This is a pre print version of the following article:

CarPatch: A Synthetic Benchmark for Radiance Field Evaluation on Vehicle Components / Di Nucci, D., Simoni, A., Tomei, M., Ciuffreda, L., Vezzani, R., Cucchiara, R.. - 14234 LNCS:(2023), pp. 99-110. (ICIAP 2023: 22nd International Conference, ICIAP 2023 Udine September 11–15, 2023) [10.1007/978-3-031-43153-1_9].

SPRINGER INTERNATIONAL PUBLISHING AG
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

11/10/2026 16:23

(Article begins on next page)

arXiv:2307.12718v1 [cs.CV] 24 Jul 2023

CarPatch: A Synthetic Benchmark for Radiance Field Evaluation on Vehicle Components

Davide Di Nucci [1], Alessandro Simoni [1], Matteo Tomei [2], Luca Ciuffreda[2], Roberto Vezzani [1], and Rita Cucchiara [1]

[1] Department of Engineering "Enzo Ferrari" (DIEF), University of Modena and Reggio Emilia, 41125 Modena, Italy
{davide.dinucci,alessandro.simoni,roberto.vezzani,rita.cucchiara}@unimore.it
[2] Prometeia, 40137 Bologna, Italy
{matteo.tomei,luca.ciuffreda}@prometeia.com

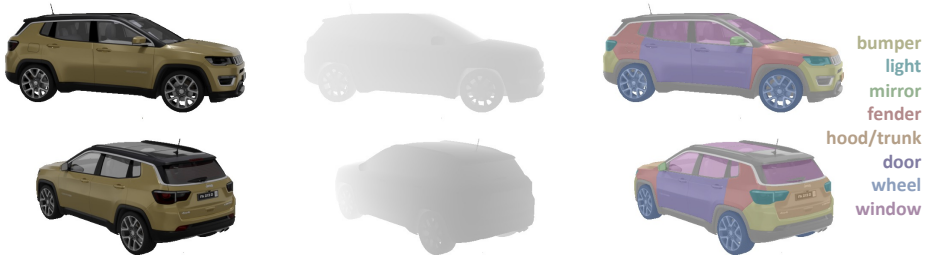

Fig. 1. A visualization of the *CarPatch* data: RGB images (left), depth images (center), and semantic segmentation of vehicle components (right).

Abstract. Neural Radiance Fields (NeRFs) have gained widespread recognition as a highly effective technique for representing 3D reconstructions of objects and scenes derived from sets of images. Despite their efficiency, NeRF models can pose challenges in certain scenarios such as vehicle inspection, where the lack of sufficient data or the presence of challenging elements (*e.g.* reflections) strongly impact the accuracy of the reconstruction. To this aim, we introduce *CarPatch*, a novel synthetic benchmark of vehicles. In addition to a set of images annotated with their intrinsic and extrinsic camera parameters, the corresponding depth maps and semantic segmentation masks have been generated for each view. Global and part-based metrics have been defined and used to evaluate, compare, and better characterize some state-of-the-art techniques. The dataset is publicly released at `https://aimagelab.ing.unimore.it/go/carpatch` and can be used as an evaluation guide and as a baseline for future work on this challenging topic.

Keywords: Synthetic vehicle dataset · 3D Reconstruction · Neural radiance fields · Volumetric rendering · RGB-D.

1 Introduction

Recent advances in Neural Radiance Fields (NeRFs) [11] strongly improved the fidelity of generated novel views by fitting a neural network to predict the volume density and the emitted radiance of each 3D point in a scene. The differentiable volume rendering step allows having a set of images, with known camera poses, as the only input for model fitting. Moreover, the limited amount of data, *i.e.* (image, camera pose) pairs, needed to train a NeRF model, facilitates its adoption and drives the increasing range of its possible applications. Among these, view synthesis recently emerged for street view reconstruction [12,19] in the context of AR/VR applications, robotics, and autonomous driving, with considerable efforts towards vehicle novel view generation. However, these attempts focus on images representing large-scale unbounded scenes, such as those from KITTI [9], and usually fail to achieve high-quality 3D vehicle reconstruction.

In this paper, we introduce an additional use case for neural radiance fields, *i.e. vehicle inspection*, where the goal is to represent an individual high-quality instance of a given car. The availability of a high-fidelity 3D vehicle representation could be beneficial whenever the car body has to be analyzed in detail. For instance, insurance companies or body shops could rely on NeRF-generated views to assess possible external damages after a road accident and estimate their repair cost. Moreover, rental companies could compare two NeRF models trained before and after each rental, respectively, to assign responsibility for any new damages. This would avoid expert on-site inspection or a rough evaluation based on a limited number of captures.

For this purpose, we provide an experimental overview of the state-of-the-art NeRF methods, suitable for vehicle reconstruction. To make the experimental setting reproducible and to provide a basis for new experimentation, we propose *CarPatch*, a new benchmark to assess neural radiance field methods on the *vehicle inspection* task. Specifically, we generate a novel dataset consisting of 8 different synthetic scenes, corresponding to as many high-quality 3D car meshes with realistic details and challenging light conditions. As depicted in Fig. 1, we provide not only RGB images with camera poses, but also binary masks of different car components to validate the reconstruction quality of specific vehicle parts (*e.g.* wheels or windows). Moreover, for each camera position, we generate the ground truth depth map with the double goal of examining the ability of NeRF architectures to correctly predict volume density and, at the same time, enable future works based on RGB-D inputs. We evaluate the novel view generation and depth estimation performance of several methods under diverse settings (both global and component-level). Finally, since the process of image collection for fitting neural radiance fields could be time consuming in real scenarios, we provide the same scenes by varying the number of training images, in order to determine the robustness to the amount of training data.

After an overview of the main related works in Sec. 2, we thoroughly describe the process of 3D mesh gathering, scene setup, and dataset generation in Sec. 3. The evaluation of existing NeRF architectures on *CarPatch* is presented in Sec. 4.

Table 1. Comparison between existing datasets used as benchmarks for neural radiance field evaluation and *CarPatch*. We provide the same scene by varying the amount of training data (40, 60, 80, and 100 images), allowing users to test the robustness of their architectures. We also release depth and segmentation data for all the images.

Dataset	Scenes	Images/scene	Depth	Segmentation
Blender [11]	8	300	✓	✗
Shiny Blender [17]	6	300	✓	✗
BlendedMVG [20]	508	200-4000	✗	✗
$CarPatch_{40}$	8	240	✓	✓
$CarPatch_{60}$	8	260	✓	✓
$CarPatch_{80}$	8	280	✓	✓
$CarPatch_{100}$	8	300	✓	✓

2 Related work

We provide a brief overview of the latest updates in neural radiance field, including its significant extensions and applications that have influenced our work. NeRF limitations have been tackled by different works, trying to reduce its complexity, increase the reconstruction quality, and develop more challenging benchmarks.

Neural scene reconstruction. The handling of aliasing artifacts is a well-known issue in rendering algorithms. Mip-NeRF [1,2] and Zip-NeRF [3] have tackled the aliasing issue by reasoning on volumetric frustums along a cone. These approaches have inspired works such as Able-NeRF [16], which replaces the MLP of the original implementation with a transformer-based architecture. In addition to other sources of aliasing, reflections can pose a challenge for NeRF. Several works have attempted to address the issue of aliasing in reflections by taking into account the reflectance of the scene [4,5,17]. Moreover, computation is a widely recognized concern. Various works in the literature have demonstrated that it is possible to achieve high-fidelity reconstructions while reducing the overall training time. Two notable works in this direction include NSVF [10], which uses a voxel-based representation for more efficient rendering of large scenes, and Instant-NGP [13], which proposes a multi-resolution hash table combined with a light MLP to achieve faster training times. Other approaches such as DVGO [15] and Plenoxels [8] optimize voxel grids of features to enable fast radiance field reconstruction. TensoRF [7] combines the traditional CP decomposition [7] with a new vector-matrix decomposition method [6] leading to faster training and higher-quality reconstruction.

In this work, in order to satisfy real-time performances for vehicle inspection, we select a set of architectures that strike a balance between training time and the quality of the reconstruction.

Scene representation benchmarks. One of the most widely used benchmarks for evaluating NeRF is the Nerf Synthetic Blender dataset [11]. This dataset

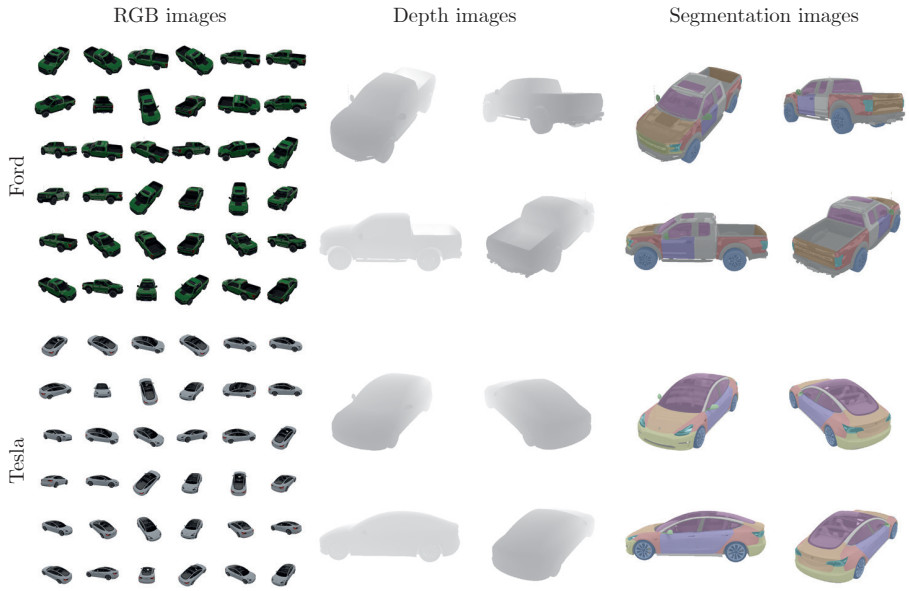

Fig. 2. Sample RGB images (left), depth data (center), and segmentation masks (right) from *CarPatch*, for different car models.

consists of 8 different scenes generated using Blender[1], each with 100 training images and 200 test images. Other synthetic datasets include the Shiny Blender dataset [17], which mostly contains singular objects with simple geometries, and Blend DMVS [20], which provides various scenes to test NeRF implementations at different scales. These works do not provide ground truth information about the semantic meaning of the images. This limitation makes it difficult to study the ability of NeRF to reconstruct certain surfaces compared to others. In our *CarPatch* dataset, we provide ground truth segmentation of vehicle components in the scene, allowing for the evaluation of architectures on specific parts. Table 1 presents a comparison between the most common datasets used as benchmarks and our proposed dataset.

3 The *CarPatch* dataset

In this section, we detail the source data and the procedure exploited for generating our *CarPatch* dataset. In particular, we describe how we gathered 3D models, set up the blender scenes, and designed the image capture process.

[1] `http://www.blender.org`

Table 2. Summary of the source 3D models from which our dataset has been generated, including their key features.

Model name	Acronym	#Triangles	#Vertices	#Textures	#Materials
Tesla Model	TESLA	684.3k	364.4k	22	58
Smart	SMART	42.8k	26.4k	0	31
Ford Raptor	FORD	257.1k	156.5k	12	50
BMW M3 E46	BMW	846.9k	442.4k	7	39
Mercedes GLK	MBZ_1	1.3M	741.4k	0	15
Mercedes CLS	MBZ_2	1.0M	667k	0	18
Volvo S90	VOLVO	3.3M	1.7M	56	44
Jeep Compass	JEEP	334.7k	189.6k	7	39

3.1 Synthetic 3D models and scene setup

All the 3D models included in *CarPatch* scenes have been downloaded from Sketchfab[2], a large collection of free 3D objects for research use. Table 2 provides a detailed list of all the starting models used. Each of them has been edited in Blender to enhance its realism; specifically, we improved the materials, colors, and lighting in each scene to create a more challenging environment.

The scenes have been set up accordingly to the Google Blender dataset [11]. The lighting conditions and rendering settings were customized to create a more realistic environment. The vehicle was placed at the center of the scene at position (0,0,0), with nine lights distributed around the car and varying emission strengths to create shadows and enhance reflections on the materials' surfaces. To improve realism, we resized objects to match their real-world size. The camera and lights were placed in order to provide an accurate representation of the environment, making the scenes similar to real-world scenarios.

3.2 Dataset building

The dataset was built using the Python interface provided in Blender, allowing us to control objects in the environment. For each rendered image, we captured not only the RGB color values but also the corresponding depth map, as well as the pixel-wise semantic segmentation masks for eight vehicle components: bumpers, lights, mirrors, hoods/trunks, fenders, doors, wheels, and windows. Examples of these segmentation masks can be seen in Fig. 1. Please note that all the pixels belonging to a component (*e.g.* doors) are grouped into the same class, regardless of the specific component location (*e.g.* front/rear/right/left door). The bpycv[3] utility has been used for collecting additional metadata, enabling us to evaluate NeRF models on the RGB reconstruction and depth estimation of the overall vehicle as well as each of its subparts.

[2] `https://sketchfab.com`
[3] `https://github.com/DIYer22/bpycv`

For the rendering of training images, the camera randomly moved on the hemisphere centered in (0,0,0) and above the ground. The camera rotation angle was sampled from a uniform distribution before each new capture. For building the test set, the position of the camera was kept at a fixed distance from the ground and rotated around the Z-axis with a fixed angle equal to $\frac{2\pi}{\#test_views}$ radians before each new capture.

In order to guarantee the fairness of the current and future comparisons, we explicitly provide four different versions of each scene, by varying the number of training images (40, 60, 80, and 100 images, respectively). Different versions of the same scene have no overlap in training camera poses, while the test set is always the same and contains 200 images for each scene.
We release the code for dataset creation and metrics evaluation at `https://github.com/davidedinuc/carpatch`.

4 Benchmark

This section presents the selection and testing of various recent NeRF-based methods [13,7,15] on the presented *CarPatch* dataset, with a detailed description of the experimental setting for each baseline. Additionally, we assess the quality of the reconstructed vehicles in terms of their appearance and 3D surface reconstruction, utilizing depth maps generated during volume rendering.

4.1 Compared methods

To overcome challenges related to illumination and reflective surfaces during the process of reconstructing vehicles, it is crucial to choose an appropriate neural rendering approach. We tested selected approaches on *CarPatch* without modifying the implementation details available in the original repositories, whenever possible. However, some parameters had to be adjusted in order to fit our models (which are larger compared to reference dataset meshes) to the scene. All tests were performed on a GeForce GTX 1080 Ti. After considering various NeRF systems, we have selected the following baselines:

- **Instant-NGP [13].** Since the original implementation of Instant-NGP is in CUDA, we decided to use an available PyTorch implementation[4] of this approach in order to have a fair comparison with the other approaches. In our experiments, a batch size of 8192 was maintained, with a scene scale of 0.5 and a total of 30,000 iteration steps.
- **TensoRF [7].** In our setting, a batch of 4096 rays was used. Additionally, we increased the overall scale of the scene from 1 to 3.5. These adjustments were made after experimentation and careful consideration of the resulting reconstructions. Training lasts 30,000 iterations.

[4] `https://github.com/kwea123/ngp_pl`

Table 3. Quantitative results on the *CarPatch* test set for each vehicle model.

Method	Metric	Bmw	Tesla	Smart	Mbz$_1$	Mbz$_2$	Ford	Jeep	Volvo	*Avg*
iNGP [13]		39.48	39.46	39.57	36.87	39.15	33.67	35.00	35.93	37.39
DVGO [15]	PSNR↑	39.91	39.89	40.34	37.45	39.37	33.82	35.32	36.28	37.80
TensoRF [7]		40.68	39.92	40.38	38.07	40.84	34.33	34.87	36.77	**38.23**
iNGP [13]		0.985	0.987	0.988	0.985	0.987	0.959	0.978	0.979	0.981
DVGO [15]	SSIM↑	0.987	0.988	0.990	0.987	0.988	0.964	0.980	0.981	0.983
TensoRF [7]		0.989	0.987	0.99	0.989	0.991	0.966	0.975	0.982	**0.984**
iNGP [13]		0.029	0.029	0.02	0.028	0.024	0.062	0.036	0.032	0.032
DVGO [15]	LPIPS↓	0.022	0.022	0.014	0.019	0.020	0.051	0.029	0.022	**0.025**
TensoRF [7]		0.023	0.026	0.017	0.02	0.017	0.051	0.039	0.027	0.028
iNGP [13]		0.640	0.369	0.377	0.496	0.500	0.406	0.558	0.674	0.503
DVGO [15]	D-RMSE↓	0.561	0.353	0.305	0.437	0.454	0.339	0.469	0.561	**0.435**
TensoRF [7]		0.590	0.357	0.335	0.467	0.482	0.375	0.536	0.626	0.471
iNGP [13]		4.24	3.38	3.41	4.26	4.13	5.15	4.60	4.67	4.23
DVGO [15]	SN-RMSE↓	4.27	3.48	3.20	4.19	4.24	5.04	4.67	4.71	4.22
TensoRF [7]		3.96	3.24	3.10	4.00	3.91	4.91	4.41	4.48	**4.00**

– **DVGO [15].** In this work, the training process consists of two phases: a coarse training phase of 5,000 iterations, followed by a fine training phase of 20,000 iterations that aims to improve the model's ability to learn intricate details of the scene. In our experiments, we applied a batch size of 8192 while maintaining the default scene size.

4.2 Metrics

The effectiveness of the chosen methods has been assessed thanks to the typical perceptual metrics used in NeRF-based reconstruction tasks, namely PSNR, SSIM [18], and LPIPS [21].

However, the appearance-based metrics are strongly related to the emitted radiance besides the learned volume density. We suggest two supplementary depth-based metrics for the sole purpose of assessing the volume density. Since it is not feasible to obtain ground truth 3D models of the vehicles in real-world scenarios, we utilize the depth map as our knowledge of the 3D surface of the objects. Specifically, we define a depth map as a matrix

$$D = \{d_{ij}\}, d_{ij} \in [0, R] \tag{1}$$

in which each value d_{ij} ranges from 0 to the maximum depth value R. Furthermore, we estimate the surface normals from the depth maps [14]. Initially, we establish the orientation of a surface normal as:

$$\mathbf{d} = \langle d_x, d_y, d_z \rangle = \left(-\frac{\partial d_{ij}}{\partial i}, -\frac{\partial d_{ij}}{\partial j}, 1 \right) \approx \left(d_{(i+1)j} - d_{ij}, d_{i(j+1)} - d_{ij}, 1 \right) \tag{2}$$

Table 4. Quantitative results on the *CarPatch* test set for each vehicle component averaged over the vehicle models.

Method	Component	PSNR↑	SSIM↑	LPIPS↓	D-RMSE↓	SN-RMSE↓
iNGP [13]	*bumper*	33.05	0.986	0.019	0.281	0.79
DVGO [15]		34.41	0.989	0.011	**0.236**	0.72
TensoRF [7]		**35.49**	**0.991**	**0.010**	0.311	**0.68**
iNGP [13]	*light*	28.71	0.993	0.009	0.421	0.48
DVGO [15]		29.10	0.995	**0.006**	**0.384**	0.43
TensoRF [7]		**29.68**	**0.996**	**0.006**	0.438	**0.38**
iNGP [13]	*mirror*	29.60	0.994	0.011	0.427	0.43
DVGO [15]		31.16	**0.996**	**0.007**	**0.345**	**0.38**
TensoRF [7]		**31.68**	**0.996**	0.008	0.372	0.39
iNGP [13]	*hood/trunk*	32.28	0.977	0.052	0.260	1.33
DVGO [15]		32.68	0.981	**0.038**	**0.259**	1.35
TensoRF [7]		**33.75**	**0.983**	0.040	0.302	**1.24**
iNGP [13]	*fender*	32.44	0.990	0.021	0.253	0.87
DVGO [15]		33.55	**0.993**	**0.013**	**0.223**	0.85
TensoRF [7]		**34.36**	**0.993**	0.015	0.267	**0.77**
iNGP [13]	*door*	34.19	0.969	0.079	0.182	0.67
DVGO [15]		35.48	0.977	**0.042**	**0.173**	0.74
TensoRF [7]		**36.25**	**0.979**	0.051	0.191	**0.62**
iNGP [13]	*wheel*	33.12	0.995	0.008	0.391	0.87
DVGO [15]		33.65	0.995	0.006	**0.267**	**0.79**
TensoRF [7]		**34.55**	**0.996**	**0.005**	0.334	**0.79**
iNGP [13]	*window*	26.44	0.897	0.166	0.879	2.52
DVGO [15]		26.54	**0.899**	**0.147**	**0.779**	2.57
TensoRF [7]		**26.74**	0.896	0.160	0.834	**2.38**

where the first two elements represent the depth gradients in the i and j directions, respectively. Afterward, we normalize the normal vector to obtain a unit-length vector $\mathbf{n}(d_{ij}) = \frac{\mathbf{d}}{\|\mathbf{d}\|}$.

We assess the 3D reconstruction's quality through the following metrics:

– **Depth Root Mean Squared Error (D-RMSE).** This metric measures the average difference in meters between the ground truth and predicted depth maps.

$$\text{D-RMSE} = \sqrt{\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}(\hat{d}_{ij} - d_{ij})^2}{M \cdot N}} \tag{3}$$

– **Surface Normal Root Mean Squared Error (SN-RMSE).** This metric measures the average angular error in degrees between the angle direction of

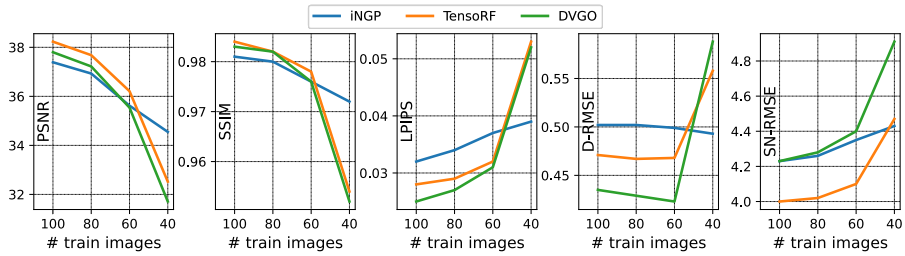

Fig. 3. Performance by varying the number of training images, in terms of PSNR, SSIM, LPIPS, D-RMSE, and SN-RMSE. Despite its lower overall performance, Instant-NGP [13] exhibits low variance with respect to the amount of training data.

the ground truth and predicted surface normals.

$$\text{SN-RMSE} = \sqrt{\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}(\arccos(\mathbf{n}(\hat{d}_{ij})) - \arccos(\mathbf{n}(d_{ij})))^2}{M \cdot N}} \tag{4}$$

D-RMSE and SN-RMSE are computed only for those pixels with a positive depth value in both GT and predicted depth maps. This avoids computing depth estimation errors on background pixels (which have a fixed depth value of 0).

4.3 Results

The following section presents both quantitative and qualitative results obtained from the selected NeRF baselines. We will discuss their performance on the *CarPatch* dataset, by analyzing the impact of viewing camera angle and the number of training images.

According to Table 3, all the selected NeRF approaches obtain satisfying results. Although the baselines demonstrate similar performances in terms of appearance scores (PSNR, SSIM, and LPIPS), our evaluation using depth-based metrics (D-RMSE and SN-RMSE) reveals significant differences in the 3D reconstruction of the vehicles. DVGO outperforms its competitors by achieving better depth estimation, resulting in a +13.5% improvement compared to iNGP and a +7.6% improvement compared to TensoRF. In contrast, TensoRF predicts a more accurate 3D surface with the lowest angular error on the surface normals.

Since our use case is related to vehicle inspection, in Table 4 we report results computed on each car component. For this purpose, we mask both GT and predictions using a specific component mask before computing the metrics. However, this would lead to an unbalanced ratio between background and foreground pixels, due to the limited components' area, and finally to a biased metric value. By computing D-RMSE and SN-RMSE only on foreground pixels (see Sec. 4.2), depth-based metrics are not affected by this issue. For PSNR, SSIM, and LPIPS, instead, we compute component-level metrics over the image crop delimited by the bounding boxes around each mask. As expected, it is

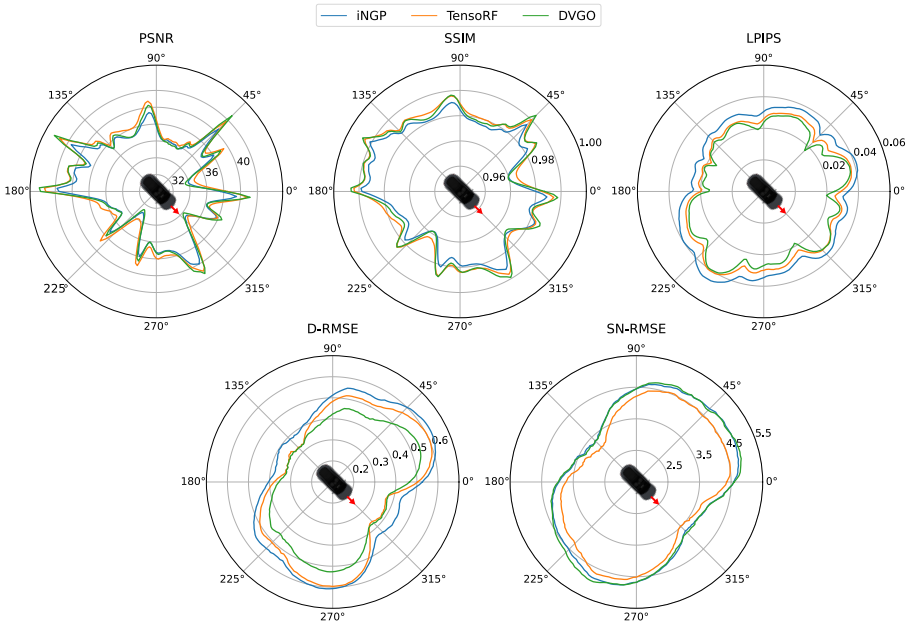

Fig. 4. Performance by camera viewing angle, in terms of PSNR, SSIM, LPIPS, D-RMSE, and SN-RMSE. Depending on the training camera distribution, all the methods struggle wherever the viewpoints are more sparse (*e.g.* between 225° and 270°). The red arrow represents where the front of the vehicle is facing.

worth noting that NeRF struggles to reconstruct transparent objects (*e.g.* mirrors, lights, and windows) obtaining the highest errors in terms of depth and normal estimation. However, over the single components, TensoRF outperforms the competitors in most of the metrics and in particular on the surface normal estimation. The errors in the reconstruction of specific components' surfaces can also be appreciated in the qualitative results of Fig. 5.

Moreover, we analyze the performances of each method in terms of the number of training images. We trained the baselines on every version of the *CarPatch* dataset and report the results in Fig. 3. It is worth noting that reducing the number of training images has a significant impact on all the metrics independently of the method. However, Instant-NGP demonstrates to be more robust to the number of camera viewpoints having a smoother drop, especially in terms of LPIPS, D-RMSE, and SN-RMSE.

Finally, we discuss how the training camera viewpoints' distribution around the vehicle may affect the performance of each method from certain camera angles. In particular, as depicted in Fig. 4, it is evident how between 180° and 270° and between 0° and 45° there are considerable variations in the metrics. Indeed, in these areas the datasets contain more sparsity in terms of camera viewpoints and, as expected, all the methods are affected.

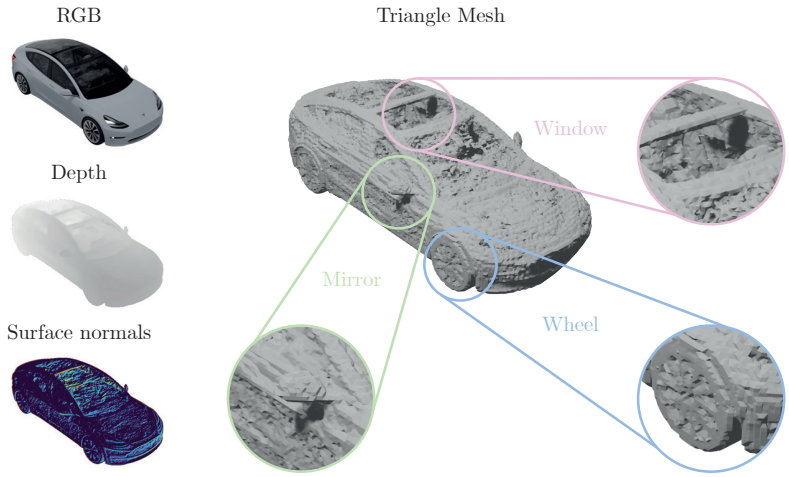

Fig. 5. Sample of 3D reconstruction of the Tesla: (left) the reconstructed RGB, depth, and surface normals, (right) the reconstructed surfaces on the triangle mesh.

5 Conclusion

In this article, we have proposed a new benchmark for the evaluation and comparison of NeRF-based techniques. Focusing on one of the many concrete applications of this recent technology, i.e. *vehicle inspection*, a new synthetic dataset including renderings of 8 vehicles was first created. In addition to the set of RGB views annotated with the camera pose, the dataset is enriched by semantic segmentation masks as well as depth maps to further analyze the results and compare the methods.

The presence of reflective surfaces and transparent parts makes the task of vehicle reconstruction still challenging. Proposed additional metrics, as well as new graphical ways of displaying the results, are proposed to make these limitations more evident. We are confident that *CarPatch* can be of great help as a basis for research on NeRF models in general and, more specifically, in their application to the field of vehicle reconstruction.

Acknowledgements. The work is partially supported by the Department of Engineering Enzo Ferrari, under the project FAR-Dip-DIEF 2022 "AI platform with digital twins of interacting robots and people".

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proc. of the IEEE/CVF ICCV (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proc. of the IEEE/CVF Conf. on CVPR (2022)

3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. arXiv:2304.06706 (2023)
4. Bi, S., Xu, Z., Srinivasan, P., Mildenhall, B., Sunkavalli, K., Hašan, M., Hold-Geoffroy, Y., Kriegman, D., Ramamoorthi, R.: Neural reflectance fields for appearance acquisition. arXiv:2008.03824 (2020)
5. Boss, M., Jampani, V., Braun, R., Liu, C., Barron, J., Lensch, H.: Neural-pil: Neural pre-integrated lighting for reflectance decomposition. Advances in Neural Information Processing Systems **34** (2021)
6. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. Psychometrika **35**(3) (1970)
7. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Proc. of the ECCV. Springer (2022)
8. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proc. of the IEEE/CVF Conf. on CVPR (2022)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proc. of the IEEE/CVF Conf. on CVPR. IEEE (2012)
10. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33** (2020)
11. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1) (2021)
12. Müller, N., Simonelli, A., Porzi, L., Bulò, S.R., Nießner, M., Kontschieder, P.: Autorf: Learning 3d object radiance fields from single view observations. In: Proc. of the IEEE/CVF Conf. on CVPR (2022)
13. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. on Graphics (ToG) **41**(4) (2022)
14. Pini, S., Borghi, G., Vezzani, R., Maltoni, D., Cucchiara, R.: A systematic comparison of depth map representations for face recognition. Sensors **21**(3) (2021). https://doi.org/10.3390/s21030944
15. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proc. of the IEEE/CVF Conf. on CVPR (2022)
16. Tang, Z.J., Cham, T.J., Zhao, H.: Able-nerf: Attention-based rendering with learnable embeddings for neural radiance field. arXiv:2303.13817 (2023)
17. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: Proc. of IEEE/CVF Conf. on CVPR (2022)
18. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4) (2004)
19. Xie, Z., Zhang, J., Li, W., Zhang, F., Zhang, L.: S-nerf: Neural radiance fields for street views (2023)
20. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: Proc. of the IEEE/CVF Conf. on CVPR (2020)
21. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. of the IEEE/CVF Conf. on CVPR (2018)