Learning Neural Radiance Fields from Multi-View Geometry

Marco Orsingher[1,2], Paolo Zani[2], Paolo Medici[2], and Massimo Bertozzi[1]

[1] Università degli Studi di Parma
[2] VisLab Srl (an Ambarella Inc company)
`marco.orsingher@unipr.it`

Abstract. We present a framework, called **MVG-NeRF**, that combines classical **M**ulti-**V**iew **G**eometry algorithms and **N**eural **R**adiance **F**ields (NeRF) for image-based 3D reconstruction. NeRF has revolutionized the field of implicit 3D representations, mainly due to a differentiable volumetric rendering formulation that enables high-quality and geometry-aware novel view synthesis. However, the underlying geometry of the scene is not explicitly constrained during training, thus leading to noisy and incorrect results when extracting a mesh with marching cubes. To this end, we propose to leverage pixelwise depths and normals from a classical 3D reconstruction pipeline as geometric priors to guide NeRF optimization. Such priors are used as *pseudo*-ground truth during training in order to improve the quality of the estimated underlying surface. Moreover, each pixel is weighted by a confidence value based on the forward-backward reprojection error for additional robustness. Experimental results on real-world data demonstrate the effectiveness of this approach in obtaining clean 3D meshes from images, while maintaining competitive performances in novel view synthesis.

Keywords: Neural Radiance Fields, Multi-View Geometry, Image-Based 3D Reconstruction, Geometric Priors, Implicit 3D Representations

1 Introduction

3D reconstruction from a set of images is a longstanding problem in computer vision, with applications in robotics [16,34], virtual reality [2,43], autonomous driving [24,25], and many other fields. There are mainly two approaches in literature for inferring 3D geometry and appearance from multi-view images: (i) the classical geometry-based pipeline with Structure From Motion (SFM) [29], Multi-View Stereo (MVS) [30] and Surface Reconstruction (SR) [10,11]; and (ii) modern deep learning methods [18,40,20], which can either replace single components in the classical pipeline or be trained end-to-end from images.

A classical image-based 3D reconstruction pipeline takes a set of images as input and produces an *explicit* representation of the scene, typically as a point cloud or as a mesh. Firstly, SFM [29] computes camera poses and calibration parameters for each input image, as well as a set of sparse keypoints. Then,

(a) Input point cloud from MVS.

(b) Mesh computed with SR.

(c) Mesh from NeRF.
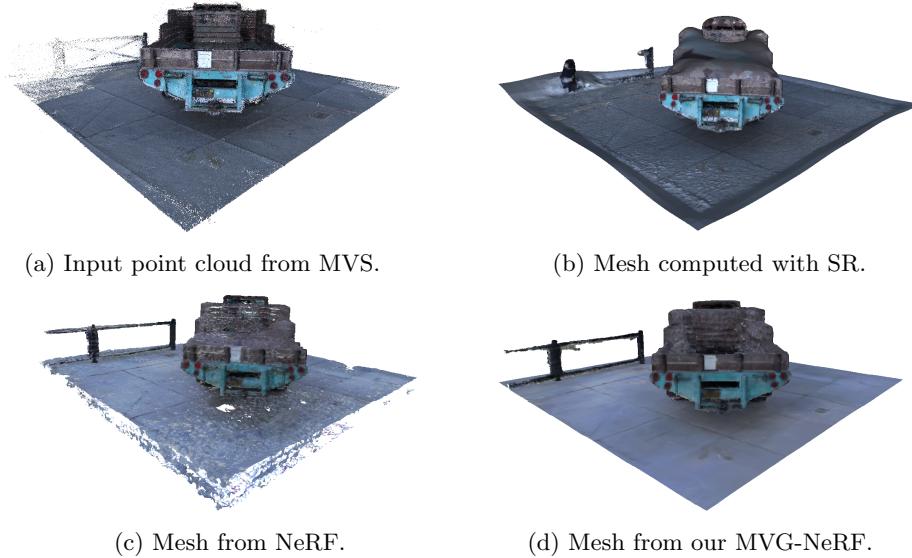
(d) Mesh from our MVG-NeRF.

Fig. 1: In complex structures such as the back of the truck, classical SR fails by hallucinating incorrect geometry, while the mesh from NeRF is very noisy. Our approach combines them to get a clean 3D model.

MVS [30] estimates pixelwise depth and normal maps for each calibrated image. Finally, a point cloud can be obtained by backprojecting in 3D multi-view consistent hypotheses and a SR algorithm [10,11] is used to compute a mesh. Despite showing promising results in the last decades, this multi-stage approach suffers from errors that propagate between each stage, without any recovery mechanism. Moreover, MVS algorithms rely on photometric matching costs that are unreliable in textureless areas and non-Lambertian surfaces. Finally, while the *discrete* representation of the scene as a point cloud is generally accurate, it is also locally sparse[1], which makes it difficult for SR algorithms to recover the correct *continuous* shape (see Fig. 1b).

In recent years, there has been a growing interest in applying modern deep learning to 3D reconstruction. While some works proposed to augment the multi-stage pipeline with learning-based components [14,40,27], another research direction is to design *implicit* 3D representations that encode shape and appearance in the weights of a neural network, such as a small MLP [42,22,18]. The concurrent development of differentiable surface [42,22] and volume [18] rendering procedures allows end-to-end training from posed images, without 3D ground truth supervision. An *explicit* representation of the scene as a mesh can then be extracted from the geometry field by querying the network with a dense grid of points and running the marching cubes algorithm [15]. On one hand, the

---

[1] This means that some areas of the scene are dense, while others are either empty or very sparse (usually where MVS fails).

main advantage of these approaches is that the 3D representation is *continuous* by construction, as the network can be queried with any point in space. On the other hand, the underlying geometry is not explicitly constrained during training, which may lead to noisy and incorrect results, especially in real-world scenarios. A typical example is shown in Fig. 1c.

We present a framework that combines neural implicit representations with classical multi-view geometry to produce clean 3D meshes from images, as shown in Fig. 1d. The key idea is to leverage pixelwise depths and normals from MVS as *pseudo*-ground truth for constraining the underlying geometry of a Neural Radiance Field (NeRF) during training. Additionally, a confidence value is estimated for each pixel to softly activate this supervision only for rays with low reprojection error. Differently from recent works that include depth priors in NeRF [6,36,28], we show that the joint optimization of normal vectors is crucial to improve the quality of the underlying surface. This is due to the fact that RGB and normals are complementary, meaning that normals can be estimated reliably in textureless regions where photometric consistency fails, while color supervision is effective in textured structures with ambiguous normals.

2   Related Work

The proposed approach is related to both classical 3D reconstruction (Sec. 2.1) and neural implicit 3D representations (Sec. 2.2), as well as the combination of these methods (Sec. 2.3). In this section, we briefly review the relevant literature in such fields.

2.1   Classical 3D Reconstruction

The problem of recovering the 3D structure of a scene from a set of images has been studied for decades. SFM algorithms, which estimate extrinsic and intrinsic parameters for each image, can be broadly divided into global [4], hierarchical [37] and incremental [29] approaches. Given poses and calibration matrices, MVS then computes pixelwise depths and normals for each image. State-of-the-art MVS methods are mostly based on the PatchMatch idea to sample and propagate good hypotheses [1,8,30,38,39], starting from a random initial solution. Finally, a 3D point cloud of the scene is given by the backprojection of multi-view consistent estimates and several SR methods have been developed to compute a continuous mesh [10,11]. However, SR often fails in intricate structures or relatively sparse regions, as shown in Fig. 1. While recent efforts focused on enhancing the classical pipeline with deep learning [14,40,27], purely geometry-based methods are still competitive on complex outdoor benchmarks [31,12], do not require any 3D supervision for training and do not suffer from generalization issues. For this reason, we propose to exploit this readily available 3D information as *pseudo*-ground truth for training a neural implicit 3D representation.
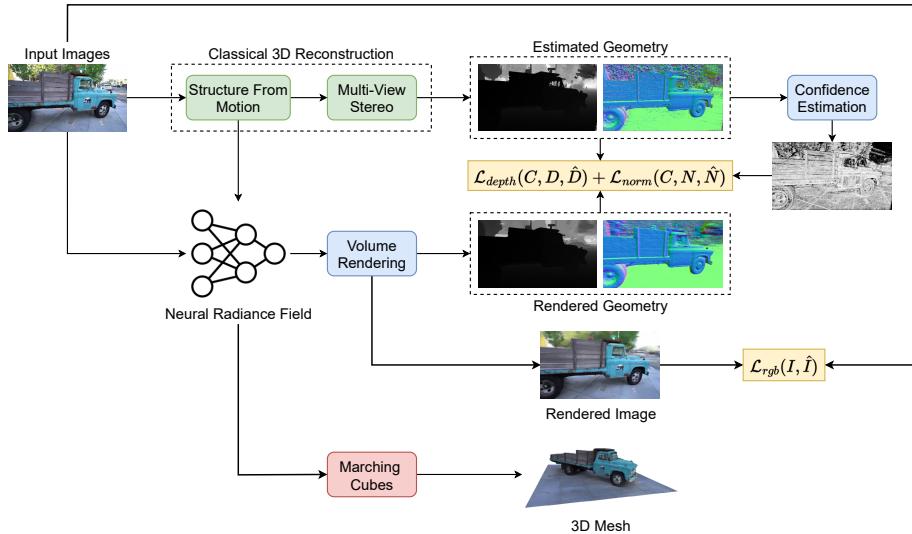
Fig. 2: An overview of the approach. NeRF training (Sec. 3.1) is guided by a confidence-aware (Sec. 3.3) geometric *pseudo*-ground truth from classical 3D reconstruction. Pixelwise depths and normals are computed from NeRF with volume rendering (Sec. 3.2) and optimized with novel geometry losses (Sec. 3.4).

2.2   Neural Implicit 3D Representations

A recent line of research proposed to encode 3D geometry and appearance in small coordinate-based neural networks that can be queried with any point in space to produce the corresponding density [18], occupancy [17] or signed distance from the surface [26]. While early approaches required 3D ground truth as supervision, differentiable surface [42,22] and volume [18] rendering techniques have been introduced to enable self-supervised training from posed images alone. In particular, NeRF [18] has shown impressive novel view synthesis results and opened several research directions to improve its training efficiency [19,21], memory consumption [32], rendering speed [19,44] and generalization [9,3]. Moreover, the combination of surface and volume rendering [23,33,41] enables significant improvements in the scene geometry, at the cost of lower performances in novel view synthesis. Our idea is to keep the original formulation of NeRF for high-quality view synthesis and to use strong geometric priors to guide the optimization towards the correct underlying surface.

2.3   Geometric Priors for NeRF

Several works have explored the possibility of introducing geometric priors while optimizing NeRF. DS-NeRF [6] pioneered the idea of exploiting sparse keypoints from SFM and proposed a probabilistic depth supervision term in the loss function. Following this insight, NerfingMVS [36] and DDP-NeRF [28] trained a

monocular depth estimation and completion network, respectively, in order to have dense depth information for NeRF. These methods have shown to reduce both the number of views required for convergence and the overall training time. Inspired by these approaches, we instead propose to leverage pixelwise depths and normals from a classical 3D reconstruction pipeline as *dense* and geometrically *accurate pseudo*-ground truth. MonoSDF [45] is a concurrent work that shows the importance of including normals as a geometric prior in the context of neural signed distance fields, but it relies on pretrained models for generating such priors.

3   Method

The proposed framework generates a 3D mesh of the scene from a set of raw images $\{I_i\}_{i=1}^N$, as shown in Fig. 2. Camera poses and calibration parameters $\{\mathbf{R}_i, \mathbf{t}_i, \mathbf{K}_i\}_{i=1}^N$ are estimated by a SFM algorithm and used as additional input for both NeRF and MVS. Then, pixelwise depths and normals $\{D_i, N_i\}_{i=1}^N$ are computed for each image with MVS and used to supervise NeRF training with confidence weights $\{C_i\}_{i=1}^N$. At the end of the training process, a 3D mesh is extracted from the density field with marching cubes [15].

After a short review of NeRF representation and color rendering procedure (Sec. 3.1), we detail how to render geometric quantities (Sec. 3.2) and how to estimate the confidence of the *pseudo*-ground truth (Sec. 3.3). Finally, loss functions are introduced and motivated in Sec. 3.4.

3.1   Review of NeRF

A Neural Radiance Field (NeRF) is an *implicit* and *continuous* representation of a 3D scene. Such field is implemented with a simple neural network F_θ that maps any point $\mathbf{x} \in \mathbb{R}^3$ in space and a viewing direction $\mathbf{d} \in \mathbb{S}^2$ to its corresponding density $\sigma(\mathbf{x}) \in \mathbb{R}^+$ and view-dependent color $\mathbf{c}(\mathbf{x}, \mathbf{d}) \in \mathbb{R}^3$:

$$F_\theta : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^3 \times \mathbb{R}^+ \tag{1}$$

This network is trained by minimizing an image reconstruction loss between the ground truth colors in the input images and the rendered colors from the radiance field. For each camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with origin $\mathbf{o}$ and oriented as $\mathbf{d}$, the corresponding color $I(\mathbf{r})$ is computed by the following integral, bounded in the interval $[t_{near}, t_{far}]$:

$$\hat{I}(\mathbf{r}) = \int_{t_{near}}^{t_{far}} w(t) \cdot \mathbf{c}(t) dt \tag{2}$$

with volumetric integration weights:

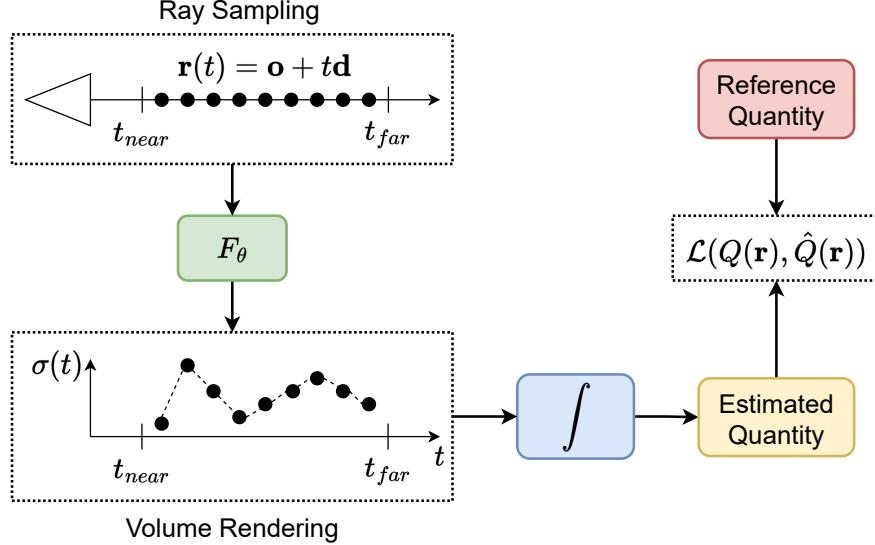$$w(t) = \exp\left(-\int_{t_{near}}^{t} \sigma(s)ds\right) \cdot \sigma(t) \tag{3}$$

Fig. 3: A visual explanation of differentiable volumetric rendering for arbitrary quantities $Q(\mathbf{r})$. A set of points is sampled along a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, with $t \in [t_{near}, t_{far}]$. The network F_θ is queried at each point to produce the corresponding value, which is then integrated with weights based on the density $\sigma(t)$ to produce the final result $\hat{Q}(\mathbf{r})$. The goal is to minimize $\mathcal{L}(Q(\mathbf{r}), \hat{Q}(\mathbf{r}))$.

In pratice, these integrals are approximated by numerical quadrature with a discrete set of samples along each ray. More details can be found in [18] and a visual explanation of the rendering procedure is provided in Fig. 3.

NeRF allows to (i) render novel views by sampling points along a ray through each pixel and integrating those samples with volume rendering; (ii) extract a 3D mesh of the scene by running the marching cubes algorithm [15] on the density field. However, the main issue is that NeRF is explicitly optimized only to produce RGB renderings and the underlying geometry is never constrained during training. This leads to the well-known phenomenon of shape-radiance ambiguity [46], which means that NeRF can hallucinate incorrect geometries as long as they explain the input views in terms of the image reconstruction loss.

3.2   Rendering Geometry from NeRF

The color rendering equation can be modified to render volumetrically any other quantity, including the underlying geometry that is learned during training. The expected depth along a ray can be computed as follows:

$$\hat{D}(\mathbf{r}) = \int_{t_{near}}^{t_{far}} w(t) \cdot t\, dt \tag{4}$$

Moreover, the unit normal vector $\mathbf{n}(\mathbf{x})$ at point $\mathbf{x} \in \mathbb{R}^3$ is given by the gradient of the density field at $\mathbf{x}$:

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} F_\theta(\mathbf{x})}{||\nabla_{\mathbf{x}} F_\theta(\mathbf{x})||} \tag{5}$$

Such gradient could be directly computed from the automatic differentiation engine of deep learning frameworks. However, in our experiments, a simple central difference approximation has proven to be more accurate and efficient. Let Δh be a small step size:

$$\nabla_{\mathbf{x}} F_\theta(\mathbf{x}) \approx \frac{F_\theta(\mathbf{x} + \Delta h) - F_\theta(\mathbf{x} - \Delta h)}{\Delta h} \tag{6}$$

Given the unit normal vectors for each sample along a ray, the final normal can be rendered as shown in Fig. 3:

$$\hat{N}(\mathbf{r}) = \int_{t_{near}}^{t_{far}} w(t) \cdot \mathbf{n}(t) dt \tag{7}$$

3.3 Confidence Estimation

The key idea of our approach is to use the result of a classical 3D reconstruction pipeline as *pseudo*-ground truth for supervising the rendered geometry from NeRF. However, the MVS stage can fail in textureless areas and non-Lambertian surfaces, thus making this supervision unreliable. For this reason, we additionally estimate a pixelwise confidence for each input image. Given a pixel (u, v), the corresponding confidence $c_{uv} \in [0, 1]$ is computed as:

$$c_{uv} = \exp\left(-\left(\frac{e_{uv}}{\bar{e}}\right)^2\right) \tag{8}$$

where e_{uv} is the top-K $(K = 4)$ forward-backward reprojection error of pixel (u, v) and $\bar{e}$ is the mean error over the entire set of observations. This error is computed as follows. A pixel (u_{ref}, v_{ref}) in the reference image is first projected in 3D according to its current depth estimate and observed by a source image k as the pixel (u_{src}^k, v_{src}^k). Then, this pixel is re-projected in 3D with the source depth hypothesis and observed again in the reference image to obtain the pixel $(\hat{u}_{ref}^k, \hat{v}_{ref}^k)$. The top-$K$ forward-backward reprojection error is simply given by:

$$e_{uv} = \frac{1}{K} \sum_{k=1}^{K} (u_{ref}^k - \hat{u}_{ref}^k)^2 + (v_{ref}^k - \hat{v}_{ref}^k)^2 \tag{9}$$

The relationship between the confidence value and such reprojection error is shown in Fig. 4. We have experimented with different confidence definitions, including a binary confidence with $c_{uv} = 1$ if the pixel (u, v) contributed to the discrete point cloud, 0 otherwise. However, we found that softly activating each pixel with continuous confidence yields to better results.
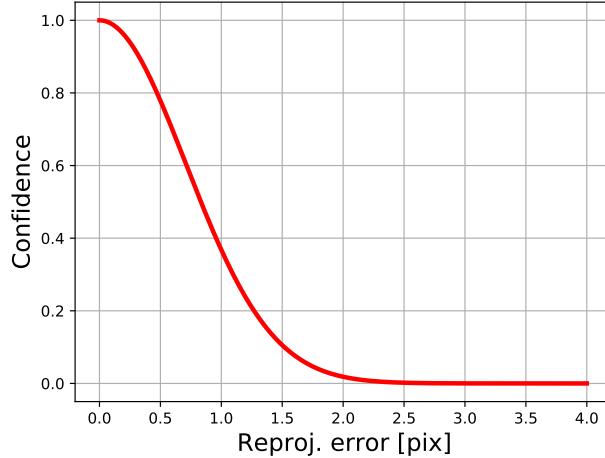
Fig. 4: Confidence value as a function of the reprojection error.

3.4   Loss Functions

At each training iteration, a random batch of rays $\mathcal{R}$ is sampled from the dataset and differentiable volumetric rendering is used to produce both colors and geometry by integrating along the ray. Then, we propose to optimize NeRF with the sum of three losses:

$$\mathcal{L} = \mathcal{L}_{rgb}(I, \hat{I}) + \lambda_{geom} \left(\mathcal{L}_{depth}(C, D, \hat{D}) + \mathcal{L}_{norm}(C, N, \hat{N}) \right) \qquad (10)$$

Consistently with the original NeRF formulation [18], the first term is the standard L_2 loss on rendered colors, which are obtained as shown in Eq. 2:

$$\mathcal{L}_{rgb}(I, \hat{I}) = \sum_{\mathbf{r} \in \mathcal{R}} ||I(\mathbf{r}) - \hat{I}(\mathbf{r})||^2 \qquad (11)$$

Moreover, we guide the optimization procedure by penalizing errors between the rendered geometry and the *pseudo*-ground truth. For both depths and normals, each ray is weighted by the corresponding confidence value to softly activate the supervision only in reliable pixels. More specifically, the learned depth values are computed from Eq. 4 and optimized as follows:

$$\mathcal{L}_{depth}(C, D, \hat{D}) = \sum_{\mathbf{r} \in \mathcal{R}} C(\mathbf{r}) \cdot \mathrm{Huber}(D(\mathbf{r}), \hat{D}(\mathbf{r})) \qquad (12)$$

Similarly, the third loss term is the confidence-weighted loss on rendered normals, which are given by Eq. 7:

$$\mathcal{L}_{norm}(C, N, \hat{N}) = \sum_{\mathbf{r} \in \mathcal{R}} C(\mathbf{r}) \cdot \mathrm{Huber}(N(\mathbf{r}), \hat{N}(\mathbf{r})) \qquad (13)$$

In both cases, the choice of the Huber loss over a standard L_2 loss is an additional step towards a more robust optimization. This function is quadratic for small errors and linear for large errors, making it less sensitive to outliers:

$$\text{Huber}(x, y) = \begin{cases} \frac{1}{2}(x - y)^2 & \text{if} \quad |x - y| < \delta \\ \delta \cdot (|x - y| - \frac{\delta}{2}) & \text{otherwise} \end{cases} \tag{14}$$

4   Experiments

In this section, we present experimental results to evaluate the proposed approach against the original formulation of NeRF. Software and hardware settings are detailed in Sec. 4.1, while quantitative and qualitative results are provided in Sec. 4.2 and Sec. 4.3, respectively.

4.1   Settings

Dataset We demonstrate the effectiveness of MVG-NeRF on the *Truck* scene from the Tanks & Temples dataset [12], as a proxy for real-world outdoor data. The scene is captured by 250 full HD images with resolution 1920×1080, as well as a high-precision laser scanner for acquiring 3D ground truth. The training set for NeRF is built by randomly selecting 90% of the images, with the remaining 10% being the test set. Similarly, only the training set of NeRF is used as input for classical 3D reconstruction.

Software The calibration parameters for NeRF and the geometric *pseudo-ground* truth are computed with COLMAP [29,30], a state-of-the-art classical 3D reconstruction pipeline. The results are obtained with the *automatic reconstruction* mode and default parameters. NeRF optimization follows an open-source PyTorch implementation[2], with custom modifications to support our confidence-aware geometric losses. After training, the mesh is extracted with a publicly available marching cubes algorithm[3] [15]. Confidence estimation is implemented as a C++ plugin after the MVS stage in COLMAP.

Implementation Details The framework has been tested on a single Nvidia V100 GPU with 32 GB RAM, but it can be adapted to run on lower tier devices with less memory. NeRF is optimized for 250000 iterations and a random batch of 1024 rays is selected at each training step. For each ray, 64 and 128 points are sampled for the coarse and fine stage of hierarchical sampling, respectively. The radiance field is approximated by a 8-layer MLP with 256 neurons each, and the geometric losses are weighted with $\lambda_{geom} = 0.1$. Finally, during the mesh extraction phase, a uniform grid of 256^3 points is fed to the marching cubes algorithm [15] and the density is thresholded at $\tau = 50$ for generating the binary occupancy field.

---

[2] https://github.com/yenchenlin/nerf-pytorch
[3] https://github.com/pmneila/PyMCubes

Method	PSNR ↑	SSIM ↑	LPIPS ↓	CD $\times 10^{-3}$ ↓
NeRF [18]	**21.2384**	**0.6526**	**0.3819**	2.3823
NeRF w/ depth	20.8911	0.6383	0.4318	<u>1.9701</u>
MVG-NeRF (ours)	<u>21.0013</u>	<u>0.6468</u>	<u>0.3971</u>	**1.8865**

Table 1: Quantitative results of NeRF with different geometric supervisions. Best and second results are **bold** and <u>underlined</u>, respectively.

4.2   Quantitative Results

In this section, a numerical comparison of our approach with the basic formulation of NeRF [18] is presented. We measure the performances of both methods in terms of the resulting 3D geometry and novel view synthesis results. Consistently with existing literature [18,21,3], three metrics are used to evaluate the quality of novel views:

- The **P**eak **S**ignal-to-**N**oise **R**atio (PSNR) is defined as follows:

$$\text{PSNR} = -\frac{10}{\log 10} \cdot \text{MSE}(I, \hat{I}) \tag{15}$$

  where $\text{MSE}(I, \hat{I})$ is the mean squared error between the rendered and the ground truth image. Higher is better.
- The **S**tructural **S**imilarity **I**ndex **M**easure (SSIM) was introduced in [35]. It considers the perceived change in structural information, thus measuring absolute errors. Higher is better.
- The **L**earned **P**erceptual **I**mage **P**atch **S**imilarity (LPIPS) was introduced in [47]. This metric computes the similarity between the activations of two image patches in a pre-trained network, such as AlexNet [13]. Lower is better.

Moreover, we want to quantify the geometric results to prove that MVG-NeRF generates better 3D models. To this end, the Chamfer distance between the point cloud from the laser scanner and the mesh vertices after marching cubes [15] is computed (lower is better):

$$\text{CD}(P, \hat{P}) = \frac{1}{|P|} \sum_{x \in P} \min_{y \in \hat{P}} ||x - y||^2 + \frac{1}{|\hat{P}|} \sum_{y \in \hat{P}} \min_{x \in P} ||x - y||^2 \tag{16}$$

The quantitative evaluation in Tab. 1 shows the comparison between our MVG-NeRF, the basic formulation of NeRF [18] and an intermediate setting where only depth from MVS is used as supervision, without pixelwise normals. It can be seen that our approach provides a better 3D geometry, while remaining competitive on the novel view synthesis task. Moreover, note that the supervision of dense depth without normals already improves significantly the quality of the underlying scene surface.

4.3   Qualitative Results

The significant improvement in geometry representation obtained by MVG-NeRF is also shown in qualitative results. Fig. 5 visualizes the output of the classical 3D reconstruction pipeline on input images, namely the geometric priors used as *pseudo*-ground truth and the confidence maps. Pixels with high confidence correspond to distinctive areas with good visibility among multiple views, while textureless regions or non-Lambertian surfaces have low confidence. This means that the loss terms on depths and normals will be softly activated only for reliable rays.

Novel RGB renderings from unseen poses are provided in Fig. 6. Despite the differences in perceptual metrics (see Tab. 1), the test views rendered from our MVG-NeRF match closely the output of basic NeRF. Note that the loss of quality in background regions is due to the limited sampling interval inherited from NeRF. This issue has already been solved by more recent algorithms, specifically designed for unbounded scenes [46], and a future research direction is to integrate such improvements within our framework (see Sec. 5).

Moreover, Fig. 7 and Fig. 8 show the pixelwise depths and normals obtained after volume rendering, respectively. It can be clearly seen that the proposed approach produces much smoother results, especially in terms of normal vectors. This finding is confirmed by the meshes visualized in Fig. 9. We obtain the cleanest 3D model, without the noise of NeRF [18] and the hallucinated geometry of COLMAP [29,30], followed by Poisson surface reconstruction [10,11].

5   Limitations and Future Work

The main limitation of MVG-NeRF is that it requires the execution of a classical 3D reconstruction pipeline before training a neural radiance field, which is a significant computational burden. For this reason, our approach is suited for offline reconstruction applications, but not for real-time graphics tasks. Moreover, we identify two main research directions that will be explored as future work.

Firstly, while we build on solid baselines for neural implicit representations [18] and geometry-based 3D reconstruction [29,30], several improvements over both NeRF and COLMAP have been presented in recent works [19,44,38,39]. In principle, this should accordingly improve their combination and our framework is flexible enough to support any version of these algorithms.

Secondly, beyond our novel idea of supervising NeRF with the results of a classical 3D reconstruction pipeline, another way for combining geometry and learning has been investigated in literature. Some works [7,5] proposed to enforce multi-view geometry constraints by warping patches as an additional loss function. This idea could be integrated in our framework, especially for pixels with low confidence. In this context, note that the consistency between deep features patches might be more robust than simply warping raw colors.
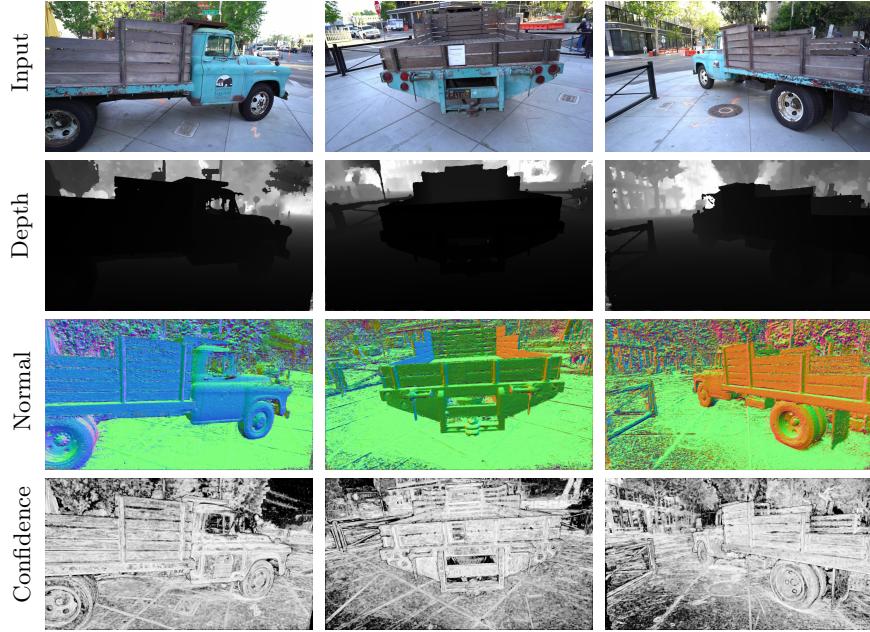
Fig. 5: Geometric priors and confidence from classical 3D reconstruction [29,30].

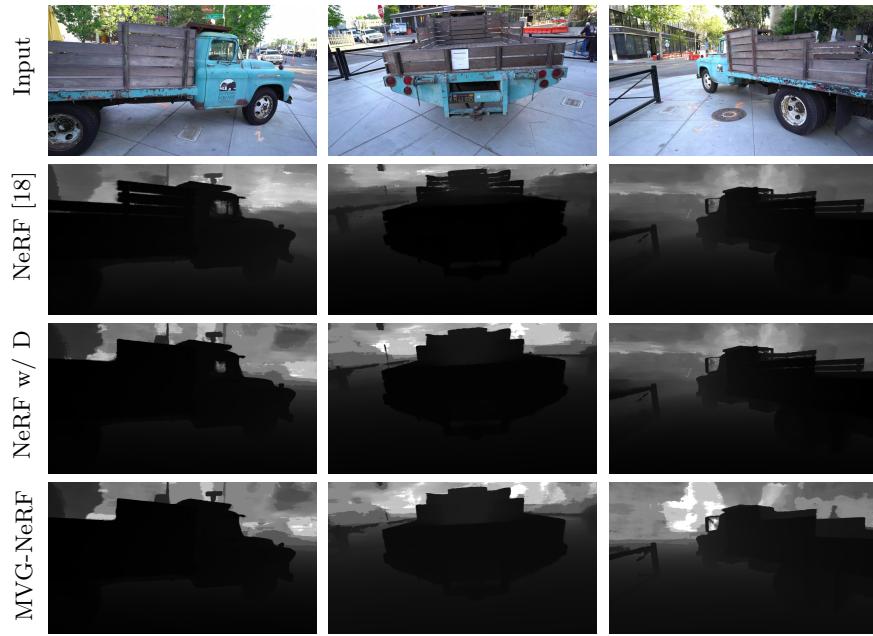Fig. 6: Qualitative comparison when rendering colors from novel views.

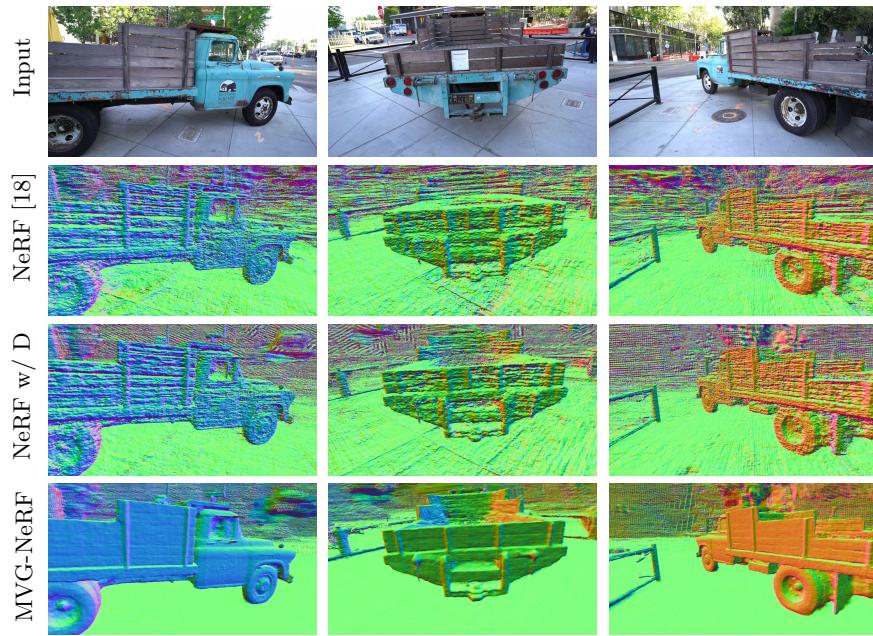Fig. 7: Qualitative comparison when rendering depths from novel views.

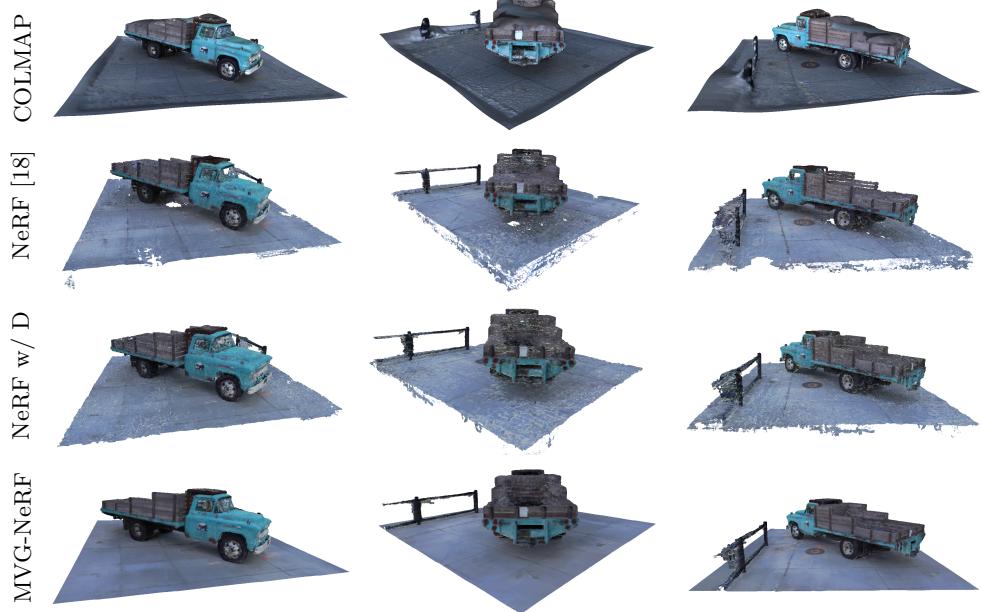Fig. 8: Qualitative comparison when rendering normals from novel views.

Fig. 9: Qualitative comparison of the resulting 3D mesh. The first row shows the result of Poisson surface reconstruction [10,11], applied on the dense point cloud from COLMAP [29,30]. Our approach removes both the noise of NeRF and the hallucinated geometry of the classical pipeline.

6   Conclusion

In this paper, we propose MVG-NeRF, a framework that effectively supervise NeRF geometry with classical 3D reconstruction during training, in order to generate cleaner and smoother 3D shapes. The key idea is to compute poses and calibration parameters with SFM, as well as pixelwise depths and normals for each input view with a state-of-the-art MVS algorithm. These geometric priors are used as *pseudo*-ground truth to guide NeRF optimization towards a multi-view consistent solution. Moreover, confidence maps are estimated to softly activated such supervision only in reliable pixels with low reprojection error. In this way, the confidence-aware geometric losses ignore the *pseudo*-ground truth in textureless areas and non-Lambertian surface, where MVS algorithms are known to fail. We show that MVG-NeRF significantly improves the resulting mesh quality on real-world outdoor scenes, while maintaining competitive performances on the novel view synthesis task.

References

1. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: Bmvc. vol. 11, pp. 1–11 (2011)
2. Cao, M., Zheng, L., Jia, W., Lu, H., Liu, X.: Accurate 3-d reconstruction under iot environments and its applications to augmented reality. IEEE Transactions on Industrial Informatics **17**(3), 2090–2100 (2021). https://doi.org/10.1109/TII.2020.3016393
3. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14124–14133 (2021)
4. Cui, Z., Tan, P.: Global structure-from-motion by similarity averaging. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 864–872 (2015). https://doi.org/10.1109/ICCV.2015.105
5. Darmon, F., Bascle, B., Devaux, J., Monasse, P., Aubry, M.: Improving neural implicit surfaces geometry with patch warping (2022)
6. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised NeRF: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2022)
7. Fu, Q., Xu, Q., Ong, Y.S., Tao, W.: Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction (2022). https://doi.org/10.48550/ARXIV.2205.15848, `https://arxiv.org/abs/2205.15848`
8. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion (June 2015)
9. Johari, M.M., Lepoittevin, Y., Fleuret, F.: Geonerf: Generalizing nerf with geometry priors. Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR) (2022)
10. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)
11. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG) **32**(3), 1–13 (2013)
12. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012), `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`
14. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In: ICCV (2021)
15. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
16. Van der Merwe, M., Lu, Q., Sundaralingam, B., Matak, M., Hermans, T.: Learning continuous 3d reconstructions for geometrically aware grasping. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 11516–11522 (2020). https://doi.org/10.1109/ICRA40945.2020.9196981
17. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)

18. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
19. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, `https://doi.org/10.1145/3528223.3530127`
20. Murez, Z., van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3d scene reconstruction from posed images. In: ECCV (2020), `https://arxiv.org/abs/2003.10432`
21. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S.M., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)
22. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020)
23. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: International Conference on Computer Vision (ICCV) (2021)
24. Orsingher, M., Zani, P., Medici, P., Bertozzi, M.: Efficient view clustering and selection for city-scale 3d reconstruction. In: International Conference on Image Analysis and Processing. pp. 114–124. Springer (2022)
25. Orsingher, M., Zani, P., Medici, P., Bertozzi, M.: Revisiting patchmatch multi-view stereo for urban 3d reconstruction. In: 2022 IEEE Intelligent Vehicles Symposium (IV). pp. 190–196. IEEE (2022)
26. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
27. Peng, S., Jiang, C.M., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape as points: A differentiable poisson solver. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
28. Roessle, B., Barron, J.T., Mildenhall, B., Srinivasan, P.P., Nießner, M.: Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2022)
29. Schönberger, J.L., Frahm, J.M.: Structure-from-Motion Revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
30. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise View Selection for Unstructured Multi-View Stereo. In: European Conference on Computer Vision (ECCV) (2016)
31. Schöps, T., Schönberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
32. Wadhwani, K., Kojima, T.: Squeezenerf: Further factorized fastnerf for memory-efficient inference (2022). https://doi.org/10.48550/ARXIV.2204.02585, `https://arxiv.org/abs/2204.02585`
33. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. NeurIPS (2021)

34. Wang, Y., James, S., Stathopoulou, E.K., Beltrán-González, C., Konishi, Y., Del Bue, A.: Autonomous 3-d reconstruction, mapping, and exploration of indoor environments with a robotic arm. IEEE Robotics and Automation Letters **4**(4), 3340–3347 (2019). https://doi.org/10.1109/LRA.2019.2926676
35. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
36. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: ICCV (2021)
37. Xu, B., Zhang, L., Liu, Y., Ai, H., Wang, B., Sun, Y., Fan, Z.: Robust hierarchical structure from motion for large-scale unstructured image sets. ISPRS Journal of Photogrammetry and Remote Sensing **181**, 367–384 (2021)
38. Xu, Q., Tao, W.: Multi-scale geometric consistency guided multi-view stereo. Computer Vision and Pattern Recognition (CVPR) (2019)
39. Xu, Q., Tao, W.: Planar prior assisted patchmatch multi-view stereo. AAAI Conference on Artificial Intelligence (AAAI) (2020)
40. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. European Conference on Computer Vision (ECCV) (2018)
41. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021)
42. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems **33** (2020)
43. Yeh, Y.J., Lin, H.Y.: 3d reconstruction and visual slam of indoor scenes for augmented reality application. In: 2018 IEEE 14th International Conference on Control and Automation (ICCA). pp. 94–99 (2018). https://doi.org/10.1109/ICCA.2018.8444222
44. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: ICCV (2021)
45. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. arXiv:2022.00665 (2022)
46. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv:2010.07492 (2020)
47. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)