





## Probabilistic volumetric speckle suppression in OCT using deep learning: supplement

**BHASKARA RAO CHINTADA**<sup>1,2,\*</sup>  **SEBASTIÁN RUIZ-LOPERA**<sup>1,3</sup>  
**RENÉ RESTREPO**<sup>4</sup>  **BRETT E. BOUMA**<sup>1,2,5</sup> **MARTIN VILLIGER**<sup>1,2</sup>   
**AND NÉSTOR URIBE-PATARROYO**<sup>1,2</sup> 

<sup>1</sup>Wellman Center for Photomedicine, Massachusetts General Hospital, Boston, MA 02114, USA

<sup>2</sup>Harvard Medical School, Boston, MA 02115, USA

<sup>3</sup>Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02142, USA

<sup>4</sup>Applied Optics Group, Universidad EAFIT, Carrera 49 # 7 Sur-50, Medellín, Colombia

<sup>5</sup>Institute of Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA 02142, USA

\*[bhintada@mgh.harvard.edu](mailto:bhintada@mgh.harvard.edu)

---

This supplement published with Optica Publishing Group on 3 July 2024 by The Authors under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) in the format provided by the authors and unedited. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Supplement DOI: <https://doi.org/10.6084/m9.figshare.26062651>

Parent Article DOI: <https://doi.org/10.1364/BOE.523716>

# Probabilistic volumetric speckle suppression in OCT using deep learning: supplemental document

The Tomographic Non-local-means despeckling (TNode) technique we previously published [1] for reducing speckle in Optical Coherence Tomography (OCT) is based on constructing a weighted similarity criterion that follows the statistics of speckle within a three-dimensional similarity window. Significant improvements have been introduced since the first implementation up to the present day regarding computational efficiency, despeckling performance, and implementation versatility. This document outlines the improvements made in the current implementation of TNode that will be publicly accessible after the review of this article.

## 1. IMPROVEMENTS IN COMPUTATIONAL EFFICIENCY

One of the most significant changes in the new implementation of TNode is its computational efficiency, which was greatly improved by implementing GPU-based processing. We extended the B-scan-wise processing to a subvolume-wise processing with a user-defined size, which can be tuned to optimize the memory overhead. Additionally, a wrapper function is now available to use multiple GPUs simultaneously.

The compounded probability of the similarity criterion is computed via summation of the logarithmic probabilities over the similarity window. In the original version TNode<sub>2018</sub>, this compounded probability is computed via convolution with a kernel using MATLAB's built-in function. In the updated version TNode<sub>2023</sub>, we use for loops to compute the convolution, which we found to perform faster given the relatively small similarity windows.

To illustrate the difference in computational efficiency, Table S1 displays the processing time of the TNode<sub>2018</sub> and TNode<sub>2023</sub> necessary to despeckle one retinal volume acquired with the ophthalmic system. The volume had dimensions  $400 \times 896 \times 960$  (depth samples  $\times$  A-lines per Bscan  $\times$  B-scans per volume) with two polarization detection channels. We processed it with two computer systems; a laptop with standard capabilities having an AMD Ryzen 9 processor and a 6 GB NVIDIA RTX 2060 GPU, and a server with advanced capabilities having an AMD Ryzen Threadripper PRO 5975WX and a 20 GB NVIDIA RTX A4500 GPU. The search and similarity windows were set to  $17 \times 17 \times 17 \text{ px}^3$  and  $7 \times 7 \times 7 \text{ px}^3$  respectively. For TNode<sub>2018</sub>, an A-line block size of 90 was used, which required a minimum of 28 GB of free RAM. For TNode<sub>2023</sub>, a subvolume block size of  $16 \times 32 \times 32 \text{ px}^3$  was used, which required at least 14.3 GB of free GPU memory.

It is worth noting that the processing time shown in Table S1 for the tomogram used in the example is significantly lower with TNode version 2023 compared to version 2018 by one order of magnitude making it more practical to use.

**Table S1. Comparison of the processing times of TNode version 2018 and 2023**

Computer system	TNode <sub>2018</sub>	TNode <sub>2023</sub>
Laptop	5136 min	655 min
Server	1996 min	230 min

### Ease of testing different $h_0$ and $h_1$ parameters

Tuning the hyperparameters  $h_0$  and  $h_1$  is important to obtain optimal results; they control the overall speckle reduction and regularize the speckle reduction as a function of signal-to-noise ratio (SNR), respectively, as explained in detailed in Ref. [1]. In order to speed up the manual tuning of  $h_0$  and  $h_1$ , TNode<sub>2023</sub> can receive multiple values of  $h_0$  and  $h_1$  at a time, which are used independently to despeckle the input tomogram in a single run, ultimately providing an output tomogram for each pair of input hyperparameters. We found this approach to be less time-consuming than making multiple individual runs.

## 2. IMPROVEMENTS IN DESPECKLING PERFORMANCE

### Apodization

We integrated new features that improve the performance of TNode. First, we added a new apodization kernel to pre-shape the similarity window, which can have a default shape such as unitary (equivalent to TNode<sub>2018</sub>), triangular, or Gaussian, or can be provided by the user as a vector. The apodization kernel is separable by construction.

### Additional compounding

Input tomograms in TNode<sub>2023</sub> can be five-dimensional. The similarity criterion is by default compounded along the entire 4th dimension. In practice, the tomograms from the two polarization detection channels of a system with polarization-diverse detection can be concatenated along the 4th dimension for compounding their probabilities. The 5th dimension is managed in the same way as the three spatial dimensions, therefore the similarity and search windows are now extended to be four-dimensional. Alternatively, if the similarity and search windows are 3D, then no compounding is performed along the 5th dimension. This feature can be used to process multiple independent tomograms in parallel in a single run.

### Pruning

We added a pruning functionality to TNode<sub>2023</sub> which was introduced in the past and shown to improve the performance of non-local means denoising without increasing processing time significantly [2]. Pruning consists of setting to zero the weights within a user-defined percentile value, thus maintaining only the voxels with significant similarity. This helps to avoid the over-filtering caused by the presence of a large number of voxels with low similarity.

### Definition of noise floor level

The hyperparameter  $h_1$  regulates the speckle reduction as a function of SNR. The computation of the SNR requires the user to input the level of the noise in the absence of a signal, commonly known as noise floor, which in TNode<sub>2018</sub> consists of a scalar value. However, it is known that the noise floor can be depth-dependent, therefore, in TNode<sub>2023</sub>, the input noise floor can be a vector with the same number of depth samples per A-line as the input tomogram to account for a depth-dependent noise floor.

### Visual comparison of TNode versions

Figure S1 shows a comparison of speckle suppression with TNode<sub>2018</sub> and TNode<sub>2023</sub> for an en-face image of the fovea of an eye acquired with the ophthalmic system. The images display capillaries that are more clearly observed in the image with TNode<sub>2023</sub>, a benefit that we mostly attribute to the pruning functionality in this particular example.

## 3. SPECKLE REDUCTION IN RETINAL NERVE FIBER LAYER USING DL-TNODE-3D

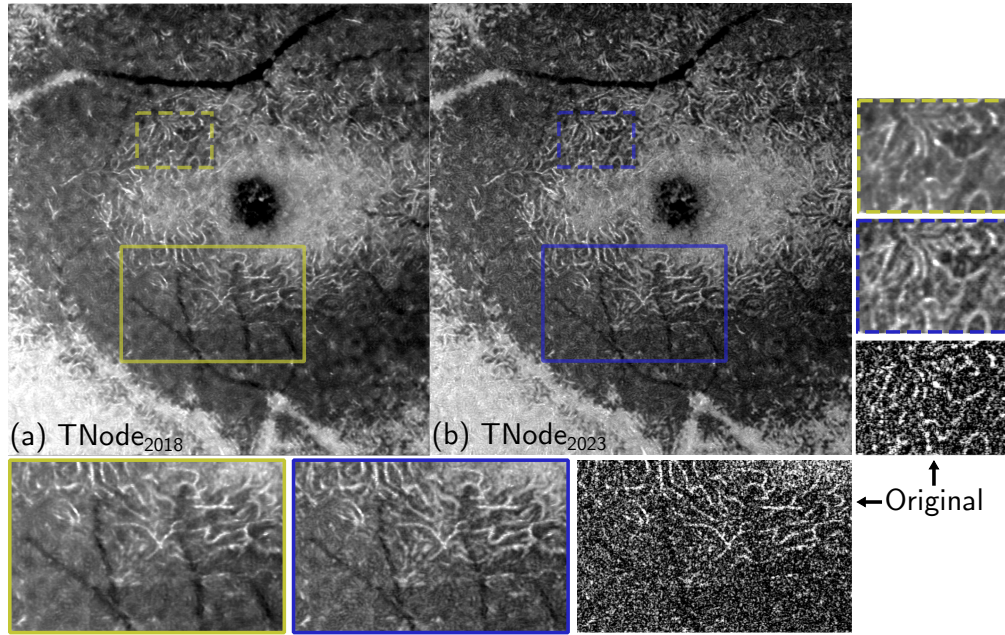
Figure S2 shows speckle reduction in retinal nerve fiber layer for the same volume shown in Figure.3 of the manuscript. DL-TNode-3D enhanced the contrast and nerve fiber bundles became much clearer and easier to identify, similar to ground-truth TNode.

## 4. DL-TNODE-3D TRAINING

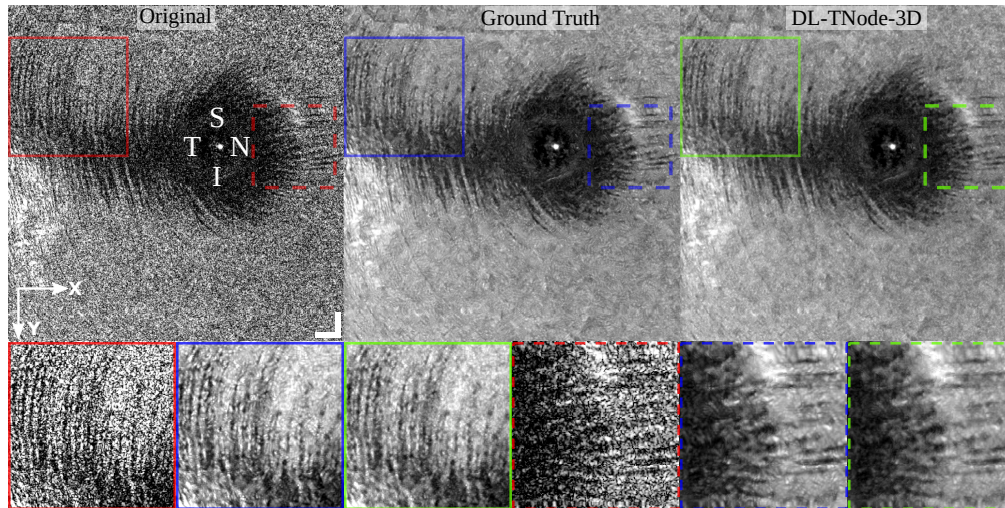
Typical training loss plots that we obtained during DL-TNode-3D training are plotted in Figure. S3. DL-TNode-3D model trained on the three OCT systems showed similar stable learning curves, where the discriminator loss exponentially reduced and converged to  $2\ln 2$ . And the generator loss, showed similar behaviour with epochs. We selected model weights corresponding to the discriminator loss  $2\ln 2$  with lower generator loss for inference in the later stages.

## REFERENCES

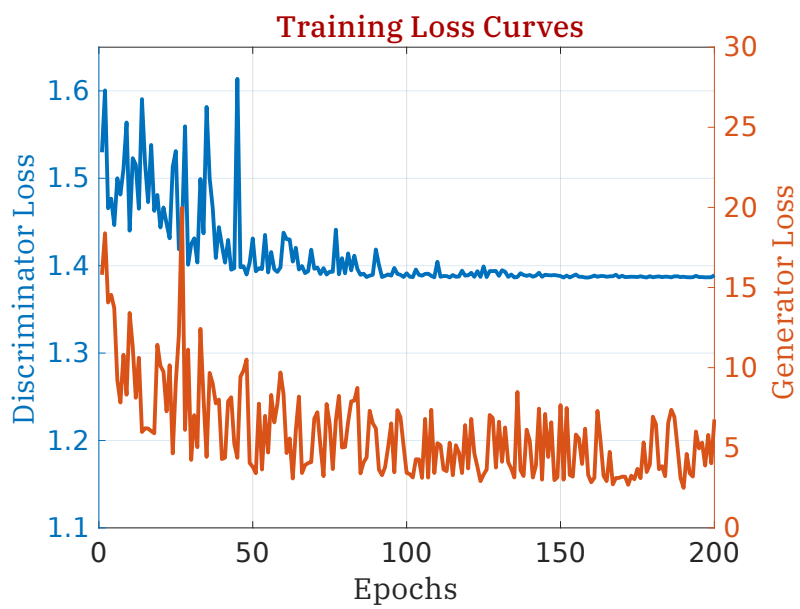
1. C. Cuartas-Vélez, R. Restrepo, B. E. Bouma, and N. Uribe-Patarroyo, "Volumetric non-local-means based speckle reduction for optical coherence tomography," *Biomed. Opt. Express* **9**, 3354–3372 (2018).
2. S. Ghosh, A. K. Mandal, and K. N. Chaudhury, "Pruned non-local means," *IET Image Process.* **11**, 317–323 (2017).



**Fig. S1.** Comparison of speckle reduction with TNode in an en-face image of the fovea of a healthy volunteer using (a) the original version TNode<sub>2018</sub> and (b) the updated version TNode<sub>2023</sub>. The capillaries in the densely populated areas that are shown in the insets are better preserved with TNode<sub>2023</sub>, furthermore, the contrast with respect to the background is slightly higher with TNode<sub>2023</sub>. We also included insets from the original tomogram.



**Fig. S2.** En-face views of retinal nerve fiber layer before and after despeckling using TNode (i.e. Ground Truth) and DL-TNode-3D. DL-TNode-3D produces OCT volumes close to the ground truth without any visible artifacts,  $y, z$  is the depth and  $x(y)$  is the fast- (slow-) scan axis direction. Directions: S: Superior, T: Temporal, N: Nasal, I: Inferior. Scale bars = 0.5 mm.



**Fig. S3.** The typical generator and discriminator loss curves of DL-TNode-3D training.