VM-NeRF: Tackling sparsity in NeRF with View Morphing

Matteo Bortolon[1,2,3,*], Alessio Del Bue[2] and Fabio Poiesi[1,2]

[1]Technologies of Vision (TeV), Fondazione Bruno Kessler (FBK), Italy
[2]Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia (IIT), Italy
[3]Department of Information Engineering and Computer Science (DISI), University of Trento, Italy

Abstract
Neural representations can encode complex signals, such as 3D space, efficiently. They demonstrate considerable capabilities in novel view synthesis (NVS), which aims to output the appearance from an unknown viewpoint given a set of images and camera poses. In novel view synthesis, they represent the state-of-the-art in terms of achieved quality. The results achieved in Novel View Synthesis can reshape media acquisition processes in professional and amateur environments. However, current state-of-the-art techniques present considerable drawbacks, such as the number of images required to achieve such results. Our solution mitigates this problem by presenting a novel approach to generate geometrically consistent image transitions between viewpoints using View Morphing. Our VM-NeRF approach does not leverage prior knowledge about the scene structure, as View Morphing is based on the general principles of projective geometry. VM-NeRF tightly integrates this geometrical view generation process during the training procedure of standard NeRF approaches. Notably, this procedure allows for improved novel view synthesis, especially when few views are available. Preliminary experiments shows a consistent improvement with respect to current methods that tackle sparse viewpoints in NeRF models. We report a gain of PSNR up to 1.8dB and 1.0dB when eight and four views are used for training, respectively.

Keywords
neural rendering, implicit representation, few-viewpoints

1. Introduction

Given a set of known camera poses and their rendered viewpoints of a scene, Novel View Synthesis (NVS) aims to generate unseen viewpoints [1]. NVS's most direct application is in the entertainment sector [2], including film, gaming, and virtual or augmented reality [3]. New advancements in NVS can significantly simplify the creation of digital twins of real subjects such as people, animals, static objects, or even entire scenes. Digital twins can have various uses, and one such application is in games, where the assets necessary can be quickly digitized from the real world and used inside. Digital twins are also helpful to quickly add digital effects to movies, as real-world actors or objects once acquired. Investing in NVS leads to faster time-to-market, lower costs, and customized products in the entertainment sector. Furthermore, advancements in NVS can help the preservation, dissemination and analysis of Italian cultural heritage, especially the less-known assets.

Novel View Synthesis state-of-the-art is currently obtained with Neural Radiance Fields (NeRF) and its evolutions [4, 5]. NeRF formulates the problem as the resolution of a volumetric function. The volumetric function is optimized on a single scene, avoiding dataset bias problems and making the training less data-hungry. It is used to predict the density and the colour of a 3D point in space. These points are sampled along a set of rays $\mathcal{R} = \boldsymbol{r}$, one for each ground-truth pixel present in the input image, extracted using standard camera geometry from the known camera poses. The predicted color of each ray $\hat{\boldsymbol{c}}(\boldsymbol{r})$ is obtained from a density-based light propagation formulation:

$$\hat{\boldsymbol{c}}(\boldsymbol{r}) = \sum_{i=1}^{\Gamma} s(i) \left(1 - e^{-\hat{\boldsymbol{\sigma}}(\boldsymbol{r})_i \boldsymbol{\delta}_i}\right) \hat{\mathbf{c}}(\boldsymbol{r})_i, \qquad (1)$$

where Γ indicates the number of points along the ray, $\hat{\boldsymbol{c}}(\boldsymbol{r})_i$ is the color and $\hat{\boldsymbol{\sigma}}(\boldsymbol{r})_i$ is the density predicted by the network at i. $\boldsymbol{\delta} i = ti + 1 - t_i$ is the distance between adjacent sampled 3D spatial locations. $s(i)$ is the inverse of the volume density that is accumulated up to the i^{th} spatial location, which is computed as

$$s(i) = e^{-\sum_{j=1}^{i-1} \hat{\boldsymbol{\sigma}}(\boldsymbol{r})_j \boldsymbol{\delta}_j}. \qquad (2)$$

The same formulation is used to predict the depth, replacing the point colour $\hat{\boldsymbol{c}}(\boldsymbol{r})_i$ with the distance from the start of the ray. The optimization loop minimizes the difference between the colour predicted by the network and the ground truth.

These solutions are typically trained with several images, for example, about a hundred images taken from different and uniformly distributed camera viewpoints around an object of interest [4]. When viewpoints are not uniformly distributed or are a few (for example, four

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
*Corresponding author.
mbortolon@fbk.eu (M. Bortolon); alessio.delbue@iit.it (A. D. Bue); poiesi@fbk.eu (F. Poiesi)
0000-0001-8620-1193 (M. Bortolon); 0000-0002-2262-4872 (A. D. Bue); 0000-0002-9769-1279 (F. Poiesi)

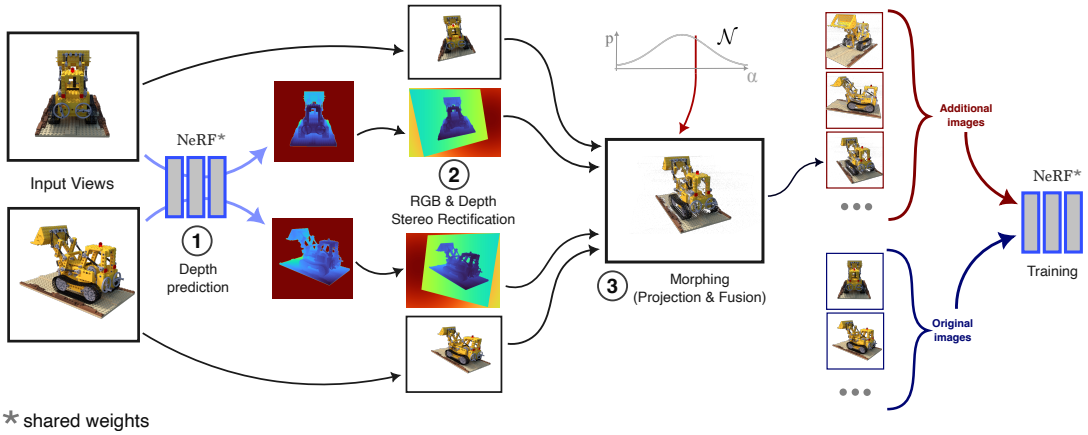

Figure 1: Block diagram of NeRF-based View Morphing (VM-NeRF). From the left, we (1) predict the depth with NeRF, (2) rectify the input images and predicted depths, and (3) compute the image morphing of a view randomly positioned between the view pair. α determines the new view position and it is sampled from a Gaussian distribution.

or eight), the resulting NeRF model may fail to produce satisfactory renderings [6, 7]. These solutions trained on few or non-uniformly distributed viewpoints present a higher likelihood of overfitting on these viewpoints, namely, the few-shot view synthesis problem [6]. This can also happen in professional environments where production requirements limit placement and amount of cameras. Few viewpoints solutions proposed so far leverage training on multiple scenes or introduce optimization constraints. Instead, we use a geometry-based strategy (NeRF-VM) that enables NeRF to learn implicit representations of scenes captured from few viewpoints.

2. Approach

Our approach, View Morphing NeRF (VM-NeRF), aims to generate additional training views by interpolating the image content of nearby ground-truth view pairs using predicted depth.

View Morphing uses pixel correspondences between images to create a smooth transition between two views without prior knowledge of the scene [8]. The first step of View Morphing is stereo-rectifying the two images to make them coplanar. Then, a new image is created by combining the two images using a pixel correspondence map, which shows how pixels in one image correspond to pixels in the other. This new image is positioned between the pair's images, and its viewpoint is on the line connecting the cameras' optical centres. In the original work, the pixel correspondence map can be input manually or created by keypoint detectors. Instead, our approach finds pixel correspondences using network-predicted depth. This lets NeRF train on known views before adding DVM-generated views into the training process.

Our approach involves three operations that allow the original View Morphing algorithm to be adapted for NeRF camera configurations. Fig. 1 shows a graphical representation of our method's steps. These three steps are: computing the depth, rectifying the images, and morphing the images based on relative depths.

The depth is predicted by rendering the ground-truth images with the learned volumetric function. The second step is rectifying the two images and the depth obtained, which makes them coplanar [9]. The last step involves image morphing, which fuses rectified images to create a new morphed image. This process is divided into three steps: *i)* finding the pixel correspondences; *ii)* computing the position of each pixel on the morphed camera; *iii)* fusing pixels that fall in the same position. We find the pixel correspondences by computing disparity maps as functions of rectified estimated depths. The position of each pixel on the morphed view is then computed, but multiple pixels may end up at the same coordinates. We resolve this issue with a GPU-optimized function that removes duplicates by selecting the one nearest to the image plane. The procedure can handle arbitrary amounts of pixels falling in the same position.

View Morphing allows the synthesis of a new view at any point on the line connecting the known cameras. We randomly sample this point during training using a Gaussian distribution centred halfway through the camera pair. Specifically, let us consider a normalized distance between the two cameras. The Gaussian distribution is centred at 0.5, and the standard deviation σ is chosen such that $3\sigma \rightarrow \epsilon$ at the optical centre positions. Therefore, we sample from $\mathcal{N}(0.5, \sigma)$. We regenerate views during training for each valid camera pair as the predicted depth improves over time.

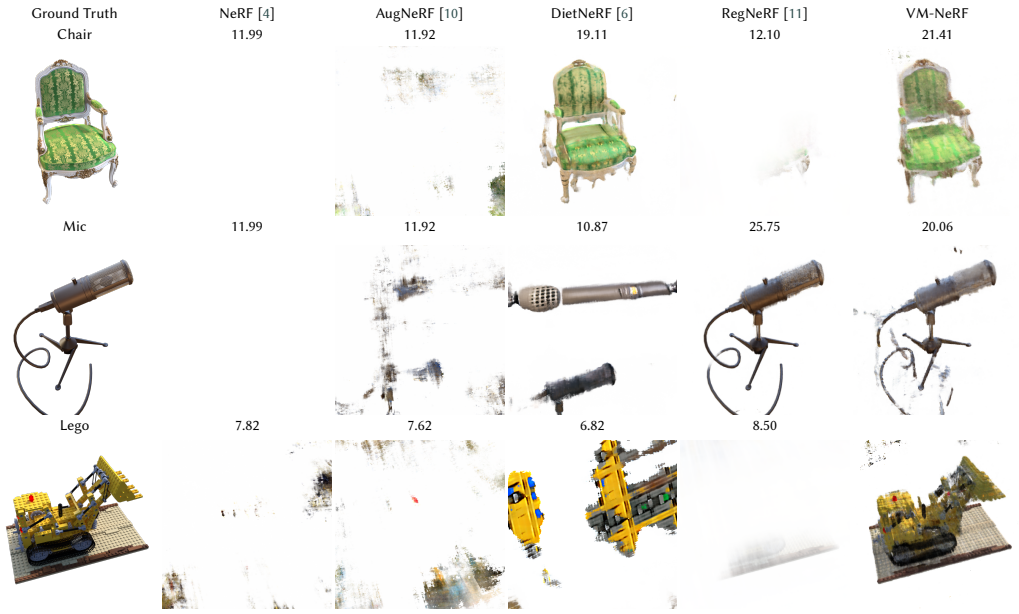

Figure 2: Comparisons on test-set views of scenes of NeRF realistic synthetic 360°, trained on 4-views. Unlike other techniques VM-NeRF help to increase the probability of building a consistent implicit representation with very sparse view. We report the PSNR that we measured for each method and for each rendered image.

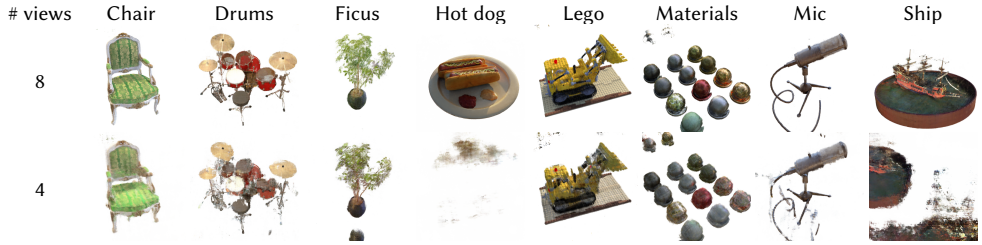

Figure 3: The figure displays examples of renderings obtained using VM-NeRF. The top row is generated using eight views as input, while the bottom row is generated using four views. Except for the hot dog and ship, VM-NeRF produces consistent results even with four views. The artefacts derive from the network's inability to establish coherence between the presented views. Consequently, it overfits by assigning the pixel colour to a random point, which will then become visible from a different viewpoint.

2.1. Results

We evaluate our method on three training setups using the NeRF realistic synthetic 360°dataset [4], which comprises eight scenes: Chair, Drums, Ficus, Lego, Materials, Ship, Mic, and Hot Dog. We select $N = 8$ and $N = 4$ views out of the 100 available for each scene using the Farthest Point Sampling (FPS) [12] method (with the first view used for FPS initialization in each scene). We test each trained model on all the test views of the NeRF realistic synthetic 360°dataset. We evaluate the rendering results using the Peak Signal-to-Noise Ratio (PSNR) score, the Structured Similarity Index Measure (SSIM) [13], and the Learned Perceptual Image Patch Similarity (LPIPS) [14]. We compare our approach quantitatively against DietNeRF [6] and RegNeRF [11], which are the most recent methods for few-shot view synthesis, and AugNeRF [10], which is to our knowledge the only data augmentation method for NeRF.

We implement NeRF and our approach in PyTorch Lightning and run experiments on a single Nvidia A40 with a batch size of 1024 rays. We can train a single scene in about two days. We use the original implementations of DietNeRF, AugNeRF, and RegNeRF to evaluate the different setups.

Table 1
Results on the NeRF realistic synthetic 360° dataset.

# views	Method	PSNR ↑	SSIM ↑	LPIPS ↓
100	*NeRF [4]*	31.21	0.9513	0.0465
8	NeRF [4]	23.45	0.8673	0.1303
	DietNeRF [6]	22.98	0.8545	0.1258
	AugNeRF [10]	10.04	0.5415	0.3866
	RegNeRF [11]	22.91	0.8667	0.1138
	VM-NeRF (ours)	24.39	0.8768	0.1146
4	NeRF [4]	10.98	0.6550	0.3620
	DietNeRF [6]	12.61	0.6591	0.3302
	AugNeRF [10]	8.14	0.3924	0.4802
	RegNeRF [11]	15.88	0.7932	0.1994
	VM-NeRF (ours)	16.90	0.7563	0.2461

Quantitative. Table 1 shows the NeRF synthetic 360°results using 4 and 8 images per object. In the eight-view setting, VM-NeRF outperforms all the other methods. Interestingly, the original version of NeRF performs second best, followed by DietNeRF and RegNeRF. AugNeRF fails to produce satisfactory results. Although noisy, we observed that VM-NeRF could effectively leverage the depth information estimated during training, resulting in higher quality results on average (24.14 vs 23.59).

In the four-view setup, we achieve an improvement of +1.02 PSNR on average. The results also show that the perturbation of the known input views, done by AugNeRF, has adverse effects in all the tested setups.

Qualitative. Fig. 2 shows some qualitative results on Chair, Mic, and Lego, where we can observe that VM-NeRF produces results with better details than DietNeRF. Fig. 3 compares our method's results with 8 and 4 views.

2.2. Conclusion

Few-viewpoint novel view synthesis is important because it enables a faster generation of 3D assets in the case of real-world applications, such as AR/VR. We presented a novel method for NeRF based on the view morphing technique [15] to tackle the problem of few-viewpoint synthesis. View morphing requires no prior knowledge of the 3D shape and is based on general principles of projective geometry. We showed how to synthesise random novel 3D projective transformations of the object between viewpoint pairs using view morphing and how to use them with NeRF. The results show that a geometric-based strategy can outperform current methods in challenging scenarios.

References

[1] O. Gallo, A. Troccoli, V. Jampani, Novel view synthesis: From depth-based warping to multi-plane images and beyond, 2020. URL: https://nvlabs.github.io/nvs-tutorial-cvpr2020/, conference on Computer Vision and Pattern Recognition.

[2] F. Devernay, A. R. Peon, Novel view synthesis for stereoscopic cinema: detecting and removing artifacts, in: Workshop on 3D Video Processing (ACMMM), 2010.

[3] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, P. Debevec, Baking neural radiance fields for real-time view synthesis, in: ICCV, 2021.

[4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, in: ECCV, 2020.

[5] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, in: ICCV, 2021.

[6] A. Jain, M. Tancik, P. Abbeel, Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis, in: ICCV, 2021.

[7] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, A. Kar, Local light field fusion: Practical view synthesis with prescriptive sampling guidelines, ACM Transactions on Graphics (TOG) 38 (2019) 1–14.

[8] S. M. Seitz, C. R. Dyer, View morphing, in: Conference on Computer graphics and interactive techniques, 1996.

[9] A. Fusiello, E. Trucco, A. Verri, A compact algorithm for rectification of stereo pairs, Machine Vision and Applications 12 (2000) 16–22.

[10] T. Chen, P. Wang, Z. Fan, Z. Wang, Aug-NeRF: Training Stronger Neural Radiance Fields with Triple-Level Physically-Grounded Augmentations, in: CVPR, 2022.

[11] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, N. Radwan, Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs, in: CVPR, 2022.

[12] C. R. Qi, O. Litany, K. He, L. J. Guibas, Deep hough voting for 3d object detection in point clouds, in: ICCV, 2019.

[13] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (2004) 600–612.

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: CVPR, 2018.

[15] S. M. Seitz, C. R. Dyer, View morphing, in: Conference on Computer graphics and interactive techniques, 1996.