Regularized Coordinate-based Neural Representation Learning for Optical Tomography

Renhao Liu

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds

Part of the Engineering Commons

Recommended Citation

This Thesis is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in McKelvey School of Engineering Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.
Regularized Coordinate-based Neural Representation Learning for Optical Tomography

by

Renhao Liu

A thesis presented to the McKelvey School of Engineering of Washington University in partial fulfillment of the requirements for the degree of

Master of Science

August 2021
Saint Louis, Missouri
List of Tables

3.1 Experiment parameters .......................................................... 20
4.1 Neural representation size and traditional reconstruction size comparison .. 29
## List of Figures

1.1 Regularized neural representation learning framework ............................................. 4
2.1 DnCNN structure ........................................................................................................... 7
2.2 DnCNN inference example .......................................................................................... 8
2.3 Illustration of IDT, mIDT, and aIDT ........................................................................ 10
3.1 Block-wise learning flowchart .................................................................................... 14
3.2 Forward model decomposition simulation ................................................................ 16
3.3 Transfer function size simulation .............................................................................. 18
4.1 3D visualization of algae structure .......................................................................... 22
4.2 Clear axial sectioning effect on IDT Algae data set ............................................... 23
4.3 Super-resolution at x,y axis ......................................................................................... 24
4.4 Axial sectioning effect on aIDT data set .................................................................. 25
4.5 Interpolation on z axis ................................................................................................. 26
4.6 High-quality reconstruction with mIDT on C. elegans data set ............................... 28
Acknowledgments

I would like to express my sincere gratitude to professor Ulugbek Kamilov for his continuous support from the beginning of my undergraduate study to the end of my master’s research. He inspired me to pursue a career in academic and generously offered me opportunities and resources to learn and grow.

Secondly, I would like to show my gratitude to Ph.D. candidate Yu Sun for his dedicated support in writing this thesis. He is not only a patient instructor in research, but also a good friend in life.

I am also grateful to Professor Kunal Agrawal for challenging me with advanced algorithms, Professor Dennis Cosgrove for helping me become a better engineer, and Professor Ning Zhang for introducing me into research. Their guidance during my undergraduate study are crucial for my early stage development in Computer Science.

Finally, I want to thank my thesis committee members Professor Jason Trobaugh and Ju Tao for their thoughtful comments and suggestions on this thesis. I also want to thank the entire Computational Imaging Lab and everyone else who have supported me and helped me.

Renhao Liu

Washington University in Saint Louis
August 2021
Dedicated to my parents.
ABSTRACT OF THE THESIS

Regularized Coordinate-based Neural Representation Learning for Optical Tomography

by

Renhao Liu

Master of Science in Computer Science

Washington University in St. Louis, August 2021

Research Advisor: Ulugbek Kamilov

Neural representation learning recently shows outstanding performance in several computer vision tasks. In this thesis, we propose a novel self-supervised neural represented reconstruction method for optical tomography. Our method uses a multilayer perceptron (MLP) network to represent the target sample without the need for any ground truth or training data. The MLP weights serve as a latent representation of the target object. Any desired permittivity information can be inferred by querying the neural network within the sample domain. We also investigate applying regularization to implicitly restrict the manifold of MLP for better performance. Our experiments produce low artifacts results with strong sectioning effects on three optical tomography modalities: fully sampled intensity diffraction tomography (IDT), multiplexed IDT (mIDT), and annular IDT (aIDT). Furthermore, since our model implicitly represents a 3D volume, it enables upsampling to any scale, despite only being optimized on discrete measurements. In addition, the disk space required to store the latent representation is much smaller than the traditional method.
Chapter 1

Introduction

1.1 Inverse problem for computational imaging

Computational imaging reconstructs samples from indirect measurement in the form below:

\[ y = A(x) + \epsilon \]  

(1.1)

Where \( y \in \mathbb{R}^p \) is the measurement, \( A \) is an approximate transformation function (TF) or forward model for the sample \( x \in \mathbb{R}^q \), and \( \epsilon \) represents noise. A common reconstruction strategy is to formulate an optimization problem of form

\[ \arg\min_x ||A(x) - y||_n^n + \lambda \mathcal{R}(x) \]  

(1.2)

where \( ||A(x) - y||_n^n \) is the data fidelity term with \( l_n \) norm, and \( \mathcal{R} \) is a regularization prior or regularizer, which is controlled by parameter \( \lambda \geq 0 \). The accuracy of \( A \) and choice of \( \mathcal{R} \) together decide the reconstruction quality in many applications.
1.2 Regularized neural representation learning

In this thesis, we propose a novel optimization framework Regularized Neural Representation Learning. Our method optimizes a set of latent values $\Theta \in \mathbb{R}^k$ that represent $x$.

$$\arg\min_{\Theta} \left|\left|A(F(\Theta)) - y\right|\right|_n^n + \lambda R(F(\Theta)) \quad (1.3)$$

Where $F \in \mathbb{R}^k \rightarrow \mathbb{R}^q$ maps the latent representation $\Theta$ to $x$, where $\Theta$ is the weights in a multilayer perceptron (MLP) neural network that maps spacial coordinates $p = (x, y, z)$ to information at that coordinate.

$$F(\Theta, p) = x_p \in x \quad (1.4)$$

The regularizer $R$ has a significant impact on the reconstruction quality. Traditional methods include Total-Variation (TV) [1,2] and Tikhonov [3] regularization. A modern framework named plug-and-play priors (PnP) [4] introduces deep image denoisers into computational imaging optimization problems [5–9]. In particular, denoising convolutional neural networks (DnCNN) [10] shows outstanding performance in various imaging tasks [11, 12]. Another modern framework regularization by denoising (RED) [13] also achieves excellent results by adding the noise residual to the optimization gradient [14,15]. An earlier work SIMBA [16] applies DnCNN to optical tomography under RED framework and demonstrates its effectiveness in removing reconstruction artifacts comparing to the traditional Tikhonov method.

Inspired by previous research, our framework explicitly penalizes the $l2$–norm of predicted noise for better reconstruction quality:

$$\arg\min_{\Theta} \left|\left|A(F(\Theta)) - y\right|\right|_n^n + \lambda \left|\left|\mathcal{D}(F(\Theta)) - F(\Theta)\right|\right|_2^2 \quad (1.5)$$

where $\mathcal{D}(F(\Theta))$ is the denoised output from DnCNN.
Figure 1.1 illustrates our framework with DnCNN regularizer. The MLP neural latent representation is optimized using only target measurements and a pre-trained DnCNN network without the need for any other experimental data. This self-supervised learning feature enables our framework to reconstruct optical tomography samples whose ground truth is difficult to acquire.

1.3 Intensity diffraction tomography

We demonstrate the effectiveness of Neural Represented Learning on intensity diffraction tomography (IDT) [17], its variants multiplexed IDT (mIDT) [18] and annular IDT (aIDT) [19]. IDT is an optical tomography method that reconstructs phase and absorption permittivity information. It illuminates a sample from multiple angles and measures intensity only information using a commercial microscope. Different IDT modalities differ from illumination methods or pattern. For each illumination pattern, IDTs approximate light diffraction per depth level using a linear model.

The ability to reconstruct phase permittivity contrast helps IDT observe samples without staining, which sometimes may damage the sample or alter its behavior [20, 21]. Comparing with phase-sensitive imaging methods that directly measure phase information [22–24], IDT features a simpler experiment setup and a faster sample rate. The classic IDT uses an LED light array and illuminates the sample one LED at a time. This method requires up to 89 measurements using 10× microscope objectives (MO) and up to 697 using 40× MO. AIDT reduces the number of measurements and increases sample rate by replacing the LED array with 24 LEDs placed on an annular ring. MIDT uses the same LED array as conventional IDT but proposes lighting up multiple LEDs together while fewer LEDs in total to reduce the number of samples required.

This thesis validates the regularized neural representation learning on an algae data set under 10× MO on IDT, a diatom data set under 40× MO on aIDT, and a C. elegans data set also under 40× MO on mIDT. We also downsample the aIDT data set by 2 to show our framework continuously represents the sample and reconstructs at any scale.
Figure 1.1: Regularized Neural Representation Learning for optical tomography framework illustration. The multilayer perceptron (MLP) neural network maps coordinates $p$ to phase (ph) and absorption (ab) permittivity information at the coordinate. Querying all coordinates in a domain infers the reconstruction $x$. Minimizing the data fidelity loss and regularization loss on $x$ optimizes the MLP network.
1.4 Contribution and thesis outline

- We propose a novel unsupervised optical tomography framework Regularized Neural Represented Learning for computational microscopy that enables continuous representation of a target sample.
- Our algorithm uses modern deep denoisers for regularizing the imaging inverse problem.
- We validate our algorithm on experimentally collected IDT data.
- We achieve high-quality reconstructions with significantly reduced memory usage.

In Chapter 2 we provide more background information on regularization, neural representation learning, and IDT. Chapter 3 describes block-wise learning algorithm and provide details on experiment settings. In Chapter 4, we validate our method on IDT, aIDT, and mIDT and present experimental results. And finally, Chapter 5 concludes this thesis.
Chapter 2

Background

In this chapter, we provide more related research on regularization, neural representation learning, and IDT. We begin by introducing modern reconstruction regularization methods and describe the DnCNN image denoiser used in this thesis. Then we summary related research on neural representation learning. Lastly, we mathematically describe different IDT forward models and traditional reconstruction method using Tikhonov regularization.

2.1 Regularization with modern regularizers

Integrating modern image denoising priors into imaging optimization problems becomes popular with recently developed plug-and-play (PnP) [4] and RED [13] frameworks. PnP generalizes the proximal operator in traditional regularization algorithms like FISTA [25] with deep image denoisers. RED proposes another approach by directly adding the predicted noise to the gradient:

$$R(x) = \frac{1}{2} x^T(x - D(x)) \quad (2.1)$$

$$\nabla R(x) = x - D(x) \quad (2.2)$$

where $x$ is the reconstruction, and $D(x)$ is denoised results. Both frameworks show promising results in various imaging tasks and theoretical converging properties are analyzed under certain assumptions [8, 12, 26–29].
Figure 2.1: DnCNN model structure. DnCNN predicts noise (residual) at pixel level using a series convolution (conv) and batch normalization (BN) layers, then remove it from input to restore a high-quality image.

One widely used image denoiser is DnCNN [10]. A typical structure of DnCNN is shown in 2.1, and image denoising example is shown in Figure 2.2. Given a noisy image $x$, DnCNN predicts noise value (residual) at each pixel then removes the noise from input and outputs a denoised image $D(x)$.

### 2.2 Neural latent representation

In most conventional imaging tasks, a reconstruction sample is represented by a set of pixel values, each corresponds to a point in voxel or frequency space. Such methods attempt to describe a continuous world using dense but discrete values. In contrast, neural representation embeds a continuous space in the weights of a neural network. A typical neural representation network maps the coordinates $p = (x, y, z)$ to physical information at the input coordinate. Many studies use neural representation learning to represent 3D objects [30–36]. A recent work *Neural Radiance Fields (NeRF)* [30] takes coordinates and observation direction of a 3D object as input and optimizes the neural network to generate novel views using ray-tracing techniques. Neural representation learning also shows promising results in other applications, such as image [37] or video [38–41] representation, re-lighting [42–45],
Figure 2.2: DnCNN denoising demonstration. A DnCNN model trained on noise level $\sigma = 15$ removes additive white gaussian noise and restores high-quality images.

and computed tomography (CT) [46]. Most current neural representation applications either have direct measurements or apply ray-tracing forward models. Our work can be viewed as a neural representation generalization from ray optics to wave optics where the light’s wave properties become crucial in reconstruction.

Previous research on neural representation learning shows that applying positional encoding to input coordinates before passing them to MLP network helps the neural network recognize high-frequency information [47]. Positional encoding maps low-frequency coordinates to a high-frequency domain. Tancik et al. [48] discusses positional encoding $\gamma$ in the form of

$$\gamma(p) = [\cos(2\pi B p), \sin(2\pi B p)]^T$$ \hspace{1cm} (2.3)

where $B$ is a Gaussian random matrix with each element drawn from $\mathcal{N}(0, \sigma^2)$. The paper shows if positional encoding ($\sigma$) is too small, the model fails to converge on high-frequency data, while if $\sigma$ is too large, the model overfits input and fails to continuously represent the target. In this thesis, we adopt another approach similar to the one used in NeRF [30]

$$\gamma(p) = (\sin(2^0 p), \cos(2^0 p), \sin(2^1 p), \cos(2^1 p), ..., \sin(2^{L-1} p), \cos(2^{L-1} p))$$ \hspace{1cm} (2.4)

Where $L$ is a parameter for number of encoding layers.
2.3 IDT forward model

An object’s scattering potential \( v \) can be described by the permittivity difference between the sample \( \epsilon \) and immersion medium \( \epsilon_0 \):

\[
v(p) = \frac{1}{4\pi} k_0^2 \Delta \epsilon(p)
\]  

(2.5)

where \( \Delta \epsilon = \Delta \epsilon_{Re} + \Delta \epsilon_{Im} = \epsilon - \epsilon_0 \). The real part \( \Delta \epsilon_{Re} \) describes the phase effect and the imaginary part \( \Delta \epsilon_{Im} \) describes the absorption. \( p \) is a point in three dimensional (3D) space \( \Omega \). \( k_0 = 2\pi/\lambda \) is the illumination wave number in free space, and \( \lambda \) is the wavelength of light. The total light field is computed as the sum of constant background input source \( u_{in} \) and diffracted light in the whole field under the first Born approximation [49]:

\[
u(p) = u_{in}(p) + \int_{\Omega} g(p - p')v(p')u_{in}(p') \, dp', \quad p' \in \Omega,
\]  

(2.6)

where \( g \) stands for the Green’s function:

\[
g(p) = \frac{e^{i\sqrt{\epsilon_0 k_0}||p||^2}}{||p||^2}
\]  

(2.7)

Assuming each LED is a point light source [50–53], the measurement intensity is computed by:

\[
I(p) = |u(p) \ast h(p)|^2
\]  

(2.8)

where \( h \) is coherent point spread function and \( \ast \) stands for convolution. The equation can be expanded into four terms:

\[
I(p) = I_{ii}(p) + I_{is}(p) + I_{si}(p) + I_{ss}(p)
\]  

(2.9)
Figure 2.3: Experiment setup, transfer function examples and measurements for IDT, mIDT, and aIDT.
\( I_{ii} \) is the constant background intensity. \( I_{ss} \) can be neglected assuming small permittivity contrast. \( I_s = I_{si}^* \) (\( * \) stands for complex conjugate) reflect the majority of permittivity information. Thus the measured intensity can be linearized as following:

\[
\tilde{I} = \tilde{I}_{ii} + \int [H_{re}(z)\tilde{\Delta \epsilon}_{re}(z) + H_{im}(z)\tilde{\Delta \epsilon}_{im}(z)] \, dz
\]  

(2.10)

where \( \tilde{\cdot} \) means Fourier transform, \( H_{re} \) is the phase transfer function (TF) and \( H_{im} \) is the absorption transfer function. Our forward model \( A \) can be obtained by discretizing Equation 2.10:

\[
A(x, l) = F^{-1} \sum_z H_{re}(z, l) \cdot \tilde{x}_{re}(z) + H_{im}(z, l) \cdot \tilde{x}_{im}(z)
\]

(2.11)

where \( F^{-1} \) represents inverse Fourier transform, \( l \) is the LED light index, and \( x_{re/im}(z) \) is the real or imaginary (\( \Delta \epsilon_{re} \) or \( \Delta \epsilon_{im} \)) part of \( x \) at slice \( z \). An illustration of IDT, aIDT, mIDT setups, transfer function example, and measurement example can be found in Figure 2.3.

2.3.1 MIDT forward model

MIDT illuminates the object with multiple lights at the same time [18]. Suppose a multiplexed measurement contains a set of LED lights \( L = \{l_1, l_2, ..., l_n\} \). Its corresponding transfer function is the sum of individual light transfer functions:

\[
H_{re}(z, L) = \sum_{l \in L} H_{re}(z, l)
\]

(2.12)

\[
H_{im}(z, L) = \sum_{l \in L} H_{im}(z, l)
\]

(2.13)

\[
A(x, L) = F^{-1} \sum_z H_{re}(z, L) \cdot \tilde{x}_{re}(z) + H_{im}(z, L) \cdot \tilde{x}_{im}(z)
\]

(2.14)

Light choice scheme and more specific information can be found in the original mIDT paper [18].
2.4 Traditional solution to IDT problems

Traditional IDT, aIDT, and mIDT reconstruct sample slice-wise using a closed form solution with Tikhonov regularization:

\[
x_{re}(z) = \mathcal{F}^{-1}\left\{ \frac{1}{N} \left[ \sum_{l=1}^{L} |H_{im}(z,l)|^2 + \tau_{im} \right] \cdot \left( \sum_{l=1}^{L} H_{re}^*(z,l) \cdot \hat{l}_l \right) \\
- \left( \sum_{l=1}^{L} H_{re}^*(z,l) \cdot H_{im}(z,l) \right) \cdot \left( \sum_{l=1}^{L} H_{im}^*(z,l) \cdot \hat{l}_l \right) \right\}
\]  

(2.15)

\[
x_{im}(z) = \mathcal{F}^{-1}\left\{ \frac{1}{N} \left[ \sum_{l=1}^{L} |H_{re}(z,l)|^2 + \tau_{re} \right] \cdot \left( \sum_{l=1}^{L} H_{re}^*(z,l) \cdot \hat{l}_l \right) \\
- \left( \sum_{l=1}^{L} H_{im}^*(z,l) \cdot H_{re}(z,l) \right) \cdot \left( \sum_{l=1}^{L} H_{re}^*(z,l) \cdot \hat{l}_l \right) \right\}
\]  

(2.16)

\[
N = \left( \sum_{l=1}^{L} |H_{im}(z,l)|^2 + \tau_{im} \right) \cdot \left( \sum_{l=1}^{L} |H_{re}(z,l)|^2 + \tau_{re} \right) \\
- \left( \sum_{l=1}^{L} H_{re}^*(z,l) \cdot H_{im}(z,l) \right) \cdot \left( \sum_{l=1}^{L} H_{im}^*(z,l) \cdot H_{re}(z,l) \right)
\]  

(2.17)

where $H^*$ represents complex conjugate and $\tau_{re}, \tau_{im}$ are the Tikhonov regularization parameters on phase and absorption respectively. Larger $\tau_{re}$ and $\tau_{im}$ constraint more energy on reconstruction results and reduces artifacts. The appropriate regularization parameter is determined visually for the lack of ground truth.
Chapter 3

Method

In this chapter, we talk about specific methods used in this thesis. In particular, we introduce a block-wise learning algorithm that significantly reduces memory usage for optimizing neural representation. In addition, we list the specific loss function, optimization parameters for each modality and experiment, and other technical details.

3.1 Block-wise learning

In this section, we propose a block-wise learning algorithm that reduces optimization memory usage. The latent representation optimization memory requirement scales linearly to the number of nodes in the neural network. A common way to reduce memory is dividing a large sample domain into small batches then optimize each batch individually [26]. In this thesis, we vertically divide a sample space into multiple predefined blocks, then optimize one block at a time. The final algorithm is described in Algorithm 1, and Figure 3.1.

Our algorithm makes three approximations to the original algorithm: dividing a sample space into blocks, introducing overlapping margin between blocks, and reducing the size of the transfer function. In the following subsections, we show those alternations are close approximations to the original problem. Thus they have little effects on final reconstruction results.
Figure 3.1: Block-wise learning flowchart. A block $x_i$ in the sample domain is optimized assuming all other blocks are fixed. The reconstruction target $y_i$ is computed by adding the global residual term and the block’s reconstruction result at the previous iteration.

Algorithm 1: Optimized block-wise algorithm

\[
\begin{align*}
    &y_{res} = y; \\
    &T_i := 0, \forall i; \\
    \text{for } i = 1, 2, ..., m \text{ do} \\
    &\quad y_i = y_{res} + T_i; \\
    &\quad \Theta = \Theta - \alpha \nabla L_\Theta(x_i, y_i); \\
    &\quad T_i := A(mask(x_i)); \\
    &\quad y_{res} = y_{res} - T_i \\
\text{end}
\end{align*}
\]

Here, A sample $x$ is divided into $m$ blocks $x_1$ to $x_m$. Each $x_i$ is obtained by querying the neural representation with all coordinates within $x_i$. We assume the network has enough weights to express the entire sample domain. During optimization, we store and update $y_{res}$ and $T_i \forall i$ as intermediate variables. $T_i = A(mask(x_i))$ is obtained when computing the gradient of the loss function $\nabla L_\Theta(x_i, y_i)$ at no extra cost. Please follow subsection 3.1.1 to 3.1.4 for more explanations.
3.1.1 Dividing sample space into blocks

We first present a naive algorithm that optimizes on an arbitrary cubical subset in a sample domain. Let $x$ be the set of all sample points in the 3D domain, $x'$ be an arbitrary subset of $x$, $\overline{x}' = x - x'$ be its complement. Since the IDT forward model is linear, we use its distributive property to separate $x'$ and $\overline{x}'$:

$$A(x) = A(x') + A(\overline{x}')$$
$$A(x') = A(x) - A(\overline{x}')$$

At each iteration, we assume $\overline{x}'$ stays the same, and optimize only on $x'$. Although $x'$ can be an arbitrary set for the forward model, we choose a cubical region for the regularizer. A naive algorithm for this step is shown in Algorithm 2. Since the transfer functions are on the Fourier domain, $x'$ and $\overline{x}$ are padded to the size of transfer functions.

Algorithm 2: Naive block-wise algorithm

 initialization;
 while not converge do
  $x' =$ a random block in $x$ ;  
  $\overline{x}' = x - x'$ ; 
  $y' = y - A(F_\Theta(\overline{x}'))$; 
  $\Theta = \Theta - \lambda \nabla L_\Theta(x', y')$;
end

3.1.2 Overlapping blocks

The previous algorithm is highly inefficient as it requires recomputing $A(\overline{x}')$. We eliminate the operation by pre-defining block locations and reusing computation results from previous iterations. We divide a sample domain evenly into vertical regions with a small overlapping margin. For instance, if a sample has width, depth, and height $(x, y, z)$ of $800 \times 800 \times 40$, we divide it into $m = 16$ blocks of $250 \times 250 \times 40$ covering the entire domain. A mask scales the overlapping margins by $\frac{1}{2}$ or $\frac{1}{4}$ depending on the number of blocks containing the region.

15
Figure 3.2: This figure shows a simulated measurement of 3 points. The computation can be divided into the sum of two parts, each contains one and a half points (in intensity). This figure also shows that the transfer function is horizontally invariant.

For example, if block $x_i$ and $x_j$ has overlapping region $x_{ij}$. Then,

$$A(x_i \cup x_j) = A(x_i) + A(x_j) - A(x_{ij})$$

$$= A(x_i - \frac{1}{2}x_{ij}) + A(x_j - \frac{1}{2}x_{ij})$$

which is equivalent to having $\frac{1}{2}$ mask on $x_{ij}$. Again, the data fidelity term does not require an overlapping region between blocks. However, without overlapping, the edge between blocks won’t be regularized. The regularization applies on $x$ before masking, thus the $\frac{1}{2}$ or $\frac{1}{4}$ mask doesn’t influence regularization results. A demonstration of this is shown in Figure 3.2. This mask mentioned is shown as mask function in Algorithm 1.

### 3.1.3 Reusing computation

Pre-defining blocks allows us to reuse previous computations and compute the complement term by checking a dictionary.

We store and update the global residual term $y_{res} = y - \sum_i A(mask(x_i))$ and predicted masked measurements $T_i := A(mask(x_i)) \forall i$ in every iteration. Then the measurement for
block $k$ can be represented as:

$$y_k = y - \sum_{i \neq k} A(\text{mask}(x_i)) = y_{res} + A(\text{mask}(x_k))$$

At each iteration for block $k$, a new $A(\text{mask}(x_k))$ is computed as an intermediate step for gradient, which is close to $A(\text{mask}(x_k))$ after optimization given small enough step size. Thus this is a close approximation to the original problem.

### 3.1.4 Reducing transfer function size

Finally, since we decreased block size, a global transfer function might not be necessary. According to Equations 2.6 and 2.7, the effect of a point in an object decays as distance increases. Thus a smaller transfer function might be sufficient for reconstruction purposes. Also, since the transfer function only considers the relative distance in the horizontal axis, all blocks share the same transfer function. Figure 3.3. demonstrates a simulation for a one-point sample and shows that a smaller transfer function is sufficient for local measurement prediction.

### 3.2 Technical details

In this section, we present the final formulation of our loss function and list the parameters used for experiments.

#### 3.2.1 Loss function

In equation 3.1, we presented regularized neural representation learning. As DnCNN only provides regularization on $x,y$ axis while we aim to reconstruct a 3D volume, we add a vertical consistency term into the loss function:
(a) Simulation of a one dot sample under different transfer function sizes.

(b) Similarity between smaller transfer function and full-scale (700) transfer function.

Figure 3.3: Transfer function size simulation. When the target is small, a smaller transfer function closely approximates the full-scale transfer function.
arg min \[ \sum_{\Theta} ||A(F(\Theta)) - y||_n^n + \alpha ||D(F(\Theta)) - F(\Theta)||_2^2 + \beta \sum_z |F_z(\Theta) - F_{z-1}(\Theta)| \] (3.1)

3.2.2 Multilayer perceptron specification

First, we separate the positional encoding \( L \) into \( L_{x,y} \) and \( L_z \), since our samples are thin in \( z \) axis and flat in \( x,y \) axis. Extra positional encoding brings no more information into the system.

\[
\gamma(p) = (\sin(2^0 p_x), \cos(2^0 p_x), \sin(2^1 p_x), \cos(2^1 p_x), \ldots, \sin(2^{L_{x,y}-1} p_x), \cos(2^{L_{x,y}-1} p_x), \\
(\sin(2^0 p_y), \cos(2^0 p_y), \sin(2^1 p_y), \cos(2^1 p_y), \ldots, \sin(2^{L_{x,y}-1} p_y), \cos(2^{L_{x,y}-1} p_y)), \\
(\sin(2^0 p_z), \cos(2^0 p_z), \sin(2^1 p_z), \cos(2^1 p_z), \ldots, \sin(2^{L_z-1} p_z), \cos(2^{L_z-1} p_z))
\] (3.2)

Secondly, we add a skip connection to the middle of the network as suggested by [30,34]. An example is shown in Figure 1.1.

3.2.3 Details and parameters

We pre-train the 8-layers DnCNN denoisers at noise level \( \sigma \) on nature images. Once the DnCNN is trained, it stays the same for all optimization tasks. During optimization, the reconstructions are normalized to a proper range for DnCNN and de-normalized for computing gradients. Specific parameters for each experiment in this thesis are listed in Table 3.1. Our experiments are conducted on two Nvidia RTX 3090 GPUs.
<table>
<thead>
<tr>
<th></th>
<th>IDT</th>
<th>aIDT</th>
<th>aIDT /2</th>
<th>mIDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total $(x, y, z) \mu m$</td>
<td>$1024 \times 1024 \times 41$</td>
<td>$700 \times 700 \times 53$</td>
<td>$350 \times 350 \times 53$</td>
<td>$800 \times 800 \times 41$</td>
</tr>
<tr>
<td>Z range $\mu m$</td>
<td>-30 to 50</td>
<td>-10 to 16</td>
<td>-10 to 16</td>
<td>-10 to 10</td>
</tr>
<tr>
<td>TF size</td>
<td>$256 \times 256 \times 41$</td>
<td>$384 \times 384 \times 53$</td>
<td>$256 \times 256 \times 53$</td>
<td>$256 \times 256 \times 41$</td>
</tr>
<tr>
<td>$L_{xy}$</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>$L_z$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Network depth</td>
<td>10</td>
<td>14</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Nodes per layer</td>
<td>256</td>
<td>150</td>
<td>256</td>
<td>200</td>
</tr>
<tr>
<td>skip layer</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>DnCNN $\sigma$</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$1e-12$</td>
<td>$1.2e-16$</td>
<td>$1e-07$</td>
<td>$1e-4$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$1e-06$</td>
<td>$7e-8$</td>
<td>$7e-08$</td>
<td>$3e-7$</td>
</tr>
</tbody>
</table>

Table 3.1: Experiment parameters
Chapter 4

Experiments

In this section, we demonstrate our result on three optical tomography modalities: conventional Intensity Diffraction Tomography, annual Intensity Diffraction Tomography, and multiplexed Intensity Diffraction Tomography. We compare our result with traditional reconstruction methods using Tikhonov regularization, which suffers from over smoothing and artifacts. Our algorithm shows better axial sectioning effect and enables arbitrary scaling by leveraging DnCNN regularization and latent representation. In the IDT Algae data set, our results clearly identify the axial location of reconstructed algae strands. In the aIDT Diatom data sets, in addition to the clear axial sectioning effect, we show our ability to continuously represent the sample by downsampling the measurement and upscaling the reconstruction. Finally, in the mIDT data set, we demonstrate how the combination of continuous representation and deep regularization make our algorithm better reconstructing granular structures in a C. elegans worm.

4.1 Strong axial sectioning effect with IDT

The proposed algorithm shows a strong axial sectioning effect with Intensity Diffraction Tomography on an algae data set. This data set is sampled using red ($\lambda = 630nm$) light under $10\times$ objective MO with $NA = 0.25$. The sample is placed 79mm below the LED array, and the background index is assumed close to water (1.33). The reconstruction uses 89 brightfield images [17].
Figure 4.1: 3D visualization of algae structure. 11 algae strands are identified from the reconstruction slices, the full scale reconstruction has size $666 \times 666 \times 80 \mu m$.

Figure 4.1 visualizes the reconstructed 3D structure. Our neural representation is optimized at every $2 \mu m$ on $z$ axis, while the 3D model is generated by querying our result at every $1 \mu m$. A total of 11 Algae strands are identified and marked in Figure 4.2 where we compare our results and traditional results. Our reconstruction shows clear axial sectioning effects. At $z = -8 \mu m$, our algorithm clearly shows algae 1 and 2 while traditional reconstruction displays noisy background artifacts from other algae. Moving forward to $z = 10 \mu m$, in our results algae 2 fades into the background, and algae 3-6 appears in the view. In traditional result, algae 2 is still barely visible and algae 7, 9, and 10 (marked in $z = 26 \mu m$) also shows up. Finally, at $z = 26 \mu m$, our reconstruction shows the end of algae 5, and algae 7-10, while the traditional reconstruction result still have algae 3,4, and 6 visible. Also, Algae 11 has not yet become visible in our reconstruction while is already visible in the traditional result.
Figure 4.2: The IDT Algae data set reconstruction slices at $z = -8 \mu m, 10 \mu m, 26 \mu m$. 11 algae strands are identified and marked in the figure. Our results show a clear axial sectioning effect, comparing to traditional reconstructions.

4.2 Continuous representation with aIDT

In this section, we show our algorithm continuous represents a diatom sample with a strong axial sectioning effect using aIDT modality. Besides scaling up the resolution, we also downsample the measurement by averaging every $2 \times 2$ pixel block into one pixel and compare the reconstruction to results without downsampling. We assume the pixel size of the camera is $2 \times$ the original size and keep other conditions fixed when generating the downsampling transfer function.

In Figure 4.3, we compare our result with the traditional result at $z = 0 \mu m$ under both original setting and downsampling settings. Our results show low artifacts at full-scale while
Figure 4.3: Example of super-resolution at x,y axis. The downsampling rows are marked with /2. The boxed areas in first column are shown at the second column, and the boxed area in the second column is shown in the third column. Dotted box and solid box are chosen based on visual effects. Both our results using original data set and downsampled data set are sharp at 6× scale, while the traditional reconstruction results are pixelized.
Figure 4.4: Demonstration of axial sectioning effect on aIDT data set. One slice at $z = -7.5\,\mu m$ and anther at $z = 8\,\mu m$ are shown in the figure. The downsampling experiments are labeled with /2. The boxed area in the left columns for each depth is scaled and presented in the right column. Our results show a clear difference between feature and background while traditional reconstruction doesn’t.
Figure 4.5: Example of interpolation on z axis. Slices at 4.5 µm and 5 µm are included in the forward model during optimization. Our neural representation is able to reconstruct slices at any other location after convergence. This example shows an interpolated slice at z = 4.75 µm (boxed) and compares it with traditional reconstruction results, which is acquired at extra optimization cost.

Traditional reconstruction results have noisy backgrounds. As we scale up the focus region, our continuous representation stays sharp and clear while the traditional reconstruction methods begin to pixelate. This effect in traditional results is especially obvious on the downsampled results.

Furthermore, we show a better axial segmentation helps identify features in Figure 4.4. In these experiment, we apply the bilinear interpolation to traditional results for better comparison. At z = −7.5 µm the the array-like diatom structure starts showing up in reconstructions. Although our downsampled result is a bit blurry, both of our results reveal the
grid-like structure. However, the structure is difficult to identify in both traditional reconstruction results. At $z = 8 \mu m$, both of our results clearly differentiate from the background while the boundary in traditional reconstruction is hard to identify.

Next, we show our continuous representation effect on $z$ axis. Our neural representation is optimized every $0.5 \mu m$ from $z = -10 \mu m$ to $16 \mu m$. However, we can acquire a reconstruction outside of those layers by querying the result with desired coordinates. For example, in Figure 4.5, we present a reconstruction at $z = 4.75 \mu m$. The accuracy of interpolation is validated by comparing it with a traditionally reconstructed slice. Our method as always shows fewer artifacts while being visually consistent with the traditional method in the focus area. Also, querying the neural network with desired coordinates is computationally simpler comparing to traditional methods reconstruction which requires re-computing the entire transfer function and applying optimization.

### 4.3 High-quality reconstruction with mIDT

We further demonstrate our high-quality reconstruction on the mIDT C. elegans data set. This data set is collected using $40 \times$ MO with $N.A. = 1.33$ and camera pixel size $= 6.5 \mu m$.

We list two slices of the C. elegans worm at $z = -0.5$ and $-7 \mu m$ in Figure 4.6. Tikhonov results are scaled using the bilinear super-resolution method for better comparison. At $z = -0.5 \mu m$, our reconstruction appears similar to Traditional results at full scale. But as we zoom in, traditional reconstruction starts getting blurry, and the advantage of continuous representation starts to show up. Finally, our reconstruction still has sharp edges and preserves details while the traditional method shows smooth edges as a result of scaling.

At $z = -7 \mu m$, traditional reconstruction result have connected and blurred features and is hard to differentiate one granular structure from another. Our result, however, shows clear edge and sharp feature segmentation despite reconstructing on a non-focal plane. An area selected in our result has five granular structures, three on the left side forming a triangular shape and two on the right of it, while in traditional reconstruction, the area is blurred into one chunk with some variation in intensity.
Figure 4.6: High-quality reconstruction with mIDT on C. elegans data set at $z = -0.5$ and $-7 \mu m$. Our continuous reconstruction results stay sharp and clear at any scale.
4.4 Size of neural representation

Finally, we list the disk size of our neural representation. We compare the reconstruction disk size between our result and traditional result under optimization resolution in Table 4.1. Our results are 15-230 times smaller than the traditional results across our experimental data sets.

<table>
<thead>
<tr>
<th>Size</th>
<th>IDT, Algae</th>
<th>mIDT, C. elegans</th>
<th>aIDT, Diatom /2</th>
<th>aIDT, Diatom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>1024 × 1024 × 41</td>
<td>800 × 800 × 41</td>
<td>350 × 350 × 53</td>
<td>700 × 700 × 53</td>
</tr>
<tr>
<td>Traditional</td>
<td>2.6 MB</td>
<td>2.4 MB</td>
<td>2.6 MB</td>
<td>2.5 MB</td>
</tr>
<tr>
<td>Traditional</td>
<td>600 MB</td>
<td>352 MB</td>
<td>41 MB</td>
<td>162 MB</td>
</tr>
</tbody>
</table>

Table 4.1: Neural representation size and traditional reconstruction size comparison. /2 stands for downsampling by 2.
Chapter 5

Conclusion

In this thesis, we propose a novel method Regularized Neural Representation Learning for optical tomography, and validate it on three Intensity Diffraction Tomography modalities: conventional IDT, mIDT, and aIDT. Our method continuously represents a sample and achieves high-resolution, low artifacts results with excellent axial sectioning effect. The generated neural representation enables arbitrary interpolation within the sample domain without the need for further optimization. We also validate the possibility of using a deep modern denoiser as regularization prior for neural representation learning in a self-supervised fashion. Our algorithm doesn’t need any training data other than target sample measurements. Furthermore, a proposed block-wise learning algorithm reduces memory usage in a fast and accurate way. Finally, our reconstruction produces a latent neural representation that is smaller in disk size comparing to traditional reconstruction results.
References


Vita

Renhao Liu

Degrees

B.S. Summa Cum Laude, Computer Science, August 2020
M.S. Computer Science, August 2021

August 2021