using information from consecutive video frames by introducing camera ego-motion. Although those unsupervised

learning based methods are able to do single image reconstruction, the results are still far from satisfactory. As shown at the left of Fig. ??, the depth results from (?) does not well represent the structure of scene, especially when visualized with computed normals. This is mostly due to photometric matching is ambiguous, *i.e.* a color in source frames can be matched to multiple similar colors in target frames. Although researchers usually use smoothness of depth (?) to reduce the ambiguity, it is often a weak constraint over neighbor pixels, yielding non-satisfied normal results.

Our work falls in the scope of learning based 3D reconstruction with videos following the work of (?), but is a step further towards learning a regularized 3D geometry with particular awareness of normal representation. We are motivated by the fact that human beings are able to explicitly point out the normal direction of each pixel in an image. Actually, we are more sensitive to normal than depth, *e.g.* one could precisely point out the normal direction of a pixel while could roughly know the exact depth. Thus, we embed the normals representation inside the network as regularization layers, and enforce the prediction consistency with depth, which we will describe in Sec. ???. There are several advantages of inducing normal representation. First, it gives explicit understanding of normal for learned models. Second, it provides higher order interaction between depth, which is beyond local neighbor relationships. Last, additional operations, *e.g.* Manhattan assumption, over normals can be further integrated. As shown at right of Fig. ??, with the help of normal representation, our recovered geometry is comparably better. We did extensive experiments over the publish KITTI and NYUv2 dataset for 3D reconstruction, 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, we converge much faster. These demonstrate the efficiency and effectiveness of our approach.

1.1 Framework

Fig. ?? illustrate an overview of our learning approach. For training, we apply supervision from view synthesis following (?). 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 maps D_t^n are 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. ??). The new depth map, combined with poses and mask predicted from the pose network (left), are then used to inverse warp the source views to reconstruct the target view, where errors are back propagate 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 use smoothness over normal, which induce higher order interaction between pixels (Sec. ??)

After the model is trained, given a new image, we first in-

fer per-pixel depth value and then compute the normal value, yielding consistent prediction between the two.

2 Related Work

Structure from feature matching and single view geometry. As discussed in Sec. ??, geometry based methods, such as SFM (?), ORB-SLAM (?), DTAM (?), are relying on feature matching, which could be effective and efficient in many cases. However, they can fail at low texture, or drastic change of visual perspective *etc.*. More importantly, it can not extend to single view reconstruction where humans are good at. Traditionally, specific rules are developed for single view geometry. Methods are dependent on either computing vanishing point (?), following rules of BRDF (?), or abstract the scenes with major plane and box representations (?; ?) *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 (?; ?; ?). The learned CNN model show significant improvement comparing to other strategies based on hand-crafted features (?; ?; ?). Others tried to improve the estimation further by appending a conditional random field (CRF) (?; ?; ?). However, most works regard depth and normal predictions as independent tasks. (?) point out their correlations over large planar regions, and regularize the prediction using a densely CRF (?), which improved the results on both depth and normal. However, all those methods are requiring dense 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 hold much 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 (?) learns to generate right view from given left view by supervision of a stereo pair. In order to do back-propagation to depth values, it quantizes the depth space and trains to select the right one. Concurrently, (?) applied the similar supervision from stereo pairs, while the depth is kept continuous. They apply Taylor expansion to approximate the gradient for depth. (?) extend Garg’s work by including depth smoothness loss and left-right depth consistency. Most recently, (?) induces camera pose estimation into the training pipeline, which makes depth learning possible from monocular videos. However, they only focus on rigid scene without the ability to deal with moving objects. At the same time, (?) proposed a network to include modeling rigid object motion. Although vastly developed for depth estimation from video, normal information, which is also highly interested for geometry prediction, has not been considered inside the pipeline. This paper fills in the missing part, and show that

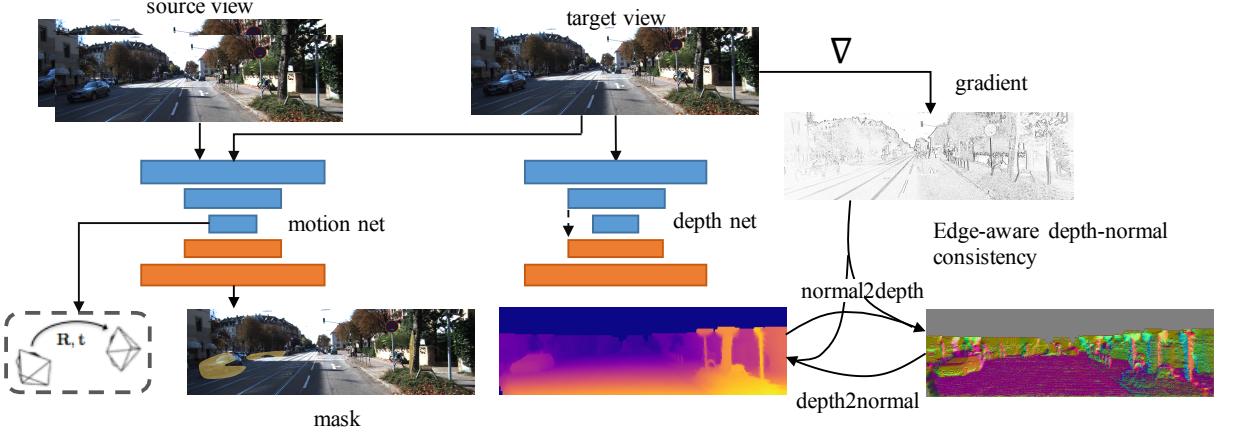


Figure 2: Framework.

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 (?) 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 (?; ?). The core idea behind, as discussed in ??, is inverse warping from target view to source view with awareness of 3D geometry, as illustrated in Fig. ??(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 $\mathbf{T}_{t \rightarrow s} = \{\mathbf{R}|t\} \in \mathcal{SE}(3)$, and $h(x)$ is the homogeneous coordinate given x . The perspective warping to localize corresponding pixels can be formulated as,

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

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 geometry.

Photometric error from view synthesis. Given pixel matching pairs between target and source view, *i.e.* I_t and I_s , we can synthesis a target view \hat{I}_s from the given source view through bilinear interpolation (?), as illustrated in Fig. ??(b). Then, under the assumption of Lambertian and a static rigid scene, average photometric error is often used to recover the depth map D for the target view and the relative pose, *e.g.* (?). However, as pointed out by (?), this assumption is not always true, due to the fact of existing moving object and occlusion. Thus, an explainability mask \mathbf{M} is induced

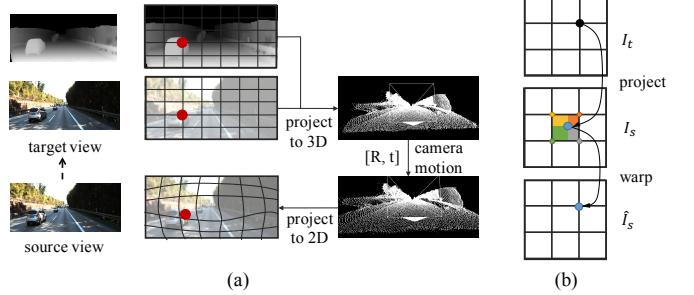


Figure 3: Illustration of (a) 3D inverse warping and (b) bilinear interpolation.

to leverage the issue. Formally, the masked photometric error is,

$$\begin{aligned} \mathcal{E}_{vs}(D, \mathcal{T}, \mathcal{M}) &= \sum_{s=1}^S \sum_{x_t} \mathbf{M}_s(x_t) |I_t(x_t) - \hat{I}_s(x_t)|, \\ \text{s.t. } d(x) > 0, \forall \mathbf{M}_s(x) \in [0, 1] \end{aligned} \quad (2)$$

where $\{I_s\}_{s=1}^S$ is the set of 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. ??, supervision based solely photometric error is ambiguous. One pixel could match to multiple candidates, especially at low-texture regions. In addition, there is trivial solution for explainability mask by setting all value to zero. Thus, to reduce depth ambiguity and encourage non-zero of masks, two regular-

ization terms are applied,

$$\begin{aligned}\mathcal{E}_s(D, 2) &= \sum_{x_t} \sum_{d \in x, y} |\nabla_d^2 D(x_t)| e^{-\alpha|\nabla_d I(x_t)|} \\ \mathcal{E}_m(\mathcal{M}) &= -\sum_s \sum_{x_t} \log P(M_s(x_t) = 1)\end{aligned}\quad (3)$$

$\mathcal{E}_s(D, 2)$ is a spatial smoothness term penalizes L1 norm of second-order gradients of depth along both x and y direction, encouraging depth value align in planar surface when no image gradient appears. Here, the number 2 represents the order of gradient w.r.t. the input.

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

$$\begin{aligned}\mathcal{E}_o(\{D_l\}, \mathcal{T}, \mathcal{M}) &= \sum_l \{\mathcal{E}_{vs}(D_l, \mathcal{T}, \mathcal{M}) + \lambda_s \mathcal{E}_s(D_l) \\ &\quad + \lambda_m \mathcal{E}_m(\mathcal{M}_l)\}\end{aligned}\quad (4)$$

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 (?), which supervises the learning process.

4 Geometry estimation with edge aware depth-normal consistency

In our scenario, given target image I , we aim to learn to estimate both depth and normal simultaneously. Formally, let N be the predicted normal from our model, we embed it into the training pipeline and make it a stronger regularization for estimating depth D , which help to train more robust model.

4.1 Depth and normal orthogonality.

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

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

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 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 linear equation for pixel x_j which we will elaborate later.

However, as introduced in Sec. ??, most supervised works try to predict the two information independently without considering their correlations, SURGE (?) 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 applied to network output, which better benefits the model learning. Specifically, to model their consistency, we developed two layers by solving Eq. (??), i.e. depth to normal layer and normal to depth layer.

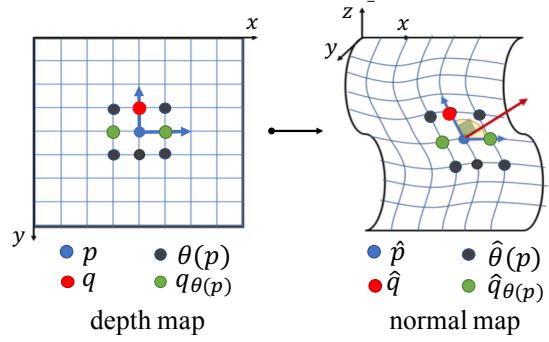


Figure 4: Depth2normal layer. $p, q, \theta(p), q_{\theta(p)}$ are 2D points, and $\hat{p}, \hat{q}, \hat{\theta}(p), \hat{q}_{\theta(p)}$ are corresponding points projected to 3D space.

Infer normal from depth. Given a depth map D , for each point x_i , in order to get $N(x_i)$ by solving Eq. (??), we need to firstly define $\mathcal{N}(x_i)$ and ω_{ji} , and then solve the set of linear equations. To deal with the first issue, we choose using the 8-neighbor convention to compute normal directions, which considerably more robust than 4-neighbor convention. However, it is not always good to equally weight all pixels due to depth discontinuity or dramatic normal changes may occur nearby. Thus, for computing ω_{ji} , we set it large neighboring pixels x_j having similar color with x_i , while smaller otherwise. Formally, $\omega_{ji} = \exp\{-\alpha|I(x_j) - I(x_i)|\}$ and $\alpha = 0.1$ in our case.

For minimizing Eq. (??), one may apply a standard singular value decomposition (SVD) to obtain the solution. However, in our case, we want to embed such an operation in the network training and back-propagate the gradient respect to input depth. SVD is computationally non-efficient for back-propagation. Thus, we choose to use mean cross-product to approximate the minimization (?), 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 w.r.t. x_i , and in a counter clock-wise order, i.e. $\mathcal{P}(x_i) = \{([m-1, n], [m, n+1]), ([m+1, n-1], [m-1, n-1])\}$. Then, for each pair, cross product of their difference vector w.r.t. x_i is computed, and the mean of computed vector is assign to the normal direction of x_i . Formally, the solver for normal is written as,

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

We show the process of calculating normal direction in Fig. ??.

Expect depth from normal. Due to the fact that we do not have ground truth normal for supervision, it is necessary to recover depth from normal to receive the supervision from photometric error as discussed in Sec. ???. To recover depth, given normal map N , we still need to solve Eq. (??). However, there is no unique solution. Thus, for simplicity,

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 neighboring pixel 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 expected 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 from its 8 neighbors. Finally, by minimum square estimation, the solution for the expected depth value is,

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

4.2 Training losses

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

Additionally, with normal representation, we additionally require smoothness over neighboring normal values, which provides high order interactive between pixels. Formally, the smoothness for normal haves the same form as \mathcal{E}_s in Eq. (??), while we apply the first order gradient, *i.e.* $\mathcal{E}_s(N, 1)$.

Last but not the least, matching corresponding pixels between frames is another central factor to find correct geometry. Additional to photometric error from matching pixel colors, matching image gradient is more robust to lighting variations, which was frequently applied in computing optical flow (?). 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,

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

In our experimental section, we observe additional benefits by having this match, especially at indoor scenario, where colors are often homogeneous on walls. In the future, we hope to investigate more in matching criteria such as with stronger descriptors like SIFT (?) or higher level convolutional features.

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

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

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

Training the model. For network architectures, similar to (?) and (?), we adopt the DispNet (?) architecture with skip connections as in (?). 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{E}_o with 5 epochs and then fine-tune it with the full loss for 1 epoch. We will provide ablation study of each term in our experiments.

5 Experiments

In this section, we introduce the training 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 present.

5.1 Implementation details.

Our framework is implemented with publicly available TensorFlow (?) platform and has 34 million trainable variables in total. During training, batch normalization (?) is used for all layers except for output layer. Adam optimizer is implemented 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.

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 target image and the other two are source images. With the settings above, the network starts to show meaningful results after 3 epochs of training, and converge at the end of 5th epoch. On a Nvidia Titan X (Pascal) GPU, the training process takes around 6 hours. The number of epochs and absolute time needed for convergence is much less than (?) (50 epochs, 25 hours) and (?) (15 epochs).

As the predicted depth is not absolute value but defined up to a scale, we correct the scale factor by enforcing the predicted median matching the ground truth median. That is, multiplying the predicted depth by a factor: $\hat{f} = \text{median}(D_{gt}) / \text{median}(D_{pred})$.

5.2 Datasets and metrics

Training. Theoretically, our framework can be trained on frame sequences captured with a monocular camera. To better compare with other methods, we train the framework on popular KITTI 2015 (?) dataset. KITTI 2015 dataset is a large dataset suite for multiple tasks, including optical flow, 3D object detection and tracking, semantic segmentations, etc. All RGB and gray-scale videos are captured by stereo cameras from 61 scenes, with a typical image being 1242×375 in original size.

Raw videos captured by both left and right cameras are downloaded, which are treated independently for training. The same training frame sequences as in (?) are used for training: training split used by Eigen *et al.*(?) excluding frames from test scenes and static sequences. This results in a total of 40,109 training sequences and 4431 validation sequences. Different from (?), no other data augmentation has been performed.

Testing. There are two splits of KITTI 2015 test data: (1) KITTI split contains 200 high-quality disparity images provided as part of official KITTI training set; (2) Eigen split contains 697 test images proposed by (?). To better compare with other unsupervised and supervised methods, we present evaluation methods on both two splits.

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 (?) and (?). 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 (?) is implemented when testing.

The normal ground truth for two splits is generated by applying depth2normal layer as described in ?? on inpainted depth ground truth. The inpainting algorithm by (?) has been used. The inpainting depth and normal results are shown in later sections for visualization. However, for evaluation, only the sparse points with depth ground truth are used.

Metrics. The same depth evaluation and normal evaluation metrics as in (?) and (?) have been implemented.

$$\text{Abs Rel: } \frac{1}{|D|} \sum_{d_{pred} \in D} |d_{gt} - d_{pred}| / d_{gt}$$

$$\text{Sq Rel: } \frac{1}{|D|} \sum_{d_{pred} \in D} \|d_{gt} - d_{pred}\|^2 / d_{gt}$$

$$\text{RMSE: } \sqrt{\frac{1}{|D|} \sum_{d_{pred} \in D} \|d_{gt} - d_{pred}\|^2}$$

$$\text{RMSE log: } \sqrt{\frac{1}{|D|} \sum_{d_{pred} \in D} \|\log d_{gt} - \log d_{pred}\|^2}$$

$$\delta < thr: \% \text{ of } d_{pred} \in D, \text{ s.t. } \max\left(\frac{d_{gt}}{d_{pred}}, \frac{d_{pred}}{d_{gt}}\right) < thr$$

$$\text{mean: } \frac{1}{|N|} \sum_{n_{pred} \in N} (n_{gt} \cdot n_{pred})$$

$$\text{median: } \text{median}([(n_{gt} \cdot n_{pred})]_{n_{pred} \in N})$$

$$\text{degree: \% of } n_{pred} \in N, \text{ s.t. } (n_{gt} \cdot n_{pred}) < \text{degree}$$

5.3 Ablation study

An ablation study is conducted to investigate how much each component contributes to the final performance, evaluated on the KITTI split.

Depth and normal geometry consistency. The impact of adding depth-normal consistency is explored by removing normal2depth layer in the framework. The inverse warping process introduced in Sec. ?? takes input image and directly predicted depth map as input. The performance of framework trained without normal2depth layer is shown as the row “Ours (no d-n)” in Tab. ???. Besides the performance gain after adding depth and normal consistency, the network converges considerably faster compared to without leveraging such consistency: the full network converges after 5 epochs of training and the network without such consistency regularization converges at 13th epoch.

Image gradient in smoothness term. We explore the effectiveness of adding image gradient into smoothness term. By setting $\alpha = 0$ in the edge-aware smoothness loss, the image gradient has no effect on the weight of smoothness

loss. The results of this variant is shown as “Ours (smooth no gradient)” in Tab. ??.

Image gradient in normal2depth layer. When setting the $\alpha = 0$ in normal2depth layer, the normal direction contributes equally to each “shifted” depth map. With image gradient in normal2depth layer, the normal direction will only contribute to those neighboring points q where there is small image gradient between central point p and q , i.e. the two points most likely lie on the same plane. With such constraint, the depth evaluation performance is better.

Normal smoothness. We explore the impact of normal smoothness term by evaluating normal performance comparing framework with and without normal smoothness loss term. The visualization results are shown in Fig. ???. When trained with normal smoothness term, the network generates more reasonable results both qualitatively and quantitatively.

5.4 Comparison with other methods

To compare with other supervised and unsupervised methods, our framework is evaluated on both KITTI and Eigen split. The depth evaluation results are shown in Tab. ???. Our method outperforms some unsupervised methods (?) and (?) and even outperforms some supervised methods (?) and (?).

Our performance is prior to two methods, (?) and (?) (supervised, semi-supervised). It is worth noting that (?) (supervised, semi-supervised) utilizes the depth ground truth and (?) takes stereo image pairs as input, which implies the camera motion between two images are known. On both two test splits, our method outperforms (?) on the Sq Rel metric. As Sq Rel metric penalizes large depth difference, our method generates depth maps without much outlier depths.

Some qualitative depth estimation results are shown in Fig. ??.

To our knowledge, there has not been works that report normal performance on KITTI 2015 dataset. We thus compare the our normal performance with normals generated by applying depth2normal layer on depth maps of two publicly available unsupervised methods (?) and (?). As shown in Tab. ???, our method outperforms both methods under all metrics.

6 Conclusion

In this paper, we propose an unsupervised learning framework for joint depth and normal estimation via edge-aware depth and 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 comparable to some supervised methods, and achieve state-of-the-art performance among unsupervised monocular methods.

Future work. In future work, we will not be limited to unsupervised learning, while being able to stick in supervised

Table 1: Ablation study of each component of the framework on Kitti test 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 (n2d no img grad)	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

Table 2: Single view depth test results on Kitti 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 training the released model 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
(?) Coarse			✓	0.214	1.605	6.563	0.292	0.673	0.884	0.957
(?) Fine			✓	0.203	1.548	6.307	0.282	0.702	0.890	0.958
(?) supervised			✓	0.122	0.763	4.815	0.194	0.845	0.957	0.987
(?) unsupervised			✓	0.308	9.367	8.700	0.367	0.752	0.904	0.952
(?)			✓	0.148	1.344	5.927	0.247	0.803	0.922	0.964
(?)				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
(?)			✓	0.124	1.388	6.125	0.217	0.841	0.936	0.975
(?)			-	-	-	0.340	-	-	-	-
(?)				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

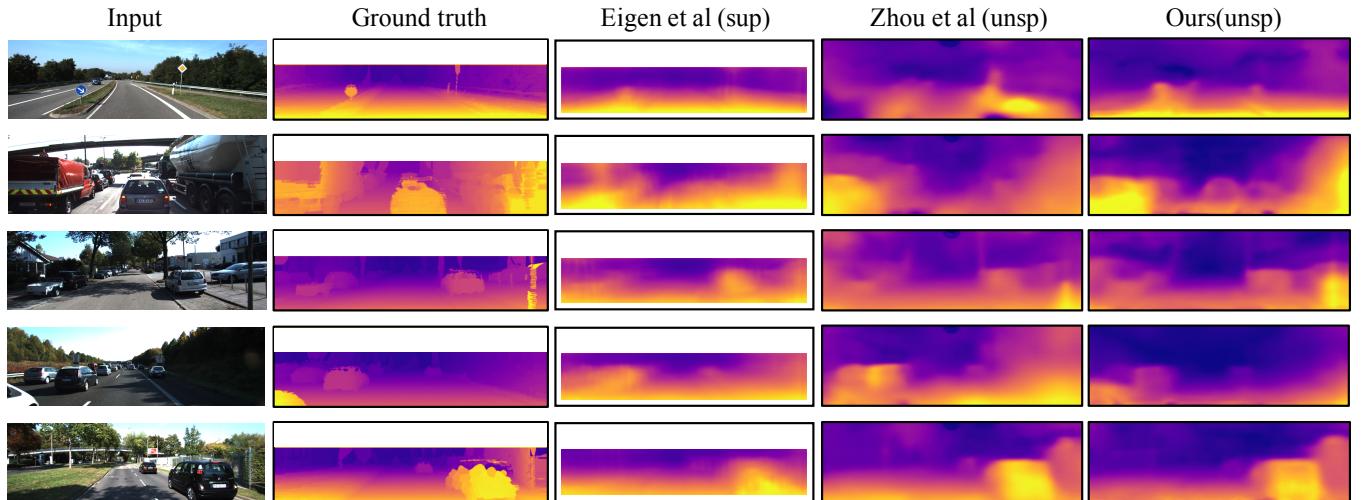


Figure 5: Visual comparison of depth estimation between (?) (supervised with depth ground truth), (?) (unsupervised) and ours (unsupervised). As the original depth ground truth comes from sparse laser measurement, the interpolated depth map is shown for better visualization.

training of joint normal and depth prediction for cross supervision, *e.g.* using normal ground truth to regularize estimated depth.