TripleDNet: Exploring Depth Estimation with Self-Supervised Representation Learning

Ufuk Umut Senturk[1,2]
ufukumutsenturk@gmail.com
Arif Akar[1,2]
arifakar@gmail.com
Nazli Ikizler-Cinbis[1]
nazli@cs.hacettepe.edu.tr

[1] Department of Computer Engineering
Hacettepe University
Ankara, Turkey

[2] ASELSAN MGEO, Inc.
Ankara, Turkey

Abstract

We propose *TripleDNet (Disentangled Distilled Depth Network)*, a multi-objective, distillation-based framework for purely self-supervised depth estimation. We add further objectives to structure-from-motion based estimation to constrain the solution space and to allow feature space disentanglement within an efficient and simple architecture. In addition, we propose a knowledge distillation objective that supports depth estimation in terms of scene context and structure. Surprisingly, we also found out that self-supervised image representation learning frameworks for model initialization outperforms the supervised counterparts. Experimental results show that proposed models trained purely in a self-supervised fashion outperform the state-of-the-art models on the KITTI and Make3D datasets compared to models utilizing ground truth segmentation maps. Codes are available at https://github.com/ufukpage/TripleD.

1 Introduction

Monocular depth estimation is a fundamental problem in computer vision due to its impact on 3D scene understanding and its critical role in practical applications including robotics, health, and autonomous driving. Gathering ground truth labels for this task is a laborious and noisy endeavor, since it requires pixelwise annotations. Recent works try to address this problem by utilizing consecutive video frame information via joint learning of ego-motion and depth prediction. Estimated relative camera transformation and depth maps are used to warp the input frames onto the neighboring frames, which is central to the Structure-from-Motion(SfM) approach [16].

Models relying on assumptions (constant illumination, static world) at the expense of self-supervision based on SfM fail disastrously in some cases, especially in textureless areas. Recent approaches [11, 30] that are masking out stationary or occluded pixels ignore the possibility that substantial signals could be lost, causing the training process to become disrupted. This leads to incorrect depth estimations in those local regions. In order to alleviate this issue, we approach the problem from an *image representation learning* (IRL) view to model scene context and keep the gradients flowing during backpropagation. This context

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

modeling helps the network to infer in a way that similar scenes would likely have similar scene representations, hence similar depth estimations. Thus, for cases where the network is not receiving gradient flowing from reprojection error, additional objectives modeling the scene are expected to provide sufficient gradient flow.

We conjecture that mutual learning of different but related tasks is likely to model good scene representations. One might think that using ground truth segmentation maps or any other scene context prior is beneficial to improve depth estimation [2, 5, 40]. However, this violates the principle of unsupervised learning, where any ground truth information should be assumed to be non-existent. To avoid using any ground truth information, we incorporate self-supervised image representation learning insight within the depth estimation framework. This insight suggests that representations learnt by utilizing pretext objective via pseudo labels should be suitable for various downstream tasks. For instance, to solve a colorization problem, a neural network needs to solve part or patch level correspondence such that pixels on the same semantic patch or part have similar colors. Even though the network does not know the ground-truth semantic label of that patch or region, it has a grasp of integrity and awareness of pixels in the same semantic area.

In the light of these insights, we propose TripleDNet (**D**isentangled **D**istilled **D**epth Network) (and variants) to obtain refined context representations and consequently, depth estimations. In this framework, we couple the depth estimation with self-supervised pretext tasks (such as autoencoding, colorization, and inpainting or masked autoencoding) to capture good semantics and to infer finer image details. We employ suitable self-supervised tasks to distill knowledge via multi-objective training. Combining those objectives naively would not perform best because the depth decoder might be enforced to decode unnecessary scene properties in the entangled latent space. Moreover, more representative features can be obtained by disentangling the scene as appearance and geometry factors[26] through those pretext tasks. Therefore, we propose a framework in which objectives can be jointly optimized thanks to disentangling features onto separate decoders. Both decoders are utilized to take on depth estimation and pretext tasks. Consequently, the final model compensates mentioned side effects while estimating better depth maps, thanks to implicit modelling of scene context that can reason about the relation between depth and latent factors of the scene. In this context, we also investigate self-supervised IRL models [3, 6, 18] for encoder initialization instead of supervised pretraining on ImageNet [7] and demonstrate their effectiveness over supervised models.

Overall, our contributions in this paper can be summarized as follows:

- We propose distillation and disentanglement mechanisms based on joint learning of novel self-supervised pretext tasks and monocular depth estimation.
- To the best of our knowledge, this is the first work to introduce and evaluate self-supervised IRL to self-supervised depth estimation in terms of unsupervised finetuning, which extends the findings of respective studies.
- Experimental results on two benchmark datasets show that the proposed approach is able to achieve state-of-the-art performance in monocular depth estimation in a fully self-supervised fashion.

2 Related Work

2.1 Self Supervised Depth Estimation

Depth estimation is a highly ill-posed problem, especially in monocular settings. One of the seminal works [12] exploits right-left consistency in the stereo camera configuration. Concurrently, another work [52] utilizes neighboring frames to constrain optimization, similar to SfM. Monodepth2[11] method has been developed as a strong baseline that offers multi-scale estimation, auto masking stationary pixels, and minimum projection loss. Following these seminal approaches, many lines of works are later proposed to improve architecture [14, 20, 50] and objectives[38, 39, 46], or enforce extra constraints [1, 32, 44, 49]. Another line of studies[5, 15, 25, 27] employs semantic priors to strengthen scene representation, producing better depth maps by fusing explicit semantic knowledge.

2.2 Knowledge Distillation on Depth Estimation

Knowledge distillation [13] is employed to have a better representation distilled from more complex models to simpler ones. Pilzer *et al.* [36] propose self-consistency and self-distillation based on stereo configuration. Subsequently, [40] jointly optimizes self-supervised optical flow and depth estimation networks with the help of a pre-trained segmentation network, which is later utilized for a self-distilled optical flow network. However, X-Distill[2] proposes distillation from a pre-trained segmentation network by introducing depth to the segmentation task quite similar to ours in terms of distillation. Key differences are that we do not use any ground truth annotations and provide disentanglement structure. [51] present another teacher depth network for distillation while regressing estimation uncertainty. In this work, we distinctively exploit cost-free labels to create better representation space rather than using a teacher network that produces depth maps which is still not good enough to be the target label to supervise distillation loss.

2.3 Self Supervised Image Representation Learning

Self-supervised image representation learning is an unsupervised learning paradigm that attempts to develop universal representations for various tasks using pretext or contrastive objectives. Since the denoising autoencoders [41], input data for unsupervised representation learning has been masked or, in a broader sense, corrupted to reconstruct input or variants of it. The first works to explore self-supervised learning (SSL) for image representation learning (IRL) are [35] using inpainting pretext task, [48] jointly training networks to optimize colorization[47] and greyscaling tasks, [34] solving jigsaw puzzle on image patches, and [10] predicting angles of rotated images. Recently, [19, 28] investigate masked image autoencoding for IRL based on masked language modelling [19, 33]. Furthermore, contrastive learning is becoming a building block for self-supervised learning frameworks due to its power of transferability and accuracy on multiple downstream tasks. This paradigm is primarily concerned with distinguishing instances from one another, optimizing contrastive objectives. MoCo[18], SimCLR [6], and SWaV [3] are some of the leading approaches. In this work, we also make use of them by initializing our models.

3 Method

Our proposed approach consists of two main components: *i)* pretext task distillation, where the estimated depth map is fed to the network that solves the pretext task, and *ii)* disentanglement, where the depth map and appearance reconstruction are separated and depth map is used as a conditional input through another neural network. Variants of our approach can be grouped into two: one is distillation-only methods((M)D2G, (M)DC2G)(Section 3.1) utilizing pretext layers and optimizing SSL losses based on pretext task, and second is TripleD

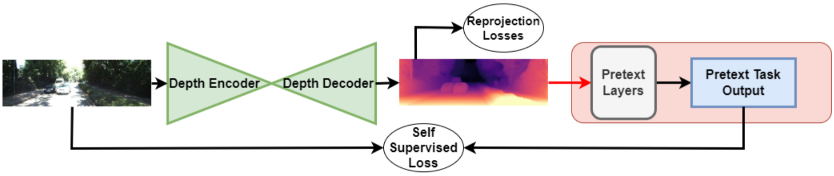


Figure 1: Only distillation-based framework. Depth predictions are forwarded from depth decoder to pretext decoder/layers via distillation connections indicated by red lines. Other skip connections are omitted for brevity.

(Section 3.2) utilizing Pretext Decoder(PD) instead of pretext layers and optimizing autoencoder loss as SSL loss. All the modules are trained in an end-to-end fashion.

3.1 Pretext Tasks Distillation

To distill knowledge from self-supervised objectives and maintain gradient flow, we aim to utilize the direct supervision signals easily extracted from the existing data. For this purpose, we mainly use four pretext tasks to refine representation while backpropagating through pretext network and depth network from self-supervised objective function. Specifically, these pretext tasks are Depth-to-Grey Scale(D2G), Depth and Grey Scale-to-Color(DG2C), Masked D2G(MD2G) and Masked DG2C(MD2C) tasks. Each task is trained and evaluated separately. The overall process is shown in Figure 1. We construct these particular tasks instead of existing self-supervised representation learning tasks such as rotation prediction[10] because of their suitability with pixel generation and the simplicity of the ideas behind them. This is because our primary motivation is not to build a complex model, but to demonstrate that even the simple elements of the IRL are sufficient to build a robust depth estimation framework. Models are illustrated in Figure 2 and pink background of Figure 1. We intentionally use a 2-layer Convolutional Network as pretext layers in this section which will be explained later. Details of one layer block in pretext layers are as follows: Conv$3 \times 3 \times 32$ →BN→ReLU, where Conv$3 \times 3 \times 32$ is 2D convolutional layer with # out channel 32 and kernel size 3×3, BN is batch normalization. Same block is used twice. Third block is a prediction layer that depends on the pretext task.

Depth-to-Greyscale (D2G): The first novel pretext task is Depth-to- Greyscale (D2G). Our intuition is similar to the colorization task, where we assume that pixels in a local neighborhood are likely to belong to the same object, hence, are likely to have similar depth values. However, direct estimation of a color image from only depth estimation would lead to poor performance, because two layers do not have enough capacity to solve that rather complex task and underfit to that task. Therefore, instead of estimating colors, we estimate the greyscale values of pixels which yields a much simpler computational task. The reason we are using simple pretext layers similar to [2] is, high capacity pretext network would weaken gradient flow to the depth network and distillation would not be done at the desired level. Conv$1 \times 1 \times 1$ is employed as prediction head in pretext layers because only greyscale version of RGB input is predicted and the following loss is employed for this task:

$$\mathcal{L}_{d2g}(x) = \sqrt{(PL(D(x)) - GS(x))^2 + \varepsilon^2} \quad (1)$$

where D is depth CNN consisting of depth encoder and depth decoder, PL is 2-layered pretext layer network and estimates greyscale version of RGB input X, and ε is a constant to avoid

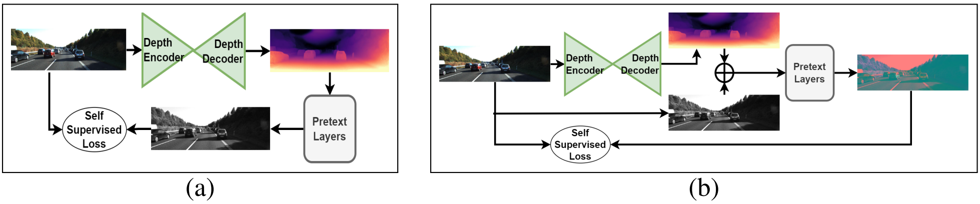


Figure 2: Variants of Pretext Task Distillation(a) Depth-to-GreyScale task, self supervised loss is calculated between L channel of RGB input and estimated greyscale image(b) Depth and Grey Scale-to-Color task, input of pretext layers concatanetion of L channel of RGB input and estimated depth map, loss is calculated over between a*b* channels of RGB input and estimated a*b* channels.

zero loss which is 1e-3 for all the variants. *GS* function converts RGB input x to *Lab* space and returns *L* channel as output.

Depth-Greyscale-to-Color (DG2C): Secondly, we employ the colorization task as yet another pretext task. Instead of inputting only a color image, we concatenate depth map and luminance *L* of the RGB input channel-wise for network input and estimate $a*b*$ channels as in [47]. We think that injecting 2.5D information as extra input for the colorization task might relax the optimization, since neighboring pixels are likely to have similar depth and intensity values. Again, RGB input x is converted to *Lab* space. *L* is utilized as greyscale input and *ab* are used for color targets. Conv$1 \times 1 \times 2$ is employed as prediction head in pretext layers and the loss function is utilized as follows:

$$\mathcal{L}_{dg2c}(x) = \sqrt{(PL(D(x) \oplus GS(x)) - AB(x))^2 + \varepsilon^2} \tag{2}$$

where $\oplus$ is channel-wise concatenation operation, *AB* is *a* and *b* channel of RGB input. This loss is similar to Equation 1. We do not use cross-entropy loss over quantized images as in [47] to keep things simple.

Masked D2G (MD2G) and Masked DG2C (MD2C): Finally, we combine inpainting insight based on prediction of masked regions to learn context representation with the D2G and DG2C tasks. In the masked version of these tasks (denoted with MD2G and MD2C respectively), we partially mask the input image by randomly zeroing out patch regions with a predefined resolution, and make the network to predict masked regions as in [35]. The inpainting/masked autoencoder task is employed to generate a representation that must understand the context of the surroundings of the missing region, and consequently, the entire image to infer the context of the missing region. By defining masked versions of these tasks, we also investigate whether combining those tasks improves depth estimation performance.

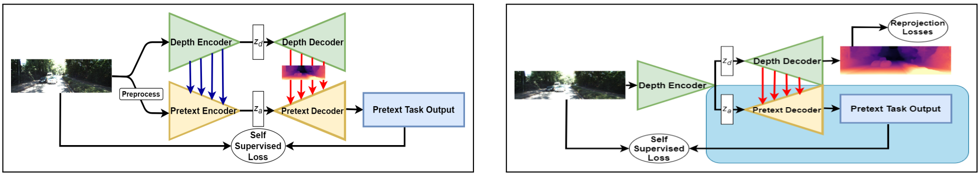
Following equation is employed as loss function for MD2G task:

$$\mathcal{L}_{md2g}(x) = \hat{M} \odot \sqrt{(PL((1-\hat{M}) \odot D(x)) - GS(x))^2 + \varepsilon^2} \tag{3}$$

where $\hat{M}$ is a binary mask where masked pixels are 1, $\odot$ is the pixel-wise product. Similarly, loss function of MDG2C task is as follows:

$$\mathcal{L}_{mdg2c}(x) = \hat{M} \odot \sqrt{(PL((1-\hat{M}) \odot (D(x) \oplus GS(x))) - AB(x))^2 + \varepsilon^2} \tag{4}$$

Note that, for masked versions of the tasks, predictions are ignored where mask pixels are 0 while calculating loss as in [35] to concentrate loss on the prediction of masked regions rather than autoencoding already visible regions. Final self-supervised/pretext task loss L_{pt} is based on the selection of pretext task.



(a) Seperate Encoder (b) Shared Encoder

Figure 3: Variants of TripleD. Red arrows indicate distillation connections that forward multi-scale depth estimations to the pretext decoder. Blue arrows forward depth encoder features to pretext encoder. Preprocess is computed based on pretext task. All fusion operations are done via channel-wise summation.

3.2 Disentangle via Pretext Task and Distill

We extend our approach in Section 3.1 with disentanglement and distillation via multiple objectives. Our intuition is that the scene can be factored into geometry and appearance components, obtained from the depth decoder and the appearance or pretext decoder, respectively. This way, the depth decoder does not have to decode irrelevant information such as color intensities. Following this intuition, we conjecture that we can reconstruct the input image with features of both networks to form auto-encoding optimization. We use two versions of this framework: *i*) using the same encoder and two decoders that are depth and color/appearance/pretext, and *ii*) where separate encoders are used.

A separate encoder allows us to use different modalities for appearance encoders, such as utilizing greyscale input and formulating colorization tasks with the help of depth estimations. For the separate encoder case (Figure 3(a)), we forward depth encoder features to separate or pretext encoder via skip connections. Simple summation between features of depth encoder and pretext encoder is applied to combine features. We formalize three main pretext tasks for separate encoder case: *i*) colorization, *ii*) inpainting and *iii*) autoencoding. Following equation is employed as the loss function for colorization pretext task:

$$\mathcal{L}_c(x) = \sqrt{(D_P(E_P(GS(x))) - AB(x))^2 + \varepsilon^2} \tag{5}$$

where E_P is pretext encoder, D_P is pretext decoder shown in Figure 3. We formalize loss functions for inpainting and autoencoding pretext tasks as follows:

$$\mathcal{L}_{mae}(x) = M \odot \sqrt{(D_P(E_P((1-M) \odot x)) - x)^2 + \varepsilon^2} \tag{6}$$

$$\mathcal{L}_{ae}(x) = \sqrt{(D_P(E_P(x)) - x)^2 + \varepsilon^2} \tag{7}$$

A shared encoder case is shown in Figure 3(b), and reprojection loss is employed as described in Section 3.3 to supervise depth estimation. In this figure, z_d and z_a are separate latent codes used for the disentanglement process. Notice that we make no guarantees about full disentanglement in feature space. Our primary focus is the rough separation of features utilized for separate tasks. We cannot change input x to form distinct pretext tasks such as colorization and inpainting. Because changing input into something so much different affects the depth estimation framework and adds an unnecessary burden to the already ill-posed problem. Therefore, we only employ autoencoding optimization for shared encoder case similar to Equation 7. Final L_{pt} is based on the encoder case or selection of pretext task.

3.3 Self Supervised Depth Estimation

To supervise the depth estimation framework, we also utilize video frames to form reprojection consistency. We use the input frame I_t for depth network and obtain the depth estimation $D_t = \mu_\theta(I_t)$ where μ is the depth network with parameters θ and use neighboring frame I_s as extra input for relative pose estimation $T_{t\rightarrow s} = \delta\gamma(I_t, I_s)$ where δ is pose network with parameters γ, following [11]. Consequently, geometric warping is modelled as follows;

$$I_{s\rightarrow t} = I_s \langle proj(D_t, T_{t\rightarrow s}, K)\rangle \tag{8}$$

where K is the camera intrinsic matrix, $proj$ is the depth coordinate projection operator, and $\langle\cdot\rangle$ is the 2D sampling operator. We can formulate reprojection objective loss $\mathcal{L}_{rp}$ as follows:

$$\mathcal{L}_{rp}(I_t, I_{s\rightarrow t}) = \psi * \mathcal{L}_{pw}(I_t, I_{s\rightarrow t}) + \lambda * \frac{1 - SSIM(I_t, I_{s\rightarrow t})}{2} \tag{9}$$

where $SSIM$ is structural similarity index, L_{pw} is pixel-wise loss defined in Equation 10, λ and ψ are scale parameters controlling contribution of losses.

$$\mathcal{L}_{pw}(x,y) = \sqrt{(x-y)^2 + \varepsilon^2} \tag{10}$$

We also utilize feature-metric loss $\mathcal{L}_{fm}$ as:

$$\mathcal{L}_{fm}(F_t, F_{s\rightarrow t}) = \mathcal{L}_{pw}(F_t, F_{s\rightarrow t}) \tag{11}$$

where F_t is encoder feature of I_t and $F_{s\rightarrow t}$ is warped version of F_s which is feature of I_s computed in a fashion similar to Equation 8. This loss is based on [38].

Following these partial loss definitions, total loss is defined as

$$\mathcal{L}_{total} = \mathcal{L}_{rp} + \alpha * \mathcal{L}_{pt} + \beta * \mathcal{L}_{fm} \tag{12}$$

where α and β are weight hyper-parameters adjusting effects of $\mathcal{L}_{pt}$ and $\mathcal{L}_{fm}$ losses. Note that multi-scale depth estimation, auto-masking stationary pixels, edge-aware loss and minimum projection loss are employed as presented in [11].

4 Experiments

4.1 Datasets

We use Eigen split[8] of KITTI dataset as depth evaluation benchmark. We utilize KITTI raw data[9] for training which consists of 39810 training, 4424 validation, and 697 test images. Besides, we experiment on the Make3D[29, 37] dataset consisting of 134 test images for depth estimation to showcase the generalizability of the model trained on the KITTI dataset. We follow the same evaluation protocol as in [11] for Make3D.

4.2 Implementation Details

Our models are trained on 4 Nvidia V100 with a total batch size of 12, learning rate 1e-4, for 20 epochs. At epoch 10, the learning rate is decreased to 1e-5. We set β as 1e-3 and α as 5-e3 in Equation 12 empirically based on cross validation, and leave $\psi = 0.15$ and $\lambda = 0.85$ as previous works [38]. We use Adam [22] optimizer with no weight decay and default parameters. We use color jittering (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1) and random vertical flip with 0.5 probability for input augmentations for depth encoder. Following [11], three neighboring frames are utilized for training, depth of the middle frame is predicted. Other frames are used for pose estimation. We utilize a shared encoder case for TripleD as default.

Backbones: We use ResNet-50(RN50) [17] based encoder for our depth estimation task. Pose Encoder is based on ResNet-18(RN18) accepts 640×192 as input resolution as shown in [38]. We use decoders similar to [11]. For all tasks utilizing masks, the input is masked by 16 patches with a 16×16 resolution quite similar to [35]. We use shared encoder case discussed in Section 3.2 as default. We use RN18 provided by [24] distilling from RN50 since no results of RN18 from respective papers. FeatDepth [38] initializes the feature-metric encoder with the supervised RN50 for $\mathcal{L}_{fm}$. Therefore, to avoid any form of supervision, we initialize all encoders with SWaV[3] unless stated otherwise. Other than SWaV[3], SimCLR[6] and MoCo[18] trained on ImageNet[7] dataset are investigated for encoder initialization. Please refer to Supplementary Material for ablation study on the feature-metric structure and encoder model initialization.

Evaluation: Estimated depth maps are capped to 80m, and median scaling is applied to depth estimations as a common practice [52].

4.3 Monocular Depth Estimation Results

We first compare our proposed method and its variants to existing SoTA methods in the literature and the corresponding results are given in Table 1. In this table, D2G, DG2C, MD2C, MDG2C corresponds to the singular pretext task distillations, whereas TripleDNet corresponds to the overall framework that includes distillation and disentanglement. Our baseline method is FeatDepth where the $\alpha = 0$ in Equation 12. We outperform our baseline for 6 out of 7 metrics with large margin. Proposed models achieve state-of-the-art results for various metrics, although many methods use semantic ground truth knowledge in some form and/or initialized with supervised pretraining. Although, DIFFNet performs relatively well, its encoder architecture is based on attention modules and HRNet[43] which explicitly utilizes built-in semantic knowledge for semantic segmentation. Our aim is not to build new architecture to improve representation, yet to construct a compact self-supervised framework. Our distillation-only models((M)D2G, (M)DG2C) also perform nicely and demonstrate that semantic knowledge extracted by ground truth labels is somewhat redundant. Generally speaking, masked versions of the D2G and DG2C performs worse than unmasked ones, this implies that the whole image is important for pixel-wise tasks as discussed in [47]. In Table 1, we also show that initializing model with supervised pretraining (TripleD(*sup.*)) performs worse than TripleD with SWaV initialization. The reason may be due to the inherent bias driven by the ground truth labels, complicating transferring knowledge from one task to a very different one.

Some methods that have RN18 backbones utilize semantic segmentation ground-truth which is direct supervision that does not need huge models. The most recent and successful related work are the ones with RN50 in Table 1. Besides, PackNet[14] is a network with ~128M parameters while our depth network have ~35M parameters. The absolute differences indeed appear to be small, however, performance gains can be observed more clearly in terms of ratios, e.g. 10.9% increase in AbsRel and 22% in SqRel between ours and the Monodepth2[11]. Table 3 also presents consistent ablation results. We note that δ_1 and δ_2 are more indicative metrics than δ_3 since δ_3 has a higher threshold (~1.95).

Table 2 demonstrates the generalizability of our approach to another dataset, namely Make3D. We observe that the proposed method outperforms current state-of-the-art methods

Method	Superv.	Encoder	Res.	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog	↑ δ_1	↑ δ_2	↑ δ_3
				Lower is better				*Higher is better*		
Wang et al.[[illegible]]	M	RN18	640x192	0.109	0.779	4.641	0.186	0.883	0.962	0.982
DDV[[illegible]]	M	RN101	640x192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
Jung et al. [[illegible]]	M+Sem	RN50	640x192	0.102	0.675	4.393	0.178	0.893	0.966	0.984
D2G	M	RN50	640x192	0.108	0.738	4.639	0.185	0.882	0.963	0.983
DG2C	M	RN50	640x192	0.107	0.742	4.607	0.183	0.886	0.964	0.983
TripleD	M	RN50	640x192	0.104	0.714	4.509	0.181	0.890	0.964	0.984
Monodepth2[[illegible]]	M	RN50	1024x320	0.110	0.831	4.642	0.187	0.883	0.962	0.982
SGDepth[[illegible]]	M+Sem	RN18	1280x384	0.107	0.768	4.468	0.186	0.891	0.963	0.982
PackNet[[illegible]]	M	PackNet	1280x380	0.107	0.802	4.538	0.186	0.889	0.962	0.981
HRDepth[[illegible]]	M	RN18	1024x320	0.106	0.755	4.472	0.181	0.892	0.966	0.984
FeatDepth[[illegible]]	M	RN50	1024x320	0.104	0.729	4.481	0.179	0.893	0.965	**0.987**
CamLessMD[[illegible]]	M	RN50	1024x320	0.102	0.723	4.374	0.178	0.898	0.966	0.983
Jung et al. [[illegible]]	M+Sem	RN18	1024x320	0.102	0.687	4.366	0.178	0.895	0.967	0.984
X-Distill[[illegible]]	M+Sem	RN50	1024x320	0.102	0.698	4.439	0.180	0.895	0.965	0.983
SGRL[[illegible]]	M+Sem	PackNet	1024x320	0.100	0.761	**4.270**	0.175	0.902	0.965	0.982
DIFFNet [[illegible]]	M	HRNet	1024x320	**0.097**	0.722	4.345	0.174	**0.907**	0.967	0.984
TripleD (*sup.*)	M	RN50	1024x320	0.103	0.726	4.437	0.180	0.896	0.965	0.983
DG2C	M	RN50	1024x320	0.099	0.668	4.448	0.176	0.893	0.966	0.985
D2G	M	RN50	1024x320	0.098	0.676	4.307	0.175	0.903	0.967	0.984
MD2C	M	RN50	1024x320	0.099	0.652	4.338	0.174	0.898	0.968	0.984
MDG2C	M	RN50	1024x320	0.099	0.651	4.336	0.173	0.897	0.967	0.985
TripleD	M	RN50	1024x320	0.099	**0.648**	4.296	**0.173**	0.901	**0.968**	0.985

Table 1: Comparison with state-of-the-art methods for depth estimation on Eigen Split of KITTI dataset. M stands for Monocular video supervision and Sem stands for semantic segmentation related supervision. **Bold** refers to best one and underline refers to second best. (*sup.*) indicates model initialization with supervised pretraining on ImageNet.

in this dataset. The main reason, we believe, is that utilizing unsupervised tasks in our framework improves the representation capability of the internal structure of the scenes.

Qualitative Analysis: In the Figure 4, we show depth maps that are consistently pleasing since our model can distinguish object boundaries better. This can also reveal the usefulness of pretext tasks for semantic segmentation that is also expected to be correlated with depth estimation. However, FeatDepth tends to mix up objects which are projected on neighboring pixels. We find that the proposed model generally produces sharper depth maps with finer details of thin objects such as trees.

Method	Superv.	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog
Monodepth [[illegible]]	S	0.544	10.94	11.760	0.193
SfMLearner [[illegible]]	M	0.383	5.321	10.470	0.478
DDVO [[illegible]]	M	0.387	4.720	8.090	0.204
Monodepth2[[illegible]]	M	0.322	3.589	7.417	0.163
X-Distill[[illegible]]	M	0.308	3.122	7.015	0.158
TripleD	M	**0.303**	**3.032**	**6.907**	**0.155**

Table 2: Comparison with state-of-the-art methods for depth estimation on Make3D.

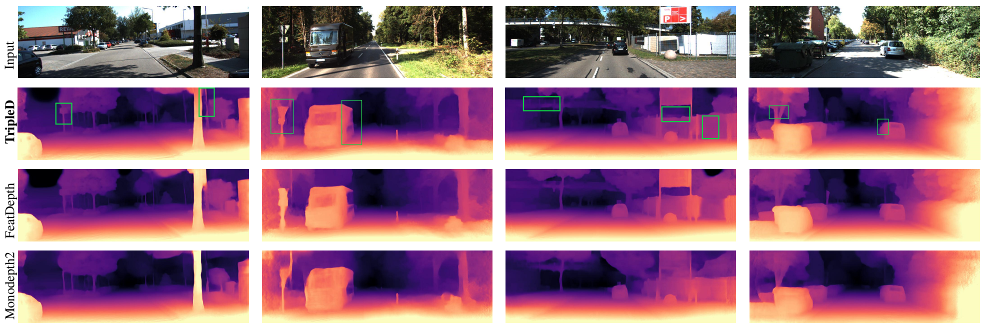


Figure 4: Qualitative Results. Green areas indicate better depth estimation.

4.4 Ablation Study

In this section (and Supplementary Material), we analyze the impact of our design decisions. Input resolution is set to 1024×320, and Eigen split of KITTI is used.

Method	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog	↑ δ_1	↑ δ_2	↑ δ_3	# params
TripleD + full disentangle	0.101	0.745	4.512	0.178	0.899	0.966	0.983	8.8M
TripleD + last 3-layer disentangle	0.101	0.635	4.337	0.176	0.893	0.968	0.985	8.9M
TripleD + last layer disentangle	**0.099**	**0.648**	**4.296**	**0.173**	**0.901**	**0.968**	**0.985**	9.1M
TripleD + no disentangle	**0.099**	0.665	4.336	**0.173**	0.899	**0.968**	**0.985**	9.6M

Table 3: Ablation study on encoder layer disentangle. # of params refer to # of decoder parameters. **Bold** refers to best one.

Layer Disentanglement: Disentanglement is made by using half of the channel features of the encoder, where those features are then forwarded through a skip connection to both of the decoders. Even with the full disentanglement, the decoder performs considerably fine as shown in Table 3. As expected, decreasing the number of separated features increases performance. It is worth noting that a model with no disentanglement performs worse than a model with 1-layer disentanglement, confirming our intuition that separating feature space according to task is likely to aid representation learning. Furthermore, one can see that # of parameters are reduced as disentangled features are increased. Even if we use an RN50 as encoder, our # of parameters (8.5M) get closer to that of [11] which uses RN18 as encoder.

Method	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog	↑ δ_1	↑ δ_2	↑ δ_3
Baseline + No Dist. Connection	0.101	0.665	4.431	0.178	0.893	0.966	0.985
Baseline + last layer Dist. Connection	0.100	0.658	4.388	0.176	0.898	0.967	0.985
Baseline + first layer Dist. Connection	0.100	0.657	4.340	0.175	0.899	0.967	0.984
Baseline + Full Dist. Connection	0.099	0.648	4.296	0.173	0.901	0.968	0.985
Baseline + Full Dist. + Encoder Skip Conn.	**0.098**	**0.667**	**4.294**	**0.174**	**0.903**	**0.968**	**0.984**

Table 4: Ablation study on distillation connection from depth decoder to appearance decoder. **Bold** refers to best one.

Distillation Connection: Table 4 analyzes the effect of distillation connections and demonstrates that it boosts performance in each metric. An important aspect is that adding skip connections from the pretext decoder to the shared encoder increases performance. That might sound counter-intuitive to our claim on depth decoder distillation. However, increasing layer size might have an undesired effect on parameter updates, since gradients start to weaken before reaching early layers. Direct skip connections to the encoder from the pretext decoder solve that problem. However, we should note that adding more connections leads towards a more multi-objective approach rather than a distillation-based method.

5 Conclusions

We demonstrate the power of a fully self-supervised framework and propose methods to improve self-supervised monocular depth estimation, shed light on important aspects of self-supervised depth estimation and impact of IRL on depth estimation. Results are promising and the proposed TripleDNet model that is purely trained in a self-supervised fashion even outperforms prior works that rely on ground truth annotations. We believe that *fully* unsupervised depth estimation framework is an attractive direction to explore in order to develop robust and generalized algorithms.

6 Acknowledgment

The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

References

[1] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian D. Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019.

[2] Hong Cai, Janarbek Matai, Shubhankar Borse, Yizhe Zhang, Amin Ansari, and Fatih Porikli. X-distill: Improving self-supervised monocular depth via cross-task distillation. In *British Machine Vision Conference (BMVC)*, 2021.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *ArXiv*, abs/2006.09882, 2020.

[4] Sai Shyam Chanduri, Zeeshan Khan Suri, Igor Vozniak, and Christian Müller. Camlessmonodepth: Monocular depth estimation with unknown camera parameters. In *British Machine Vision Conference (BMVC)*, 2021.

[5] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Y. Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2619–2627, 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231 – 1237, 2013.

[10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[11] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3827–3837, 2019. doi: 10.1109/ICCV.2019.00393.

[12] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, 2017. doi: 10.1109/CVPR.2017.699.

[13] Jianping Gou, B. Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *ArXiv*, abs/2006.05525, 2021.

[14] Vitor Campanholo Guizilini, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2482–2491, 2020.

[15] Vitor Campanholo Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. *ArXiv*, abs/2002.12319, 2020.

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

[17] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *ArXiv*, abs/2111.06377, 2021.

[20] Adrian Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4755–4764, 2020.

[21] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12622–12632, 2021.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[23] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. *ArXiv*, abs/2007.06936, 2020.

[24] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Compress: Self-supervised learning by compressing representations. *Advances in neural information processing systems*, 2020.

[25] Varun Ravi Kumar, Marvin Klingner, Senthil Kumar Yogamani, Stefan Milz, Tim Fingscheidt, and Patrick Mäder. Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 61–71, 2021.

[26] Chung-Sheng Lai, Zun-Zhi You, Ching-Chun Huang, Yi-Hsuan Tsai, and Wei-Chen Chiu. Colorization of depth map via disentanglement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020.

[27] Seokju Lee, François Rameau, Fei Pan, and In-So Kweon. Attentive and contrastive learning for joint depth and motion field estimation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4842–4851, 2021.

[28] Zhaowen Li, Zhiyang Chen, F. Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Jinqiao Wang. Mst: Masked self-supervised transformer for visual representation. In *NeurIPS*, 2021.

[29] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:2024–2039, 2016.

[30] Yao Lu, Xiaoli Xu, Mingyu Ding, Zhiwu Lu, and Tao Xiang. A global occlusion-aware approach to self-supervised monocular visual odometry. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2260–2268, May 2021.

[31] Xiaoyang Lyu, L. Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *AAAI*, 2021.

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

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[35] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[36] Andrea Pilzer, Stéphane Lathuilière, N. Sebe, and Elisa Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9760–9769, 2019.

[37] Ashutosh Saxena, Min Sun, and A. Ng. Learning 3-d scene structure from a single still image. *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[38] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020.

[39] Jaime Spencer, R. Bowden, and Simon Hadfield. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14390–14401, 2020.

[40] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi di Stefano, and S. Mattoccia. Distilled semantics for comprehensive scene understanding from videos. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4653–4664, 2020.

[41] Pascal Vincent, H. Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML '08*, 2008.

[42] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[43] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3349–3364, 2021.

[44] Lijun Wang, Yifan Wang, Linzhao Wang, Yu-Wei Zhan, Ying Wang, and Huchuan Lu. Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12707–12716, 2021.

[45] Lijun Wang, Yifan Wang, Linzhao Wang, Yunlong Zhan, Ying Wang, and Huchuan Lu. Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12727–12736, October 2021.

[46] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016.

[48] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 645–654, 2017.

[49] Hang Zhou, David Greenwood, Sarah L. Taylor, and Han Gong. Constant velocity constraints for self-supervised monocular depth estimation. *European Conference on Visual Media Production*, 2020.

[50] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021.

[51] Hang Zhou, Sarah Taylor, and David Greenwood. Sub-depth: Self-distillation and uncertainty boosting self-supervised monocular depth estimation. *ArXiv*, abs/2111.09692, 2021.

[52] Tinghui Zhou, Matthew A. Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017.