TripleDNet: Exploring Depth Estimation with Self-Supervised Representation Learning-Supplementary Material

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

1 Overview

In Supplementary Material, we provide more insights into the proposed method with extended experiments. For all experiments input resolution is set to $1024x320$, and Eigen split[4] of KITTI[6] results are reported. Imagenet[3] is utilized as pretraining dataset for SWaV[1], SimCLR[2], MoCo[5] and supervised case. We use a share encoder case and all encoders are initialized with SwAV unless stated otherwise.

2 Quantitative Metrics

Common metrics are used defined below, which compute the error between estimated depth value $\hat{d}$ from a set of $\hat{D}$ consisting of all predicted depth values of an image and ground truth d value. Lower is the better since those are error metrics.

Absolute Relative Error(Abs Rel): $\frac{1}{|\hat{D}|}\sum_{\hat{d}\in\hat{D}}\frac{|d-\hat{d}|}{d}$

Squared Relative Error(Sq Rel): $\frac{1}{|\hat{D}|}\sum_{\hat{d}\in\hat{D}}\frac{||d-\hat{d}||^2}{d}$

Root Mean Squared Error(RMSE): $\sqrt{\frac{1}{|\hat{D}|}\sum_{\hat{d}\in\hat{D}}||d-\hat{d}||^2}$

log of RMSE (RMSElog): $\sqrt{\frac{1}{|\hat{D}|}\sum_{\hat{d}\in\hat{D}}||logd-log\hat{d}||^2}$

Below metric computes the ratio between pixels that are in a range defined by t from 1. Higher is better for those metrics since it somewhat classifies pixels.

δ_t**:** $\frac{1}{|\hat{D}|}|\{\hat{d}\in\hat{D}|max(\frac{d}{\hat{d}},\frac{\hat{d}}{d})\}<1.25^t|$ x 100%

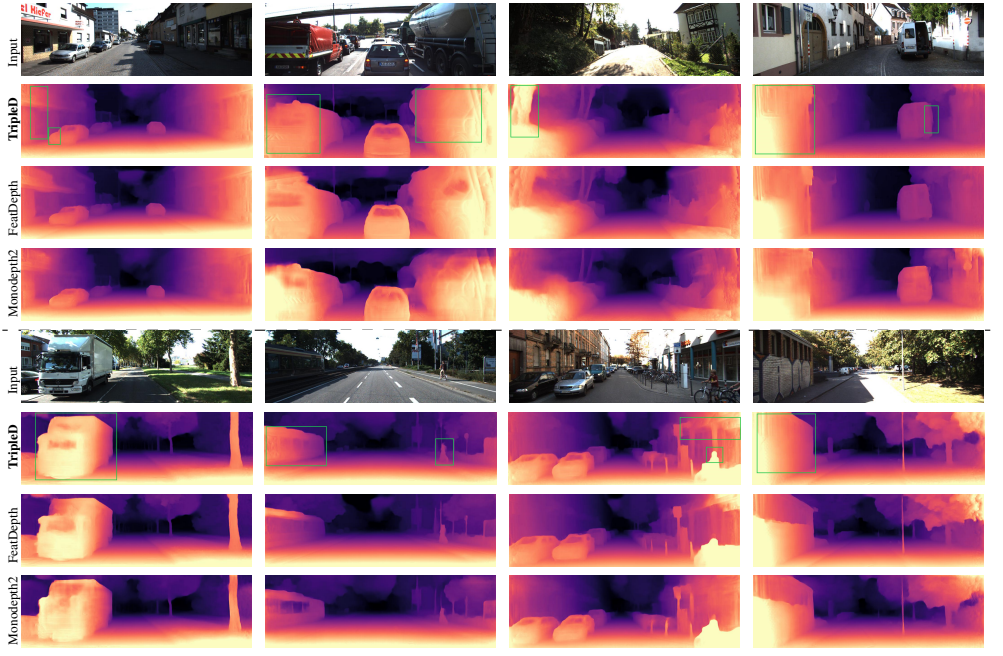


Figure 1: Qualitative Results. Green boxes indicate better depth estimation.

3 Self-Supervised Objective Selection

For the only-pretext task case, self-supervised objective is picked as follows:

$$L_{pt} = \begin{cases} L_{d2g} & \text{if task} = \text{D2G} \\ L_{dg2c} & \text{if task} = \text{DG2C} \\ L_{md2g} & \text{if task} = \text{MD2G} \\ L_{mdg2c} & \text{if task} = \text{MDG2C} \end{cases} \tag{1}$$

where L_{d2g}, L_{dg2c}, L_{md2g}, L_{mdg2c} are defined in main paper. For the TripleD case, self-supervised objective is:

$$L_{pt} = \begin{cases} L_{c} & \text{if task} = \text{colorization \& encoder} = \text{separate} \\ L_{mae} & \text{if task} = \text{inpainting \& encoder} = \text{separate} \\ L_{ae} & \text{if task} = \text{autoencoding \& encoder} = \text{separate} \\ L_{sae} & \text{if task} = \text{autoencoding \& encoder} = \text{shared} \end{cases} \tag{2}$$

where L_c, L_{mae}, L_{ae}, L_{sae} are defined in main paper.

4 Different Objectives For Separate Encoder

Using a separate/pretext encoder for pretext tasks gives us the flexibility to change objective functions rather than autoencoding shown in Figure 2. For the shared encoder case, we could also apply the inpainting task end-to-end; however, using partially masked input for depth estimation would add unnecessary complexity to an ill-posed problem. Adding an extra

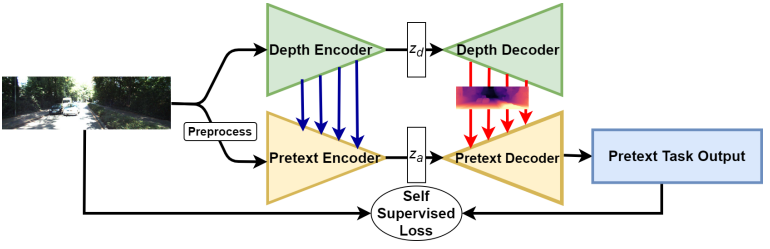


Figure 2: Separate Encoder Case for TripleD. Red arrows indicate distillation connections that forward multi-scale depth estimations to the pretext decoder. Blue arrows forward depth encoder features to pretext encoder. All fusion operations are done via channel-wise summation.

Method	Pretext Objective	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog	↑ δ_1	↑ δ_2	↑ δ_3
Separate Encoder	Autoencoding	0.103	0.682	4.324	0.175	0.896	0.968	0.985
Separate Encoder	Inpainting	0.101	0.656	4.407	0.178	0.893	0.966	0.984
Separate Encoder	Colorization	**0.099**	0.657	4.341	0.175	**0.902**	0.968	0.984
Shared Encoder	Autoencoding	**0.099**	**0.648**	**4.296**	**0.173**	0.901	**0.968**	**0.985**

Table 1: Ablation study on separate encoder with different objectives.

encoder decreases performance as expected in Table 1. It removes a burden out from the depth encoder to itself to solve pretext tasks, and disrupts the distillation and disentanglement mechanism. Hence, colorization task performs better than others in separate encoder, since it utilizes whole image instead of partially masked one.

5 Depth Encoder Initialization

Method	*Shared Encoder Init*	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog	↑ δ_1	↑ δ_2	↑ δ_3
FeatDepth	Supervised	0.104	0.725	4.485	0.179	0.894	0.964	0.987
FeatDepth	SwAV	0.104	0.729	4.481	0.179	0.893	0.965	0.987
TripleD	-	0.120	0.881	4.913	0.199	0.859	0.954	0.980
TripleD	Supervised	0.103	0.726	4.437	0.180	0.896	0.965	0.983
TripleD	MoCo	0.103	0.735	4.482	0.178	0.899	0.965	0.984
TripleD	SimCLR	0.101	0.695	4.435	0.178	0.894	0.966	0.984
TripleD	**SwAV**	**0.099**	**0.648**	**4.296**	**0.173**	**0.901**	**0.968**	**0.985**

Table 2: Ablation study on model initialization.

Transfer learning is currently one of the primary practices in machine learning, shortens training time for various tasks. Thus far, supervised trained models are utilized for encoder initialization in self-supervised depth estimation. However, trained models by ground truth supervision have so much bias driven by the labels and complicate transferring knowledge from one task to a very different one. Thus we change the model from supervised to unsupervised for the task at hand. We initialize both the depth and pose encoder with the same unsupervised method specified in Table 2 and use feature metric loss using model initial-

ized with SwAV. Note that the architecture of the depth encoder is ResNet-50, and the pose encoder is ResNet-18. Initialization with any method is a huge performance boost as expected. In Table 2, SwAV outperforms other methods by a large margin as it outperforms in the image classification task. Furthermore, we test SwAV initialization on FeatDepth, which reveals that it does not necessarily improve FeatDepth's performance on each metric.

6 Feature-Metric Loss

We also utilize feature-metric loss $\mathcal{L}_{fm}$ presented by FeatDepth[8] to analyze effect on our structure and image representation learning(IRL) perspective. As [8] propose, separate feature encoder is trained with image reconstruction $\mathcal{L}_{rec}$, discriminative $\mathcal{L}_{dis}$ in Equation 3. and convergent $\mathcal{L}_{cvt}$ loss in Equation 4.

$$\mathcal{L}_{dis} = -\sum_{p} e^{|\nabla^1 I(p)|_1} |\nabla^1 \phi(p)|_1 \tag{3}$$

$$\mathcal{L}_{cvt} = \sum_{p} |\nabla^2 \phi(p)|_1 \tag{4}$$

where $\nabla^1 I(p)$ is image gradient with respect to pixel p, $\nabla^1 \phi(p)$ is feature gradient with respect to pixel p, $\nabla^2 \phi(p)$ is second order feature gradient and ϕ is feature encoder.

Then, neighboring images are fed to this encoder, and obtained feature maps are warped to compute reprojection error in feature space. However, this feature encoder is initialized with a model trained on ImageNet ground truth supervision. Thus, we initialize this encoder with self-supervised IRL models, and further, we replace $\mathcal{L}_{rec}$ with masked image reconstruction $\mathcal{L}_{mask-rec}$. This loss computed as follows;

$$\mathcal{L}_{mask-rec}(x) = \hat{M} \odot \sqrt{(x - F((1-\hat{M}) \odot x))^2 + \varepsilon^2} \tag{5}$$

where $\hat{M}$ is a binary mask where masked pixels are 1, $\odot$ is the pixel-wise product, x is the input image, F is the convolutional neural network consisting of decoder and feature encoder ϕ, and ε is a constant to avoid zero loss which is 1e-3. This loss is used for all cases utilizing masks in this study. When we do not use a mask, for instance for D2G or DG2C task, we utilize similar following pixel-wise loss;

$$\mathcal{L}_{pw}(x,y) = \sqrt{(x-y)^2 + \varepsilon^2} \tag{6}$$

where y is the estimation based on the task such as any pretext or autoencoding.

In Table 3, $\mathcal{L}_{mask-rec}$ increases(or does not change) performance of different initializations consistently. Surprisingly, $\mathcal{L}_{fm}$ is not necessary for encoder initialized with Even if we change feature encoder initialization with a supervised or SimCLR while keeping shared encoder initialization as SwAV, it does not have a negative impact. That implies representation capability of SwAV initialization is best for our framework. However, $\mathcal{L}_{fm}$ boosts performance so much for other initializations. Nevertheless, we demonstrate that unsupervised methods can replace supervised models for model initialization and loss on representation space[9] somewhat similar to perceptual loss [7].

Shared Encoder Init	*Feature Encoder Init.*	$\mathcal{L}_{fm}$	$\mathcal{L}_{mask-rec}$	↓ Abs Rel	↓ Sq Rel	↓ RMSE	↓ RMSElog	↑ δ_1	↑ δ_2	↑ δ_3
Supervised	-			0.106	0.755	4.499	0.187	0.892	0.964	0.983
Supervised	Supervised	✓		0.103	0.736	4.443	0.180	0.896	0.965	0.983
Supervised	Supervised	✓	✓	0.103	0.726	4.437	0.180	0.896	0.965	0.984
MoCo	-			0.105	0.752	4.483	0.178	0.899	0.965	0.982
MoCo	MoCo	✓		0.103	0.748	4.489	0.182	0.898	0.964	0.984
MoCo	MoCo	✓	✓	0.103	0.736	4.486	0.177	0.899	0.964	0.984
SimCLR	-			0.103	0.703	4.451	0.179	0.895	0.898	0.984
SimCLR	SimCLR	✓		0.101	0.700	4.445	0.176	0.894	0.966	0.984
SimCLR	SimCLR	✓	✓	0.101	0.699	4.443	0.176	0.895	0.967	0.984
SwAV	-			0.099	0.652	4.314	0.173	0.901	0.967	0.985
SwAV	SwAV	✓		0.099	0.655	4.300	0.173	0.901	0.967	0.985
SwAV	**SwAV**	✓	✓	**0.099**	**0.648**	**4.296**	**0.173**	**0.901**	**0.968**	**0.985**
SwAV	SimCLR	✓	✓	0.101	0.663	4.386	0.176	0.895	0.967	0.984
SwAV	Supervised	✓	✓	0.099	0.667	4.361	0.175	0.900	0.968	0.984

Table 3: Ablation study for TripleD on feature metric loss initialization.

7 Qualitative Results

In Figure 1, we show extended results of our approach. Generally, the proposed model completes objects such as gas tankers, and many-windowed walls or trucks while keeping finer details and smoothens those objects realistically perspective-wise. For those examples, other models produce unnecessary and false depth maps with large edges for even the flat regions. Interestingly, our method distinctly generates a depth map by recognizing an object in the low-light scene(2nd row, 3rd column).

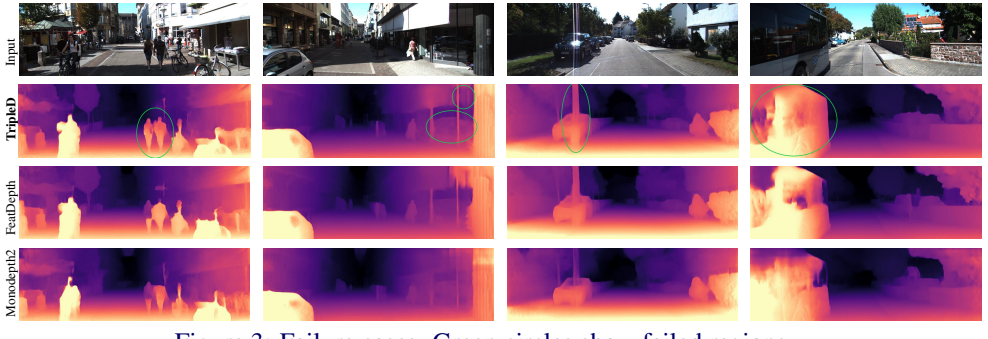
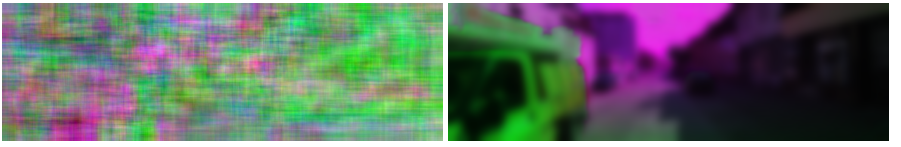


Figure 3: Failure cases. Green circles show failed regions.

However, our model fails for some cases shown in Figure 3. Depth values of people are produced very well for many samples. However, overly vertical smoothing is a problem in some cases because of bias based on the dataset of scenes consisting of sky and road consistently. High-intensity and mirror reflections from a vehicle or building glass are the most common failure cases which can be solved by further abstraction reasoning.

7.1 Disentanglement Effect on Pretext Task

We note that our primary focus is only a rough separation of feature space, we make no guarantees about full disentanglement. We can demonstrate this rough disentanglement by zeroing out the depth estimates in the input of pretext decoder during inference. For this purpose, we carried out a small experiment, where we replace depth estimates with zeros for pretext decoder. Since condition is done via summation, we prevent effect of estimated depth



(a) W/o Depth Map (b) W/ Depth Map

Figure 4: Pretext Decoder outputs, scaled for visualization.

maps. An output as in Figure 4(a) that do not have any geometric detail, only random colors are obtained. On the contrary, feeding estimated depth maps onto pretext decoder produces an output as in Figure 4(b) that is quite similar to a depth map.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

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

[9] 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.