2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Geometric Pretraining for Monocular Depth Estimation

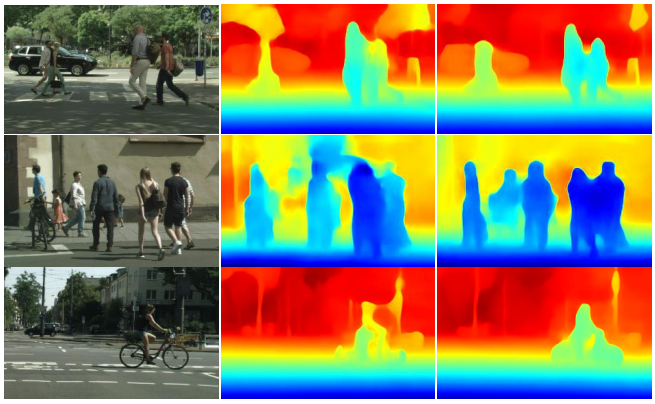
Kaixuan Wang[1], Yao Chen[2], Hengkai Guo[2], Linfu Wen[2] and Shaojie Shen[1]

***Abstract*— ImageNet-pretrained networks have been widely used in transfer learning for monocular depth estimation. These pretrained networks are trained with classification losses for which only semantic information is exploited while spatial information is ignored. However, both semantic and spatial information is important for per-pixel depth estimation. In this paper, we design a novel self-supervised geometric pretraining task that is tailored for monocular depth estimation using *uncalibrated* videos. The designed task decouples the structure information from input videos by a simple yet effective conditional autoencoder-decoder structure. Using almost unlimited videos from the internet, networks are pretrained to capture a variety of structures of the scene and can be easily transferred to depth estimation tasks using *calibrated* images. Extensive experiments are used to demonstrate that the proposed geometric-pretrained networks perform better than ImageNet-pretrained networks in terms of accuracy, few-shot learning and generalization ability. Using existing learning methods, geometric-transferred networks achieve new state-of-the-art results by a large margin. The pretrained networks will be open source soon[1].**

I. INTRODUCTION

Estimating depth maps of images is of vital importance in computer vision and robotics. Benefiting from the development of deep learning, many methods have been proposed to estimate the depth map using a single input image. These methods can be deployed easily and used in a variety of applications such as visual odometry [1], [2], sensor fusion [3], and augmented reality [4].

Although generating impressive results, learning-based methods need a large amount of data for training. Per-pixel depth annotating of real-world images is almost impossible as LiDAR only provides sparse depth measurements, and time-of-flight (ToF) cameras have limited ranges. The KITTI stereo dataset [5] uses CAD models to densify depth measurements of cars but only contains hundreds of images. Yang *et al.* [6] propose DrivingStereo with more than $180k$ frames fused from multi-frame LiDAR measurements. Despite the high accuracy depth measurement in DrivingStereo, the density of annotated pixels is less than 15%. Recently, many self-supervised works [7]–[16] have been proposed to train networks using *calibrated* stereo images or monocular videos. These methods are built on the assumption that images from nearby views can be synthesized correctly if the scene geometry and camera motion are estimated correctly. Compared with using active sensors (e.g., LiDAR and ToF), training with stereo images or monocular videos offers a number of advantages. First and foremost, training data can be captured much more easily without depth labeling, thus a larger variety of data can be used for training. Second, the depth of every pixel can be supervised by minimizing photometric errors. Despite the success of recent self-supervised methods, the geometric view synthesize process needs camera intrinsic parameters, which requires offline calibration. Because of this, current self-supervised methods cannot utilize the almost unlimited videos from the internet and train a 'universal' monocular depth estimation model. To overcome the training data limitations, using networks pretrained on ImageNet [17] classification tasks as training initialization becomes a default choice.



(a) Input images (b) ImageNet-transferred (c) Geometric-transferred

Fig. 1. Comparison of depth estimation using different pretrained networks. Networks are trained using the same method proposed in monodepth2 [18] but are *initialized differently*. The middle column is the result of the official model initialized with ImageNet-pretrained networks. The right column is the result of using the proposed geometric-pretrained networks as initialization. As shown, networks transferred from the proposed geometric-pretrained models generate more accurate and sharper depth maps.

The success of transferring ImageNet-pretrained models to other classification tasks has been widely demonstrated and studied [19], [20]. However, transferring networks trained on ImageNet classification tasks for dense depth map estimation has several potential problems. First, classification tasks emphasize semantic feature maps, while the spatial information is ignored. When transferred to spatial-sensitive tasks, e.g. keypoint detection, He *et. al.* [20] show that ImageNet-pretrained networks have limited benefits. Secondly, images from the ImageNet dataset focus on object detection and are very different from images for depth estimation.

In this paper, we propose a novel pretraining task that uses wild videos from the internet to capture both semantic and spatial information. Based on the fact that the optical flow between two images is determined solely by the geometric structure and motions (both camera and object motion), we use a conditional encoder-decoder to separate structure information and motion information from uncalibrated videos.

Authors[1] are with the Department of Electronic and Computer Engineering, HKUST, Hong Kong, SAR China. {kwangap, eeshaojie}@ust.hk. Authors[2] are with ByteDance AI Lab.

[1]https://github.com/HKUST-Aerial-Robotics/GeometricPretraining

The structure information of the scene is encoded from a single image, and motion information is estimated from two adjacent images. The optical flow between two images is finally reconstructed by using the estimated structure information conditioned on the motion vector. Since optical flow can be supervised without camera intrinsic or extrinsic parameters, the pretraining task can utilize unlimited videos from the internet and learn a variety of structures. After the pretraining stage, the encoder network can be easily transferred to depth estimation tasks.

Extensive experiments are used to demonstrate the transferred performance of the proposed geometric-pretrained networks. In Figure 1, we show estimated depth maps of ImageNet-pretrained networks and geometric-pretrained networks. Compared with ImageNet-pretrained networks, the proposed networks generate more accuracy and sharper depth estimations and can be generalized to other scenes or datasets.

II. RELATED WORK

In this section, we first briefly review recent developments of depth learning from calibrated stereo images or monocular videos. Since many works use ImageNet-pretrained models as training initialization, we also review works investigating the effectiveness of transfer learning.

A. Self-supervised Depth Learning

Due to the limited quantity of images with depth annotation, using stereo images or monocular videos to learn the depth map has become attractive to researchers and industry. The core concept of self-supervised learning is that with accurate geometric prediction and poses from calibration or estimation, images from nearby views can be correctly reconstructed. Garg *et. al.* [7] propose a pioneering method that minimizes the photometric error between reconstructed images and the second view image. Subsequent works [21]–[23] improve the depth quality by incorporating discriminating losses and improving the depth resolution. Compared with rectified and synchronized stereo images, calibrated monocular videos are easier to capture. SfMLearner [8] extends the self-supervised method from synchronized stereo images to calibrated monocular videos by estimating the camera motion between two frames by a deep network. However, in monocular cases. pixels of dynamic objects cannot be reconstructed correctly using only the camera motion. A number of works [15], [18], [24] have been proposed to mask dynamic objects or explicitly estimate the motion of each *rigid* dynamic object using semantic masks. GLNet [25] further estimates intrinsic parameters such that it can be trained on uncalibrated pinhole images.

B. Transfer learning

Although self-supervised methods have improved in recent years, the available training data is not comparable to that of classification tasks. For example, KITTI Eigen full split [26] has 45k image pairs while ImageNet 2011 [17] has more than 14M images, which is 300 times larger. To accelerate the training and improve the accuracy, many self-supervised methods [18], [27]–[29] use ImageNet-pretrained models as initialization.

Transferring learning with pretrained models has been widely studied. Kornblith *et. al.* [19] found a high correlation between transferred accuracy and ImageNet accuracy in classification tasks. He *et. al.* [20] further systematically studied the effectiveness of ImageNet-pretrained networks. Experiments show that ImageNet-pretrained models show clear benefits when transferred with small training datasets. However, these benefits are limited in classification tasks. Tasks that require spatial information (e.g., keypoint detection) benefit little from the pretraining. By extensive experiments, Zamir *et. al.* [30] points out that there exists a relationship structure between different visual tasks, and they call this structure Taskonomy. Taskonomy shows that depth estimation is more related to normal estimation, occlusion estimation, but less related to semantic segmentation, object classification tasks. With limited training data, networks can be (first and higher-order) transferred from related pretrained networks and achieve competitive results.

Due to the benefits of using pretrained networks as initialization, many works focus on designing pretraining tasks specifically for downstream use. Agrawal *et. al* [31] use egomotion classification losses to pretrain networks, and then apply the network to different tasks. Mahajan *et. al.* [32] pretrain networks using billions of images annotated with hashtags and fine-tune the networks using the ImageNet dataset. Zhan *et. al.* [33] use Conditional Motion Propagation (CMP) models to reconstruct dense optical flow given a few sparse flow vectors. In this manner, CMP models are forced to capture kinematic properties and can be transferred to segmentation tasks. However, CMP uses discretized optical flow from LiteFlowNet [34] to supervise the training such that the efficiency and precision of the training are limited. Similar ideas also appear in natural language processing (NLP) areas. BERT [35] and XLNet [36] are proposed to pretrain networks using large-scale unlabeled text from the internet, and achieve new state-of-the-art performances.

In this work, we employ the idea of pretraining task design for monocular depth estimation, and propose a geometric-pretraining task. Using large-scale *uncalibrated videos* from the internet, we first train encoder networks to capture structure information of images and then fine-tune pretrained networks using existing methods. Extensive experiments are used to demonstrate that the proposed pretraining task improves the performance of depth estimation in terms of both accuracy and generalization ability.

III. GEOMETRIC PRETRAINING

Given two adjacent frames of a video, the optical flow between two frames is caused by the moving camera and the motion of independent objects. Denote $\mathbf{D}$ as the depth map of the source frame, and $\mathbf{K}$ as the camera intrinsic matrix. The motion of the camera and objects is represented

(a) Geometric pretraining

(b) Fine-tune for depth estimation

Fig. 2. Overview of the proposed geometric pretrain and further transfer learning. (a), the proposed geometric pretraining uses a simple but effective conditional encoder-decoder structure. Input during the pretraining is two-frame image pairs. Structure information is estimated from the reference image and the motion information is estimated from two images. Using both the encoded structure and motion, the optical flow between two frames is reconstructed. Since the encoder only sees the reference image, it is forced to learn motion-invariant structure information. (b), pretrained encoder networks can be further transferred for depth estimation using current methods (e.g., monodepth2).

in homogeneous coordinates,

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{R}$ and $\mathbf{t}$ is the rotation matrix and translation vector, respectively. We use ${}^{c}\mathbf{T}_t^{t+1}$ for camera motion and ${}^{o}\mathbf{T}_t^{t+1}$ for object motion from frame t to $t+1$.

The optical flow of a pixel $\mathbf{p}$ from I_t to I_{t+1} can be calculated as:

$$\mathbf{f}_\mathbf{p} = \lambda([\mathbf{K}|0]\, {}^{c}\mathbf{T}_t^{t+1} {}^{o}\mathbf{T}_t^{t+1} \begin{bmatrix} \mathbf{K}^{-1}\mathbf{p} \cdot \mathbf{D}(\mathbf{p}) \\ 1 \end{bmatrix}) - \mathbf{p}, \tag{2}$$

where $\lambda(\cdot)$ is the normalization function. For pixels of static objects, ${}^{o}\mathbf{T}_t^{t+1} = \mathbf{I}_4$ is an identity matrix. Compared with $\mathbf{D}$, which is a *dense* depth map with the same size of images, the motion matrix $\mathbf{T}$ can be parameters with only 6 numbers. In dynamic environments, the number of different motion matrices ${}^{o}\mathbf{T}_t^{t+1}$ is equal to the number of independent moving objects and is far less than the number of pixels in the image.

The core of the proposed geometric pretraining task is to separate the structure information from the optical flow. Using a conditional encoder-decoder, the optical flow is reconstructed using the structure information from a single image conditioned on the motion information from two images. By compressing the motion information through a low-dimensional bottleneck, the structure encoder network is forced to captures motion-invariant structure information so that the optical flow can be correctly estimated.

A. Framework

The system framework is shown in Figure 2. The proposed pretraining task consists of two encoder networks and one decoder.

1) Structure Encoder: The structure encoder takes the source image as the input and outputs feature maps for the optical flow decoder. Since the trained structure encoder will be used as the backbone network for depth estimation, no specific architecture is required. In this work, we use standard ResNet-18 [37] for most of the experiments to be consistent with monodepth2.

2) Motion Encoder: The motion encoder takes two images as the input and outputs a compact motion vector. The purpose of the motion encoder is similar to that of pose networks in GeoNet [11] that estimate the motion information. Different from pose networks, the proposed motion encoder generates motion information implicitly using latent vectors. We follow monodepth2 and use a modified ResNet [37] network with bottleneck layers to generate the latent motion vectors with the dimension of only 128. Compared to the dimensions of recovered optical flow, for example, $640 \times 192 \times 2$, the motion vector only accounts for less than 0.1 ‰ of the data, such that motion vectors only encode the necessary motion information.

3) Flow Decoder: The flow decoder is designed to fuse the information from the structure encoder and the motion encoder. The motion vector is upsampled and concatenated with the feature map from the structure encoder. The decoder consists of several nearest-upsample layers and uses skip connections to reconstructs the optical flow between I_t and I_{t+1}. Compared to the encoder network, the decoder contains 14 layers and is relatively small. To speed up the training convergence, the optical flow $\mathbf{f}$ is estimated in a coarse-to-fine manner.

Overall, the proposed structure is similar to monocular-supervised depth learning, which contains a depth network, pose network and an image warping module. However, our pretraining structure aims to learn general structures, thus does not explicitly estimate depth and poses.

B. Loss Function

The optical flow between two frames can be calculated by the photometric consistency assumption. In the pretraining, the loss function consists of one photometric term and one smoothness term:

$$L = L_{pho} + \alpha L_{smooth}, \tag{3}$$

where α is the weight of the smoothness term. The photometric error measures the difference between the source image I_t and the warped image $\hat{I}_t$ with $\hat{I}_t(\mathbf{p}) = I_{t+1}(\mathbf{f}(\mathbf{p}) + \mathbf{p})$. Recent development of the monocular depth estimation has

shown that a combination of $l1$ loss and **ssim** loss [38] offers sharp estimation of depth maps. Here, we adopt the idea and use:

$$L_{pho}(\hat{I}_t, I_t) = \beta \frac{1 - \textbf{ssim}(\hat{I}_t, I_t)}{2} + (1-\beta)|\hat{I}_t - I_t|. \quad (4)$$

Smoothness loss is defined as

$$L_{smooth}(\mathbf{f}) = |\partial_x \mathbf{f}| e^{-|\partial_x I_t|} + |\partial_y \mathbf{f}| e^{-|\partial_y I_t|}. \quad (5)$$

Similar to previous works [7], [11], we use $\alpha = 0.001$ and $\beta = 0.85$ to balance loss terms.

C. Pretraining Dataset

In this work, we use data from three sources for the pretraining: KITTI [39], CityScapes [40] and YouTube videos. Different combinations of these datasets are used in this work to study the effect of the pretraining dataset.

1) KITTI: The KITTI dataset is widely used in monocular depth learning works because of the available LiDAR measurement for benchmarking. We use KITTI Zhou split [8] that contains image triplets. In Zhou split, static frames with an average optical flow of less than 1 pixel are removed leading to 40k triplets for training and 4k triplets for testing.

2) CityScapes: The CityScapes dataset is widely used in semantic understanding. Compared with the KITTI dataset, the CityScapes dataset contains much more dynamic moving objects (e.g., moving cars and walking pedestrians) but does not have any LiDAR measurement. *leftImg8bit_sequence* captures images in $17Hz$ and is used for the pretraining. We subsample the sequence into $8.5Hz$ and follow Zhou's method to extract image triplets that have sufficient optical flow. In total, we extract $40k$ triplets for training and $4k$ triplets for evaluation.

3) Driving Videos: The KITTI dataset and CityScapes dataset are calibrated carefully but are limited in size. On the contrary, on the internet, there exist countless driving videos captured by cameras on cars without any calibration or labels. These videos cover cities all around the world and a variety of urban structures. We downloaded 87 sequences in total from YouTube and subsampled images in $10Hz$. To ease the following experiments, we extract two YouTube-based datasets of different sizes. The smaller one (D_s) contains $18k/1k$ images for training/testing and the larger one (D_l) contains $38k/2.7k$ triplets. Figure 3 shows image samples of the extracted dataset. As shown, extracted images cover dynamic scenes and a variety of structures (e.g., bike on a car in the image).

D. Training Details

A number of encoders are trained using different combinations of the datasets. Networks are also initialized differently to study the effect of second-order transfer learning (e.g., ImageNet→Geometric-pretraining→depth-learning). Details are listed in Table III-D. *ImageNet* indicates the model is trained using ImageNet-pretrained networks as the initialization. The models are implemented in PyTorch [41] and optimized using Adam [42]. The batch size is selected to maximize the usage of a single GTX-TITAN Xp: 20 images for the resolution of 640×192, and 8 images for 1024×320. The learning rate is 10^{-4} throughout the training for simplicity.

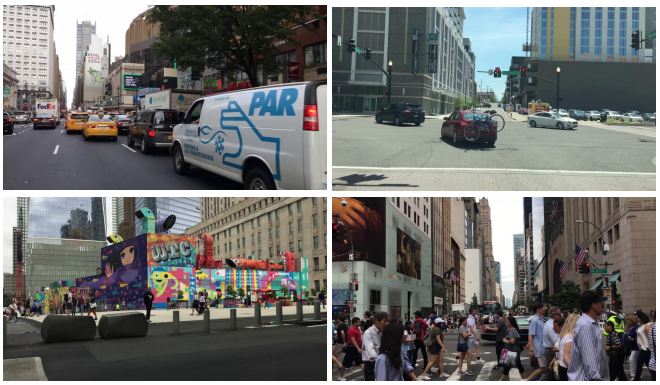


Fig. 3. Samples of the driving video dataset from the internet.

TABLE I
DETAILS OF PRETRAINED MODELS

Name	ImageNet	Resolution	Backbone	K	CS	D_s	D_l	Num.
kcd	✓	640×192	ResNet18	✓	✓	✓	✗	$98k$
kc	✓	640×192	ResNet18	✓	✓	✗	✗	$80k$
d_noI	✗	640×192	ResNet18	✗	✗	✗	✓	$38k$
d	✓	640×192	ResNet18	✗	✗	✗	✓	$38k$
kcd_hd	✓	1024×320	ResNet50	✓	✓	✗	✓	$118k$

K: KITTI, CS: CityScapes
D_s: Driving Videos small, D_l: Driving Videos large
Num.: Number of samples in the pretraining dataset

IV. Depth Learning Fine-Tune

We transfer the geometric pretrained networks into monocular depth estimation using monodepth2 [18]. Monodepth2 is a state-of-the-art framework that supports self-supervised monocular depth learning using stereo or monocular images. Compared with methods that have complex structures or loss functions, monodepth2 only optimizes photometric error and depth map smoothness. We choose monodepth2[2] for its simplicity such that the effectiveness of pretrained networks can be well studied.

In stereo supervised learning, the source image is reconstructed by sampling the image on the other side along the epipolar line. The photometric error measures the difference between the source image and the reconstructed image.

In monocular supervised learning, image triplets, $\{I_{t-1}, I_t, I_{t+1}\}$, are taken as inputs. A network is used to estimate the camera poses between adjacent frames: ${}^{c}\mathbf{T}_{t}^{t+1}$ and ${}^{c}\mathbf{T}_{t}^{t-1}$. Using the estimated camera poses and the depth map, optical flows can be calculated using Equation 2 (the dynamic term ${}^{o}\mathbf{T}$ is ignored in monodepth2). The reference image can be reconstructed: I'_t, I''_t using I_{t-1} and I_{t+1}, respectively. Different from the photometric loss that is applied in the pretraining, monodepth2 optimizes the per-pixel minimum photometric error to handle occlusion and dynamic objects:

[2]https://github.com/nianticlabs/monodepth2

TABLE II

QUANTITATIVE COMPARISON OF DEPTH LEARNING MODELS

idx	Method	Init	Train	Lower is better				Higher is better		
				Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
1	Superdepth+pp[23]		S	0.112	0.875	4.958	0.207	0.852	0.947	**0.977**
2	Monodepth2	scratch	S	0.130	1.144	5.485	0.232	0.831	0.932	0.968
3	Monodepth2	d_noI	S	0.114	0.953	5.070	0.211	0.856	0.947	0.975
4	Monodepth2	ImageNet	S	0.109	0.873	4.960	0.209	0.864	0.948	0.975
5	Monodepth2	kc	S	0.107	0.822	4.925	0.204	0.862	0.950	**0.977**
6	Monodepth2	kcd	S	0.107	0.840	4.881	**0.200**	0.865	**0.952**	**0.977**
7	Monodepth2	d	S	**0.105**	**0.816**	**4.820**	0.204	**0.869**	**0.952**	0.976
8	Zhou [8]		M	0.183	1.595	6.709	0.27	0.734	0.902	0.959
9	GeoNet [11]		M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
10	LEGO [14]		M	0.162	1.352	6.276	0.252	-	-	-
11	EPC++ [16]		M	0.141	1.029	5.350	0.216	0.816	0.945	0.979
12	GLNet(w/o ref) [25]	ImageNet	M	0.135	1.070	5.230	0.210	0.841	0.948	0.980
13	Monodepth2	scratch	M	0.132	1.044	5.142	0.210	0.845	0.948	0.977
14	Monodepth2	d_noI	M	0.117	0.832	4.790	0.194	0.864	0.958	**0.982**
15	Monodepth2	ImageNet	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
16	Monodepth2	kc	M	0.121	0.967	5.044	0.200	0.866	0.956	0.980
17	Monodepth2	kcd	M	0.117	0.943	5.003	0.199	0.871	0.957	0.979
18	Monodepth2	d	M	**0.112**	**0.820**	**4.707**	**0.189**	**0.879**	**0.961**	**0.982**
19	EPC++ [16]		MS	0.128	0.935	5.011	0.209	0.831	0.945	0.979
20	Monodepth2	scratch	MS	0.127	1.031	5.266	0.221	0.836	0.943	0.974
21	Monodepth2	d_noI	MS	0.114	0.905	5.024	0.206	0.856	0.950	0.978
22	Monodepth2	ImageNet	MS	0.106	0.818	4.750	0.196	0.874	0.957	0.979
23	Monodepth2	kc	MS	0.105	**0.800**	4.773	0.196	0.874	0.957	0.979
24	Monodepth2	kcd	MS	0.105	0.804	**4.693**	**0.193**	0.874	**0.958**	**0.980**
25	Monodepth2	d	MS	**0.103**	0.809	4.761	0.194	**0.876**	**0.958**	**0.980**
26	Monodepth2↓	obj	MS	0.104	0.786	4.656	0.193	0.879	0.958	0.980
27	Monodepth2↓	ImageNet	MS	0.103	0.781	4.644	0.193	0.880	0.958	0.980
28	Monodepth2↓	kcd_hd	MS	**0.099**	**0.757**	**4.547**	**0.187**	**0.888**	**0.961**	**0.981**
29	Monodepth2↓*	obj	MS	0.099	0.744	4.447	0.188	0.886	0.962	0.981
30	Monodepth2↓*	ImageNet	MS	0.098	0.742	4.477	0.187	0.889	0.962	0.981
31	Monodepth2↓*	kcd_hd	MS	**0.093**	**0.704**	**4.367**	**0.183**	**0.896**	**0.964**	**0.982**

Methods are trained and evaluated on the KITTI 2015 dataset. All methods are evaluated without post-processing or online refinements except Superdepth (idx-1).

The geometric-transferred networks achieves all best performances and most of the second-best performance.

Legend:
Init: the initialization model.
Scratch: the model is initialized randomly.
ImageNet: the model is pretrained on ImageNet classification task.
S: stereo-supervised
M: monocular-supervised
MS: monocular-stereo-supervised
Best results are **bold**.
Second-best results are underlined.
↓ : uses ResNet50 as the backbone network.
* : the model is trained and evaluated on 1024x320.

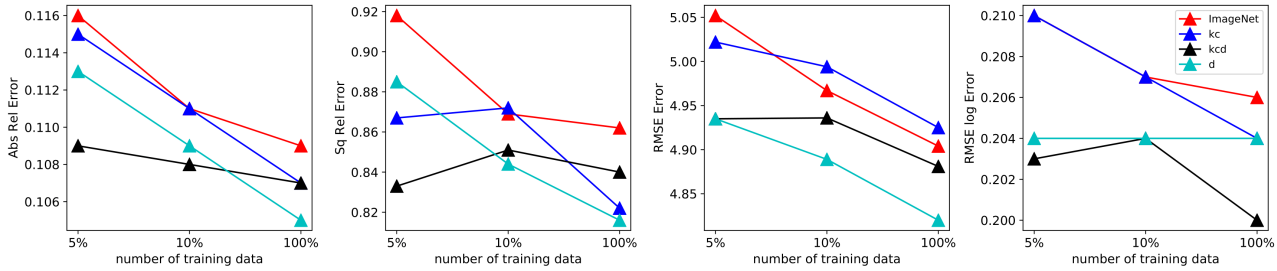


Fig. 4. Few-shot depth learning using pretrained networks. Different sizes of the KITTI dataset are used to train depth estimation networks. The legend indicates the pretrained network used for training initialization. As shown, geometric-pretrained networks perform better than ImageNet-pretrained networks in most of the cases. Specifically, the network transferred from *pretrained network d* using 10% training data performs consistently better than ImageNet-transferred network using 100% training data.

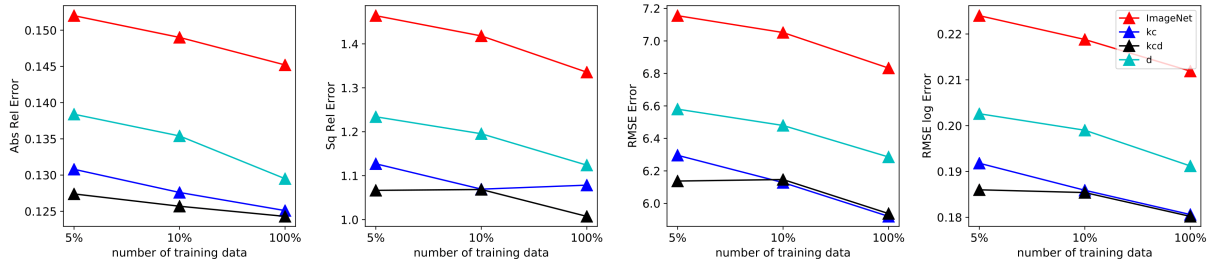


Fig. 5. Generalization ability of depth estimation networks on the CityScapes dataset. Geometric-pretrained networks perform consistently better than ImageNet-transferred networks. With only 5% of the KITTI fine-tune data, geometric-pretrained networks can generalize better than ImageNet-pretrained networks fine-tuned with 100% data.

$$L_{pho} = min(L_{pho}(I'_t, I_t), L_{pho}(I_{t-1}, I_t), L_{pho}(I''_t, I_t), L_{pho}(I_{t+1}, I_t)). \quad (6)$$

V. EXPERIMENTS

In this section, we demonstrate that the geometric pretraining:

1) improves the accuracy of fine-tuned models,
2) helps few-shot learning with much less (e.g., 5%) data,
3) improves the generalization ability.

A. Geometric Pretraining Improves Accuracy

We transfer the pretrained structure encoders into depth estimation networks using monodepth2 [18] without any

modification. Specifically, we use the Eigen full split to train stereo-supervised depth estimation and use the Zhou split to train monocular-supervised or monocular-stereo-supervised depth estimation. To evaluate the performance of trained networks, we follow the standard approach that caps depth values to 80 m. Monocular-supervised depth maps are median scaled [8] before evaluation to correct the unknown scale in monocular sequences. We evaluate recent state-of-the-art methods including our pretrain-transferred models in Table II. We also evaluate the transferred performance of the ResNet50 from object detection for driving scenes [43] (denoted as *obj* in the table).

When initialized with geometric-pretrained models, the model accuracy *improves consistently* compared with random initialized models (e.g. comparing *idx-2* with *idx-3*). However, the improvement is not comparable with that from ImageNet-pretrained models. This can be explained by the fact that d_noI is trained with only $38k$ samples. On the contrary, public pretrained ResNet models are trained on ImageNet with $1.3M$ images that may provide more accurate low-level feature representation. Using ImageNet-pretrained networks as the starting point of the geometric pretraining overcomes the limit of the current small driving video dataset. When initialized with ImageNet→Geometric-pretrained networks, the trained networks show superior results compared to the original monodepth. Although designed for driving applications, *obj*-transferred models do not performs well compared to ImageNet and the proposed networks. The performance gap between *obj*-networks and the pretrained networks proves the importance of the task-specific pretraining design.

Interestingly, the pretrained network using only drive large dataset (*d*) brings the most benefits compared with other pretrained networks. On the other hand, geometric-pretrained models that use the KITTI dataset (*kc* and *kcd*) do not perform well when transferred using only monocular images as supervision. This can be explained by the fact that even the pretraining and fine-tuning are two different tasks, they are highly related thus networks suffer from overfitting when the same supervision is used twice (e.g., for *kc* and *kcd*, KITTI monocular supervision is used twice).

In short, using geometric pretrained networks as the initialization for depth learning brings more accuracy compared to ImageNet-initialized models. To overcome the small size of pretraining datasets, second-order transfer learning (ImageNet→Geometric-pretraining→depth-learning) can be adopted. We also achieve the new state-of-the-art results by a large margin using geometric-pretrained ResNet50 as the encoder in a higher resolution.

B. Geometric Pretraining Improves Few-shot Learning

Since encoder networks learn the structure information during the geometric pretraining, depth maps can be learned using small size training data. To prove the ability of few-shot learning, we train a number of stereo-supervised depth estimating networks using different sizes (100%, 10%, and 5%) of the KITTI dataset. Since training with small datasets may lead to overfitting, in Figure 4, we report the best performance of each model among the training epochs.

As shown, few-shot depth learning using geometric-pretrained networks shows advantages compared to using ImageNet-pretrained networks. Comparing the performance of *kc* (blue line) and *kcd* (black line), the few-shot learning improves when the size of the pretraining dataset increases. Even without the KITTI dataset in the pretraining, the network *d* that is pretrained using internet images performs consistently better than ImageNet-pretrained networks.

C. Geometric Pretraining Improves Generalization

Here, the generalization ability of depth estimation models is tested. We use the CityScapes [40] dataset to evaluate the performance of trained models from Section V-B. Because depth maps provided by the dataset are calculated by stereo matching methods and contain outliers, we remove the largest 5% error terms before evaluation. Depth maps are median-scaled [8] to match the difference of the intrinsic parameters in KITTI and CityScapes.

The results are shown in Figure 5. Geometric-pretrained networks show a better generalization ability than ImageNet-pretrained networks by a large margin. Specifically, networks (*kc* and *kcd*) that are pretrained using CityScapes show the best results. Although camera intrinsic parameters are not utilized during the pretraining and only optical flow is estimated, these networks show superior generalization abilities on the dataset. This suggests another way to generalize depth estimation networks into certain scenes in which only *uncalibrated* images are available (using available images to pretrain the backbone networks). Without CityScapes images in the pretraining dataset, the model *d* still performs consistently better than ImageNet-pretrained networks indicating that pretrained networks learn generalizable structure information. More generalization examples can be found at the project page.

VI. Conclusion and Future Work

We propose a simple but effective pretraining task, called geometric pretraining, that is designed for monocular depth learning. The pretraining task only requires *uncalibrated* monocular image sequences and thus can utilize unlimited resources from the Internet. Extensive experiments are used to prove that, in terms of depth transfer learning, the proposed geometric pretrained model performs better in accuracy, few-short learning, and generalization ability compared to ImageNet-pretrained networks. The pretraining task can also be used to generalize depth learning into specific scenes without the calibration or synchronization of cameras.

In the future, we plan to improve the pretraining tasks that can further decouple each independent moving object. The driving videos dataset will also be further expanded due to the observation that a larger dataset benefits the performance of transferred models.

VII. Acknowledgement

This work was supported from ByteDance.

REFERENCES

[1] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: Real-time dense monocular slam with learned depth prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] N. Yang, R. Wang, J. Stückler, and D. Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse. In *European Conference on Computer Vision (ECCV)*, 2018.

[3] J. M. Facil, A. Concha, L. Montesano, and J. Civera. Single-view and multi-view depth fusion. *IEEE Robotics and Automation Letters*, 2(4):1994–2001, 2017.

[4] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *arXiv preprint arXiv:1905.08598*, 2019.

[5] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[10] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] Y. Zhou, Z. Luo, and J. Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision (ECCV)*, 2018.

[14] Z. Yang, P. Wang, Y. Wang, W. Wu, and R. Nevatia. LEGO: Learning edge with geometry all at once by watching videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] A. Ranjan, V. Jampani, L. Balles, D. Sun, K. Kim, J. Wulff, and M. J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Wu, R. Nevatia, and A. Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[18] C. Godard, O. Aodha, M. Firman, and G. Brostow. Digging into self-supervised monocular depth estimation. In *International Conference on Computer Vision (ICCV)*, 2019.

[19] S. Kornblithand J. Shlens and Q. V. Le. Do better ImageNet models transfer better? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] K. He, R. Girshick, and P. Dollar. Rethinking ImageNet pre-training. In *International Conference on Computer Vision (ICCV)*, 2019.

[21] I. Mehta, P. Sakurikar, and P. Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *International Conference on 3D Vision (3DV)*, 2018.

[22] M. Poggi, F. Tosi, and S. Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *International Conference on 3D Vision (3DV)*, 2018.

[23] S. Pillai, R. Ambrus, and A. Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *Proc. of the IEEE Int. Conf. on Robot. and Autom.*, 2019.

[24] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[25] Y. Chen, C. Schmid, and C. Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *International Conference on Computer Vision (ICCV)*, 2019.

[26] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision (ICCV)*, 2015.

[27] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin. Single view stereo matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] I. Alhashim and P. Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2019.

[29] W. Yin, Y. Liu, C. Shen, and Y. Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *International Conference on Computer Vision (ICCV)*, 2019.

[30] A. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *International Conference on Computer Vision (ICCV)*, 2015.

[32] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, 2018.

[33] X. Zhan, X. Pan, Z. Liu, D. Lin, and C. Loy. Self-supervised learning via conditional motion propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] T. Hui, X. Tang, and C. Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[35] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[36] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[37] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[38] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Transactions on Image Processing*, 2014.

[39] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Research*, 32(11):1231–1237, 2013.

[40] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[41] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[42] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[43] P. Li, X. Chen, and S. Shen. Stereo R-CNN based 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.