3D Deconvolution with Deep Learning

Abhijeet Phatak[1,2]
Solgaard Lab[1].$Department of Electrical Engineering$[2]
Stanford University, 350 Serra Mall, Stanford, CA 94305
aphatak@stanford.edu

Abstract

3D deconvolution is a challenging problem faced by astronomers and micrsoscopists. The goal of deconvolution algorithms is to reverse the effects of convolution on the observed data. Convolution can be understood as a blurring operation on the ground truth that results in a relatively blurred/unclear observation. It is challenging to model the exact convolution operation because of various factors like noise, optical aberrations and the complexity of optical setup. The Point Spread Function (PSF) is the impulse response of the imaging system which helps us to model a major part of the image forming process.

In this report, we look at various methods used for deconvolution of 3D microscopy images. In particular, we compare the performance of typical non-blind iterative algorithms like Richardson-Lucy (RL) and Alternating Direction Method of Multipliers (ADMM) that require knowledge of the PSF. The second set of techniques are based on Deep Learning methods that use Convolutional Neural Networks (CNNs) to learn a non-linear mapping from the observed data to the ground truth. In this context, we consider two network architectures and analyze the results of all the methods.

1. Introduction

Three dimensional (3D) deconvolution is used to deblur micrsoscopy images by eliminating the out-of-focus light. The amount of blur in spatially variant in most optical systems because of the spatially varying PSF of the micrsocope. Usually the PSF doesn't vary much in the lateral direction but varies more strongly in the axial direction. This is because most optical aberrations like spherical aberration, axial color, coma and astigmatism have a stronger axial component in typical imaging systems. Moreover, most 3D microscopes scan lateral planes at regular axial intervals. This causes further under-sampling along the axial (z) direction and leads to pooretr axial resolution. The lateral resolution is usually determined by the numerical aperture of the objective, effective magnification of the system, pixel size and the wavelength of light.

Typical algorithms used for deconvolution like RL, ADMM try to formulate the image formation process as a linear convolution with the optical PSF which is then transformed by a Poisson process. This is usually a good assumption in the case when the Shot noise is the dominating term and the Gaussian noise is minimal and holds good for most microscopes. Both the algorithms then iteratively try to minimize the residual between the ground truth by estimating the effects of convoltuion. ADMM is a much more powerful proximal-operator based optimization technique that can help us model different terms like data fidelity, noise model and sample-specific prior. ADMM tries to decouple these terms and optimize them separately in order to minimize the cost function.

However, both the above methods do not specifically address the issue of sub-sampling along the axial direction. In this report, I have tried to implement two different methods which employ CNNs and train them to perform deconvolution. The idea is to train the network to directly learn the mapping from the observed data to the ground truth without explicitly giving any image formulation model for noise. However, there are certain assumptions made about the PSF of the system. Before we discuss the methods in detail, let us understand the image formulation models used.

1.1. Image Formation and Convolution

The 3D captured dataset, $M(r)$, is created by recording optical sections of the three dimensional volume at different focal planes. Let $PSF(r)$ be the 3D PSF associated with the microscope system. Assuming linearity, the relationship between $M(r)$ and the 3D sharp object, $O(r)$, can be represented as a convolution operation:

$$M(r) = \int O(s)PSF(r-s)ds \quad (1)$$

$$= O(r) \otimes PSF(r) \quad (2)$$

$M(r), O(r), PSF(r)$ are 3D matrices written as $\mathbf{b}, \mathbf{x}, \mathbf{c}$, respectively, since the captured data is discrete. The convo-

lution operation can now be represented as:

$$\mathbf{b} = \mathbf{c} * \mathbf{x} \tag{3}$$

Now, we represent the convolution kernel $\mathbf{c}$ as a block-circulant matrix $\mathbf{A}$ such that

$$\mathbf{b} = \mathbf{A}\mathbf{x} \tag{4}$$

We have therefore incorporated the PSF of the optical system in this model. However, we also have noise due to detection at the sensor and emission by the fluorescent dyes. Both of these processes can be modelled by a possion process. Thus we can write,

$$\mathbf{b} = \mathcal{P}(\mathbf{A}\mathbf{x}) \tag{5}$$

For Isonet, we consider a slightly modified version of the image formation model which also includes a subsampling operator along with this image formation process.

1.2. Dataset

The dataset used for all experiments is taken from Cell Imaging Library which is an open-source repository for images of biological specimen. The dataset [2] shows microtubules in a Drosophila S2 cell. It contains a stack of 44 wide-field (WF) images and 44 structured-illumination (SIM) images of size 1904x1900 which makes the volume size to be 1904x1900x44. Due to limitations of computing power and ease of iterating, the images were resized in the lateral directions. Some basic image thresholding and corrections were made with exisiting pipelines in Fiji (ImageJ) [5]. It is a reasonable assumption to make that the SIM images are the ground truth and the the WF images are blurred observed data.

2. Methods for Deconvolution

2.1. Richardson-Lucy

Richardson-Lucy deconvolution is very widely used in microscopy because of its simplicity and effectiveness for Poisson noise. Using the same formulation as above, the Richardson-Lucy deconvolution in 3D with a Total-Variation (TV) prior gives us an iterative update strategy as [3]:

$$\mathbf{x}^{(q+1)} = \frac{\mathbf{A}^T\left(\frac{b}{\mathbf{A}x}\right)}{\mathbf{A}^T\mathbf{1} - \lambda\left(\frac{\mathbf{D_x x}}{|\mathbf{D_x x}|} + \frac{\mathbf{D_y x}}{|\mathbf{D_y x}|} + \frac{\mathbf{D_z x}}{|\mathbf{D_z x}|}\right)}\mathbf{x}^{(q)}$$

In our implementation, because of the large scale of problem, we do not implement the total variation prior and instead use $\lambda = 0$. Richardson-Lucy algorithm usually takes more number of iterations to converge.

2.2. Alternating Direction Method of Multipliers

ADMM is a variant of the augmented Lagrangian scheme that uses partial updates for the dual variables which is why it is a very popular and powerful technique for optimization. We can pose deconvolution [8] as an ADMM problem by considering the following optimization problem. Proximal methods have been shown to perform better than RL for large-scale problems by converging to a lower residual in fewer iterations and also by being more flexible towards the choice of prior.[6]

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \sum_i g_i(\mathbf{z_i}) = g_1(\mathbf{z_1}) + g_2(\mathbf{z_2}) + g_3(\mathbf{z_3}) \\ &\text{subject to} \quad \underbrace{\begin{bmatrix}\mathbf{A}^T & \mathbf{I}^T & \mathbf{D}^T\end{bmatrix}^T}_{\mathbf{K}}\mathbf{x} - \underbrace{\begin{bmatrix}\mathbf{z_1}^T & \mathbf{z_2}^T & \mathbf{z_3}^T\end{bmatrix}^T}_{\mathbf{z}} = 0 \end{aligned}$$

Here the goal is to minimize the cost function which includes the negative log-likelihood of the Poissonian observations, the non-negativity constraint and the prior (if used) respectively. $\mathbf{D}$ is the differential convolution operator and depends on the regularization/prior used.

The solution to this optimization problem can be iteratively found using ADMM as:

Initialization: $\mathbf{x}_{init} = \mathbf{b}, \mathbf{v} = \mathbf{z} = \mathbf{u} = 0, \rho > 0$

for k = 1 to MaxIter

$\mathbf{x} \leftarrow \mathbf{prox}_{quad}(\mathbf{v}),\ \mathbf{v} = \mathbf{z} - \mathbf{u}$

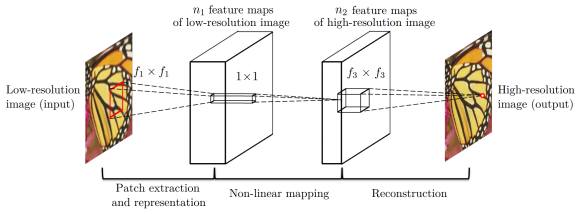
$\mathbf{z} \leftarrow \mathbf{prox}_{g_i,\rho}(\mathbf{v}),\ \mathbf{v} = \mathbf{K}\mathbf{x} + \mathbf{u}$

$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{K}\mathbf{x} - \mathbf{z}$

end

2.3. Super-Resolution Convolutional Neural Network

Recently, CNNs have shown great promise in creating super-resolved images from low-resolution input images. This is achieved by training a neural network with many such pairs of low resolution, high resolution images. The idea is that the network learns a non-linear mapping between features of the low-resolution inputs and the corresponding features in the high-resolution space. This illustration taken is taken from the Super-resolution CNN (SRCNN) paper [4] and shows intuitively how the network works.



The network architecture was adapted as-is from the paper with the values $f1 = 9, f2 = 1, f3 = 5, n1 = 64, n2 = 32$ for the filters and number of layers. For this

work, I applied a naïve 2D SRCNN model frame-by-frame. Then the output was stacked together. There were two sets of training images used. First one were natural images provided in the supplemental information of the original paper. For the second test, I chose to train the neural network on half of the images and used the other half as the test set. For each layer, ReLU activation was used and the exact protocol mentioned in the SRCNN paper was followed.

2.4. Isonet

Uptil now, we have not considered the down-sampling in the axial (z) direction because of the stretched PSF and also due to the mechanical scanning modality along z-axis for most micrscopes. Both these affect the axial resolution to decrease by a factor of 2 to 4. Recently, two new CNN architectures were proposed for deconvolution [7] which try to solve a super-resolution problem on sub-sampled data and adeconvolution problem to counter the effects of the optical PSF by exploiting the anisotropy of resolution along the lateral and axial directions.

The Isonet protocol can be intuitively understood with the help of the below illustration.

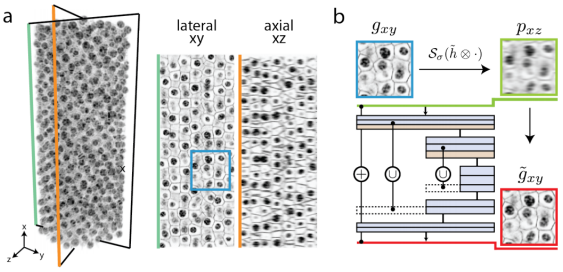


Figure (a) shows how the lateral and axial resolution is different. Figure (b) shows the network architecture of Isonet-2. First, a rotated PSF is considered that is used to convolve the lateral slices to estimate the sub-sampled axial slice. Then such training sets are passed to the network to train the model to quantify the anisotropy. To understand the exact procedure, let us consider a revised image formation model as shown below.

$$g = \mathcal{P}\left[\mathcal{S}_\sigma(h \otimes f)\right] + \eta \qquad p_{xy} = \mathcal{S}_\sigma(\tilde{h} \otimes g_{xy})$$

$$C_{64,9,9} - C_{32,5,5} - C_{1,5,5} - C_{1,1,1}$$

$$\mathcal{L} = \sum_n -[20 \log_{10} \max g_{xy,n} - 10 \log_{10} |g_{xy,n} - \tilde{g}_{xy,n}|^2]$$

Here we assume that the ground truth (f) gets convolved with the PSF (h) and is then sub-sampled by the slicing operator S_σ. p_{xy} is the inverse map which is found by convolving the axial slice with a rotated PSF ($\tilde{h}$). The loss function L is then minimized to train the model which basically tries to minimize the pixel wise PSNR. $\tilde{g_{xy}}$ is the output of the network when applied to p_{xy}.

The two architectures can be summarized with conventional notation as below:

$C_{n,w,h}$: Convolutional layer with n filters of size $(w,\ h)$

$M_{p,q}$: Max pooling layer with a subsample factor of $(p,\ q)$

$U_{p,q}$: Upsampling layer with a subsample factor of $(p,\ q)$

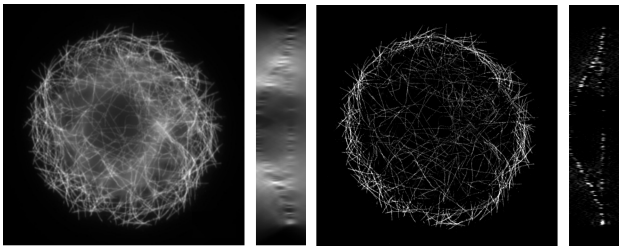
Isonet-1 : $C_{64,9,9} - C_{32,5,5} - C_{1,5,5} - C_{1,1,1}$

Here, the first layer extracts features and maps the output non-linearly to resulting image estimate.

Isonet-2 : $C_{16,7,7} - M_{2,2} - C_{32,7,7} - M_{2,2} - C_{64,7,7} - U_{2,2} - C_{32,7,7} - U_{2,2} - C_{16,7,7} - C_{1,1,1}$

3. Results and Discussion

Let us first take a look at the initial dataset. The below images are the Widefield (blurred) and SIM (ground truth) images in the focal stack. Clearly the spatial resolution is higher in the SIM image.



(a) WF (b) SIM

The results of all the methods are given in the table below.

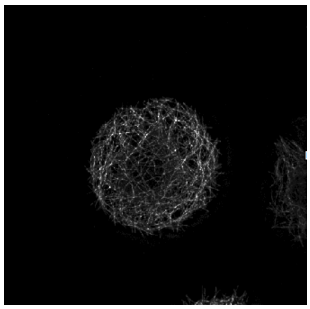
3.1. Typical methods

Both RL and ADMM techniques work very well for Poissonian noise which is observed in this dataset. They also converge smoothly. The algorithms were run for 100 iterations both. The convergence plots and log of residual plots are also shown. It can be clearly seen that ADMM converges much faster than RL and hence use of proximal algorithms is better than conventional algorithms for large scale problems. Although 100 iterations of ADMM require more time, we can see that it converges to a decent residual value in fewer iterations. ρ was chosen to be 0.25 for the given dataset. Please note that these are the maximum intensity projections of the 3D stack.

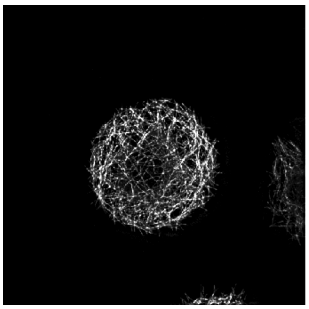
Algorithm	PSNR	Time(s)
RL	20.842	326
ADMM	22.151	418

3.2. Deep Learning based methods

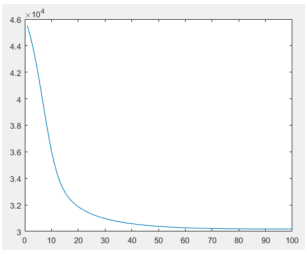
Deep learning based approaches are expected to be more flexible and robust as they try to learn the mapping from observed data to ground truth directly and create a sample specific prior automatically. I tried two things with the SRCNN model. First, I tried to see if I can directly use natural



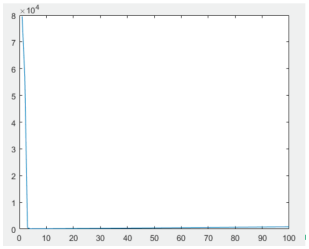
(c) RL Output



(d) ADMM Output

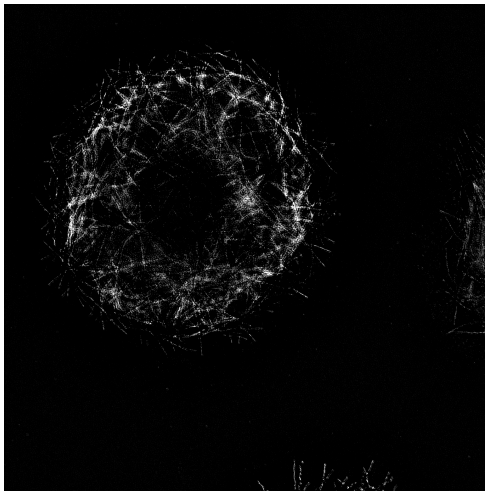


(e) RL Residual

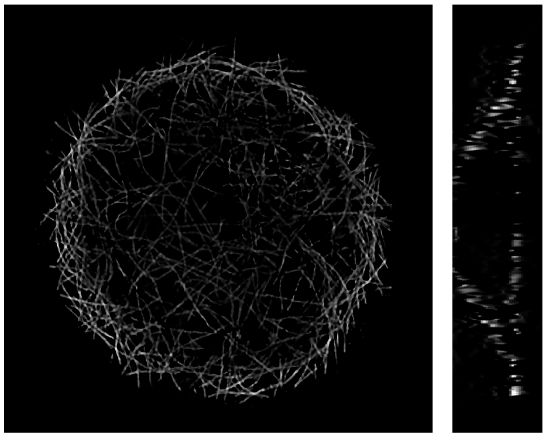


(f) ADMM Residual

images for training the model and apply it for deconvolution of WF images. However, as expected this method performs very poorly because there is a lot of difference between natural images and microscopy data. The distribution of data is significantly different in real world and micrsocopy . To see the effectiveness of a naïve SRCNN, I tried to train the network using half of the dataset and then used it for deconvolving the other half of the dataset. It is also important to note that the training data for both the SRCNN implementations was limited and not sufficient for deep networks. Better performance can be expected if we have more training data.



Isonet performs better than SRCNN as expected. As per the the paper [7], it is also clear that a Isonet-2 will give better results than Isonet-1. However, it is still sub-optimal because it assumes that the PSF is constant throughout the volume and is not true for this dataset. Usually it is not true for any microscopy setup but is not a terrible approximation.



Method	PSNR	Approx. Training Time (min)
Naive SRCNN	18.192	6
Isonet-1	23.151	13

We note that although the training process takes time, inference is much faster.

4. Conclusion

To summarize, we saw how novel techniques like deep learning with CNNs can be used to solve complex problems like 3D deconvolution effectively. Such methods can also be modified for other problems like segmentation, feature extraction and custom analysis on 3D datasets. We also show comparison of the performance of the algorithms.

5. Future work

For this project, I estimated the PSF using the blind deconvolution algorithm used by MATLAB's `deconvblind` function to estimate the PSF [1]. However, since we know the ground truth and the observed data, we could have formulated it using a primal-dual problem. This would have let us model the PSF and noise model more efficiently as blind deconvolution does not guarantee that the resulting PSF is the actual PSF. I would also like to work on better CNN architectures for image segmentation and deconvolution. Also since there is no ground truth available in microscopy, it is also interesting to try simulating samples and then training the model with the simulated samples with different range of noise and PSF so as to make the architecture more general and not sample specific. It would be also interesting to see the robustness of the model by using a model to deconvolve different datasets.

6. Hardware and Software

MATLAB 2017a was used for implementing RL and ADMM. SRCNN and Isonet was implemented using Python and tensorflow-gpu as the neural network framework. Keras was used to wrap the tensorflow model. System configuration - 16GB RAM, 6GB Nvidia 1060 GPU, Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz processor. OS - Linux Ubuntu 16.04 LTS 4.13.0.27 generic - Xenial Xerus. All the codes have been uploaded to github (https://github.com/abhijit-15/Deconvolution3D). Some operations like image thresholding, image correction wer done with the open source ImageJ software (also known as Fiji). Please feel free to reach out to me in case of any issues.

7. Acknowldgements

I would like to thank Prof. Gordon Wetzstein for interesting lectures, discussions and for the opportunity to work on this project. I would also like to thank Hayato Ikoma who was my project mentor and for giving me valuable feedback and advice.

As part of Solgaard lab, I would like to thank Prof. Olav Solgaard for the opportunity to work on the Light Sheet Fluorescence Microscopy project. I thank Joseph Landry for frutiful discussions on deconvolution and also letting me work with him on the light sheet microscope project.

I would also like to thank Martin Weigert (MPI-CBG) for clarifications regarding the Isonet paper and the assumptions they have made.

References

[1] MATLAB blind deconvolution method. `http://www.mathworks.com/help/images/ref/deconvblind.html`.

[2] V. Cazares. Microtubules in a Drosophila S2 cell. `http://www.cellimagelibrary.org/images/36797`.

[3] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia. Richardson–lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microscopy research and technique*, 69(4):260–266, 2006.

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014.

[5] W. Rasband. ImageJ. US National Institutes of Health, Bethesda, MD, 1997.

[6] P. Varma and G. Wetzstein. Efficient 3d deconvolution microscopy with proximal algorithms. In *Imaging Systems and Applications*, pages JT3A–44. Optical Society of America, 2016.

[7] M. Weigert, L. Royer, F. Jug, and G. Myers. Isotropic reconstruction of 3d fluorescence microscopy images using convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 126–134. Springer, 2017.

[8] G. Wetzstein. Deconvolution. `http://stanford.edu/class/ee367/reading/lecture6_notes.pdf`.