3D Scene Reconstruction with Neural Radiance Fields

Ka Ho Lee	Helbert Paat	Hong Wing Pang	Ka Chun Shum	Tun Jian Tan
20342478	20837409	20315504	20467981	20393609

Abstract

The recent success of Neural Radiance Field has drawn much attention in the field. While most of the works are on object rendering, only a few works are on large scene rendering. Since some of our group mates work on autonomous driving, we thought it would be interesting if we could develop an implicit-based cityscape navigation project similar to the GoogleMap's road map navigation. We found an indoor scene rendering work particularly suitable as a starting point of our project. Based on this work, we worked on two dimensions to explore our idea on the work. Specifically, we tested on a few model architectures on its original dataset and, explored the model performance with different KITTI-like datasets. Our results shows the possibility of performing implicit rendering for outdoor complex cityscape, which can be referenced for further development.

1. Introduction

Neural Radiance Field has been shown effective in encoding 3D volumetric models from a few images during training. [17] depicts the possibility to reconstruct the 3D scenes by extracting dynamic objects from the scene. Generative Scene Networks (GSN) [4] has successfully modelled 3D scenes from a sequence of images taken around an enclosed area. Combination of these works to model city scenes would allow virtual exploration of the map, automate HD map construction, and urban planning.

We would like to experiment on various real and synthetic datasets to reconstruct a 3D cityscape from the scenes. We will start with a relatively static scene (with fewer moving objects) in the dataset for the neural rendering experiments. We will also explore the potential of incorporating depth information from the dataset to boost model performance. Our final goal is to train a neural network for reconstructing the static scenes from the dataset.

In this work, we aim to achieve the following:

- Apply GSN [4] on various unconstrained and realistic outdoor dataset, the KITTI-360 dataset [7].
- Apply GSN [4] onto a synthetic outdoor and complex dataset generated through CARLA [5].
- Improve on the above baseline, e.g. deal with dynamic objects in training data [13].
- Further improvement by incorporating depth data [2].

2. Background

Novel view synthesis. Some neural rendering approaches have tackled the problem of view synthesis by encoding a continuous radiance field in neural network parameters where high resolution views of realistic scenes train the network [12, 14, 15]. The drawback with this approach is that multiple scenes cannot be represented within the same model, i.e. for every scene, it requires optimizing a new model (where each optimization takes days on commodity hardware). Hence, prior distribution over multiple scenes cannot be learned by the model.

Moreover, generative adversarial approaches in neural radiance field modeling have been studied by some recent works [16, 18]. These models incorporate more scene compositional attributes into the generative networks that enables more controllable image and view synthesis. However, these method are based on an single-object assumption: the reconstructed scene consists of a single object, and the novel views are usually constrained in a viewing sphere or limited angles.

Free camera movement for view synthesis. In many fields such as visualization or game design, it is often desirable to explore scenes with a freely moving camera view. Synthesizing novel views for a freely moving camera that explores an entire scene rather than a camera moving on a sphere oriented towards a single object is also an area that has been explored recently. This setup presents a more complex problem because multiple objects in the scene must be learned by the model.

One approach to solving this problem is to represent the environment using a dictionary-based memory where the camera poses serve as keys and latent observation representations serve as values [6]. Given a query viewpoint at inference, the model predicts the observation by querying the memory. Generated Scene Networks (GSN) [4] is one

of the first works exploring free camera movement using the radiance field approach; this is done by decomposing the scene rendering task into learning an array of radiance fields, each representing a local part of the scene. We build our project upon this work and provide a detailed description of the methodology used in the next section.

Autonomous driving dataset. Since autonomous driving is a trending field, a number of autonomous driving dataset that facilitates the related research are published. Proceeding novel view synthesis task on autonomous driving dataset requires both image sequences and cameara information. KITTI and its related dataset [7, 8, 10] provides high-resolution realistic RGB images, depth images, and the methods to calculate camera calibration. Due to the expensive cost of building real-world dataset, Carla and its related dataset [3, 5] instead provides synthetic RGB and depth views, and as well the ground-truth camera calibration.

3. Methodology

Generative Scene Networks (GSN) [4] is the first generative model for unconstrained scene-level radiance field which allows view synthesis of a freely moving camera in an open environment by decomposing a scene into many local radiance fields. The major contributions of GSN over existing view synthesis methods are twofold:

- Learn a prior distribution for indoor scenes with complex structures, using an adversarial generator-discriminator approach; as well as synthesizing new scenes by sampling from it
- Allowing unconstrained camera control over a given scene, by rendering from different localized radiance fields based on the camera position

In this project, we are interested in the second property, i.e. to model complex scenes using the decomposed local radiance fields approach. While our goal is to reconstruct a single scene instead of synthesizing completely new scenes from scratch, we believe it is possible to produce meaningful results by directly adopting the GSN method, but only train on one single scene.

Global generator. GAN networks typically synthesizes images from a single *latent code* $\mathbf{z}$, a 1-D vector sampled from a simple prior distribution. Each latent code can be viewed as a compact encoding of the generated image. While this is the case for prior works on generative NeRF models, such as GRAF [19] and pi-GAN [1], the authors of GSN hypothesized that a single vector lacks the capacity to encode an entire indoor scene, as opposed to most other NeRF methods that aim to reconstruct a single object only. Therefore, a *global generator* $g : \mathbf{z} \rightarrow \mathbf{W}$ is proposed to model a mapping from a single latent code $\mathbf{z}$ to a 2D array of latent codes, $\mathbf{W}$. Each latent code w_{ij} in $\mathbf{W}$ corresponds to the localized radiance field in a particular location in the generated scene. Conceptually, $\mathbf{W}$ can be thought of as a *floor plan* of the generated scene.

The global generator takes in a 4×4 learned constant input and passes it through multiple modulated convolution (ModConv) blocks, and produces $\mathbf{W}$ as the output. Each convolution block is "modulated" with $\mathbf{z}$, by performing channel-wise multiplication with the post-convolution feature maps. The implementation of ModConv follows directly from StyleGAN2 [9].

Local generator. Like other NeRF-based methods, GSN learns the radiance field with a MLP network f that predicts the radiance for a 5D input $(\mathbf{p}, \mathbf{d})$, at position $\mathbf{p} \in \mathbb{R}^3$ and camera angle $\mathbf{d} \in \mathbb{R}^2$. The model predicts two variables: the RGB appearance $\mathbf{a} \in \mathbb{R}^3$ and the density value $\sigma \in [0, 1]$. In addition, this generator is also conditioned on the local radiance field vector w_{ij}, sampled from $\mathbf{W}$ obtained from the previous generator. Given a global position $\mathbf{p}$, the corresponding w_{ij} is selected from $\mathbf{W}$, and each linear layer in f is modulated by f. Furthermore, to ensure that the radiance fields learned by f represents the local area corresponding to w_{ij}, the global position $\mathbf{p}$ is first translated to the local coordinate $\mathbf{p}'$ with respect to w_{ij}, before it is passed in to f.

Datasets. The original GSN are trained from two datasets: *VizDoom* [21] is collected from in-game scenes from a FPS shooter, and *Replica* [20] is a collection of reconstructions of indoor scenes. In both cases, sequences of RGB images and depth maps are collected from these two sources, as well as the trajectories of extrinsic matrices of the camera while navigating in the scene. In section 4.3, we will elaborate on how we collect training data from autonomous driving sources, and adapt them in the format of VizDoom / Replica to facilitate training on GSN.

Training. As a generative model, GSN is trained adversarially with a discriminator network $D(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^{4 \times w \times h}$ is a concatenated pair of RGB image and depth map (normalized to the range $(0, 1)$). Ablation studies in [4] show that concatenating depth information is essential for training GSN. The architecture of D follows directly from StyleGAN2. The adversarial loss term is given by

$$\begin{aligned} \mathcal{L}_{adv} = \mathbf{E}_{\mathbf{z} \sim p_z, \mathbf{T} \sim p_T}[h(D(G(\mathbf{z}, \mathbf{T})))] \\ + \mathbf{E}_{\mathbf{X} \sim p_S}[h(-D(\mathbf{X}))] \end{aligned} \tag{1}$$

where $\mathbf{z}$ is a random vector sampled from a normal distribution p_z; $\mathbf{T}$ is a camera pose sampled from the distribution

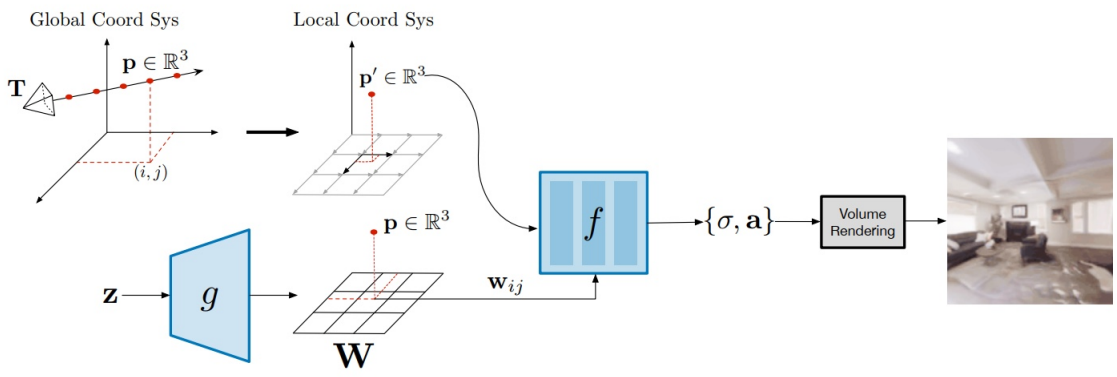

Figure 1. Architecture of GSN generator, image source: [4].

of trajectory poses p_T in the dataset; and $\mathbf{X}$ is the ground truth RGB-D data viewed from $\mathbf{T}$, sampled from the distribution of scenes p_S in the dataset. The softplus penalty is used, with $h(u) = -\log(1 + \exp(-u))$.

The final discriminator loss is regularized with an additional R1 gradient penalty, as well as a decoder reconstruction loss following the implementation of [11]. Since we aim to reconstruct a single scene without the need to synthesize new scenes, we attempted to replace the GAN loss with a simple L2 loss between the generated image $G(\mathbf{T})$ and the ground truth image $\mathbf{X}$, as well as replaced components modulated by the random vector $\mathbf{z}$ with regular convolution layers, since the random vector $\mathbf{z}$ is theoretically not needed for our application. The resulting model failed to train, so it appears it is not trivial to directly adapt the GSN architecture to a non-conditional setting.

4. Experiments

4.1. Benchmarks

To train the model for view synthesis, GSN requires sequences of RGB images with the corresponding depth map for each image. Moreover, the intrinsic parameters and camera pose for each image are also necessary.

For the experiments, we evaluate the performance of the model in terms of generating fake images and the view synthesis task.

We train GSN using both a single sequence and multiple sequences consisting of 100 frames. Images are resized to 64×64 resolution for all generation experiments. The dataset provides intrinsic parameters and the camera pose for each frame. Using the available velodyne point clouds, the depth map is generated for each frame.

4.2. Metrics

We evaluate the generation performance of GSN on the different datasets using the Frechet Inception Distance (FID) metric, which are measures of how varied the real distribution and the generated images in the pretrained image embedding spaces. The results are shown in Table 1.

Dataset	Epoch	FID
KITTI-360	399	222.9
KITTI Odometry	80	97.81
KITTI-CARLA Single	119	87.16
KITTI-CARLA Stereo	119	80.99
CARLA Single	99	213.03
CARLA Stereo	99	179.18

Table 1. FID metrics of models trained on KITTI-360, KITTI Odometry, KITTI-CARLA and CARLA. FID is evaluated at different epoch as models are trained separately with limited amount of training time.

4.3. Datasets

Our datasets are either adopted from existing KITTI-related datasets, generated on our own or adopted from the paper.

4.3.1 KITTI Odometry Dataset

KITTI Odometry is a conventional benchmark driving dataset for autonomous driving related researches [8]. The dataset has been well-maintained that the poses of the vehicle to minimize, ground truth poses and the sensor poses have been carefully calibrated from a subset of the raw KITTI driving sequences. This dataset is chosen for it contains the ground truth information which are needed as an input to the GSN model.

We chose the left RGB camera (cam2) as the camera of interest in our experiments. The input intrinsic matrix and

extrinsic matrix for the camera are computed from the cam2 projection matrix and the inverse of the rectified camera pose relative to the global coordinates respectively. The 5*th* driving sequences of trajectories is selected to train the GSN model because of suitable driving speed, less dynamic objects and less street view occlusions.

To obtain reasonable result, we fit our sequence to the GSN input format so that the model performance won't deviate to much from the original setting. Since the GSN model only takes in square images by default, we chose to first crop the image sequence with the camera principle axis located at the center of the cropped image. Then, the intrinsic matrix is corrected by modifying the center pixel offset component, while the camera focal length remains the same as the image plane position in the global coordinate system is not shifted. For the input depth map, we obtained the sparse depth map by back-projecting the Velodyne LiDAR point cloud onto the image plane of cam2, similar to 4.3.2.

4.3.2 Kitti-360 Dataset

KITTI-360 is a suburban driving dataset consisting of better input modalities, semantic instance annotations, and accurate localization to further research in vision, graphics, and robotics [10].

To use the driving scenes in this dataset, we have first preprocessed the data. The input extrinsic matrices are computed from the provided camera poses with respect to the world frame (*'cam0_to_world.txt'*). We use the provided calibration matrix in the dataset as the intrinsic matrix. For the RGB images, we use only data from a subset of a single driving sequence in the dataset. Specifically, from the raw, unrectified stereo pairs, we only used the left images. Since the width and height of the images in the raw data are not equal, we crop the images to produce a square and choose the image's center view. Since the depth data is not provided, we use the corresponding velodyne scans in binary format defined in Velodyne coordinates to project the corresponding depth map for each image.

4.3.3 KITTI-CARLA Dataset

In parallel with the train on the real world dataset, we would like to also test it on synthetic data in case the real world dataset training fails to learn. KITTI-CARLA [3] is a dataset built from the CARLA v0.9.10 simulator using a vehicle with sensors identical to the KITTI dataset. The sensors settings, paramters, and the positions are the same as the setup used in KITTI.

The objective of this dataset was originally intended for transfer learning researches from synthetic to real dataset. However, we found that it can also be a good comparison with the real world KITTI odometry datset. Our preliminary hypothesis for the model performance is that synthetic dataset should give a better result for being less noisy and visually simpler compared to the real world dataset.

There 7 sequences with 5000 frames in each sequence in the 7 maps of CARLA providing different environments. We selected the Town02 sequence (suburban area environment) to match with the suburban sequence that we chose for the KITTI odometry dataset. Specifically, we selected 300 frames from the 10fps sequence that are without traffic light stopping and with less dynamic objects.

However, later in our experiment, we found that the provided grayscale depth map precision is low. The author carelessly provided a logarithmic grayscale depth map that are converted from a 3-channel depth map only for display purpose. The objects that are close to the vehicle simply the same grayscale value. We have emailed the author to request for the high precision depth map. Unfortunately, he had cleaned up the settings. Now he is busy re-synthesizing the KITTI-CARLA dataset.

4.3.4 Carla Dataset

Since KITTI-CARLA dataset's author was not able to provide high precision depth map within our project time constrain, we decided to generated our own dataset following KITTI's setting.

Carla is an open-source autonomous driving simulator that supports creating synthetic autonomous driving scenes, setting up driving parameters, and providing camera calibration [5].

We built a dataset by generating driving sequences with random vehicle models and initial points, then preferably cut out segments with more buildings in view. The speed of the car is controlled under 25 km/h. During the driving time, 2 RGB and depth cameras are fixed at the front of the vehicle facing forward. We limited the maximum rendering distance of the sensors to be 1000 meters. The frame rate is set to be 5 frames per second. However, in practise, fps seems to vary on different CARLA settings. In order to adapt to GSN, the resolutions are set as 64×64 or 128×128. Since Carla provides APIs to access camera calibration, the 4 by 4 extrinsic matrix and 3 by 3 intrinsic matrix of each frame is deterministically calculated and written to the dataset.

4.4. Results

4.4.1 KITTI Odometry

The GSN model is trained on a similar configuration of Visdoom training setting in the GSN model. We trained the model on 100 frames and their corresponding depth maps from the 5*th* driving sequence of KITTI Odometry. With limited computation resources and time constraint, the model was trained till epoch 80. Fig. 4 shows the image synthesized during training.

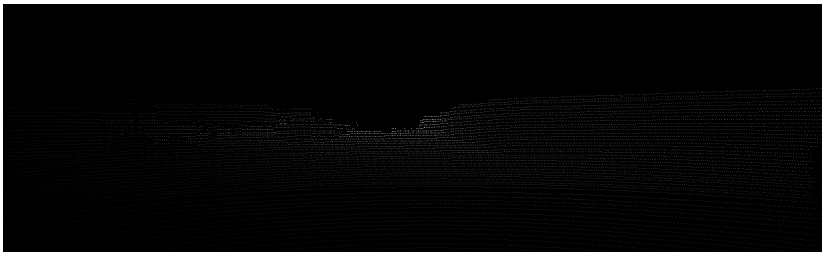

Figure 2. Sparse depth map generated from projecting the lidar point cloud onto the image plane.

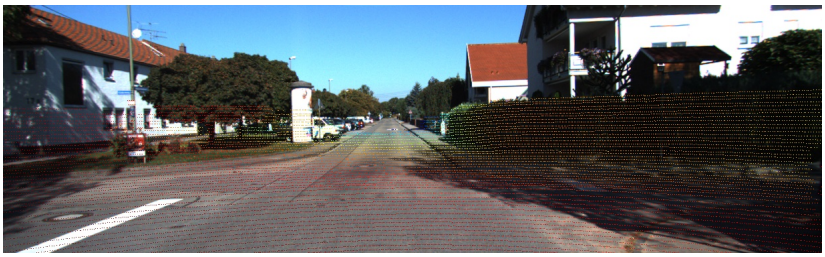

Figure 3. Visualization of the projected lidar point cloud onto the camera2 image in KITTI Odometry Dataset.

Figure 4. Synthesised images generated by GSN during KITTI Odometry dataset training. The model is trained for 80 epochs.

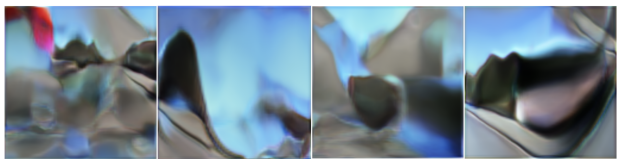

Figure 5. Image synthesised by the GSN model trained on KITTI Odometry dataset using *walkthrough_demo.ipynb*.

After training, we used the *walkthrough_demo.ipynb* to explore the model. The results is shown in Fig. 5.

4.4.2 KITTI-360

First, we trained the model on this dataset using a single sequence of images consisting of 100 frames and their corresponding depth maps. As seen in Fig. 6, the model was not able to generate high quality images at 109th epoch. After training 389 epochs, we can see that the image quality has improved and has become closer in terms of visual appearance to outdoor driving scenes of the KITTI-360 dataset. However, training the model for hundreds of epochs requires days.

To see the generation performance of the model when trained on multiple sequences of outdoor driving scenes, we also trained the model on few sequences of the KITTI-360 dataset each consisting of 100 frames. As seen in Fig. 7, this results in poorer image generation quality. The fake images generated at epoch 109 and epoch 389 are of low quality.

Figure 6. Fake images generated by GSN during KITTI-360 dataset training. The two leftmost images were generated at the 109th epoch and the two rightmost images were generated at the 389th epoch.

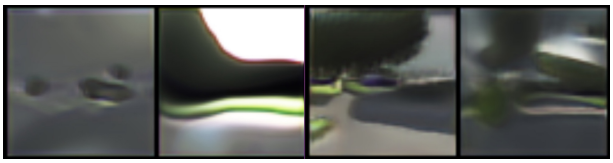

Figure 7. Fake images generated by GSN at using multiple sequences of KITTI-360 dataset for training. The two leftmost images were generated at the 109th epoch and the two rightmost images were generated at the 389th epoch.

4.4.3 KITTI-CARLA

We adopted the setting of Vizdoom training pipeline for training on KITTI-CARLA dataset. We selected 300 frames

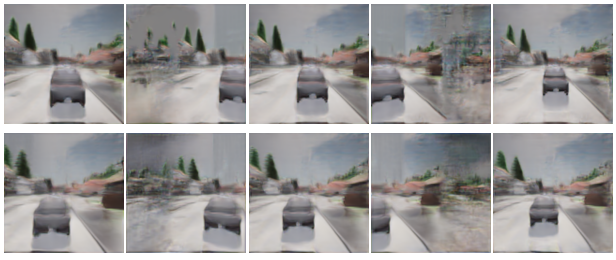

Figure 8. Navigation results generated from model trained on KITTI-CARLA dataset using *walkthrough_demo.ipynb*. First row is trained on single-view sequence. Second row is traind on dual-view sequence

Figure 9. Navigation results generated from model trained on our own CARLA generated dataset using *walkthrough_demo.ipynb*. First row is trained on single-view sequence. Second row is traind on dual-view sequence

from the Town02 driving sequence. The sensors sampled at 10Hz. Therefore, the sequence represents 30 seconds driving time in the real world. Since the dataset follows KITTI'S setting, it provides stereo camera driving sequence. Moreover, the synthetic data also allow us to get accurate depth map for the RGB images. Therefore, we trained two models for single and stereo sequence to see if stereo input can help the model to learn. Both models are trained till 109 epoch.

4.4.4 Own Dataset Generated From CARLA

As we have stated in form the KITTI-CARLA dataset, although it follows KITTI settings, precision of the offered depth map is low. KITTI-CARLA's author was not able to re-synthesis high precision depth map for us by the deadline. Therefore, we generated on our own and test out the model performance on high precision depth map, with both single-view and dual-view sequence.

4.5. Analysis

From the experiments, we can see during the train stage that both real-world and synthetic models are able to generate scenes from their training camera poses, albeit low quality. However, in the testing stage, all models trained on real-world KITTI datasets failed to synthesize reasonable visualization. After exhaustive experiments on hyperparameter tuning, dataset sequence selection and debugging, we came up with 4 possible reasons.

1. The depth map that we computed from LiDAR point cloud is sparse. Most image pixels ($\sim 90\%$) cannot be covered by the point cloud (see Fig. 3).

2. KITTI's real-world scenes are too complex for the model. The camera trajectory distribution is hugely different from the typical datasets. The trajectory for autonomous driving is mostly front forward looking, while for the latter ones, the camera usually moves freely in the scenes. This could Moreover, the real-world outdoor scene 3D shape and texture distribution is more complex than synthetic scenes.

3. There are always uncontrollable noise from the real-world dataset:

   - Fast driving speed in a sequence results in a blurry side view at the edge pixels when objects pass by the vehicle.
   - Despite carefully selected training sequence, dynamic objects always exist in the scenes
   - Vehicles driving in front of the vehicle does not only occlude the cityscape but also adversely affect model's 3D geometry learning capability.

4. The world, camera and image plane coordinate system defined in GSN paper is probably different from the standard OpenCV's definition judging from their Replica and Vizdoom datasets' K and Rt matrix. Unfortunately, GSN paper did not provide any details on its coordinate system definition.

All these uncontrollable factors lead to significantly deteriorated 3D-aware representation learning for the model. From our experience on GAN related project, it is very possible that the generator only learnt the 3D scene as a 2D scene, which make it unable to synthesize views for the camera trajectories that are not in the training set.

For the comparison on the performance of single view and dual-view sequences, we can observe that the FID is lower on stereo input sequence compared to a mono-view sequence. Also, the visualization results have shown that the dual-view trained mode synthesises higher quality images.

Moreover, the transition between camera pose is more consistent. Thirdly, the model is still able to generate reasonable visualization even when we rotate the navigation at a larger degree.

However, the performance of GSN on synthetic dataset such as KITTI-CARLA and CARLA datsets is promising. For the comparison on the performance of low and high depth precision sequences, the results have shown that the latter one synthesized images with higher quality. Moreover, the transition between camera pose is more consistent. Also, the model is able to generate reasonable visualization even when we navigate far from the training set's trajectory.

Comparing all the models' performance, the synthesized dataset with high depth precision gives the most promising result.

4.6. Challenges

Along the way, there are some challenges that we have faced in this project. This is our first time working on 3D data and NeRF architecture. Conversion between the coordinate system of dataset and the coordinate system of the GSN model has posed a challenging tasks for us to fully understand them in given time constraints. Moreover, iterating through the models requires a long time, it takes 1 day to get to a point to understand if the model is able to converge. It requires 3 days of training to know if the model is really able to fully learn to model/ overfit to the sequences.

5. Conclusion

We have seen that even if GSN is successful in generative modeling of unconstrained, complex, and realistic indoor scenes, they fail to render smooth and consistent trajectories on outdoor driving scenes where the freely moving camera is not operating within a closed space.

There are two major directions that we are interested to further explore and investigate:

1. The information of LiDAR point cloud in both real-world and synthetic datasets are not fully exploited. Currently, the model only takes the RGB image sequence and the pixels' corresponding depth as the inputs. However, LiDAR point cloud provides 360 deg spare depth in the KITTI datasets, which could be exploited.

2. Some prior knowledge in outdoor driving can be incorporated into this task for outdoor cityscape rendering. For KITTI datasets, sunlight is the light source in most of the driving sequences, a far lighting point. Given that the vehicle drives straightly within a short time interval, the change of color intensity of a far object point on surrounding architectures should change in a linear of scene. We could pre-train a model to learn such relationship to assist model rendering in larger variant of camera pose.

References

[1] Eric Chan, Marco Monteiro, Peter Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. *https://arxiv.org/abs/2012.00926*, 2020. 2

[2] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *CoRR*, abs/2107.02791, 2021. 1

[3] Jean-Emmanuel Deschaud. KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator. *arXiv e-prints*, 2021. 2, 4

[4] Terrance Devries, Miguel Ángel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. *ArXiv*, abs/2104.00670, 2021. 1, 2, 3

[5] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1, 2, 4

[6] Marco Fraccaro, Danilo Rezende, Yori Zwols, Alexander Pritzel, SM Ali Eslami, and Fabio Viola. Generative temporal models with spatial memory for partially observed environments. In *International Conference on Machine Learning*, pages 1549–1558. PMLR, 2018. 1

[7] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1, 2

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 3

[9] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 2

[10] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv.org*, 2109.13410, 2021. 2, 4

[11] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized {gan} training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2021. 3

[12] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268*, 2020. 1

[13] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215, 2021. 1

[14] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 1

[15] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1

[16] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1

[17] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2856–2865, June 2021. 1

[18] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *arXiv preprint arXiv:2007.02442*, 2020. 1

[19] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3D-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. 2

[20] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2

[21] Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games*, 2018. 2