

# 多重焦点画像列を用いた Optical Projection Tomography の DNN ライブラリ実装

石井 直行<sup>1,a)</sup> 長原 一<sup>2,b)</sup> 諸岡 健一<sup>1,c)</sup>

**概要:** 我々の研究グループでは、多重焦点画像列を用いた Optical Projection Tomography による細胞の 3 次元画像を復元する手法を構築している。本稿では、一般に用いられている DNN 用ライブラリを利用して、細胞の 3 次元画像構築の高速化法を提案する。具体的には、DNN 用ライブラリに適用するための処理の再定式化と、3 次元画像構築に適した最適化アルゴリズムの検証を行った。シミュレーションモデルを用いて、最適化アルゴリズムごとの透過率の推定精度や推定速度を比較した。

## 1. はじめに

Optical projection tomography(OPT)[1][2]とは、複数枚の画像から物体の内部構造を含めた 3 次元形状を推定する手法の一つである。OPT では、計測対象物体をボクセルに分割し、各ボクセルに格納された光線の吸収係数を求める。主にレンズ光学系を用いるため透視投影法となる近赤外線、可視光および紫外線をテレセントリック光学系であるとみなし、近似的に X 線 CT における画像再構築 [4]と同様の方法で物体の 3 次元形状を推定する。

一方、我々の研究グループでは、一般的な光学顕微鏡を用いて、異なる焦点面で細胞を撮影した多重焦点画像列から OPT による細胞の 3 次元画像復元手法を構築している [5]。細胞の 3 次元画像とは、細胞を覆う直方体領域をボクセルの集合で表現し、各ボクセルに存在する物質の透過率を可視化したものである。文献 [5] は、多重焦点画像列から各ボクセルの透過率を、最小二乗法による最適解探索を用いて推定することにより実現している。しかし、この手法は計算量が多く、実用的な時間で 3 次元画像構築を行うためにも、値の収束は早いほうが望ましい。

ここで、深層学習では、ネットワークの重みの最適値を最小二乗法で求める。また、近年 DNN 用ライブラリにより、多数の重みの最適解を効率的に求めることが可能となっている。

そこで、本稿では、一般に用いられている DNN 用ライブラリを利用して、細胞の 3 次元画像構築の高速化法を提案

する。具体的には、以下の 2 つのことを行った。1 つ目は、DNN 用ライブラリに適用するための処理の再定式化である。DNN 用ライブラリを適用させる利点としては、自動微分を行う機構があり勾配を容易に取得できるほか、豊富な最適化関数や損失関数の選択の自由度が高いことである。使用する DNN 用ライブラリは、Python のオープンソースライブラリの PyTorch である。2 つ目は、3 次元画像構築に適した最適化アルゴリズムの検証である。Gradient Descent(GD), Adam, Adadelta, RMSprop の各最適化アルゴリズムを、OPT の問題に適用した収束挙動を検証する。また、人工的に作成した細胞のシミュレーションモデルを用いて、最適化アルゴリズムごとの透過率の推定精度や推定速度を比較した。

## 2. 関連研究

### 2.1 Optical Projection Tomography

通常の OPT では、図 1 の A のような装置を用いて、推定する対象物体を回転させて、複数の画像を取得する。光学系は、被写界深度に限りがあり、対象物体の全体をカバーすることができない。そこで、図 1 の B のように、焦点面が対象物体の前半分に合うように位置させる。これにより、対象物体の前半分の焦点の合ったデータと、対象物体の後ろ半分の焦点の外れたデータの両方を含む画像が取得できる。また、図 1 の C では、標本全体での平行投影データへの近似法を示す。

対して、文献 [5] では、一般的な光学顕微鏡を用い、標本台を操作しながら細胞を上下に動かすことで、焦点の合った位置が異なる複数の画像を取得する。詳細は 3.1 にて述べる。

<sup>1</sup> 岡山大学

<sup>2</sup> 大阪大学

a) n.ishii@s.okayama-u.ac.jp

b) nagahara@ids.osaka-u.ac.jp

c) morooka@okayama-u.ac.jp

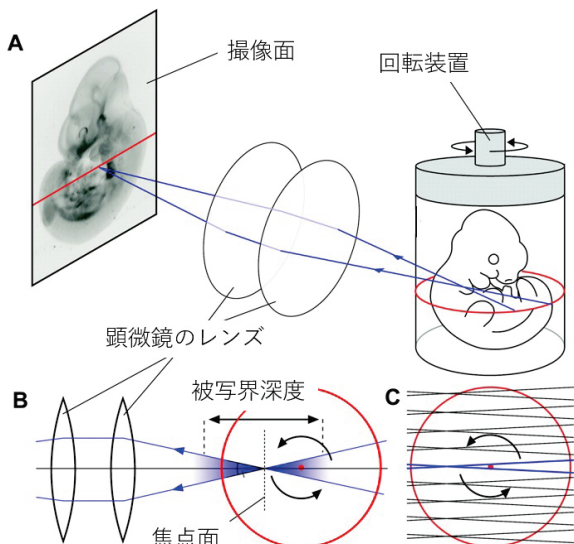


図 1 OPT 顕微鏡

## 2.2 DNN ライブラリ実装

複数の画像から対象の 3 次元形状を推定する別の手法として、Diffuse Optical Tomography(DOT)[6]がある。DOT は、表面下の光の伝搬を測定することによって、皮膚下の 3 次元構造を復元する方法である。この復元法では、皮膚下の体内を異種散乱媒体とみなし、光を皮膚に照射すると、その光が体内を通る過程を、異種散乱媒体内の拡散光子の伝搬によりモデル化している。このモデルをフォワードモデルとすると、3 次元構造構築は、取得した画像と、異種散乱媒体内の拡散光子の伝搬を表すフォワードモデルの差を最小化することで求める。通常、この最小化は繰り返し処理によって行われるが、文献 [6] では、この繰り返し処理を、入力データである光の伝搬情報に、各ボクセルが持つ光の吸収率を用いた畳み込み近似で表現し、それを PyTorch を用いて実装することによって、計算時間の高速化を実現している。[6] で用いている計算過程は、すでに我々が提案している手法 [5] の透過率の推定法と類似している。また、フォワードモデルは、光の経路に関する情報とボクセルごとに格納する値との線形結合である。我々の手法でも、多重焦点画像列の画素値は、光線の強さと透過率を用いた線形結合で表される。そこで、文献 [6] に基づき、細胞の 3 次元画像構築の高速化を行う。

## 3. 多重焦点画像列からの 3 次元画像構築

### 3.1 顕微鏡画像の撮像モデル

本研究では、図 2 に示す撮像モデルによって顕微鏡画像が得られると仮定する。つまり、ある焦点位置において標本台の下側から光を照射し、その光線が細胞内を通過する。この時、細胞内の物質によって光線は減衰し、その減衰した光線がレンズにより集光されることでその焦点位置での標本画像が得られる。

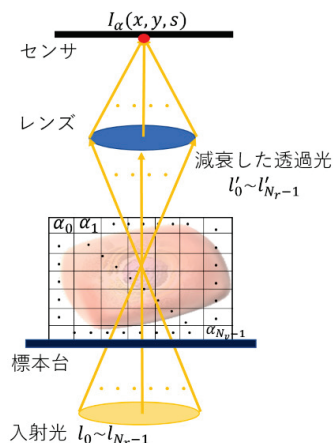


図 2 撮像モデル

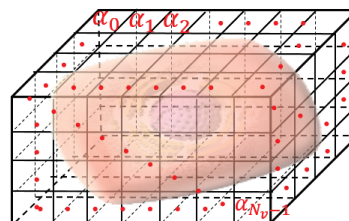


図 3 細胞が存在するボクセル領域

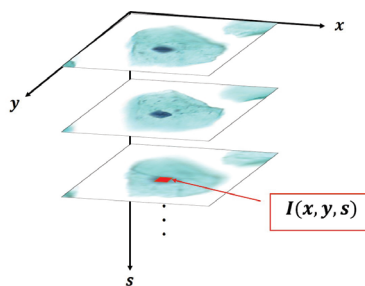


図 4 観測される多重焦点画像列

この撮像モデルを計算機で実現するために、まず、細胞を覆う直方体領域をボクセルの集合で表現し、 $i$  ( $i = 0, 1, 2, \dots, N_v - 1$ ,  $N_v$ :全ボクセル数) 番目のボクセルに存在する物質の透過率を  $\alpha_i$  ( $0 < \alpha_i \leq 1$ ) とする (図 3)。また、本撮像モデルでは固定された光学系に対し、標本台を操作しながら観察する細胞を上下に動かすことで、焦点位置を変化させる。この時、図 4 に示すように、多重焦点画像列中の  $s$  枚目の画像の画素  $(x, y)$  の輝度値を  $I(x, y, s)$  とする。さらに、本研究では得られた多重焦点画像列の枚数とボクセル領域の高さ方向の層数は同じとする。

次に、本来連続的である入射光を、 $N_r$  本の光線の集合として離散化する。この時、 $j$  ( $j = 0, 1, \dots, N_r - 1$ ) 番目の光線の明るさを  $l_j$ 、画像中の背景部分の輝度値を  $I_{back}$  を用いた式 (1)、次式を満たすように  $l_j$  の値を定める：

$$\sum_j l_j = I_{back}. \quad (1)$$

具体的には、まず入射光によって作られる三角錐の底面を、

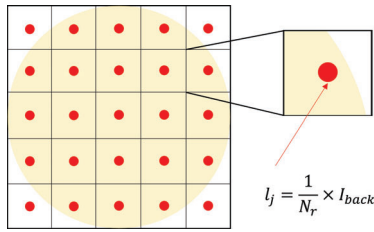


図5 入射光の設定

図5に示すように正方形に分割する．次に，各光線は分割した正方形の中心から出ていると仮定し， $l_j$ を式(2)と定める：

$$l_j = \frac{1}{N_r} \times I_{back}. \quad (2)$$

この時，分割した正方形の中心が三角錐の底面の外側の場合，その正方形の部分からは光線は出ていないものとする．

$l_j$ は，ボクセルを通過する度に，そのボクセルの透過率と，そのボクセルを通過した距離に応じて減衰し $l'_j$ となる．図6に示すように， $j$ 番目の光線が $i$ 番目のボクセルを通過する距離 $d_{ji}$ を使うと，減衰した光線 $l'_j$ は次式のように表される：

$$l'_j = l_j \times \prod_i \alpha_i^{d_{ji}}. \quad (3)$$

ただし， $j$ 番目の光線が $i$ 番目のボクセルを通過しない場合は $d_{ji} = 0$ である．ここで，式(3)の両辺の自然対数をとると，

$$\log l'_j = \log l_j + \sum_i d_{ji} \log \alpha_i \quad (4)$$

となる．画素ごとに各光線がボクセル領域を通過する経路も異なるため， $d_{ji}$ と $l'_j$ は $x, y, s$ の関数とみなせる．よって式(4)は式(5)となる．

$$\log l'_j(x, y, s) = \log l_j + \sum_i d_{ji}(x, y, s) \log \alpha_i. \quad (5)$$

観測される画像の輝度値は減衰した光線の総和とみなせるので，本撮像モデルに基づいて計算される画像の輝度値 $I_\alpha(x, y, s)$ は式(6)で求められる：

$$I_\alpha(x, y, s) = \sum_j l'_j(x, y, s). \quad (6)$$

### 3.2 フォワードモデル

式(6)を展開しまとめると，式(7)となる：

$$I_\alpha = \mathbf{L}^T \mathbf{X}. \quad (7)$$

ただし，

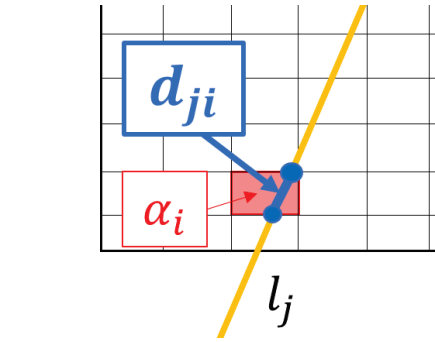


図6 光線 $l_j$ が $i$ 番目のボクセルを通過する距離 $d_{ji}$ の具体例

$$\mathbf{L} = [l_0 \quad \cdots \quad l_{N_r-1}]^T; \quad (8)$$

$$\mathbf{A} = [\log \alpha_0 \quad \cdots \quad \log \alpha_{N_v-1}]^T; \quad (9)$$

$$\mathbf{D}_i(x, y, s) = \begin{bmatrix} d_{i0}(x, y, s) \\ \vdots \\ d_{i(N_v-1)}(x, y, s) \end{bmatrix}; \quad (10)$$

$$\mathbf{X} = \begin{bmatrix} \exp(\mathbf{A}^T \mathbf{D}_0(x, y, s)) \\ \vdots \\ \exp(\mathbf{A}^T \mathbf{D}_{N_v-1}(x, y, s)) \end{bmatrix}. \quad (11)$$

式(7)において，各変数を，行列を用いた計算式に置き換えることで，DNN用ライブラリに適用できるようにした．

### 3.3 ボクセル値の再構築

撮像モデルに基づいた最適化手法における目的関数は式(12)である：

$$F(\mathbf{A}) = \sum_s \sum_y \sum_x (I(x, y, s) - I_\alpha(x, y, s))^2 + \sigma TV(\mathbf{A}). \quad (12)$$

この目的関数の最小化により $\mathbf{A}_t$ の最適値を求める．ここで， $\sigma$ は正則化項の重み，正則化項として Total variation(TV)ノルムを用いる．目的関数の最小化には3.4節で述べる4つの最適化アルゴリズムを用いる．実装にはPyTorchを使用した．PyTorchでは，自動微分を行う機構があり勾配を容易に取得できるほか，豊富な最適化関数や損失関数の選択の自由度が高い．

### 3.4 最適化アルゴリズム

本稿では，以下の4つの最適化アルゴリズムを用いて検討する．

#### • Gradient Descent(GD)

Gradient Descentは代表的な最適化手法であり，求めた勾配方向にその大きさだけパラメータを更新する方法である． $t$ 回目のパラメータ $\theta_t$ の更新式は式(13)である：

$$\theta_t = \theta_{t-1} - \gamma \cdot g_t. \quad (13)$$

$g_t$  はパラメータ  $\theta$  による損失関数の勾配,  $\gamma$  は学習率である.

- RMSprop

学習率を移動平均で割ることで, 勾配の大きさに応じて学習率を調整する. RMSProp の更新式は式 (14) である:

$$\theta_t = \theta_{t-1} - \frac{\gamma}{\sqrt{\beta \cdot E[g^2]_{t-1} + (1-\beta) \cdot g_t^2 + \epsilon}} g_t \quad (14)$$

$E[g^2]$  は勾配の二乗の累積,  $\beta$  は学習率を調整する係数,  $\epsilon$  はゼロ除算を避けるための極めて小さい値である.

- Adadelta[7]

RMSprop や Adagrad[9] などの最適化アルゴリズムは次元数が揃っていないという問題点がある. その次元数を揃うように改良したものが Adadelta である. Adadelta の更新式は式 (15) である:

$$\theta_t = \theta_{t-1} - \frac{\sqrt{\beta \cdot E[\Delta\theta^2]_{t-1} + (1-\beta) \cdot \Delta\theta^2 + \epsilon}}{\sqrt{\beta \cdot E[g^2]_{t-1} + (1-\beta) \cdot g_t^2 + \epsilon}} g_t \quad (15)$$

$\Delta\theta$  はパラメータ  $\theta$  の更新量である.

- Adam[8]

Adam では学習段階で継続的に学習率を下げる間, 学習の収束段階を早めるために, Momentum と Adagrad の長所を統合したものである. よって, 学習率を調整する最適化アルゴリズムと比較して, 収束段階が速く, 学習性能も高い. Adam の更新式は式 (16) である:

$$\theta = \theta - \gamma \cdot \frac{[\beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t] / (1-\beta_1^t)}{\sqrt{[\beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2] / (1-\beta_2^t) + \epsilon}} \quad (16)$$

$\beta_1$  と  $\beta_2$  は学習率を調整する係数,  $m_t$  と  $v_t$  はそれぞれ  $t$  回目における勾配の移動平均と勾配を二乗したものの移動平均である.

これら 4 つの最適化アルゴリズムの選択理由は, 3 次元復元を行ううえで結果がでていたこと [6] と. 同じ最適化アルゴリズムを用いて比較実験を行っていた例 [10] があったためである.

## 4. 実験

DNN ライブラリ実装による高速化実験と, 3 次元画像構築に適した最適化アルゴリズムの検証を行った. 使用した計算機は, CPU: Intel(R)Core(TM)i7-7820X CPU@3.60GHz, GPU: NVIDIA GeForce GTX 1080 Ti, 物理メモリ: 32GB である.



図 7 細胞モデルの多重焦点画像列

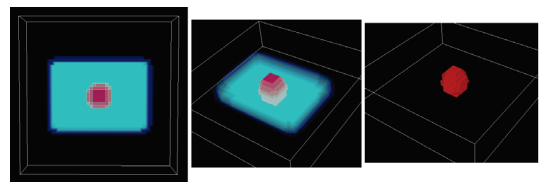


図 8 使用する細胞モデル

### 4.1 シミュレーションモデル

実験に用いる細胞の多重焦点画像列 (図 7) は, 図 8 に示す 3 次元ボクセルモデルから, 3.1 節で述べた撮像モデルに基づいて生成した. 図 8 のボクセルモデルの色は, 透過率を表しており, 赤いほど透過率が低く, 青いほど透過率は 1 に近い. 色がない部分は, 細胞がその領域にはないため透過率は 1 である. また, 生成した多重焦点画像列 (図 7) は,  $1[\mu m]$  ごとに焦点面をずらしながら撮影した 11 枚の画像で構成される. 各画像のサイズは  $50 \times 50[\text{pixel}]$ , 輝度値は 0 から 255 のグレースケール画像である.

### 4.2 DNN ライブラリ実装による高速化

文献 [5] と, DNN ライブラリ実装との, 計算時間比較による高速化実験を行った. 計測した時間は, フォワードモデルと最適化アルゴリズムを用いて  $A_t$  の最適値を更新する反復的なプロセスである. 用いた最適化アルゴリズムは, Gradient Descent であり, 学習率は  $6.0 \times 10^{-3}$  である. また, 反復回数は 1000 回, 正則化項の重み  $\sigma$  は  $1.0 \times 10^{-3}$  である. 透過率の初期値は文献 [5] で提案した方法を用いて, 入力の多重焦点画像列の輝度値をもとに, 式 (17) によって定める:

$$\alpha_i^{(0)} = \sqrt[N_s]{I(x, y, s)}. \quad (17)$$

この時, 光線数が 21 本, 437 本の場合について行い, 表 1 はそれぞれの計算時間を示す. 表 1 より, DNN ライブラリ実装により文献 [5] と比べ計算速度が速くなった. 光線数が 21 本の時は, 約 2 倍. 光線数が 437 本の時は, 約 10



表 1 計算時間の比較

	光線数 [本]	計算時間 [秒]
文献 [5]	21	62
	437	1181
DNN ライブラリ実装	21	28
	437	115

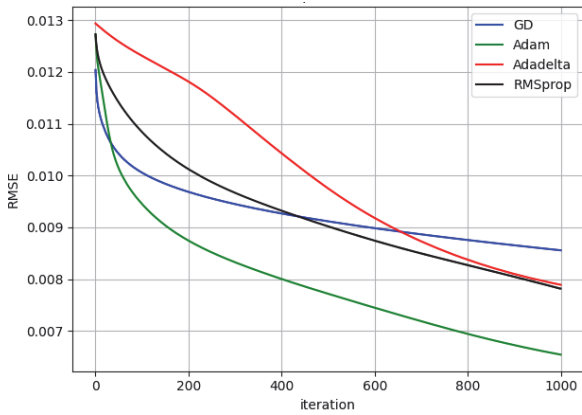


図 9 最適化アルゴリズムごとの RMSE の収束過程

倍となった。

### 4.3 最適化アルゴリズムの検証

3.4 節で述べた 4 種類の最適化アルゴリズム (Gradient Descent, Adam, Adadelta, RMSProp) を用いた 3 次元画像構築を行い、透過率の推定精度の比較・検証を行った。方法としては、最適化アルゴリズムとその時の学習率のみを変化させて、最適化アルゴリズムごとの収束過程を比較した。また、学習率はそれぞれ Gradient Descent :  $6.0 \times 10^{-3}$ , Adam :  $1.0 \times 10^{-3}$ , Adadelta :  $1.0 \times 10^{-2}$ , RMSprop :  $1.0 \times 10^{-3}$  とした。最適化関数の学習率以外のパラメータは PyTorch でのデフォルトの値を用いた。光線数は 437 本、反復回数は 1000 回、正則化項の重み  $\sigma$  は  $1.0 \times 10^{-3}$  である。透過率の初期値は、4.2 節で述べた、設定方法を使用した。以上の条件で行い、反復回数ごとの真値との平均平方根二乗誤差 (Root Mean Square Error : RMSE) を図 9 に示す。

図 9 より、どの最適化アルゴリズムでも RMSE の値は減少傾向にある。中でも、Adam が他の最適化アルゴリズムと比べて収束段階が速い。

最適化アルゴリズムを用いてパラメータ値を推定する上で、初期値は収束までの時間短縮につながるため重要な要素である。初期値の設定方法による影響を考慮して、初期値を全て 0.75 に設定し実験を行った。その結果を、図 10 に示す。また、二つの方法における、1000 回の反復回数での RMSE を表 2 に示す。

図 10 より、どの最適化アルゴリズムでも RMSE の値は減少傾向にあるが、図 9 の時と比べて差が出ることが分かった。特に、RMSprop と Adadelta がその差が顕著に表

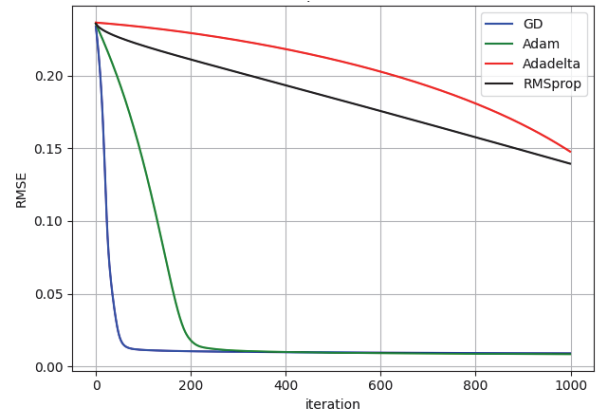


図 10 最適化アルゴリズムごとの RMSE の収束過程

表 2 各モデルの最適化アルゴリズムごとの推定精度

最適化アルゴリズム	初期値の設定	RMSE [ $\times 10^{-3}$ ]
GD	入力画像の輝度値	8.55
	全て 0.75	8.98
Adam	入力画像の輝度値	6.54
	全て 0.75	8.47
Adadelta	入力画像の輝度値	7.89
	全て 0.75	147.76
RMSprop	入力画像の輝度値	7.81
	全て 0.75	139.46

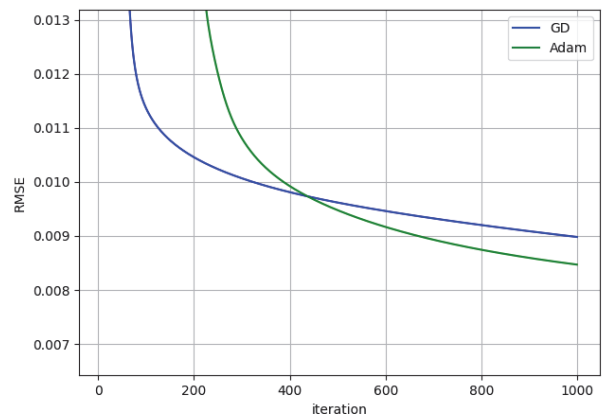


図 11 最適化アルゴリズムごとの RMSE の収束過程

れている。図 10 から RMSprop と Adadelta を除き、図 9 の縦軸の値域に合わせた RMSE の収束過程を図 11 に示す。表 2 より、Adam は初期値を変化させても 1000 回の反復回数で、どの最適化アルゴリズムよりも推定精度が高いことが分かった。

## 5. まとめ

本研究では、一般に用いられている DNN 用ライブラリを利用して、細胞の 3 次元画像構築の高速化法を提案した。具体的には、DNN 用ライブラリに適用するための処理の再定式化と、3 次元画像構築に適した最適化アルゴリズムの検証を行った。シミュレーションモデルを用いて、最適

化アルゴリズムごとの透過率の推定精度や推定速度を比較した。結果としては、RMSprop と Adadelta が初期値の設定によって、収束過程に差がでること、Adam が精度と収束が早い観点から適していることを示した。

## 参考文献

- [1] James Sharpe, Ulf Ahlgren, Paul Perry, Bill Hill, Allyson Ross, Jacob Hecksher-Sørensen, Richard Baldock, and Duncan Davidson. *Optical projection tomography as a tool for 3d microscopy and gene expression studies*. Science, Vol. 296, No. 5567, pp. 541-545, 2002.
- [2] Udo Jochen Birk, Alex Darrell, Nikos Konstantinides, Ana Sarasa-Renedo, and Jorge Ripoll. *Improved reconstructions and generalized filtered back projection for optical projection tomography*. Applied optics, Vol. 50, No. 4, pp. 392-398, 2011
- [3] Koskela, Olli, et al. *Gaussian light model in bright-field optical projection tomography*. Scientific reports 9.1 (2019): 1-14.
- [4] Brenner, David J., and Eric J. Hall. *Computed tomography—an increasing source of radiation exposure*. New England journal of medicine 357.22 (2007): 2277-2284.
- [5] Yamaguchi H, Nagahara H, Morooka K, Nakashima Y, Uranishi Y, Miyauchi S, Kurazume R: *3D Image Reconstruction from Multi-focus Microscopic Images*, Pacific-Rim Symposium on Image and Video Technology(PSIVT2019), pp.73-85, 2019.10(Sydney, Australia).
- [6] Chao Liu, Akash K. Maity, Artur W. Dubrawski: *High Resolution Diffuse Optical Tomography using Short Range Indirect Subsurface Imaging*, The IEEE International Conference on Computational Photography(ICCP)(2020).
- [7] Zeiler, Matthew D: *Adadelta: an adaptive learning rate method*. arXiv preprint arXiv:1212.5701 (2012).
- [8] D. P. Kingma and J. Ba: *Adam: A method for stochastic optimization*. in Proc. ICLR (Poster), 2015, pp. 115.
- [9] S. Ruder, *An overview of gradient descent optimization algorithms*, 2016, arXiv:1609.04747. [Online]. Available: <https://arxiv.org/abs/1609.04747>
- [10] Wang, Yixiang, et al. *Assessing optimizer impact on dnn model sensitivity to adversarial examples* IEEE Access 7 (2019): 152766-152776.