

Efficient Simulation of Multidimensional Optical Imaging Using Tensor Train Decomposition

Pandhittaya Noikorn and Sherif S. Sherif

Abstract – Large multidimensional optical problems could require impractically large computer resources (e.g., storage and time). However, representing multivariable functions using tensors (multidimensional arrays) could greatly reduce these computational difficulties. In this paper, we present novel formulations of diffraction by a planar aperture and imaging by a thin lens of three-dimensional scalar optical fields using tensor train decompositions. Numerical simulations using our novel tensor train-based formulations compared with directly using original formulations demonstrated a significant reduction in both computational times (27–76 times) and computer storage requirements (413–1576 times).

1. Introduction

Optical mathematical models typically involve multivariable functions. For example, a simple physical optics problem could possibly involve time, three spatial variables and wavelength. Computations involving such an N -dimensional model with a large number of computational samples, $\{I_1, \dots, I_N\}$, would usually necessitate impractically large computer storage and/or computational time. This is because the storage complexity of this problem would increase exponentially, that is, $O(I^N)$, where I is the maximum number of computational samples. This practical constraint is referred to as the *curse of dimensionality* [1].

Multivariable functions could be represented as multidimensional arrays (tensors), so different tensor decompositions could alleviate these computational difficulties. Recently, such tensor decompositions have been playing a significant role in the computational aspects of many research fields (e.g., Signal Processing, Machine Learning, and Statistics) [2]. A function with N variables could be represented as an N^{th} order tensor having N modes, where the vectors along its n^{th} mode and their corresponding number of samples are referred to as *mode- n fibers* and their *mode- n dimension*, respectively [3].

A classic tensor decomposition is the *Tucker decomposition* (TD) [2–4]. TD decomposes an N^{th} -order tensor to an N^{th} -order core tensor, typically

smaller than the original tensor, along with N -factor matrices. This decomposition could be interpreted as a multilinear transformation of the core tensor by the factor matrices. The storage complexity of TD is $O(NIR + R^N)$, where R is the maximum dimension of the modes of the core tensor. Therefore, the potential reduction in size compared with the original tensor is essentially dependent on the dimensions of the resulting core tensor. We note that TD does not truly overcome the curse of dimensionality as the core tensor is still an N^{th} -order one.

A significantly more computationally efficient decomposition compared with TD is the recent tensor train decomposition (TTD) [5]. TTD could truly overcome the curse of dimensionality, as it represents an N^{th} -order tensor as a sequence of third-order core tensors. The storage complexity of TTD is $O(NIR^2)$, where R is the maximum rank of the matrices resulting from a mode-2 *matricization* of its third-order core tensors. We note that *Mode- n matricization* of a higher-order tensor ($N \geq 3$) is an operation that represents it by placing its mode- n fibers along the columns of a matrix.

In this paper, we present a novel TTD-based formulation and computationally efficient simulations of diffraction by a planar aperture and imaging by a thin lens of three-dimensional scalar optical fields.

2. Tensor Train Decomposition

The TTD of an N^{th} -order tensor, $\mathfrak{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, is given by

$$\mathfrak{X} \cong \mathfrak{X}^{(1)} \times_3^1 \dots \times_3^1 \mathfrak{X}^{(N)} = \left[\left[\mathfrak{X}^{(1)}, \dots, \mathfrak{X}^{(N)} \right] \right], \quad (1)$$

where $\mathfrak{X}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$ is a third-order core tensor, $n = 1, \dots, N$, R_0, \dots, R_N are the ranks of the matrices resulting from a mode-2 matricization of these core tensors ($R_0 = R_N = 1$), and \times_3^1 is a mode-(1, 3) tensor product [6]. The mode-(m , n) tensor product of $\mathfrak{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and $\mathfrak{B} \in \mathbb{R}^{J_1 \times \dots \times J_M}$ with common modes $I_n = J_m$ is given by an $(N + M - 2)$ order tensor

$$\mathfrak{C} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1}, I_{n+1}, \dots, I_N, J_1, \dots, J_{m-1}, J_{m+1}, \dots, J_M},$$

where

$$c_{i_1, \dots, i_{n-1}, i_{n+1}, \dots, i_N, j_1, \dots, j_{m-1}, j_{m+1}, \dots, j_M} = \sum_{i_n=1}^{I_n} a_{i_1, \dots, i_{n-1}, i_n, i_{n+1}, \dots, i_N} b_{j_1, \dots, j_{m-1}, i_n, j_{m+1}, \dots, j_M} \quad (2)$$

Manuscript received 15 April 2024. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

Pandhittaya Noikorn and Sherif S. Sherif are with the Department of Electrical and Computer Engineering, University of Manitoba, 75 Chancellors Circle, Winnipeg, MB, R3T 5V6, Canada; e-mail: noikornp@myumanitoba.ca, Sherif.Sherif@umanitoba.ca.

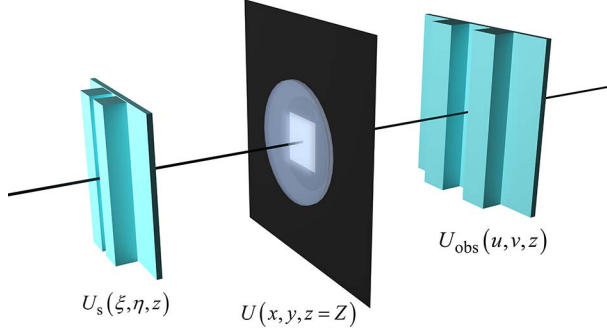


Figure 1. Setup of diffraction or imaging of three-dimensional scalar optical fields.

As convolutions and Hadamard products (element-wise multiplication) are common in mathematical formulations of multidimensional physical optics problems, their efficient computations using tensors and corresponding TTDs are very important.

The convolution of two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times \dots \times J_N}$ that have the same order and are represented by their TTDs, is given by

$$\mathcal{C} = \mathcal{A} * \mathcal{B} \cong \left[\left[\mathcal{A}^{(1)} \square_2 \mathcal{B}^{(1)}, \dots, \mathcal{A}^{(N)} \square_2 \mathcal{B}^{(N)} \right] \right], \quad (3)$$

where \square_2 denotes a partial mode-2 tensor convolution [6]. We note that compared with full convolutions of tensors partial mode- n convolutions described in (3) have significantly lower computational complexities as they involve a single mode- n only.

The Hadamard product of two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ that have the same order and same mode- n dimensions and are represented by their TTDs, is given by

$$\mathcal{C} = \mathcal{A} \bullet \mathcal{B} \cong \left[\left[\mathcal{A}^{(1)} \odot_2 \mathcal{B}^{(1)}, \dots, \mathcal{A}^{(N)} \odot_2 \mathcal{B}^{(N)} \right] \right], \quad (4)$$

where \bullet denotes a Hadamard product, and \odot_2 denotes the mode-2 tensor Khatri-Rao product [6].

3. Computational Complexity of Tensor Train Decomposition

The TTD of an N^{th} -order tensor is typically computed using singular value decomposition (SVD) or QR decomposition of N matrices representing its

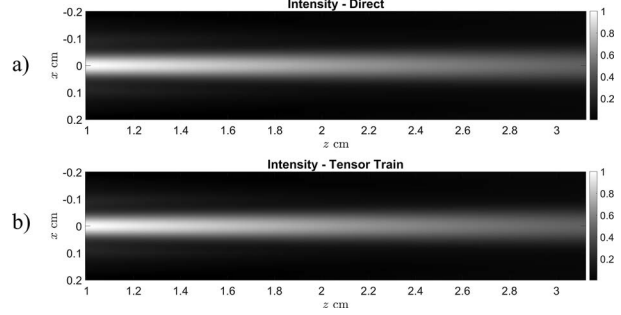


Figure 2. Optical intensity of the diffracted field along the z -axis using (a) direct formulation and (b) our TT formulation.

different mode- n matricizations [5–7]. The SVD-based TTD is called tensor trains (TT)-SVD, where its computational complexity depends on the chosen SVD algorithm. One computationally efficient algorithm that uses a stochastic method could be used to compute SVD. Therefore, it could significantly reduce the required TTD computational complexity to $O(N(Q^2K + Q^3I))$, where K is the maximum number of non-zero entries of the N^{th} -order tensor, and Q is the maximum rank of all mode-2 matricizations of the resulting TT core tensors [8].

The computational complexities of mathematical operations between tensors that are represented as TTs are typically linear with order N . Thus, one-time changes of the original tensor representations to TT could significantly reduce computational complexities for different operations. For example, the computational complexity of convolving two N^{th} -order tensors is $O(I^{2N})$ [9]. However, the computational complexity of such convolution using TTs is $O(N^2Q^3)$. Also, a significant computational reduction using TTs, instead of the original representation, is possible when computing the Hadamard product of two N^{th} -order tensors, where the computational complexity is reduced from $O(I^N)$ to $O(NIQ^3)$ [5].

4. Diffraction and Imaging of Three-Dimensional Scalar Optical Fields

As per the setup and coordinates shown in Figure 1, free-space Fresnel propagation of a three-dimensional scalar optical field from a source, $U_s(\xi, \eta, z)$, could be written as its convolution

Table 1. Simulation results of optical field diffraction

Size of source optical field, voxels	Computational time, s			Storage compression ratio	Complex field ℓ_2 -norm error
	Direct method	TT method	Speed up		
$561 \times 561 \times 121$	421	12	35×	839×	4.2369×10^{-31}
$749 \times 749 \times 157$	1777	36	49×	1224×	1.9905×10^{-30}
$899 \times 899 \times 186$	4915	182	27×	1576×	4.4705×10^{-30}

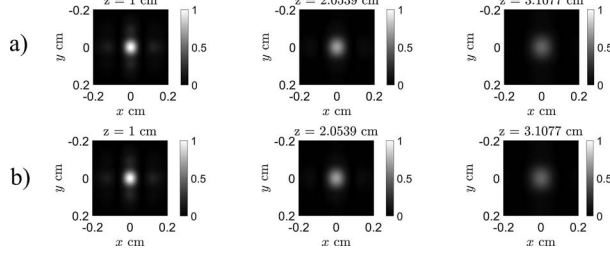


Figure 3. Lateral optical intensities of the diffracted field along the z -axis using (a) direct formulation and (b) our TT formulation.

$$U(x, y, z) = [U_s(\xi, \eta, z) *_{\xi, \eta, z} h(\xi, \eta, z)](x, y, z), \quad (5)$$

with the free-space Fresnel propagation kernel

$$h(\xi, \eta, z) = \frac{\exp(jkz)}{jkz} \exp\left(\frac{jk}{2z}(\xi^2 + \eta^2)\right). \quad (6)$$

At a fixed distance $z = Z$, we introduce a planar (two-dimensional) rectangular aperture, $P'(x, y)$, and a thin lens with focal length f , to obtain

$$P(x, y, z = Z) = P'(x, y) \exp\left[-\frac{jk}{2f}(x^2 + y^2)\right]. \quad (7)$$

Therefore, the resulting optical field at a three-dimensional observation region is given by

$$U_{\text{obs}}(u, v, z) = [\{U(x, y, z = Z)P(x, y, z = Z)\} *_{x, y} h(x, y, z)](u, v, z), \quad (8)$$

where U represents the two-dimensional field immediately before the plane $z = Z$, and $UP *_{x, y} h$ represents further propagation from this plane to the three-dimensional observation region.

5. Tensor Train-Based Formulation of Diffraction and Imaging of Scalar Three-Dimensional Optical Fields

We start by representing the optical field at the source, U_s , free-space Fresnel propagation kernel, h , planar rectangular aperture, P' , aperture with a thin lens, P , and optical field at the observation region, U_{obs} , as tensors, $\mathcal{U}_s, \mathcal{H}, \mathcal{P}', \mathcal{P}, \mathcal{U}_{\text{obs}}$, respectively. Furthermore, we represent these tensors by their TTDs

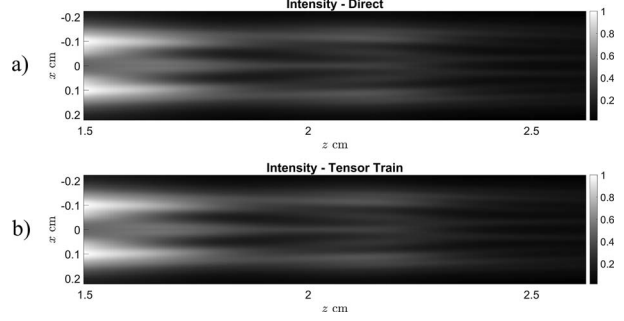


Figure 4. Optical intensity of the imaged field along the z -axis using (a) direct formulation and (b) our TT formulation.

using the same mathematical notation used in (1) [10].

Using (3) and (5), the TTD of the two-dimensional optical field, \mathcal{U} , at the plane immediately before the aperture ($z = Z$), is given by

$$\mathcal{U} \cong \left[\left[\mathcal{U}_s^{(\xi)} \square_2 \mathcal{H}_{(s \rightarrow Z)}^{(\xi)}, \mathcal{U}_s^{(\eta)} \square_2 \mathcal{H}_{(s \rightarrow Z)}^{(\eta)}, 1 \right] \right] (x, y, z = Z). \quad (9)$$

Using (4) and (8), the TTD of the two-dimensional optical field at the plane immediately after this aperture ($z = Z$) is given by

$$\mathcal{U}' \cong \left[\left[\left(\mathcal{U}_s^{(\xi)} \square_2 \mathcal{H}_{(s \rightarrow Z)}^{(\xi)} \right) \odot_2 \mathcal{P}^{(x)}, \left(\mathcal{U}_s^{(\eta)} \square_2 \mathcal{H}_{(s \rightarrow Z)}^{(\eta)} \right) \odot_2 \mathcal{P}^{(y)}, 1 \right] \right]. \quad (10)$$

Similar to (9), the propagation of this transmitted field, \mathcal{U}' , to a three-dimensional observation region is described by another two-dimensional convolution. Therefore, the TTD of this observed field is given by [11]

$$\mathcal{U}_{\text{obs}} = \left[\left[\left(\mathcal{U}_s^{(\xi)} \square_2 \mathcal{H}_{(s \rightarrow Z)}^{(\xi)} \right) \odot_2 \mathcal{P}^{(x)} \right] \square_2 \mathcal{H}_{(Z \rightarrow \text{obs})}^{(u)}, \left[\left(\mathcal{U}_s^{(\eta)} \square_2 \mathcal{H}_{(s \rightarrow Z)}^{(\eta)} \right) \odot_2 \mathcal{P}^{(y)} \right] \square_2 \mathcal{H}_{(Z \rightarrow \text{obs})}^{(v)}, \mathcal{H}_{(Z \rightarrow \text{obs})}^{(z)} \right]. \quad (11)$$

Table 2. Computation results of optical field imaging

Size of input optical field, voxels	Computational time, s			Storage compression ratio	Complex field ℓ_2 -norm error
	Direct method	TT method	Speed up		
$561 \times 561 \times 121$	584	26	70×	413×	1.2227^{E-30}
$749 \times 749 \times 157$	2981	39	76×	593×	1.0064^{E-29}
$899 \times 899 \times 186$	8029	208	39×	757×	1.692^{E-29}

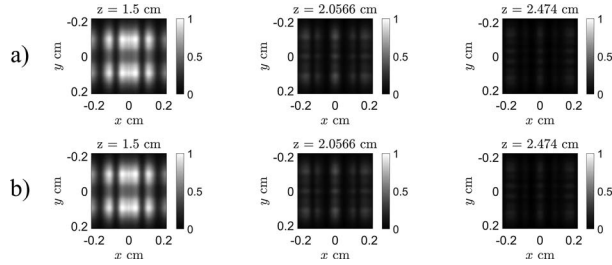


Figure 5. Lateral intensities of the imaged field along the z -axis using (a) direct formulation and (b) our TT formulation.

6. Numerical Examples

We compared different results of simulations of (8) and (11), where we refer to (8) as the direct formulation and (11) as the TT formulation, which we computed using the TT-SVD algorithm. Our first numerical example used the setup in Figure 1 without a lens to simulate the diffraction of a three-dimensional source optical field by a rectangular aperture. This scalar source field was numerically generated using a finite-difference frequency-domain method [12]. Different computational results of these simulations are shown in Table 1. We note that as the dimensions of the source optical field increase, the storage requirement using the TT formulation becomes significantly less than the storage requirement using the direct formulation. In Figure 2, we show optical intensities of the diffracted fields resulting from a three-dimensional source field ($899 \times 899 \times 186$ voxels) along the z -axis (relative to the aperture located at $z = Z$) that were obtained using both direct and our TT-based formulations. Lateral cross-sections of these diffracted fields at different locations along z relative to the aperture are shown in Figure 3. To obtain these diffraction simulation results, the required computer storage for the direct formulation was 16.67 GB, while our TT formulation required 10.83 MB only.

Our second numerical example used the same setup as in Figure 1 but with an added lens having a focal length of 2 cm at the aperture to simulate imaging of the same three-dimensional source optical fields used in the first example above. Different computational results of these simulations are shown in Table 2. Similar to our first numerical example, we note that as the dimensions of the source optical field increase, both computational times and storage requirements using the TT formulation become significantly less than these requirements using the direct formulation. In Figure 4, we show optical intensities of the fields resulting from imaging a three-dimensional source field ($899 \times 899 \times 186$ voxels) along the z -axis (relative to the aperture located at $z = Z$) that were obtained using both direct and our TT-based formulations. Lateral cross-sections of these imaged fields at different

locations along z relative to the aperture are shown in Figure 5. To obtain these imaging simulation results, the required computer storage for the direct formulation was 22.38 GB, while our TT formulation required 30.28 MB only.

7. Conclusions

Multidimensional optical simulations could involve impractical computational times and storage requirements as the number of computational samples typically increases exponentially with the number of variables. Multidimensional array (tensor)-based representations of multidimensional functions and their decompositions could be used to overcome these computational difficulties. TTD efficiently represents a high-order tensor into a sequence of third-order tensors, significantly reducing computational times and storage requirements. We presented novel formulations of diffraction by a planar aperture and imaging by a thin lens of three-dimensional scalar optical fields using TTD. Our numerical simulation examples using our novel TT-based formulations demonstrated a significant reduction in both computational times (27–76 times) and computer storage requirements (413–1576 times) compared with directly using original formulations. Our results confirm the feasibility and computational advantages of our TTD-based formulations.

8. References

1. R. E. Bellman, *Adaptive Control Processes*, Princeton University Press, Princeton, NJ, 1961.
2. N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, et al., “Tensor Decomposition for Signal Processing and Machine Learning,” *IEEE Transactions on Signal Processing*, **65**, 13, July 2017, pp. 3551–3582.
3. T. G. Kolda and B. W. Bader, “Tensor Decompositions and Applications,” *SIAM Review*, **51**, 2009, pp. 455–500.
4. A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, et al., “Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis,” *IEEE Signal Processing Magazine*, **32**, 2, March 2015, pp. 145–163.
5. I. Oseledets, “Tensor-Train Decomposition,” *SIAM Journal Scientific Computing*, **33**, 5, 2011, pp. 2295–2317.
6. A. Cichocki, N. Lee, I. Oseledets, A-H. Phan, Q. Zhao, et al., “Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions,” *Foundations and Trends in Machine Learning*, **9**, 4-5, September 2016, pp. 249–429.
7. I. Oseledets, oseledets/TT-Toolbox, <https://github.com/oseledets/TT-Toolbox>, 2023.
8. B. Huber, R. Schneider, S. Wolf, “A Randomized Tensor Train Singular Value Decomposition,” in: H. Boche, G. Caire, R. Calderbank, M. März, G. Kutyniok, et al. (eds), *Compressed Sensing and its Applications. Applied and*

- Numerical Harmonic Analysis*. Birkhäuser, Cham, 2017, Chapter 9 page 261.
9. K. Pavel and D. Davi, "Algorithms for Efficient Computation of Convolution," *InTech*, 2013.
 10. P. Noikorn and S. S. Sherif, "Multidimensional Optical Diffraction using Tensor Train Decompositions," 2023 XXXVth General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS), Sapporo, Japan, August 19–26, 2023.
 11. P. Noikorn, *Arbitrary Mode-n Convolution of Physical Tensors with Applications in Optics*, M.Sc., thesis, University of Manitoba, Canada, 2022.
 12. R. Rumpf, *Electromagnetic and Photonic Simulation for the Beginner: Finite-Difference Frequency-Domain in MATLAB*, Norwood, Artech, 2022.