2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Depth Generation Network: Estimating Real World Depth from Stereo and Depth Images*

Zhipeng Dong[1], Yi Gao[1], Qinyuan Ren[2], Yunhui Yan[1] and Fei Chen[3], *Member, IEEE*

***Abstract*— In this work, we propose the Depth Generation Network (DGN) to address the problem of dense depth estimation by exploiting the variational method and the deep-learning technique. In particular, we focus on improving the feasibility of depth estimation under complex scenarios given stereo RGB images, where the stereo pairs and/or depth ground-truth captured by real sensors may be deteriorated; the stereo setting parameters may be unavailable or unreliable, hence hamper efforts to establish the correspondence between image pairs via supervision learning or epipolar geometric cues. Instead of relying on real data, we supervise the training of our model using synthetic depth maps generated by the simulator, which deliver complex scenes and reliable data with ease. Two non-trivial challenges, i.e., (i) attaining reasonable amount yet realistic samples for training, and (ii) developing a model that adapts to both synthetic and real scenes arise, whereas in this work we mainly deal with the later one yet leveraging state-of-the-art Falling Things (FAT) dataset to overcome the first. Experiments on FAT and KITTI datasets demonstrate that our model estimates relative dense depth in fine details, potentially generalizable to real scenes without knowing the stereo geometric and optic settings.**

I. INTRODUCTION

Perceiving the depth of the scene is crucial for robotic tasks such as manipulation [1], [2], auto-pilot [3], [4], and navigation [5]. Although state-of-the-art sensors including Time-of-Flight (ToF) cameras, structured light cameras or Light Detection And Ranging (LiDAR) can serve as the quick solutions for depth sensing, their outputs are relatively sparse and may contain defects for many reasons, which is a well-known issue in the robotics community [6]. Stereo RGB cameras, on the other hand, provide a good alternative. It is probably safe to say that compared with the counterparts, they can adapt to highly dynamic situations because they project no light but merely receiving them passively, and thus will not fail due to missing requisite feedback.

We use our stereo vision system—eyes to perceive the world in 3D every day, although we cannot tell the distance accurately by observation as a LiDAR do, we are not likely to directly walk into a house with the glass door closed. Our eyes provide dense depth estimations in complex environments filled with thin, shiny, transparent, and textureless objects, whereas to date most artificial stereo vision systems do not. Despite the extensive progress made in this field [7], [8], in intricate scenes it still remains as an open problem.

(a) Target Scene (b) Ground Truth Depth (c) Our Depth Estimation

Fig. 1. Our depth estimation result on Falling Things (FAT) dataset.

Recent progress in the deep learning technique [9], [10] has cast light on this field, pointing out some promising solutions for this problem, such as establishing the correlation from the left image to the right and vice versa, hence inherently estimating the depth without referring to the ground truth [11], [12]. These works, however, require explicit information about the geometric and optic settings (baseline, camera parameters, etc.) of the stereo vision system to establish correct matching or at least calibrate image pairs, wherein some scenarios, such information is hard to acquire online, resulting in unforeseen matching results.

In this work, our dense depth estimation is investigated with a novel perspective. Besides caring about the accuracy of depth estimation, we ask ourselves whether the proposed method can adapt to sophisticated scenes, requiring no prior knowledge about the stereo configurations, yet hallucinate depth estimations with a relatively small trade-off in accuracy, just as our eyes and brain do. To this end, we assume our input stereo pairs are not calibrated beforehand, which means we are about to estimate the depth without benefiting from epipolar geometry constraints, whereas enduring hazardous factors such as fine details, extremely high or low lighting (shadows), textureless, or transparent materials present within the scene.

Inspired by [14], we opt for tackle the problem by revising the so-called Generative Query Network (GQN). According to their research, a latent representation of the 3D environment can be learned by the variational deep learning model, which has been proved to be helpful for rendering the scene in new viewpoints. This scheme is relatively close to our objective, except that we care more about converting the

*This work was supported by the National Key Research and Development Program of China (2017YFB0304200), the Fundamental Research Funds for the Central Universities (N150308001, N170304014) and Shenzhen Peacock Plan (KQTD2016112515134654)

[1]School of Mechanical Engineering and Automation, Northeastern University, No. 3-11, Wenhua Road, Heping District, Shenyang, Liaoning, P. R. China zhipengdong@stumail.neu.edu.cn, yigaoyi@stumail.neu.edu.cn, yanyh@mail.neu.edu.cn

[2]Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, P. R. China latepat@gmail.com

[3]Active Perception and Robot Interactive Learning Laboratory, Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego, 30 16163 Genova, Italy fei.chen@iit.it

Codes available at https://github.com/DrawZeroPoint/dgn-pytorch

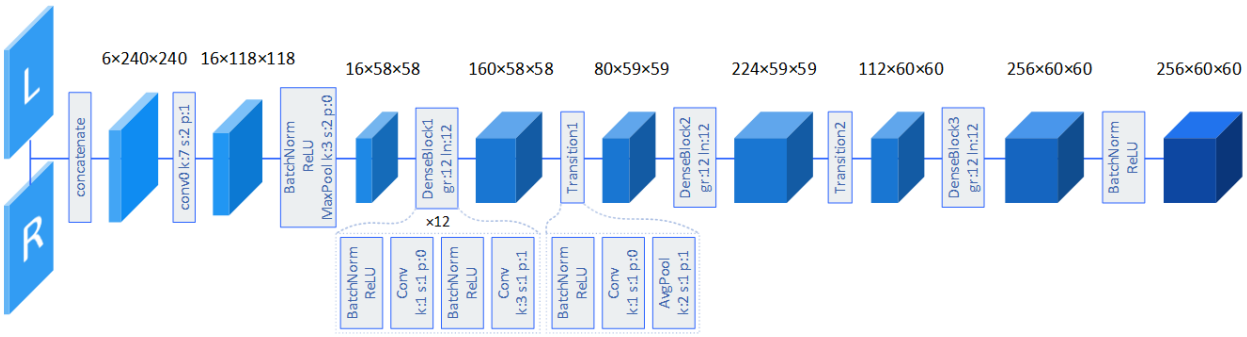

Fig. 2. The architecture of the representation part of DGN. The input is a stereo pair. The intermediate tensors are illustrated with 3D boxes with their dimensions marked atop. Note that the width of the box is not in proportion with the channel size. The neural layers depicted in flat boxes are arranged from left to right following the data flow direction. In this figure k is for kernel size; s for stride; p for padding; gr and ln respectively denote the growth rate and layer number of the dense block. The inner structure of the first dense block and transition layer are shown underneath them. For detailed explaination on dense block and transition layer please refer [13].

latent 3D representation into the depth estimation, rather than sampling that in a new perspective. Our proposed model, the Depth Generation Network (DGN), is trained using a realist synthetic dataset named Falling Things (FAT) [15]. As its name suggested, it records frames when objects fall, whereas currently we only use the footages independently, exploiting no physical characteristics or temporal relations. Moreover, we benchmark the trained model with KITTI 2015 stereo dataset [16] without fine-tuning, showing that our model is improved by training on synthetic data even the environment and the stereo settings are completely different between FAT and KITTI. Fig. 1 illustrates a sample of the result of our approach.

Our contributions are: (i) A deep variational model that estimates dense depth map. We use this model to archive reasonable accuracy without knowing the settings of the stereo vision system; (ii) An generic loss function and corresponding optimization paradigm; and (iii) Experimentally verified the feasibility of generalizing the model for real-world depth estimation by only training on synthetic data.

II. METHODOLOGY

In this section, we present the Depth Generation Network (DGN). We describe its workflow during training and deployment, the optimization and regularization strategies, and the data augmentation.

A. Depth Generation Network

Given a couple of stereo images (L, R), the purpose of our task is predicting the dense depth map $\hat{D}$ observed in the left camera's perspective (our method should have no trouble adapting to the right camera's view, yet we obey the convention in this area) without knowing the baseline distance or the camera parameters. Most existing supervised stereo matching methods minimize the loss between the estimated depth map and the depth map captured by high-end sensors(e.g., LiDAR). However, in complex scenes exhibiting transparent, specular, or areas with fine details, the depth sensation can be unreliable, therefore consequently disabling the methods that rely on the supervision. Although our method also entitles supervision, alternatively, we use simulated depth to train the model, which avoids the distortion factors.

As humans, we can perceive the depth of a scene even from a photo of it, because we leverage prior knowledge and infer the unseen area by reasoning from the adjacent, or even distant part of the scene [17]. As a capable function fitter, DGN has the potential to incorporate 3D senses via training and predicts the depth by inferring to the latent 3D model. What endows it with such ability is a variational Bayesian model contains a interpreter network named f and a generation network named g. We use the strategy of maximizing the likelihood of the distribution over the generated depth, and given the synthetic ground truth, following the Expectation-Maximization (EM) scheme to organize and optimize the proposed model with the latent 3D representations (denoted as $\mathbf{r}$) of the input (L, R).

B. Representation Architecture

In both of the training and testing phase, $\mathbf{r}$ is represented as:

$$\mathbf{r} = f(L, R), \tag{1}$$

in which f is a typically convolutional neural network (CNN). We use the *Dense Convolutional Network (DenseNet)* architecture proposed in [18] instead of the *Tower* CNN used in [14] which can be adopted with minor modifications in our configuration, because it strengthens feature propagation, and reuses feature-maps throughout the network. Other alternatives [19], [20] can also be tested in future work. We believe properly designed representation network is vital for the model to detect the correlations between distant regions of different scales, and to promote the quality and robustness of stereo matching. The comparison on the performance of different representation structures is detailed in Sec. III-B.

Fig. 2 illustrates the representation network framed in the DenseNet. We use this framework for all the experiments in this article except the one specifically claimed. As interpreted in Fig. 2, the input of f, i.e., the left and right images of size $3 \times 240 \times 240$ are concatenated at the beginning, and then fed into the body of the network containing 3 dense blocks. The output is a tenor $\mathbf{r}$ of size $256 \times 60 \times 60$, being the input of the generation network described below.

C. Generation Architecture

With the representation $\mathbf{r}$ and the scaled (pixel values range in $[0, 1]$, detailed in Sec. II-F) the left ground-truth depth map D, we formulize the conditional latent variable models g_θ that implicitly describe densities $g_\theta(D|\mathbf{r})$ over ground truth depth map D, given $\mathbf{r}$, though a marginalisation over a set of latent variables $\mathbf{z}$. θ is the parameter set of this network:

$$g_\theta(D|\mathbf{r}) = \int g_\theta(D|\mathbf{z}, \mathbf{r})\pi_\theta(\mathbf{z}|\mathbf{r})d\mathbf{z}, \tag{2}$$

where $g_\theta(D|\mathbf{z}, \mathbf{r})$ is a conditional density referred to as the depth generation model, $\pi_\theta(\mathbf{z}|\mathbf{r})$ is a conditional prior.

Similar to [14], we parametrize the conditional density $g_\theta(D|\mathbf{z}, \mathbf{r})$ and $\pi_\theta(\mathbf{z}|\mathbf{r})$ with recurrent latent Gaussian models [21], where the vector of latent variables $\mathbf{z}$ is split into K groups $\mathbf{z}_k, k \in [0, K)$ and the density over $\mathbf{z}_k$ is constructed sequentially. According to this sequential architecture, the prior $\pi_\theta(\mathbf{z}|\mathbf{r})$ can be written as an auto-regressive density:

$$\pi_\theta(\mathbf{z}|\mathbf{r}) = \prod_{k=0}^{K-1} \pi_{\theta_k}(\mathbf{z}_{k+1}|\mathbf{r}, \mathbf{z}_k) \tag{3}$$

where θ_k refers to the subset of parameters θ that are used by the conditional density at step k.

During *training*, the generation procedure can be defined by a sequence of conditional computations expressed by the following equations:

$$\pi_{\theta_k}(\cdot|\mathbf{r}, \mathbf{z}_k) = \mathcal{N}(\cdot|\eta^\pi_\theta(\mathbf{h}^g_k)) \tag{4}$$

$$(\mathbf{c}^g_{k+1}, \mathbf{h}^g_{k+1}, \mathbf{u}_{k+1}) = C^g_\theta(\mathbf{r}, \mathbf{c}^g_k, \mathbf{h}^g_k, \mathbf{u}_k, \mathbf{z}_k) \tag{5}$$

$$\hat{D} \sim \mathcal{N}(D|\mu = \eta^g_\theta(\mathbf{u}_k), \sigma = \sigma_s) \tag{6}$$

where the convolutional network $\eta^\pi_\theta(\mathbf{h}^g_k)$ map its respective inputs to the means and standard deviations of a Gaussian density and η^g_θ maps its inputs to the mean of the Gaussian density that generates $\hat{D}$. $\mathbf{z}_k$ is sampled from the posterior distribution as explained in Sec. II-D. The core C^g_θ is a skip-connection Convolutional Long Short-Term Memory network [14]. The initial states of standard LSTM [22] state variables $\mathbf{h}^g_k$, $\mathbf{c}^g_k$, and $\mathbf{u}_k$ are

$$(\mathbf{h}^g_0, \mathbf{c}^g_0, \mathbf{u}_0) = (\mathbf{0}, \mathbf{0}, \mathbf{0}), \tag{7}$$

and $\mathbf{u}_k$ is updated as

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Delta(\mathbf{h}^g_{k+1}), \tag{8}$$

where $\Delta(\mathbf{h}^g_{k+1})$ is a transposed convolution upsampling the image.

Please note that during *deployment*, we sampling $\mathbf{z}_{k+1}$ from the prior distribution as $\mathbf{z}_{k+1} \sim \pi_{\theta_k}(\cdot|\mathbf{r}, \mathbf{z}_k)$, because the trained model should produce analogical prior density to posterior one. Besides, we let $\hat{D}$ equal to $\eta^g_\theta(\mathbf{u}_k)$ instead of sampling $\hat{D}$ with (6), because μ is actually the unbiased estimation of $\hat{D}$, yet sampling breaks the concentration to $\hat{D}$ as proofed by experiments.

D. Inference architecture

In this part, we fomulize the variational posterior density $q_\phi(\mathbf{z}|D, \mathbf{r})$ parameterized by a sequential neural network that shares some of its parameters with the generative network. In analogy to the prior $\pi_\theta(\mathbf{z}|\mathbf{r})$, $q_\phi(\mathbf{z}|D, \mathbf{r})$ is written as an auto-regressive density:

$$q_\phi(\mathbf{z}|D, \mathbf{r}) = \prod_{k}^{K-1} q_{\phi_k}(\mathbf{z}_{k+1}|D, \mathbf{r}, \mathbf{z}_k) \tag{9}$$

where ϕ_k refers to the subset of parameters ϕ that are used by the conditional density at step k. The variational posterior can be expressed by the following equations:

$$(\mathbf{c}^e_{k+1}, \mathbf{h}^e_{k+1}) = C^e_\phi(D, \mathbf{r}, \mathbf{c}^e_k, \mathbf{h}^e_k, \mathbf{h}^g_k, \mathbf{u}_k) \tag{10}$$

$$q_{\phi_k}(\cdot|D, \mathbf{r}, \mathbf{z}_k) = \mathcal{N}(\cdot|\eta^q_\phi(\mathbf{h}^e_k)) \tag{11}$$

$$\mathbf{z}_{k+1} \sim q_{\phi_k}(\cdot|D, \mathbf{r}, \mathbf{z}_k) \tag{12}$$

where we use superscript e to denote the relevant variables of the inference LSTM. The convolutional network η^q_ϕ map its inputs to the sufficient statistics of the posterior distribution, from which the variational hidden parameter $\mathbf{z}$ can be derived by sampling during training. Please note that during deployment, the inference part is detached from DGN and hence receives no information about D.

E. Optimization

Training the depth generation model on a dataset $\mathcal{S} = \{(L_i, R_i, D_i)\}$ meant to maximize $\sum_i \ln g(D_i|\mathbf{r}_i)$, where $\mathbf{r}_i = f(L_i, R_i)$. Since

$$\begin{aligned}
\sum_i \ln g_\theta(D_i|\mathbf{r}_i) &= \sum_i \int q_\phi(\mathbf{z}_i) \ln g_\theta(D_i|\mathbf{r}_i) d\mathbf{z}_i \\
&= \sum_i \int q_\phi(\mathbf{z}_i) \ln \frac{q_\phi(\mathbf{z}_i) g_\theta(D_i, \mathbf{z}_i|\mathbf{r}_i)}{q_\phi(\mathbf{z}_i) g_\theta(\mathbf{z}_i|D_i, \mathbf{r}_i)} d\mathbf{z}_i \\
&= \sum_i \int q_\phi(\mathbf{z}_i) \ln \frac{g_\theta(D_i, \mathbf{z}_i|\mathbf{r}_i)}{q_\phi(\mathbf{z}_i)} d\mathbf{z}_i + \\
&\quad \sum_i \int q_\phi(\mathbf{z}_i) \ln \frac{q_\phi(\mathbf{z}_i)}{g_\theta(\mathbf{z}_i|D_i, \mathbf{r}_i)} d\mathbf{z}_i \\
&= ELBO(\phi, \theta) + \\
&\quad \sum_i KL(q_\phi(\mathbf{z}_i)||g_\theta(\mathbf{z}_i|D_i, \mathbf{r}_i)),
\end{aligned} \tag{13}$$

where the density $q_\phi(\mathbf{z}_i)$ is an approximation to the true posterior denstity, ϕ is the set of parameters of density q. $ELBO(\phi, \theta)$ is the evidence lower bound (ELBO) w.r.t both ϕ and θ, and $KL(q_\phi(\mathbf{z}_i)||g_\theta(\mathbf{z}_i|D_i, \mathbf{r}_i)$ is the KL-divergence of the approximate posterior and the ground-truth. Notice that KL-divergence is always ≥ 0, while $\ln g_\theta(D_i|\mathbf{r}_i) \in (-\inf, 0)$, hence maximizing $\sum_i \ln g_\theta(D_i|\theta)$ is equivalent to minimizing $-ELBO(\phi, \theta)$, and the loss can be defined as

$$\mathcal{L} = -ELBO(\phi, \theta) + \sum_i KL(q_\phi(\mathbf{z}_i)||g_\theta(\mathbf{z}_i|D_i, \mathbf{r}_i)) \tag{14}$$

where the gradients of $-ELBO(\phi, \theta)$ w.r.t ϕ and θ are approximated in an unbiased manner by drawing a small number of samples from $q_\phi(\mathbf{z}|D, \mathbf{r})$.

We use the same annealing strategy as [14] to decrease the standard derivation (SD) σ_s in (6) over the duration of training

$$\sigma_s = \max(\sigma_f + (\sigma_i - \sigma_f) * (1 - s/s_t), \sigma_f), \tag{15}$$

where $\sigma_i = 2.0$, $\sigma_f = 0.7$ are respectively the initial and final SD of the normal distribution in (6), s_t is the total training steps. This strategy converges $-ELBO(\phi, \theta)$, encouraging the model to stabilize the depth distribution globally in the beginning and refine the details after the global structures are properly learned.

F. Data Augmentation and Regularization

We use several data augmentation strategies to expand the variety of the scene given the FAT dataset, which by and large contains not sufficient samples for training a deep learning model. We sequentially transform the input (L, R, D) with *RandomCrop*, *Resize*, and *RandomVerticalFlip*. For *RandomCrop*, identical square crop box of size c is applied to the images, where $c \in (240, 540)$ is a random integer; The size of the original image is 540×960 Since we randomize both the cropping location and c, the model will not concentrate only on a fixed part of the image. After cropping, the images are resized to 240×240 using *Resize*; and finally, we apply *RandomVerticalFlip* to them with 50% chance to break the pattern of the training set, wherein the foreground tend to locate on the lower part of the image.

Before feeding (L, R) into the network, we normalized their values from $[0, 255]$ to $[0, 1]$ by multiplying $1/255$. We also normalized the ground truth depth map ranging in $[0, 65535]$ to $[0, 1]$, but with the following equation:

$$d' = \frac{d - d_{min}}{d_{max} - d_{min}}, \tag{16}$$

where d' is the normalized version of d, whose maximum and minimum are repectively d_{max} and d_{min}. We found this strategy particularly helpful for stabilizing the training procedure, yet as a trade-off, it make our model only estimate the relative depth of a scene.

III. EXPERIMENTS

A. Traning Details

We implement DGN using PyTorch. Our models are sorely trained from scratch on a subset of FAT footages (detailed below) shuffled into a list. We used a batch size of 8 and trained for 200k steps, which took almost 150 hours on a desktop with Intel Xeon E5-2620v3, a NVIDIA GeForce 1080Ti GPU with 11GB memory, and 64GB RAM. Adam [23] was used for optimization. We determine the learning rate δ as $\max(\delta_f + 8 * (\delta_i - \delta_f) * (1 - s/s_t), \delta_f)$, where $s_t = 2e^6$; δ_i and δ_f are the initial and final learing rate. We keep other hyperparameters (β and ϵ) as default.

There are 21 household objects generated in virtual environments built with Unreal Engine 4 (UE4) in the Falling Things dataset. The number of annotated photos is 60k. The stereo RGB pairs along with registered dense depth pairs were captured simultaneously, making it ideal for training or testing our model on. Moreover, this dataset provides high-fidelity models and quality images with a variety of 3D backgrounds, lighting conditions, and shadows. All these features potentially enable the trained model to generalize to other environments including the real world.

TABLE I
EVALUATION ON FAT-EVAL DATASET

Methods	RMS[a]	ACC[b] $\lambda < 1.25$	ACC[b] $\lambda < 1.25^2$
kitti-resnet [12]	0.314	25.7	39.6
kitti-stereo [12]	0.338	26.1	37.5
DGN-Tower (ours)	0.177	36.7	53.3
DGN-DenseNet (ours)	**0.158**	**38.1**	**59.2**

[a] Lower is better
[b] Higher is better

We selected only the *single* subset of FAT containing $31,500$ training pairs as our training set, denoted as FAT-train, of which each scene contains only one object from the 21 categories with the presence of all three backgrounds (kitchen, forest, temple). Whereas we used the rest samples, i.e., the *multiple* subset denoted as FAT-eval ($30,000$ pairs in total, each frame contains multiple objects) for evalutaion.

B. Evaluation on FAT-eval

This work compares the proposed method against one of the state-of-the-art stereo matching algorithm, i.e., Monodepth [12] on FAT-eval. This method is chosen because: (1) It provides an open-source implementation with multiple pre-trained models, among which *model_kitti* and *model_kitti_stereo* were directly used without fine-tuning in our experiments; (2) It is proved by the authors to be superior to both learning based [11], [24] and nonlearning based pioneers [25] regarding the accuracy and adaptiveness; (3) Similar to our work, it also tried to establish a latent depth prediction model (although following a different scheme), and used stereo RGB pairs during training; (4) It is featured by the ability of generalizing to unseen environments.

Since our models are trained on FAT-train, there should be no surprise that our models outperform Monodepth on FAT-eval. Fig 3 shows some qualitative results. While quantitive comparisons go to Table I, for which we stochastically drew 100 samples from FAT-eval for 10 times, and calculated the averaged Root Mean Square (RMS) and accuracy (ACC) following the metrics reported in [26]:

$$\text{RMS}: \sqrt{\frac{1}{T}\sum_{i \in T} ||d_i - d_i^{gt}||^2} \tag{17}$$

$$\text{ACC}: \%\ of\ d_i\ s.t.\ \max(\frac{d_i}{d_i^{gt}}, \frac{d_i^{gt}}{d_i}) = \lambda < \Lambda \tag{18}$$

where T denotes the total of pixels within the depth maps. Regaring the results, we confirmed that (1) our model properly handled the intrinsic 3D features at least within the synthetic scenes, and (2), using an elaborate representation

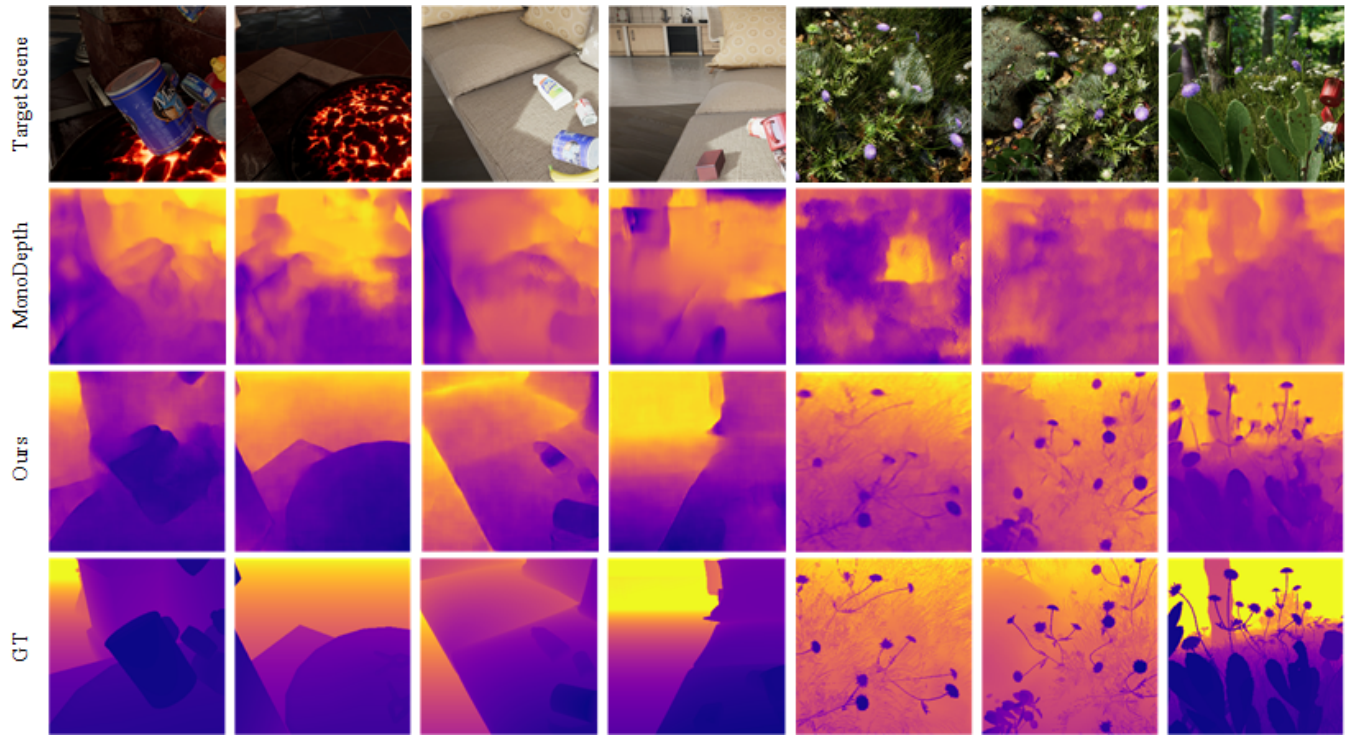

Fig. 3. The results of Monodepth and our purposed DGN on FAT-eval dataset.

network such as DenseNet is benifical comparing with the shallow models such as *Tower*. It is worth noting that for DGN models d_i and d_i^{gt} were all normalized using (16), while Monodepth is essentially a monocular deep estimation method that produces disparities, we calculated the relative depth map d' from disparities as

$$d' = 1 - \frac{p - p_{min}}{p_{max} - p_{min}}, \tag{19}$$

such that the disparity map $p \in [p_{min}, p_{max}] \mapsto [1, 0]$, where p_{max} and p_{min} are repectively the maximum and minimum of the disparity image.

C. Evaluation on KITTI

The performance of our model is validated on the KITTI scene_flow/training split. The official 400 stereo pairs for training are exploited as the testing set. We randomly drew 100 pairs for testing and repeated this procedure for 10 times just as the scheme in Sec. III-B. Despite that we never trained DGN (the sparse ground truth depths generated by LiDAR violate the principle of telling the "truth" to the robot) on KITTI dataset, the depth estimation is relatively reasonable as shown in Table 2 and with some example outputs visualized in Fig. 4. It is worth noting that for this comparison we directly use the results of Monodepth as the ground truth, by this means we can reveal that DGN performed better when a new environment is given, since it archived 48.7% ACC ($\lambda < 1.25$) on KITTI, while Monodepth only gain 25.7% on FAT-eval. Other metrics also supported that our model implicitly learned the basic primitives forming the 3D scene, and is able to infer the depth not only from pixel level but also in a regional perspective. Please note that our model use no post-processing or consistency check as in [12], [27], yet the depth preserves the consistency on flat surfaces, yet is sharp on edges.

TABLE II
EVALUATION ON KITTI DATASET

Methods	RMS	ACC $\lambda < 1.25$	ACC $\lambda < 1.25^2$
kitti-resnet[a][12]	0	100	100
DGN-DenseNet (ours)	0.194	48.7	76.1

[a] Here we used the results of Monodepth as a reference of our method, since the dense depth ground truth of KITTI is unavailable

IV. CONCLUSIONS

The idea of promoting the flexibility of stereo vision, and providing robots with synthetic *true* depth data to overcome hazardous factors in stereo matching have driven us to present the Depth Generation Network for dense depth estimation from stereo pairs. Our generalizable model shows the potential of being employed in the real world even it has only been trained with synthetic data. What unveiled to be crucial for adapting such ability is its novel structure, wherein the feature sensitive dense network extracts latent representations, while the recurrent generation models adaptively phase that to potential distributions coordinating with the ground truth depths. This novel architecture comprehensively understands the structure and geometric relationships within the scene, such that a reasonable depth hallucination of the environment can be formed without accessing stereo settings or post-processing schemes. In this sense, it is applicable

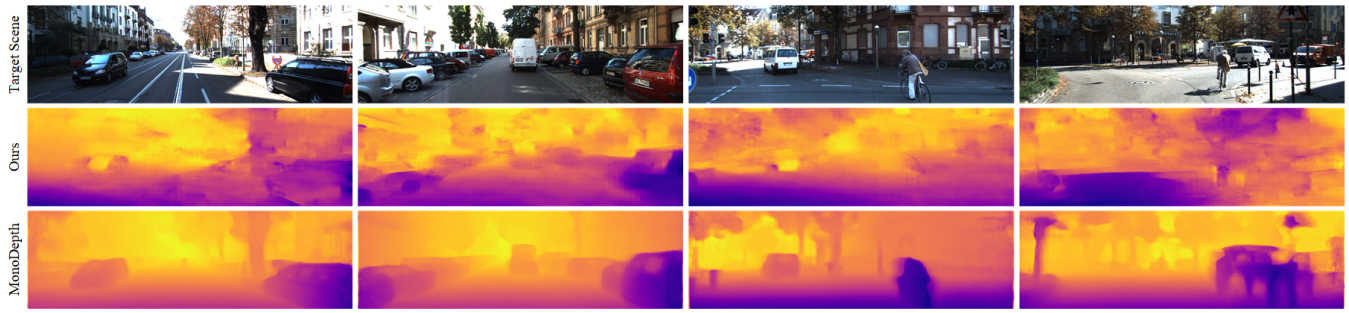

Fig. 4. The results of Monodepth and our purposed DGN on KITTI dataset.

in scenarios where the stereo settings can be unreliable or unavailable, or cost-sensitive applications requiring only loosely depth estimations.

In future work, we are about to evolute DGN in more complicate synthetic worlds to make the domain transformation more robust and smooth. Besides, our model may reach out to videos instead of still images by adding a time dimension into the input, and by this means, we may preserve the temporal consistency of depth estimations. Finally, our proposed method may also be leveraged to predict the instance segmentations as we developed at [28] (code avaliable at `https://github.com/DrawZeroPoint/hope`) or 6 Degree-of-Freedom (DOF) poses of the objects by training on FAT, wherein the training data have been provided, such that it can go beyond as a depth estimator, but conducts certain robotic tasks in an end-to-end fashion.

REFERENCES

[1] P. Ramon Soria, B. C. Arrue, and A. Ollero, "Detection, location and grasping objects using a stereo sensor on uav in outdoor environments," *Sensors*, vol. 17, no. 1, p. 103, 2017.

[2] F. Chen, M. Selvaggio, and D. G. Caldwell, "Dexterous grasping by manipulability selection for mobile manipulator with visual guidance," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1202–1210, 2019.

[3] L. Meier, D. Honegger, V. Vilhjalmsson, and M. Pollefeys, "Real-time stereo matching failure prediction and resolution using orthogonal stereo setups," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5638–5643.

[4] N. Smolyanskiy, A. Kamenev, and S. Birchfield, "On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach," *arXiv preprint arXiv:1803.09719*, 2018.

[5] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2371–2378.

[6] V. De Silva, J. Roche, and A. Kondoz, "Robust fusion of lidar and wide-angle camera data for autonomous mobile robots," *Sensors*, vol. 18, no. 8, p. 2730, 2018.

[7] A. Geiger, M. Roser, and R. Urtasun, *Efficient large-scale stereo matching*. Springer, 2010, pp. 25–38.

[8] R. A. Jellal, M. Lange, B. Wassermann, A. Schilling, and A. Zell, "Ls-elas: Line segment based efficient large scale stereo matching," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, Conference Proceedings, pp. 146–152.

[9] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *Journal of Machine Learning Research*, vol. 17, no. 1-32, p. 2, 2016.

[10] X. Liu, Y. Luo, Y. Ye, and J. Lu, "Mc-dcnn: Dilated convolutional neural network for computing stereo matching cost," in *International Conference on Neural Information Processing*. Springer, 2017, Conference Proceedings, pp. 249–259.

[11] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, Conference Proceedings, pp. 740–756.

[12] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, vol. 2, 2017, Conference Proceedings, pp. 270–279.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[14] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, and K. Gregor, "Neural scene representation and rendering," *Science*, vol. 360, no. 6394, pp. 1204–1210, 2018.

[15] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2038–2041.

[16] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 60–76, 2018.

[17] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1788–1792, 2017.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, vol. 1, 2017, Conference Proceedings, p. 3.

[19] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Transactions on Image Processing*, 2018.

[20] B. Zhang, J. Gu, C. Chen, J. Han, X. Su, X. Cao, and J. Liu, "One-two-one networks for compression artifacts reduction in remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[24] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, Conference Proceedings, pp. 2366–2374.

[25] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2144–2158, 2014.

[26] F. Liu, C. Shen, G. Lin, and I. D. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, 2016.

[27] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2018, Conference Proceedings.

[28] Z. Dong, Y. Gao, J. Zhang, Y. Yan, X. Wang, and F. Chen, "Hope: Horizontal plane extractor for cluttered 3d scenes," *Sensors*, vol. 18, no. 10, p. 3214, 2018.