Neural 3D Implicit Modeling

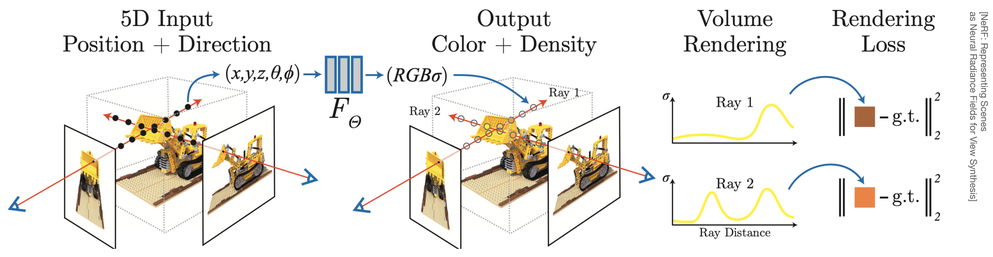
Karim Kassab

Supervisors
Jean-Yves Franceschi[2]
Laurent Caraffa[1]
Jeremie Mary[2]
Valérie Gouet-Brunet[1]

[1] IGN
[2] Criteo

Paris Est Sup — École Doctorale Mathématiques et STIC
Université Gustave Eiffel

Introduction: Neural Radiance Fields (NeRFs)



- NeRF enables a learned representation.
- Three key aspects make it work:
  - Ray marching & Volume rendering
  - Stratified sampling
  - Positional Encoding

Limitations

- Slow optimization
- No generalization
- Static scenes only
- Rigid bodies only
- Requires many training images
- Requires precise camera parameters
- Non-robust towards variable illuminations
- Non-robust towards transient occluders

- NeRF models a scene as a 5D function $F_\Theta : (x, y, z, \theta, \phi) \rightarrow (R, G, B, \sigma)$ where Θ are its learnable parameters.
- F_Θ is approximated using a standard MLP, with some few added tricks (see below).
- For each point (x, y, z) and viewing direction (θ, ϕ), NeRF attributes a color (R, G, B) and a density (σ).

Primary Problem

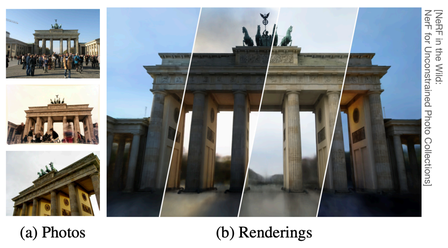
- The key problem that this PhD tackles, and that is being faced by both IGN and Criteo, is the 3D modeling and generation of assets in unconstrained environments.
- Both Criteo and IGN own datasets of images that are taken in uncontrolled setups that exhibit illumination changes as well as transient occluders and dynamic backgrounds.
- The main objective of this PhD is to improve upon the limitations of NeRFs in these contexts in order to eventually better fit these use-cases.

Perspectives & Future Work

- NeRFs have been used in very diverse applications, from 3D object generation to dynamic scene modeling and interpolation, and it has also seen a lot of works that focus on overcoming its main limitations, as can be seen in the state-of-the-art illustration above.
- In this section, we enumerate some of our targeted applications at IGN as well as at Criteo.

Modeling 3D monuments using unconstrained photo collections

- An interesting application of NeRFs is 3D monument modeling using publicly available images taken by tourists.
- 3D monument modeling can also be exploited for improving the indexing and the structuring of photographic collections through data augmentation



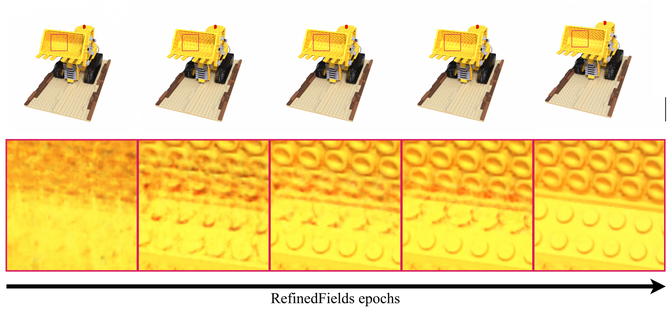
(a) Photos (b) Renderings

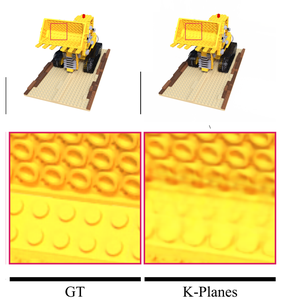
Modeling distractor-free objects in 3D

- In the context of advertising, modeling 3D objects captured in dynamic contexts (varying occlusions and illuminations) will allow the use of these assets in 3D ads (e.g. video games, metaverse)

Proposal 1 RefinedFields: Radiance Fields Refinement for Unconstrained Scenes

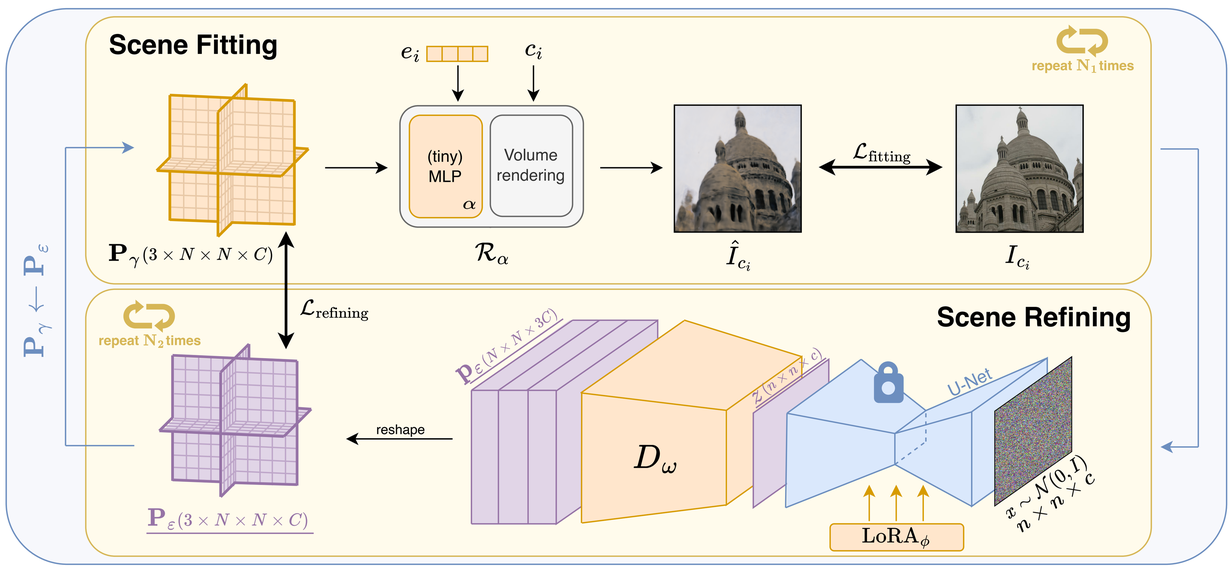
- We introduce **RefinedFields**, a novel way to refine scene representations. This is, to the best of our knowledge, the first method leveraging pre-trained networks for novel view synthesis in-the-wild.
- We show that the proposed scene refining pipeline, coupled with our training procedure, makes RefinedFields outperform the state-of-the-art on the task of novel view synthesis in-the-wild.





Synthetic toy example. The training set is constrained to 50% of its initial size for both RefinedFields and K-Planes. RefinedFields refines the K-Planes representation enabling the proper reconstruction of details in the scene.

Qualitative Results. Given images of the Trevi fountain from Phototourism, as well as a pre-trained model, our method leverages the pre-trained model and refines K-Planes with finer details that are under-represented when optimizing the same K-Planes on the images alone.



Scene learning procedure. The K-Planes P_γ, the MLP with trainable parameters α, and the appearance embeddings e_i are learned during scene fitting. The LoRA parameters ϕ as well as the decoder D_w are learned during scene refining. The pre-trained U-Net is frozen. At each iteration, new planes P_ε are inferred and assigned to P_γ, which are then corrected by scene fitting.

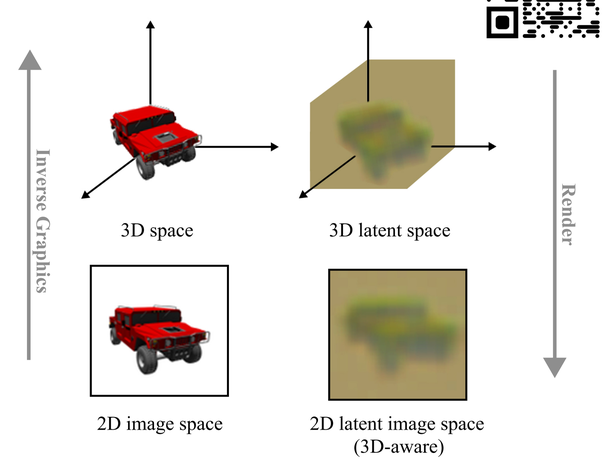
	Brandenburg Gate		Sacré Coeur		Trevi Fountain	
	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)
NeRF (Mildenhall et al., 2020)	18.90	0.8159	15.60	0.7155	16.14	0.6007
NeRF-W (Martin-Brualla et al., 2021)	24.17	0.8905	19.20	0.8076	18.97	0.6984
Ha-NeRF (Chen et al., 2022b)	24.04	0.8773	20.02	0.8012	20.18	0.6908
CR-NeRF (Yang et al., 2023)	26.53	**0.9003**	22.07	**0.8233**	21.48	0.7117
K-Planes (Fridovich-Keil et al., 2023)	25.49	0.8785	20.61	0.7735	22.67	0.7139
K-Planes-SS (Fridovich-Keil et al., 2023)	24.48	0.8629	19.86	0.7419	21.30	0.6627
RefinedFields-noFinetuning (ours)	25.39	0.8834	21.41	0.8059	22.54	0.7324
RefinedFields-noPrior (ours)	25.42	0.8822	21.17	0.7978	22.16	0.7251
RefinedFields (ours)	**26.64**	0.8869	**22.26**	0.8176	**23.42**	**0.7379**

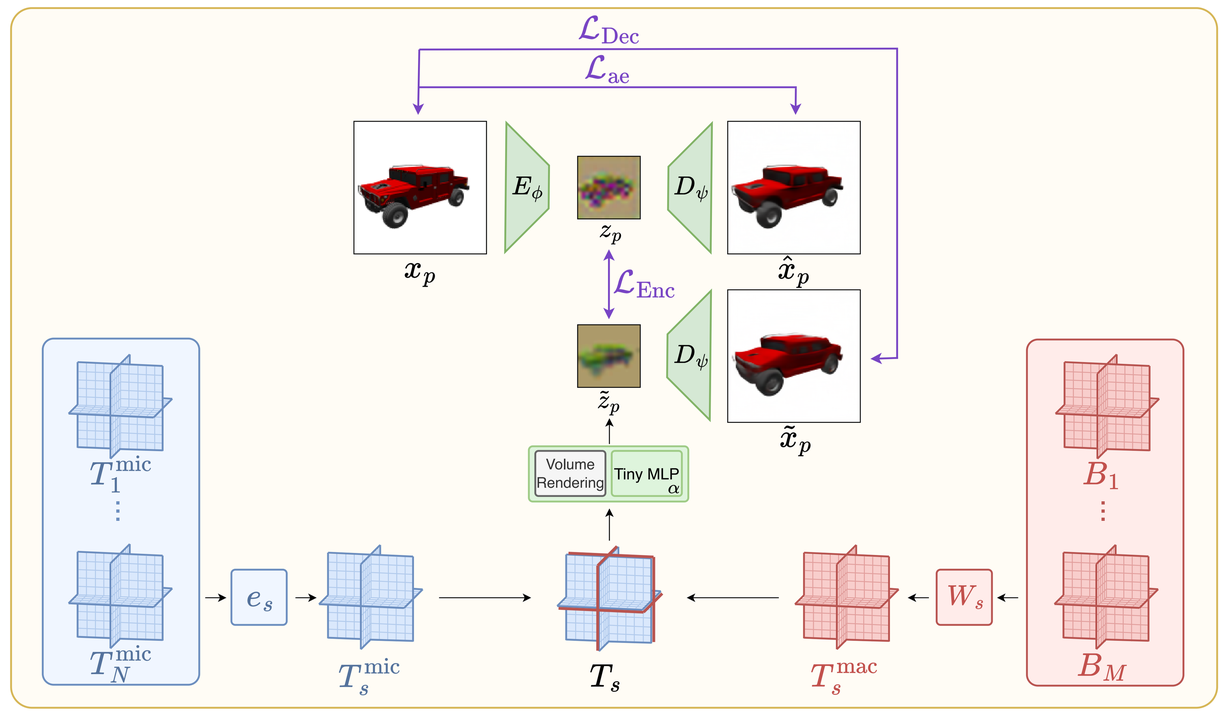
Quantitative results. Results on three real-world datasets from Phototourism. The **bold** and underlined entries respectively indicate the best and second-best results.
Our method demonstrates state-of-the-art performance on the task of NVS-W.

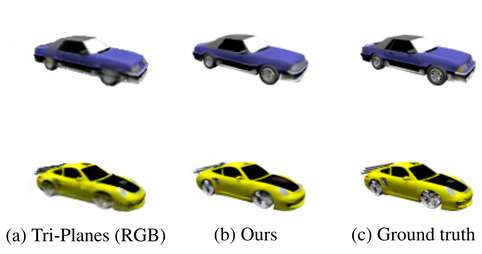
Proposal 2 Exploring 3D-aware Latent Spaces for Efficiently Learning Numerous Scenes

- **Inverse graphics problem:** How to model a scene using its captured images?
- **Scaled inverse graphics problem:** How to model abundantly many scenes at once?

- We build a 3D-aware latent space in which neural scene representations can be trained,
- We present an approach to minimize the capacity needed to model a latent scene by sharing common globally-trained scene representations across scenes,
- Our work can learn 1000 scenes with 86% less time and 44% less memory than our base representation.



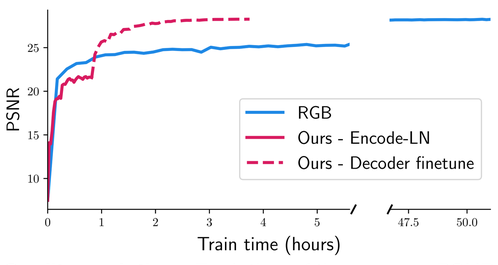




Visual comparison. Visual comparison of novel view synthesis quality for our method and Tri-Planes (RGB).

Experiment	Latent Space	Micro-Planes	Macro-Planes	Train scenes	Exploit scenes
Ours-Micro	✓	✓	✗	26.52	26.95
Ours-Macro	✓	✗	✓	25.67	26.10
Tri-Planes-Macro (RGB)	✗	✗	✓	27.84	28.00
Tri-Planes (RGB)	✗	✓	✗	**28.24**	28.40
Ours-No-Prior	✓	✓	✓	27.72	28.13
Ours	✓	✓	✓	28.05	**28.48**

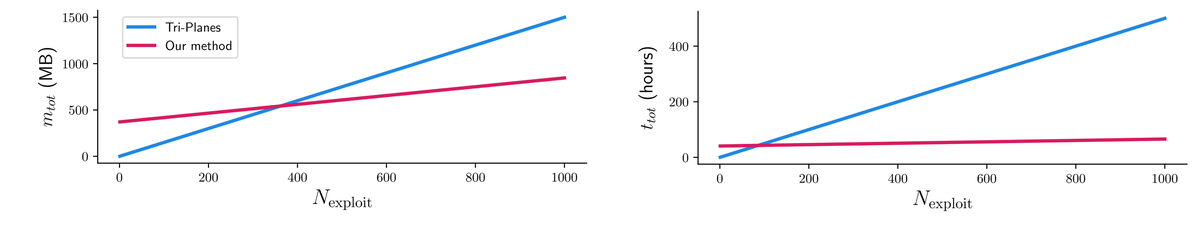
Quality comparison. Average PSNR demonstrated by our method with a comparison to Tri-Planes and ablations of our pipeline. All metrics are computed on never-seen test views. Here, we consider $N_{\text{train}} = 500$, $N_{\text{exploit}} = 100$, and $M = 50$.



Quality evolution. Evolution of the average PSNR demonstrated by our method compared to Tri-Planes.

	t_{scene} (min)	$t_{\text{scene}}^{\text{eff}}$ (min)	m_{scene} (MB)	$m_{\text{scene}}^{\text{eff}}$ (MB)	Rendering Time (ms)	Rendering Resolution
Encoder	—	—	0	0.13	—	—
Decoder	—	—	0	0.19	9.7	128×128
Tri-Planes (RGB)	32	32	1.5	1.5	23.3	128×128
Our method	2	4.5	0.48	0.84	11.0	128×128

Cost comparison. Per scene cost comparison with Tri-Planes (RGB). Here, we consider $N_{\text{train}} = 500$, $N_{\text{exploit}} = 1000$, $t_{EC} = 40$ hours, $M = 50$, $F^{\text{mac}} = 22$.



Cost evolution. Total memory and train time evolution when scaling the number of trained scenes N_{exploit}.
Our method demonstrates more favorable scalability properties as compared to Tri-Planes (RGB).