A generic and flexible regularization framework for NeRFs

Thibaud Ehret, Roger Marí and Gabriele Facciolo
thibaud.ehret@ens-paris-saclay.fr

Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, 91190, Gif-sur-Yvette, France

https://github.com/tehret/diffnerf

NeRF regularization

NeRFs [1] learned from few views (e.g. three views), are often noisy, present inconsistent geometry or fail to converge. Here, we present a generic geometry regularization framework that takes advantage of the differentiable property of MLPs (when replacing ReLUs by Softplus).

Let $(\mathbf{o}, \mathbf{v}) = C(x, y)$ be the origin and direction of the ray at position (x, y) and associated to the camera configuration C. We also note d the depth function for a ray and $\tilde{d}(x, y) = d(C(x, y))$.

Depth regularization (first order).

$$L_{depth} = \sum_{(x,y)} \text{clip}(\|\nabla \tilde{d}(x, y)\|^2, g_{max}). \quad (1)$$

$\nabla_{(x,y)}\tilde{d}(x, y) = \mathbf{J}_C(x, y)\nabla_{\mathbf{v}} d(\mathbf{o}, \mathbf{v})$
depends on the Jacobian of the camera function C.

Normal regularization (second order).

$$L_{normals} = \sum_{(x,y)} \|J_{\tilde{n}}(x, y)\|_F^2 \quad (2)$$

where $J_{\tilde{n}}$ is the Jacobian of the map of normals.

Camera independent regularization (depth).

$$L_{depth} = \sum_{(\mathbf{o},\mathbf{v})\in\mathcal{R}} \|\nabla_{\mathbf{o}} d(\mathbf{o}, \mathbf{v}) - \langle \nabla_{\mathbf{o}} d(\mathbf{o}, \mathbf{v}), \mathbf{v}\rangle \mathbf{v}\|^2 \quad (3)$$

does not depend on the camera definition anymore (see article to find out where this approximation comes from).

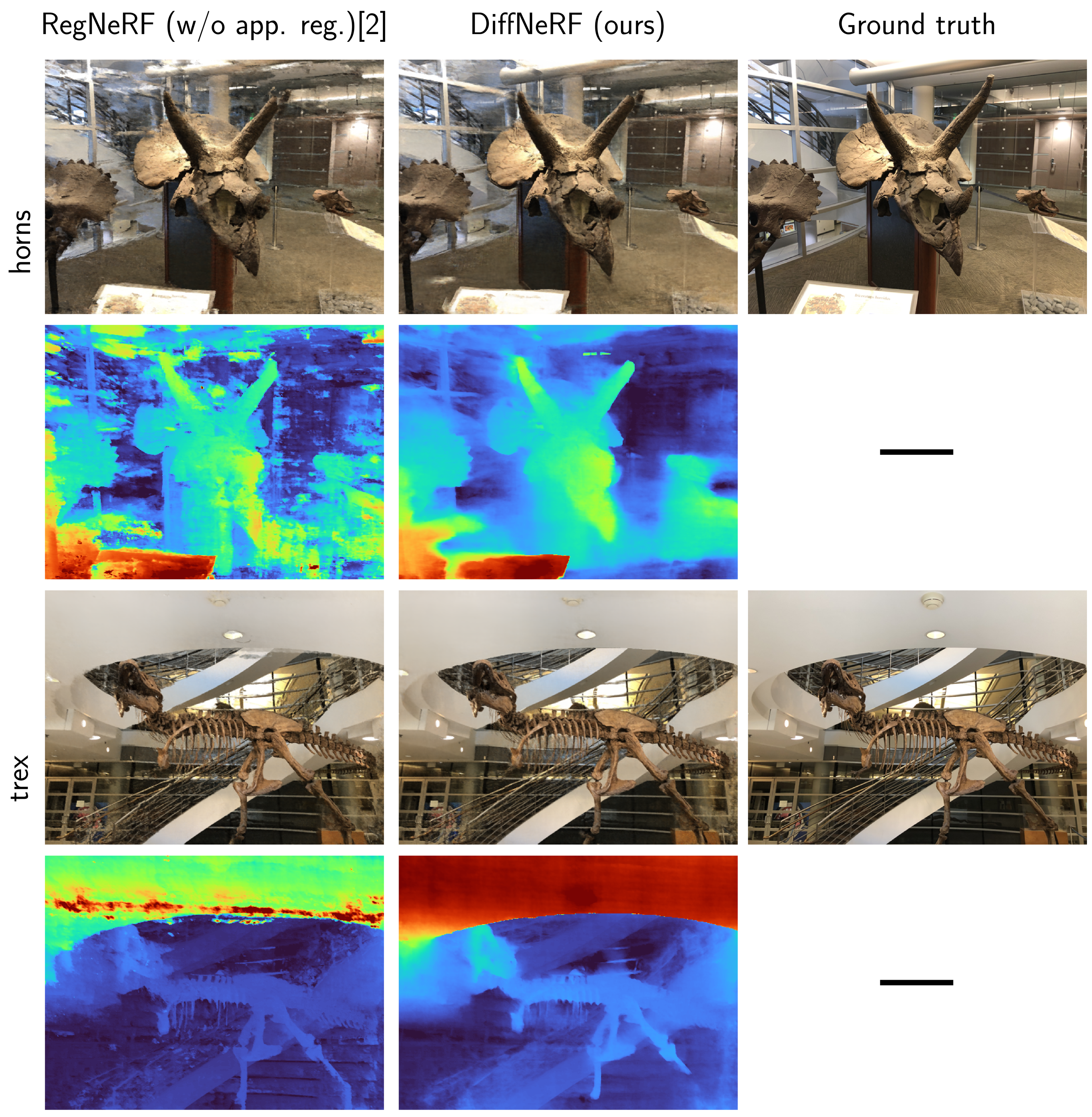

Figure 1: Novel view synthesis for the *horns* (top) and *trex* (bottom) sequences of the LLFF dataset after training with three views. The depth map produced by the proposed DiffNeRF (using Eq. (3)) is more regular than those produced RegNeRF. It also recovers more details both in the foreground (see the sign panel on the left or the triceratops' left horn) but also in the background (see the glass panels and the handrails).

Extension to surface models

Surface based methods like VolSDF [3] learn the surface by means of a signed distance function (SDF). Using the SDF, the surface $\mathcal{S}$ is defined implicitly as the set of points $\{\mathbf{x} \in \mathbb{R}^3 \mid F(x) = 0\}$. It is possible to compute other differential quantities related to surface regularity, such as the curvature.

Surface curvature. Mean and Gaussian curvatures are respectively defined as

$$\gamma_{mean} = -\textbf{div}\left(\frac{\nabla F}{\|\nabla F\|}\right) \quad (4)$$

and

$$\gamma_{gauss} = \frac{\nabla F \times H^*(F) \times \nabla F^t}{\|\nabla F\|^4}, \quad (5)$$

where H^* is the adjoint of the Hessian of F (see [4]).

Regularizing the curvature. Using (4) and (5), we define

$$L_{curv}(\kappa_{curv}) = \mathbb{E}_{x\in\mathcal{S}}\left[\min\left(|\gamma(x)|, \kappa_{curv}\right)\right]. \quad (6)$$

where γ can be either γ_{mean} or γ_{gauss}, depending on the preferred behavior, and κ_{curv} is a clipping value. The final loss to train a regularized VolSDF model becomes

$$L = L_{RGB} + \lambda_{SDF} L_{SDF} + \lambda_{curv} L_{curv}(\kappa_{curv}). \quad (7)$$

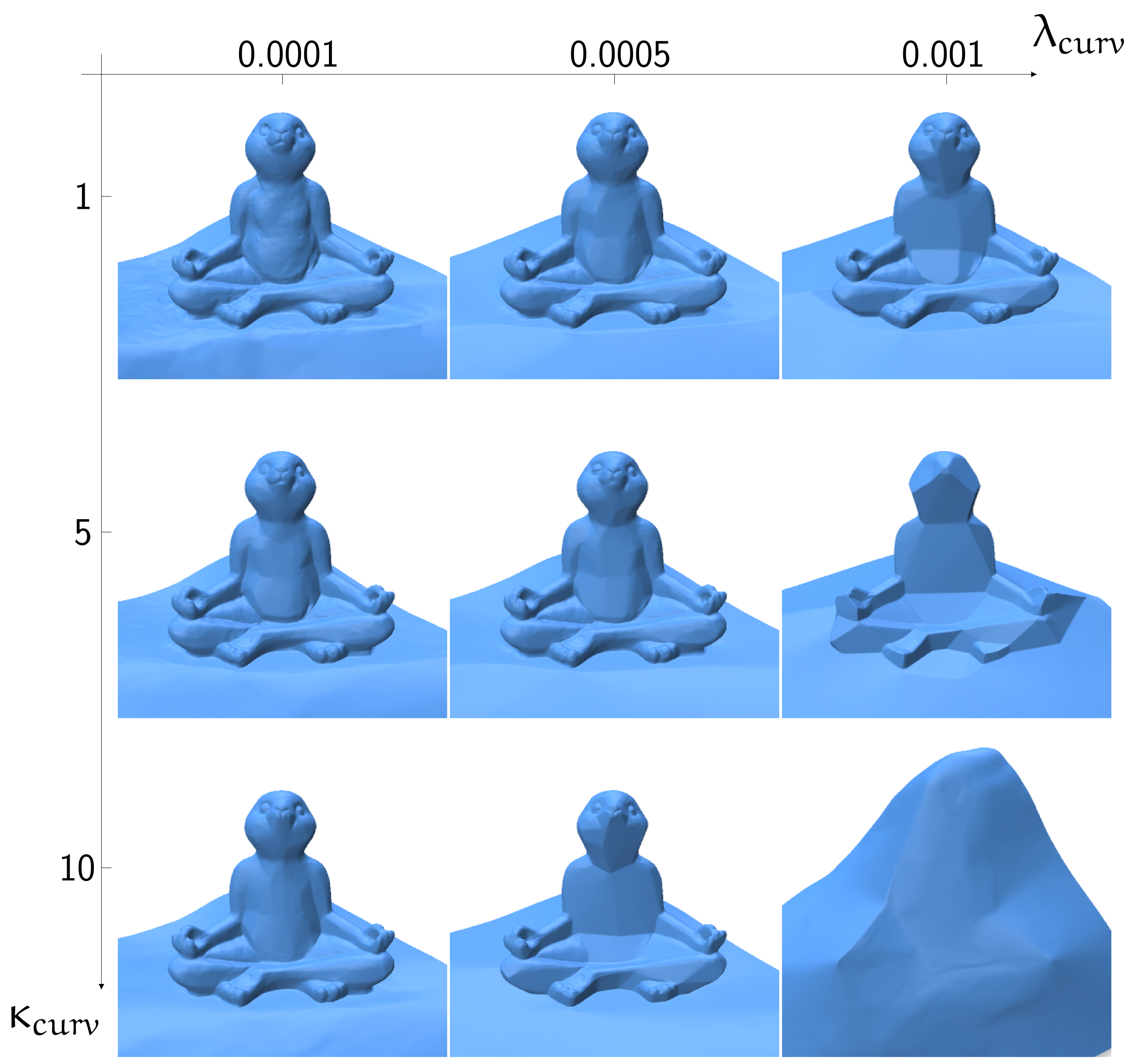

Figure 2: Study of the impact of the two parameters controlling the strength of the surface regularization, i.e. the regularization weight λ_{curv} and the clipping value κ_{curv}. These results were computed using the Gaussian curvature.

References

[1] Ben Mildenhall et al. "NeRF: Representing scenes as neural radiance fields for view synthesis". ECCV, 2020.

[2] Michael Niemeyer et al. "RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs". CVPR, 2022.

[3] Lior Yariv et al. "Volume rendering of neural implicit surfaces.". NeurIPS, 2021.

[4] Ron Goldman. "Curvature formulas for implicit curves and surfaces". Computer Aided Geometric Design, 2005.