Supplementary Material for Robust and Efficient Edge-guided Pose Estimation with Resolution-conditioned NeRF

Liesbeth Claessens[1]
liesbeth.claessens@mavt.ethz.ch
Fabian Manhardt[2]
fabianmanhardt@google.com
Ricardo Martin-Brualla[2]
rmbrualla@google.com
Roland Siegwart[1]
rsiegwart@ethz.ch
Cesar Cadena[1]
cesarc@ethz.ch
Federico Tombari[2]
tombari@google.com

[1] Autonomous Systems Lab
ETH Zurich
Zurich, Switzerland
[2] Google, Inc.
Zurich, Switzerland

1 Architectural details

We use a multiresolution hash grid with 16 levels with number of entries ranging from 4096 to 524288. The features encoded at each resolution level have a dimensionality of 2, leading to a total encoding size of 32. For the MLPs, we follow the architecture used in [2], which consists of two concatenated MLPs:

1. The density MLP - which has one hidden layer of size 64 with ReLu activations - maps the encodings to 16 output values, the first of which is the log-density.

2. The color MLP - which has two hidden layers of size 64 with ReLu activations - then maps the outputs of the density MLP and the viewing direction to the color.

2 Pose inference details

The optimisation strategy used differs slightly on a per-parameter basis. Nevertheless, the core structure of all parameter optimisations is the same: an exponentially decayed learning rate with which the updates are computed using Adam [1]. Additionally, in order to counteract numerical instability, we set undefined values to zero and clip the gradients to 1 at every time step of the optimisation for all parameters.

The formula for computing the exponentially decayed learning rates α_t at timestep t is:

$$\alpha_t = \alpha_0 * r^{(t/s)}$$

In order to compute the parameter parameter updates u_t at timestep t with Adam[1].

$$\begin{aligned}
m_t &\leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \\
v_t &\leftarrow \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot {g_t}^2 \\
\hat{m}_t &\leftarrow m_t / (1-\beta_1^t) \\
\hat{v}_t &\leftarrow v_t / (1-\beta_2^t) \\
u_t &\leftarrow \alpha_t \cdot \hat{m}_t / \left(\sqrt{\hat{v}_t + \bar{\varepsilon}} + \varepsilon \right) \\
S_t &\leftarrow (m_t, v_t).
\end{aligned}$$

Detailed per parameter optimisation settings of the optimisation are detailed in Table 1.

3 Motivation of the edge sampling strategy

Our edge sampling strategy is motivated by the fact that the gradients are located around the edges and textured areas of the object. Figure 1 provides a visualization of a snapshot of the location of the gradient for our method. For all parameters, the learning signal is located around edges and texture in the current position of the object in the optimisation (which can be seen most clearly for the rotation). Nevertheless, despite the apparent simplicity of the visualisation, analyzing the learning behaviour of the method is intricate. One factor that contributes to this is that there seems to be gradients around the edges that sent the optimisation in the wrong direction (as can be seen for the translation parameters in Figure 1). We attribute convergence despite these factors to higher order effects of the optimisation.

4 Backbone and convergence speed

The convergence speed of the method at pose inference time results from its raw speed in terms of optimisation steps per second and how much these steps can contribute to retrieving the correct pose. Without reparametrisation, using Instant NGP as a backbone for iNeRF[3] does not lead to good pose optimisation performance. The screw axis variant of our method follows iNeRF[3] with an Instant NGP backbone and its performance can be compared to the results for iNeRF with random sampling as reported in iNeRF[3]. Since both of these methods use random sampling and run on the same hardware, it seems that even though the Instant NGP backbone allows our method to process larger batches faster while running on the same hardware as iNeRF[3], it is only in combination with reparametrising that we reach similar accuracy (see Table 1).

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning*

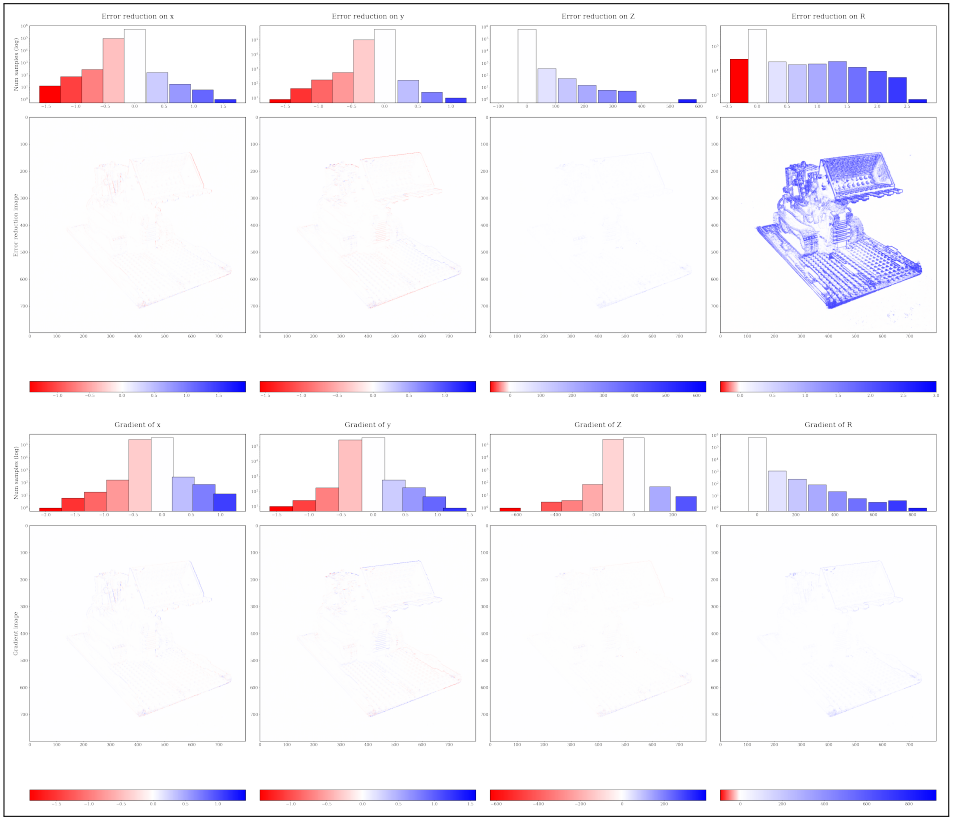

Figure 1: **Visualisation of the error correction per pixel induced by the gradients and their distribution.**

Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL http://arxiv.org/abs/1412.6980.

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4): 102:1–102:15, July 2022. doi: 10.1145/3528223.3530127.

[3] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330, 2021. doi: 10.1109/IROS51168.2021.9636708.

Parameter	Configuration	Setting
Scale	*Learning rate schedule*	Exponential decay
	Initial value (α_0)	0.01
	Transition steps (s)	100
	Decay rate (r)	0.8
	Optimisation algorithm	Adam[1]
	First moment decay rate (β_1)	0.9
	Second moment decay rate (β_2)	0.99
	Epsilon (ε)	1e-8
	Epsilon root ($\bar{\varepsilon}$)	0.0
	Initial estimate of the first moment (m_0)	0.0
	Initial estimate of the second moment (v_0)	0.0
Pixel translation	*Learning rate schedule*	Exponential decay
	Initial value (α_0)	0.01
	Transition steps (s)	100
	Decay rate (r)	0.8
	Optimisation algorithm	Adam[1]
	First moment decay rate (β_1)	0.9
	Second moment decay rate (β_2)	0.99
	Epsilon (ε)	1e-8
	Epsilon root ($\bar{\varepsilon}$)	0.0
	Initial estimate of the first moment (m_0)	0.0
	Initial estimate of the second moment (v_0)	0.0
	Additional aspects	The updates are scaled by 300
Rotation	*Learning rate schedule*	Exponential decay
	Initial value (α_0)	0.005
	Transition steps (s)	100
	Decay rate (r)	0.8
	Optimisation algorithm	Adam[1]
	First moment decay rate (β_1)	0.9
	Second moment decay rate (β_2)	0.99
	Epsilon (ε)	1e-8
	Epsilon root ($\bar{\varepsilon}$)	0.0
	Initial estimate of the first moment (m_0)	0.0
	Initial estimate of the second moment (v_0)	0.0

Table 1: **Detailed overview of the per parameter optimisation configuration for the pose inference.** Note that the only difference between the parameter optimisations are that we use a scaling operation for the translation optimisation and a different initial learning rate for the rotation optimisation. The settings reported in this table were used for all the experiments.

	Inference (s)	Rotation error $< 5^\circ$	Translation error < 0.02 Units
iNeRF	50 s	≤ 0.7	$\leq 0.7^*$
Ours (screw axis)	4.4 s	0.44	0.2
Ours (reparametrised)	4.4s	**0.79**	**0.76**

Table 2: The translation error metric reported for iNeRF is a loose upper bound to our translation error due to the larger error range used (if the meter unit reported in iNeRF corresponds to the COLMAP Unit retrieved for our scene).