



Enabling real-time light-field 3D fluorescence microscopy through computational microscopy and deep learning

Josué Page Vizcaíno

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:

Prof. Dr. Cristina Piazza

Prüfende der Dissertation:

1. Priv.-Doz. Dr. Tobias Lasser
2. Prof. Dr. Paolo Favaro,
Universität Bern

Die Dissertation wurde am 22.02.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 21.06.2023 angenommen.

Abstract

Microscopy has been one of the pinnacles of the functional understanding of biological specimens for centuries. And for a good reason, being able to visualize and study how cells, bacteria, and viruses, among others., develop, interact, and strongly influence our day-to-day is not only mesmerizing but necessary. Industries and research directions indispensable to humankind depend on it, like the alimentary and pharmaceutical industry, medicine, and genetics. In parallel, computational tools have become essential for microscopy analysis, processing, and optimization. This work integrates computational sciences and microscopy to develop tools for imaging living specimens' 3-dimensional (3D) dynamics. Due to biology being a continuous 3D process, having access to real-time 3D information is essential for the understanding of biological systems.

Within this doctorate project, tools for 3D real-time fluorescent imaging were investigated. Mainly focused on the light field microscope (LFMic) and its image formation model. The LFMic is a scan-less fluorescent microscope capable of acquiring images at camera speed and reconstructing 3D volumes in a post-processing step, enabled by a micro lens array (MLA) within the optical path. As the reconstruction procedure is lengthy and far from real-time, the first approach tried was a deep learning (DL) model that showed substantial speed and quality improvements. However, due to the lack of certainty metrics, the biological validity of the DL reconstruction could not be determined, which is a priority when working with bio-medical data. Hence, motivating the usage of Bayesian learning, specifically of Normalizing Flows (NFs), where fast 3D reconstructions are possible with a certainty metric and out-of-distribution (OOD) detection. Meanwhile, exploring different optical architectures, like light field microscopy (LFM) and Fourier light field microscopy (FLFM), in fluorescently labeled mice brain vasculature and image sequences of zebrafish neural activity.

During this process, an important notion came alight, the joint optimization of the optical system and the reconstruction algorithm is a powerful concept. Tunable elements like a spatial light modulator (SLM) optimized with either a neural network or a reconstruction algorithm allow the algorithm to pick which setup and features are best for a given task *e.g.*, 3D reconstruction. With this vision in mind, we developed the WaveBlocks Python framework, an auto-differentiable wave-optics simulator. And put it into practice for aberration correction and point spread function (PSF) recovery with a SLM on a bright field microscope.

Finally, I present the research workflow based on tools and paradigms crucial to project organization, like GitHub, Notion, and the Pomodoro method. And tools like SLURM and guild.ai for handling the computational needs and experiments in a reproducible and organized way.

Zusammenfassung

Die Mikroskopie ist seit Jahrhunderten einer der Höhepunkte des funktionalen Verständnisses biologischer Exemplare. Und das aus gutem Grund: Es ist nicht nur faszinierend, sondern auch notwendig, zu sehen und zu studieren, wie sich Zellen, Bakterien und Viren entwickeln, miteinander interagieren und unser tägliches Leben stark beeinflussen. Industrien und Forschungsrichtungen, die für die Menschheit unverzichtbar sind, sind davon abhängig, genauso wie die Lebensmittel- und Pharmaindustrie, die Medizin und die Genetik. Parallel dazu sind computergestützte Werkzeuge für Mikroskopieanalyse, -verarbeitung und -optimierung unverzichtbar geworden. Diese Arbeit integriert Computerwissenschaften und Mikroskopie, um Werkzeuge zur Abbildung der 3-dimensionalen (3D) Dynamik lebender Proben zu entwickeln. Da die Biologie ein kontinuierlicher 3D-Prozess ist, ist der Zugang zu 3D-Informationen in Echtzeit für das Verständnis biologischer Systeme unerlässlich.

Im Rahmen dieses Promotionsprojekts wurden Werkzeuge für die 3D-Echtzeit-Fluoreszenzbildgebung untersucht. Der Schwerpunkt lag dabei auf dem Lichtfeldmikroskop (LFMic) und seinem Bildgebungsmodell. Das LFMic ist ein scanloses Fluoreszenzmikroskop, das in der Lage ist, Bilder mit Kamerageschwindigkeit aufzunehmen und 3D-Volumen in einem Nachbearbeitungsschritt zu rekonstruieren, was durch ein Mikrolinsen-Array (MLA) im optischen Pfad ermöglicht wird. Da das Rekonstruktionsverfahren langwierig und weit davon entfernt ist, in Echtzeit abzulaufen, wurde zunächst ein Deep-Learning-Modell (DL) ausprobiert, das erhebliche Geschwindigkeits- und Qualitätsverbesserungen zeigte. Aufgrund des Fehlens von Sicherheitsmetriken konnte die biologische Validität der DL-Rekonstruktion jedoch nicht bestimmt werden, was bei der Arbeit mit biomedizinischen Daten eine Priorität ist. Dies motiviert den Einsatz von Bayesian Learning, insbesondere von Normalizing Flows (NFs), bei denen schnelle 3D-Rekonstruktionen mit einer Gewissheitsmetrik und Out-of-Distribution (OOD) Erkennung möglich sind. In der Zwischenzeit wurden verschiedene optische Architekturen, wie Lichtfeldmikroskopie (LFM) und Fourier-Lichtfeldmikroskopie (FLFM) in fluoreszenzmarkierten Hirngefäßen von Mäusen und Bildsequenzen neuraler Aktivität von Zebrafischen untersucht.

Während dieses Prozesses kam ein wichtiger Gedanke zum Vorschein: Die gemeinsame Optimierung des optischen Systems und des Rekonstruktionsalgorithmus ist ein leistungsfähiges Konzept. Abstimmbare Elemente wie ein räumlicher Lichtmodulator (SLM), der entweder mit einem neuronalen Netz oder einem Rekonstruktionsalgorithmus optimiert wird, ermöglichen es dem Algorithmus, die am besten geeigneten Einstellungen und Merkmale auszuwählen. Mit dieser Vision im Hinterkopf haben wir das WaveBlocks Python-Framework entwickelt, einen auto-differenzierbaren Wellenoptik-Simulator. Wir

Zusammenfassung

haben ihn für die Aberrationskorrektur mit einem SLM an Fluoreszenzmikroskopen in die Praxis umgesetzt.

Zuletzt stelle ich den Forschungsworkflow vor. Dieser basiert auf Tools und Paradigmen, die für die Projektorganisation entscheidend sind, wie GitHub, Notion und die Pomodoro-Methode sowie auf Tools, wie SLURM und guild.ai, für die reproduzierbare und organisierte Handhabung der Berechnungsanforderungen und Experimente.

Acknowledgement

The five years of my Ph.D. went through smoothly, with interesting projects and collaborating with amazing people. I would like to thank those that accompanied me on this journey.

First, I'm very thankful to my supervisor Tobias Lasser, for allowing and trusting the team and me to organize our time and energy ourselves, and, encourage a work-life balance. This flexibility allowed me to conduct the Ph.D. almost stress-free, with large motivation for research. Also, we had the luck to have a supervisor always there for us; with tight supervision, I learned so much during this period. Thank you! I would like to thank the team at the CIIP group, Jonas, Alessandro, David, Theo, John, and Erdal, for all the interesting talks, the amazing atmosphere to work, and the growing friendships. Also, to Anca Stefanoiu, with whom, since the beginning, I shared my passion for light, our project, and sharing bike rides. Even though Panagiotis Symvoulidis wasn't directly in the group, your supervisor has always been unprecedented, always willing to help, clarify, and collaborate with good energy. Thank you!

Furthermore, I was lucky to be part of projects abroad. For example, the year in Bern, where I shared a scientific journey with Fede and Yury and an amazing supervision by Paolo Favaro. With a very positive approach to the projects and life in general, the team was just on point, I had so much fun, and the trust and friendship of all of you made my stay there a fantastic adventure and allowed me to nourish my expertise further. Thank you!

Or the visit to the Marine Biology Lab, where Rudolf Oldenbourg, Amith, Geneva, and Grant welcomed me and made me feel like my second research home. So many afternoons in the quiet and windy Woods Hole and 'Pine in the sky', discussing all sorts of natural phenomena, physics, and the beauty of life. Thank you all for the fantastic time!

Living so far from home, I think daily of my family, who have endured the distance and have always been there to support me. Always received me with open arms and encouraged me to follow my instincts and live life at its maximum potency. Also, my family from other parents, my beloved friends. Muchas gracias!

And my second family here in Europe, like Max, Martin, Gonzalo and others, in those uncountable river nights, full of our ideas about the universe, the world, and ourselves, have nourished my soul and curiosity to further investigate enormously. Or my climbing friends, that are always up to some mountain adventures. Furthermore, I would like to thank Thesi, who has always encouraged me and followed me toward the adventure on the good and bad days. Thank you so much for your love and patience!

Of course that it is hard to thank everyone in one's life, but if you aren't mentioned here, don't feel bad, I am thankful to you too!

Contents

Abstract	iii
Zusammenfassung	v
Acknowledgement	vii
Contents	ix
List of Figures	xiii
List of Tables	xix
Acronyms	xxi
1 Introduction	1
1.1 Microscopy in life sciences	1
1.2 Real-time 3D microscopy in live specimens	1
1.3 Why light-field microscopy for 3D imaging?	2
1.4 The role of Computational microscopy	3
1.5 Goals of the dissertation	3
1.6 Thesis outline	4
2 Background	5
2.1 The 3D fluorescence microscope image formation model	5
2.1.1 Types of fluorescence	5
2.1.2 The image formation process	5
2.1.3 Mathematical interpretation	6
2.1.4 3D reconstruction or inverse problem	7
2.2 Scanning microscopes	7
2.3 Scan-less or single shot microscopes	8
2.3.1 Light field microscopy	8
2.3.1.1 LF and LF2.0	9
2.3.2 The Fourier light-field microscope (FLFM)	11
2.3.3 Alternative optical designs	11
2.4 3D reconstruction techniques	13
2.4.1 Deconvolution	13
2.4.1.1 The Richardson-Lucy deconvolution	14

CONTENTS

2.4.1.2	Deconvolution of LFM images	14
2.4.1.3	Deconvolution of FLFM images	15
2.4.2	Deep-learning for 3D reconstruction	15
2.4.2.1	Learning priors from the data	15
2.4.2.2	Including physical model priors	16
2.4.2.3	Pros and cons of DL for 3D reconstruction	16
2.4.3	Further DL approaches for 3D microscopy	17
2.4.4	Bayesian networks for 3D reconstruction	17
2.4.4.1	Mathematical derivation	18
2.4.4.2	Training a conditional normalizing flow	19
3	Traditional 3D reconstruction	21
3.0.1	3D deconvolution with oLaF in Matlab	21
3.0.1.1	LFM images pre-processing	21
3.0.1.2	Point Spread function simulation	21
3.0.1.3	Memory and computation optimization	22
3.0.1.4	3D reconstruction methods	23
3.0.1.5	Optical setups available	23
3.0.2	WaveBlocks and 3D deconvolution in Python	23
3.0.2.1	Explicitly computing the update step as in the Richardson-Lucy method	23
3.0.2.2	Exploiting PyTorch auto-differentiability	24
4	Deep-learning-based 3D reconstruction	27
4.1	Traditional deep learning approaches	27
4.1.1	Learning to reconstruct confocal microscopy stacks from single light field images	27
4.1.1.1	Methods	29
4.1.1.2	Experiments	40
4.1.1.3	Discussion	48
4.1.2	Real-Time light field 3D microscopy via sparsity-driven learned deconvolution	49
4.1.2.1	Methods	51
4.1.2.2	Experiments	59
4.1.2.3	Discussion	62
4.2	Bayesian learning for reconstruction and out-of-distribution detection	63
4.2.1	Fast Light-field 3D microscopy and out-of-distribution detection enabled by conditional normalizing flows	64
4.2.1.1	Methods	64
4.2.1.2	Experiments	70
4.2.1.3	Discussion	72

5	Mix of both worlds: Joint optimization	77
5.1	Using auto-differentiability for inverse problems	77
5.1.1	Wave-blocks: Learning to model and calibrate optics via a differentiable wave optics simulator	77
5.1.1.1	Methods	79
5.1.1.2	Experiments	81
5.1.1.3	Discussion	84
6	Toolkit for research project management	85
6.1	Time management	85
6.2	Progress tracking	85
6.2.1	Github project	87
6.2.2	Notion	87
6.3	Experiment repeatability	88
6.3.1	Copy and store the used code and arguments manually	88
6.3.2	The guild.ai framework	88
6.3.2.1	Running a script	89
6.3.2.2	Listing runs status	89
6.3.2.3	Visualizing in the browser	89
6.4	Hardware management in shared computed servers	90
6.4.1	Guild queues	90
6.4.2	SLURM	90
7	Conclusion & closing thoughts	93
	Bibliography	95
A	Appendix	109
A.1	LFM geometry	109
A.1.0.1	Field of view of an LF image	109
A.1.0.2	Blur of an object at the MLA plane	109
A.2	List of publications	111
A.2.1	Peer-reviewed conference submissions	111
A.2.2	Peer-reviewed journal submissions	111
A.2.3	Peer-reviewed abstract submissions and posters	111

List of Figures

1.1	Fluorescent microscope simplest design, also known as bright field microscope	2
1.2	A Fourier light field microscope	2
2.1	Geometric optics analysis of the LFM. This scheme illustrates the effect of the axial position of a point-light source on the blur produced at the MLA plane.	9
2.2	Spatial and angular LF representations. (a) Spatial representation of the LF (raw LFM image). (b) Angular representation of the LF. (c) The magnification of the region is shown in (b). The number of micro-lenses is the number of pixels in each sub-image.	10
2.3	3D reconstructions using a simulated forward projection as the measurement image. Top: Light field microscope. Bottom: Fourier light field microscope. All volumes are shown as maximum intensity projections. . .	12
2.4	Conditional normalizing flow internal functionality.	20
3.1	Pre and post-rectification of LFM image. The red lines show the rotation correction applied. Images from [1]	22
3.2	PSF storage representation of a 5×5 voxels per microlens system, where each cell is a PSF for a position in front of a single microlens. Number 6 is the central voxel. Note how we only need to compute one-fourth of the PSFs, and the rest can be computed by flipping the pre-computed PSF. . .	25
4.1	Comparison of a confocal stack scan vs. a scan with our LFM. Memory and time measurements refer to a volume with $1287 \times 1287 \times 64$ voxels. Notice how the LFM data acquisition (bottom row) is faster at capturing and reconstructing and requires much less storage than the confocal stack scan (top row).	28
4.2	Grid evaluation of the MLA-to-sensor distance in the LFM for different depths. The white dotted line is the location where the focal point on an LF-2.0 LFM [2,3] would be located. The red horizontal line shows our selected setting, equivalent to an LF-1.0 microscope, with the camera sensor placed at the focal plane of the MLA. On the right side of every plot is the depth-wise sum. The green dot in each plot indicates which configuration maximizes the corresponding metric across our selected depth range. . . .	31

LIST OF FIGURES

4.3 PSNR between Confocal and brightfield stacks: By comparing the confocal and brightfield z-stacks, we can observe how even though both get attenuated deeper into the tissue, brightfield stacks suffer from poorer signal due to scattering. 32

4.4 Focal stack comparison of a mice brain slice. 33

4.5 Data set sample acquisition and alignment: **(a)** LF image and **(b)** average of the confocal stack along the z axis. Both images are obtained by stitching multiple acquisitions of the same brain slice sample. **(c)** Single LF image tile to be aligned to the confocal volume. **(d)** Deconvolved [4] volume from the LF image in (c) averaged along the z-axis. **(e)** Correlation map between (b) and (d). The region in the correlation map with the highest peak is highlighted with a green box. **(f)** Corresponding position of the tile found in the confocal scan. **(g)** **(h)** LFM image crop aligned with 3D Confocal stack crop. The 4D LFM image and the corresponding confocal stack are then stored in the database for training. 34

4.6 Proposed LFMNet architecture. We show two different versions of the U-Net: One is the **full** version, with four down-sampling convolutions, and the other is the **shallow** version, shown in the blue dotted rectangle. The dimensions of the tensors can be found in Table 4.1. 36

4.7 Reconstruction comparison when using the **shallow** U-Net vs the **full** U-Net. Left column: input LF image (top) and ground truth volume average z projection from the confocal scan (bottom). Middle column: patch-wise reconstruction with the **full** U-Net (top) and the **shallow** U-Net (bottom). Right column: Fully convolutional reconstruction (20 ms). The reconstruction with the **full** U-Net (top) shows artifacts. . . . 38

4.8 Comparison between **(a)** the simulated MTF and **(b)** our measured MTF. Measuring the recovered frequencies from reconstructions of an USAF resolution target. 40

4.9 42

4.10 Reconstruction comparison between the proposed LFMNet, the V2C network [5] trained on simulated LFs and LF deconvolution [6]. The green line in sample 2, is used for further analysis in Fig. 4.11. 43

4.11 Brain vessel axial image comparison. **(a)-(d)** shows the projection of a blood vessel with different reconstruction methods. The projections are taken from the green line (and indicated by the white arrow) shown in Fig. 4.10, Sample tile 2. **(e)** shows the intensity profiles through the middle of the blood vessel (green dotted line in (a)-(d)) of different methods. **(f)** shows the full width at half maximum comparison. 44

4.12 Learning rate optimization: learning rate vs. MSE (in log scale for visualization). The dotted line shows the selected learning rate, where the steepest MSE change is present, as in [7]. 45

4.13 Simulated beads image comparison. **(a)-(d)** show zoomed projection of simulated fluorescent beads with different reconstruction methods. **(e)** shows the mean lateral and axial FWHM, extracted from each depth of the reconstructions generated with different methods. The missing points represent depths where no beads could be found, due to artifacts. 47

4.14 Statistical comparison of data sets. **(a)** and **(b)**: Block histograms showing the relative importance of each PCA coefficient on VGG-16 relu1 (left) and relu2 (right) features. **(c)** and **(d)**: Coefficients of variation of VGG-16 relu1 (left) and relu2 (right). 48

4.15 **Top**: Diagram of the extended field-of-view light field microscope (XLFM) used in this work. The microlens array was conjugated to the back focal plane of the objective lens through a 4-f lens pair. Excitation light from a 470 nm LED was projected on the sample through the objective lens using a dichroic mirror (DM). An sCMOS camera recorded all the sub-images formed behind the microlens array. **Bottom**: A comparison between the state-of-the-art SDLFM reconstruction (in orange) and the proposed method (in blue). Both methods first compute a sparse representation of an XLFM time series stack and later perform a 3D reconstruction of the sparse images. 50

4.16 Proposed network architecture, where the SLNet performs a sparse decomposition of an image time series and XLFMNet the 3D reconstruction of the sparse component. The number of depths can be controlled by the parameter n , and the amount of channels per convolutional layer per depth is controlled by the parameter w 51

4.17 53

4.18 Activity of a single neuron and the three frames (M_{t-100} , M_{t-50} , M_t) used as input for SLNet to compute the sparsity of frame M_t 53

4.19 XLFMNet training data generation pipeline: From left to right, 3D reconstruction is applied to every image k in a time series. The sparse component of the time series is also extracted with the SLNet and reconstructed. In the middle, for each augmentation n , both the dense and the sparse reconstructions are fed to T , where the same random transformation is applied to both volumes. The dense augmented volume R_M^n is forward projected to image space. In the last step, the dense image M_a^n and the sparse volume R_s^n are stored to train the XLFMNet. 54

4.20 **(a)** Sparseness progression along a SLNet training with $\mu = 2$. Compared against the SD result using the augmented Lagrangian. **(b)** Mean rank comparison and min/max results between SD method and SLNet with $\mu = 2$ when evaluating 12 images in the test set. 56

4.21 Comparison between SLNet trained with different μ values and the SD method based on the augmented Lagrangian. 57

LIST OF FIGURES

4.22 Spatial resolution estimation for different reconstruction methods. The top panel shows reconstructed zebrafish brain volumes’ temporal and axial max intensity projections. The bottom panel shows 3D MTFs (displayed in log scale) that show the spatial frequency support of each method. 62

4.23 The Conditional Wavelet Flow Architecture and workflow: **(a)** Dataset preparation with the SLNet [8] and performing 3D reconstructions using the Richardson-Lucy (RL) algorithm. The volume **(d)** and image **(b)** pairs are used to train the Conditional Wavelet-Flow (CWF)s. Training is performed in each CWF individually and consists in feeding V_0 and the processed condition **(b)** $\Omega_1(C)$ to the CWF 1, generating outputs V_1 and z_1 . The latter is used in loss function 2.10. V_1 is fed to the next CWF, which is trained similarly. This is repeated **(g)** until reaching the lowest resolution output. This is used to train the deterministic Convolutional Neural Network (CNN) Ω_n **(h)**. 3D reconstruction is performed by running the flow inversely. First, by reconstructing V_n with the CNN **(h)**, feed this to the last flow **(g)**, sample z_{n-1} from a normal distribution **(i)**, and run the CNF_n inversely to generate the Haar detail coefficients D_n used to up-sample the low-resolution volume V_n into V_{n-1} . This process is repeated for each up-sampling step **(g, f, e)** until reaching the desired resolution **(d)**. 65

4.24 Out-of-distribution detection workflow when using the Conditional Wavelet-Flow Architecture (CWFA) 67

4.25 Single conditional normalizing flow used within the CWFA, also present in Fig. 4.23 as Conditional Normalizing-Flow (CNF)1,2, etc. In blue, the CAT block is responsible for computing a scaling and translation from the condition and applying it to the input. 69

4.26 3D reconstruction comparison of sparse zebrafish images with different methods. On the top row, the volumes considered ground truth (GT) generated with 100 iterations of the RL algorithm using a measured PSF. On each column, a different zebrafish acquisition. The following rows show reconstructions with different methods. The left-most column shows the performance metrics used for comparison: mean absolute percentage error (MAPE), followed by the mean Pearson correlation coefficient (PCC) of 100 frames from the same fish acquisition measured on the top 50 most active neurons per fish. The bottom arrows show which direction the metrics are considered better. 71

4.27 Neural activity comparison with different methods on a single fish acquisition. In **(a)**, the MIP of the GT volume, with the top 50 most active neurons highlighted. A single reconstructed frame with different methods is in the **(b)** row. In **(c)**, the neural potentials of 3 neurons in 100 frames (10 seconds). **(d)** shows the mean PCC with four different fish and the three methods. 73

4.28 Domain Shift Detection (DSD) performance of the CWF proposed method. In (a) the MIP of the GT and predicted CWF volumes. In (b), 3 types of datasets to test the DSD. In (c), the Mahalanobis distance from a sample negative Log-Likelihood training mean to different samples per dataset, measured in training standard deviations. A threshold of $2.5 \sigma_{\text{train}}$ was used to discern out-of-distribution samples. Each CWF in the architecture has access to the NLL. Hence the Mahalanobis approach can be applied to discern OOD samples, and in (c), the output distributions of step 1 are shown. In (d), the performance of all the down-sampling is presented, where each dot represents the performance when training the CWF in a different fish. 74

4.29 Grid search for hyper-parameter tuning for up-sample step-5,, conducted with guild.ai and visualized in tensorboard. In our experiments, we optimized for the highest PSNR value. The optimized settings for each step can be found in Table. 4.7 75

5.1 Fluorescent microscope recreated with WaveBlocks. A bright-field PSF is propagated through the air by WP1, then imaged by the first camera (C1). Alternatively, the wavefront continues to the 4-f system (L1-2) with a phase mask (PM) placed at its Fourier plane. Later, the second Camera (C2) convolves the object and the PSF from the back focal plane of L2. Each camera (C1-C2) is used for a separate experiment in sec. 5.1.1.2. . . . 78

5.2 Microscope used for the experiments. See Fig. 5.1 for a definition of the optical elements involved. 78

5.3 First row: comparison between the PSF (at $0\mu m$) of an ideal microscope and the recovered PSF obtained through our approach. Second row: comparison between a constant PM pattern and the recovered one. 83

5.4 Top: Comparison of depth prediction using the initial PSF or the optimized one. Bottom: Comparison of depth prediction using the recovered PSF and either a constant PM as displayed in the SLM or the recovered PM. 84

6.1 Pomodoro cycle statistics from the last 9 months 86

6.2 Github project planning views 87

6.3 Notion project organization views. 91

A.1 Blur at MLA vs. the number of lenslets. The number of micro-lenses overlapping with the blur from an object with size $O_s = \frac{MLpitch}{M} = \frac{112}{40} \mu m = 2.8\mu m$ placed at different depths (O_1) in front of the microscope. The red lines show the depths used in our setup (-28.8 to 28.8 μm). In this range, the blur covers approximately 20 micro-lenses. 110

List of Tables

4.1	Tensor sizes for the LFMNet, whose architecture is shown in Fig. 4.6. A_i and S_i are the angular and spatial coordinates, nD the number of depths and $O_i = S_i - \text{fov} + 1$ the output spatial size.	36
4.2	Performance results for different LFMNet configurations. The top part shows ablation results. The bottom part shows the final network compared with previous work. The best results per metric are in boldface. In orange, we highlight the chosen network configuration. The testing is performed on ten full LF images with shape $33 \times 33 \times 39 \times 39$	39
4.3	3D reconstruction of different samples. 100 frames of the first sample were used for training, as described in sec. 4.1.2.1. The next 100 frames and the first 100 frames of the remaining three samples were used for testing. The neural activity sections are taken from the white areas numbered 1 to 4 for each sample.	55
4.4	Results of XLFMNet ablation study, as described in sec. 4.1.2.1. The row in orange is the setting used for our final tests, achieving the best performance in two out of the four performance metrics.	57
4.5	Generalization analysis with beads. Reconstructed with conventional deconvolution (left) and the proposed XLFMNet (right). The lateral and axial plots show the full width at half maximum for every depth and a detectability histogram, <i>i.e.</i> the number of beads found per depth with each method. The beads were not part of the training set of the XLFMNet.	61
4.6	Datasets used: 4 networks were trained, each on a single fish from the 'Used for training list.' We re-trained all networks with each fish for the out-of-distribution experiment.	68
4.7	Result of hiper-parameter optimization on each CNF step.	72
5.1	Mean and standard deviation of NMSE error between the image stack (i) captured by $C1$ and $C2$ and the ground truth stack (i^{gt}). The first row is the PSF in experiment one, and the second row is the PM in experiment two.	82

Acronyms

3D	3-dimensional.
BNNs	Bayesian neural networks.
CNF	Conditional Normalizing-Flow.
CNN	Convolutional Neural Network.
CWF	Conditional Wavelet-Flow.
CWFA	Conditional Wavelet-Flow Architecture.
DL	deep learning.
DSD	domain shift detection.
FLFM	Fourier light field microscopy.
FLFMic	Fourier light field microscope.
FWHM	full width at half maximum.
GPU	graphics processing unit.
GT	ground truth.
INN	Invertible Neural Network.
LF	light field.
LFM	light field microscopy.
LFMic	light field microscope.
LL	log-likelihood.
MAPE	mean absolute percentage error.
ML	micro lens.
MLA	micro lens array.
MTF	modulation transfer function.
NF	normalizing flow.
NFs	Normalizing Flows.
NLL	negative-log-likelihood.
OOD	out-of-distribution.

Acronyms

OODD	out-of-distribution detection.
PFLFM	polarization light field microscopy.
PFLFMic	polarization light field microscope.
PSF	point spread function.
RL	Richardson-Lucy.
ROI	region of interest.
SD	sparse decomposition.
SDLFM	sparse decomposition light field microscopy.
SLM	spatial light modulator.
WF	wave-front.
WF	Wavelet-Flow.
XLFM	extended field-of-view light field microscope.

1 Introduction

1.1 Microscopy in life sciences

Microscopic living and non-living entities significantly impact our daily life, from the food we eat, our gut microbiome, the bio-chemistry in our brains, etc. Basically influencing every aspect of human existence. And it wasn't until the invention of diffraction optics that this notion became apparent to humankind. Realizing that the world had much more than we were aware of. With the invention of the microscope, our understanding and technology started to shift and have brought us to the current state.

The influence of the optical microscope within life sciences has been essential and will continue to be, for instance, in fluorescence microscopy. For example, in applications like neural-activity analysis or spatial transcriptomics, where imaging fluorescently labeled specimens allow the spatial and functional understanding of biology and chemistry.

However, life happens in dynamical 3D environments, where capturing only a single image doesn't gather enough information for functional analysis and further understanding. This is where 3D fluorescence microscopy comes into play.

1.2 Real-time 3D microscopy in live specimens

A straightforward approach to acquire 3D information is to use a traditional fluorescence microscope, in which a sample is illuminated through a set of lenses, which then capture the fluorescence emanating from the sample and conduct the light towards a camera, as seen in Fig. 1.1. With this setup, a stack of images can be acquired by focusing on multiple focal points and capturing images. This setup focuses the illumination into a depth in the sample but also illuminates other planes where it goes through. Affecting the measurement such that the camera capture fluorescence incoming from out-of-focus planes.

Due to this disadvantage, a more common approach to acquiring volumetric information is to scan the sample volume at multiple planes or points in space and reduce the out-of-focus fluorescence. This, of course, involves more complex systems such as the confocal, light-sheet, and two-photon microscopes [9–11]. Scanning can achieve a high spatial resolution, but at the expense of time-consuming volumetric acquisitions, a substantial amount of storage usage for the acquired data, and high excitation power, resulting in high photo-toxicity and photo-bleaching of the sample.

These shortcomings motivated my work towards finding a scan-less approach capable of achieving 3D microscopy in real-time.

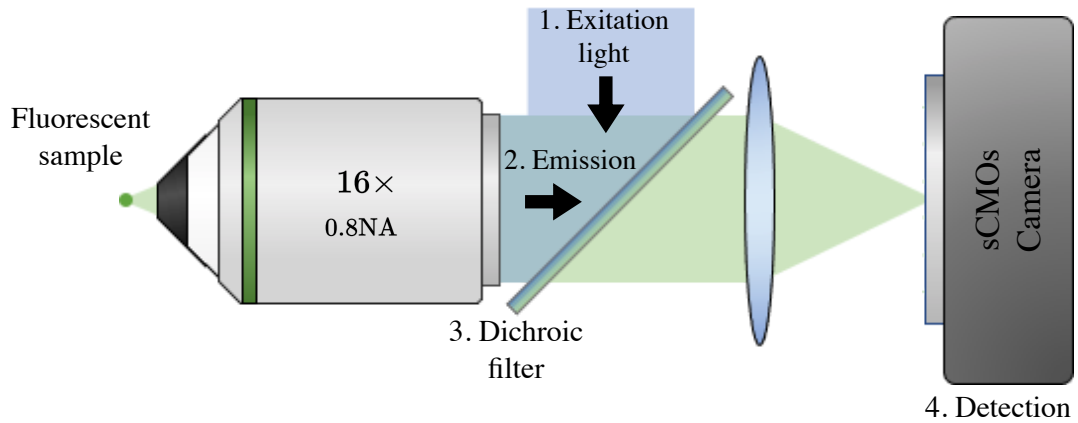


Figure 1.1: Fluorescent microscope simplest design, also known as bright field microscope

LFM [12] is a technique that turns any bright-field microscope into a scan-less single-shot 3D microscope. It offers low phototoxicity, reduced data storage, and high capture frame rates. However, it suffers from lower spatial resolution compared to scanning microscopes and from time-consuming 3D reconstruction post-processing.

1.3 Why light-field microscopy for 3D imaging?

In LFM, the key component is a MLA, which is placed in the optical path of a bright-field microscope, as seen in Fig. 1.2. The MLA embeds volumetric information about a sample by mapping its brightness for all combinations of incoming directions on a grid of locations on the camera sensor. Thus embedding 4D information (two parameters for the direction in 3D and two parameters for the location on the grid) in 2D space (the camera sensor coordinates). In the second step, the fluorescence volume is recovered from the captured LF image by using a reconstruction algorithm [6,13–15]. This approach enables advanced computational imaging applications, such as *in-vivo* calcium 3D imaging of neural activity in animals that are immobilized [16–22] or freely moving [23–26], and for imaging the scattering tissue of the mouse brain up to $380\mu\text{m}$ in-depth [18].

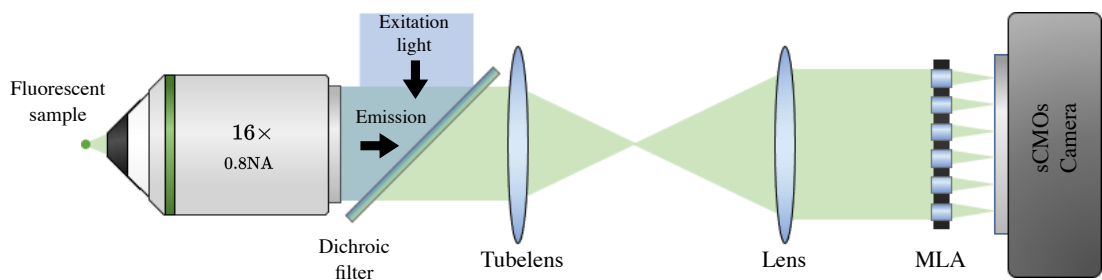


Figure 1.2: A Fourier light field microscope

The task of recovering a 3D volume from a single LF image is an ill-posed inverse problem [12], which suffers from several shortcomings, including an-isotropic lateral and axial resolutions [3, 6, 14, 27] and artifacts in the recovered volumes [4, 28]. Most volume reconstruction methods are based first on devising an image formation model, which boils down to describing the optics via a point spread function (PSF), and then on inverting such model through an optimization procedure, typically iterative, called *deconvolution* [6, 13–15]. Such reconstruction algorithms are often rather slow due to their iterative nature and the high computational complexity of the imaging model, particularly the space-variant PSF used to describe LF microscopes. To aid the reconstruction problem and further constrain the solution space, explicit statistical data priors are used, such as a Poisson distribution [29] for the measured fluorescent counts or regularization during the deconvolution process [30]. The accuracy of the mathematical model in fitting a real microscope and of the priors in capturing the statistical structure of the observed data determines the quality of the recovered volumes. Unfortunately, in practice, fitting the model and priors to the microscope and data is a challenging and sometimes impossible task, as the exact aberrations of the microscope are usually unknown.

1.4 The role of Computational microscopy

In the years previous to accessible computational power, microscopy was bounded to manual analysis and optimization, with a limit to the achievable tasks but setting a strong base into the mathematics, physics, and concepts needed for the future.

With this base in place, when computational power started to be available and continued to evolve, concepts of microscopy that couldn't be implemented before became a reality, for example, by having access to deconvolution in bright-field microscopy and being able to remove the out of focus light from captures. However, the strongest impact was on techniques where a computational method is required to access the underlying information, like scanning methods, where multiple captures and synchronization have to be orchestrated to get a 3D volume, as in confocal, light-sheet, and multi-photon microscopy. Or LFM, where accessing 3D information relies completely on a computational deconvolution.

The work presented in this dissertation is based on the computational influence on microscopy and presents novel tools and paradigms that integrate these in an even tighter matter, where joint optimization of optics and reconstruction algorithms start to become a reality.

1.5 Goals of the dissertation

In this project, we had the aim of creating computational tools to complement and aid 3D real-time fluorescence microscopy, mainly through the following incremental steps:

1. **Increase 3D reconstruction speed of LFM:** Due to the lack of simple methods capable of 3D microscopy, expanding the scannless approaches is essential.

1 Introduction

2. **Generate reliable reconstructions, suitable for biomedical applications:** Even though neural networks and other "black box" algorithms can generate good quality reconstructions, having a certainty metric that the network is doing what is trained to do is essential. Due to the low tolerance of false positives in the biomedical realm.
3. **Explore and expand the computational microscopy realm:** Having computational tools for microscopy has been of crucial importance, and now they are more integrated than ever. Pushing this frontier is a necessary and engaging activity that motivated me through the PH.D.

1.6 Thesis outline

This thesis is divided into multiple sections guiding the reader. First, in the background section, we introduce fluorescence microscopy, its image formation model, and the mathematics used to model it as an inverse problem, followed by a brief review of different reconstruction methods, such as deconvolution, deep learning, and Bayesian learning. Also, a review of computational microscopy, where computer science and microscopy integration become evident once the algorithms and microscope are optimized jointly in a task-specific design workflow. It is important to mention that in this work, I try to avoid redundancy in the mathematical background. For example, for subjects like deconvolution or DL we refer the reader to already developed works. However, a deeper mathematical insight is presented for contributions like the sparse-low rank decomposition and the CWFA.

After the background, in chapters 3 and 4, I present incremental steps toward the mentioned goals (sec. 1.5) explaining the importance of the different requirements and tackling them progressively.

Followed by a chapter presenting the basic concepts of computational microscopy and the implementation of a joint-optimization Python framework (chapter. 5) aimed towards joint optimization of optical elements (*e.g.* SLM) and algorithms towards a goal (*e.g.* 3D deconvolution).

The last chapter (6) presents a set of tools and techniques that were fundamental for the research project management during my Ph.d. E. g., time and hardware management, project planning and execution, and supervision of teams and students.

2 Background

2.1 The 3D fluorescence microscope image formation model

A 3D fluorescent microscope uses fluorescence as a contrast media to visualize biological and non-biological specimens. Fluorescence is the property of certain molecules to emit light when excited by a specific light wavelength known as the excitation wavelength. The emitted light has a longer wavelength, known as the emission wavelength, and can be captured by the microscope to form an image.

2.1.1 Types of fluorescence

In nature, fluorescence can be observed in beings like fungi, algae, marine animals like medusa, etc. And in the lab and research, fluorescence can be introduced to biological systems using the following [31]:

- Fluorescent staining: Molecules designed to attach to specific structures. Like cell walls or nucleic acid.
- Immuno-fluorescence: This technique uses highly specific binding of an antibody to its antigen to label specific proteins or other molecules within the cell, *e.g.* micro-tubules.
- Fluorescent proteins: Genetically engineered animals like zebrafish, fruit flies, *C-Elegans*, etc., can be modified to transcript a fluorescent protein reporter, like a Calcium indicator on zebrafish neurons.

2.1.2 The image formation process

The image formation, also called the forward model in a 3D fluorescent microscope, can be summarized into the following steps and visualized in Fig. 1.1:

1. Excitation: A light source, typically a laser, provides the excitation wavelength required to activate the fluorescence in the sample. The excitation light is directed onto the sample using an objective lens, which focuses the light onto a small area.
2. Emission: When the fluorophores in the sample are excited, they emit light at the emission wavelength. The emitted light is collected by the objective lens and directed to a series of filters.

2 Background

3. Filtering: The filters separate the fluorescence signal from the excitation light and other background noise. The filters typically include a dichroic mirror, which reflects the excitation light and transmits the fluorescence signal, and an emission filter, which only allows the light of the desired emission wavelength to pass through.
4. Detection: The filtered fluorescence signal is directed to a detector, such as a CCD or sCMOS camera, which converts the light into an electronic signal, later converted into an image by a computer.

2.1.3 Mathematical interpretation

The image formation process can be modeled mathematically as a linear system. This means that the fluorescence signal observed by the detector can be modeled as a linear combination of the fluorophore distribution in the sample and the system response of the microscope. Mathematically, this can be expressed as: $I = HV + n$

Where I is the observed fluorescence signal, V is the fluorophore distribution in the sample, H is the system response of the microscope, and n is the noise in the measurement.

The system response H includes all of the optical elements in the microscope, such as the objective lens, filters, and detector, and their associated transfer functions. It represents how the fluorophore distribution in the sample is transformed into the observed fluorescence measurement. The noise n includes all measurement error sources, such as background noise, photobleaching, and shot noise.

It is important to note that the image formation model is linear because it assumes that the system's response to different fluorescence levels is proportional. This means that the observed fluorescence signal is proportional to the fluorescence in the sample and that adding two fluorescence signals together results in an observed fluorescence signal equal to the sum of the individual signals.

In practice, the linearity of the image formation model may not hold exactly, and non-linear effects, such as saturation of the detector, photo-bleaching of the fluorophores, or the misalignment of optical components, may need to be considered. However, the linear model provides a good first approximation for many applications, and it can be used to obtain an initial estimate of the fluorophore distribution in the sample. And, it is often computationally implemented as a convolution operation (\otimes), which makes it highly compatible with graphics processing unit (GPU) devices.

The detection, excitation, and emission of fluorophores are often modeled as a Poisson process, as the number of excited fluorophores and photons detected on the camera are discrete numbers. Furthermore, modeling this as a Poisson distribution has desirable mathematical properties, such as being easy to manipulate and having a simple likelihood function, which makes it a convenient choice for image analysis.

2.1.4 3D reconstruction or inverse problem

The inverse action of computing a 3D volume out of one or many measured images is called 3D reconstruction, which is an inverse problem.

It consists of recovering the fluorophore distribution V by taking one or more measurement images and inverting the system matrix H . In other words, tracing the light that arrived at a pixel back to the voxels it came from. This might be a one-to-many problem, and H might be ill-posed or inaccessible as a single matrix. Hence, an iterative method is usually employed to recover V , as discussed in sec. 2.4.

As the image formation model can be modeled as a Poisson process, reconstruction methods suited for this are used. Like expectation maximization [32], maximum likelihood estimation [33], also called RL deconvolution [34,35], etc. RL is the most commonly used, further described in sec. 2.4.1. And for a deeper analysis of the image formation model and the RL deconvolution, please refer to the dissertation from A. Ștefănoiu [36].

2.2 Scanning microscopes

3D microscopes are widely available for non-moving samples, where the only temporal constraint is sample photobleaching. These include microscopes like computing axial stacks with a bright-field microscope [37], confocal [38], light-sheet [39], multi-photon [40], Fourier ptychography [41], fluorescence correlation spectroscopy [42], fluorescence lifetime imaging [43], and super-resolution microscopes [44] like stimulated-emission-depletion fluorescence [45], Structured Illumination [46], stochastic optical reconstruction [47, 48], etc.

However, the need to investigate moving specimens in 3D with high spatiotemporal resolution has fostered the development of other alternatives. For instance, modified versions of light-sheet [49], reverberation two-photon [50–52] and spinning disk confocal microscopy (SDCM, also known as Nipkow disk confocal) [53]. The latter, for example, uses a fast-rotating disc that improves acquisition speed with multiple pinholes (ca. 1000) that allow for acquisition with multiple excitation spots. This solution grants a theoretical imaging time of only 10ms per field of view. Thus, compared to laser scanning confocal, spinning disk confocal acquisition speed is increased by two orders of magnitude. Moreover, in SDCM, digital cameras are adopted as detectors, improving the speed and efficiency of acquisition compared to photo multipliers commonly implied in laser scanning confocal microscopes. Additionally, SDCM acquisition reduces photobleaching and photo-toxicity compared to laser scanning confocal, thus being a better alternative for live imaging.

Despite the significant increase in xy-scanning speed, SDCM, multi-photon, and light-sheet still require z-scanning to produce 3D images and rely on complex and expensive imaging setups with either convoluted optics or specialized fluorescent dyes. Additionally, all rely on computational post-processing for getting an actual 3D volume.

2.3 Scan-less or single shot microscopes

Regarding imaging real-time biological processes, delay matters, and scanning a sample introduces unavoidable delays. Hence, the idea behind pushing microscopes that rely on a single snapshot per frame that are scanless by nature.

To capture a snapshot of the volumetric information, two main approaches come in handy:

- Introducing a MLA in the optical path, like in LFM and FLM.
- Other types of wave-front coding. Like the Fourier Nets approach that uses an SLM.

Both approaches involve modifying the PSF of the system in some way, either to diversify the information arriving at the camera or specifically select the information relevant to the reconstruction algorithm. This is also called PSF engineering.

With scanless methods, a 3D volume is obtained from reconstruction from a single-shot acquisition, subject to a post-processing reconstruction. For this reason, compared to SDCM and the previously mentioned methods, scan-less volumetric acquisition speed is increased proportionally to the number of axial planes acquired. Thus, reducing the required storage space, sample photo-toxicity, and photo-bleaching. In practical applications, acquisition speed is only limited by the speed of the camera and by the intensity of the fluorescence signal of the sample, which impacts the required exposure time. So, for a proper time comparison, one would need to use the same camera in both LFM and the other methods.

2.3.1 Light field microscopy

Light field microscopy was first proposed by Levoy et al. in 2006 [12], as a technique that multiplexes the camera sensor between spatial and angular information and allows for 3D reconstruction with a post-processing deconvolution step. This initial approach used a measured PSF, for a single voxel behind the central micro lens (ML) and synthetically refocused it along the axial dimension to get a 3D PSF.

In later work, the reconstruction quality was significantly enhanced by Broxton et al. [6] by using a wave-optics model to simulate the LFM PSF considering multiple voxels behind a ML. Thus showing the limits of the angular and spatial resolution while increasing the sampling rate to 16 times that of the lenslet.

Unfortunately, when the tube-lens is focused on the MLA, all the angular information at the focal plane in object space is lost [2, 15] and only MLA resolution is possible (*i.e.*, only one voxel per microlens at the focal plane). Indicating that the achievable x-y resolution varies with depth, as seen in Fig. 4.8. Hence, the regular sampling used by Broxton et al. [6] introduces artifacts in the reconstructions, specifically on the focal plane. As a solution, Stefanoiu et al. [4] introduced a pre-filtering depth-dependent step that removes artifacts by allowing each depth to be reconstructed only to its maximum possible resolution.

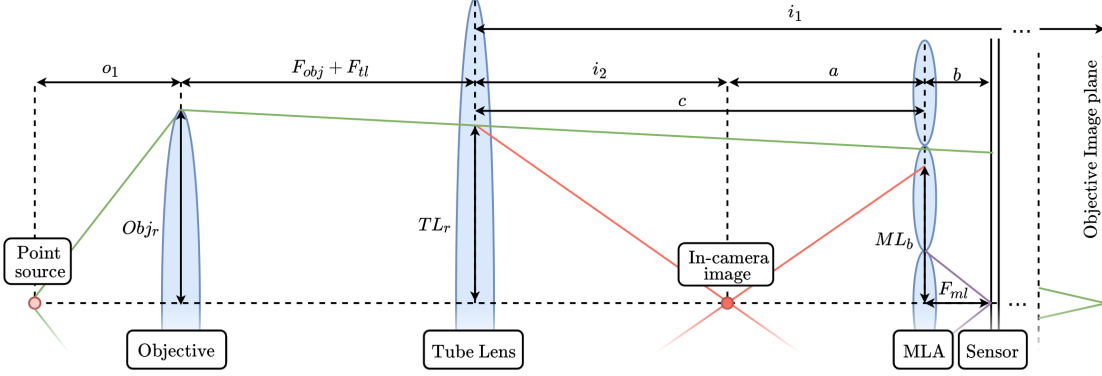


Figure 2.1: Geometric optics analysis of the LFM. This scheme illustrates the effect of the axial position of a point-light source on the blur produced at the MLA plane.

The remarkable speed of LFM makes it a powerful tool in neuroscience for high-speed neural activity imaging, especially in small transparent animal models. This capability was first demonstrated in 2014 by imaging the whole brains of *C. Elegans* and larval zebrafish [54] and further enhanced in recent years [10, 19, 21, 22, 26, 55–58]. It has also been applied to mice [17, 23, 59] and drosophila [25] neural activity imaging. Furthermore, pan-neuronal, high temporal resolution data from LFM has fueled new understandings of the principles underlying key cognitive processes, such as decision making [60].

However, two drawbacks have limited the applicability of LFM. First, compared to scanning 3D imaging methods, the inferior spatial resolution of classical LFM methods is sometimes insufficient to resolve, for instance, individual neurons in larval zebrafish brains. Second, current methods for LFM volumetric reconstruction can require days of data processing (see sec. 2.4.1).

2.3.1.1 LF and LF2.0

In the first LFM design, the MLA was placed at the focal plane of the tube lens and the sensor at the focal plane of the MLA. This setting is called LF 1.0 [61]. Georgiev and Lumsdaine [2, 3] propose instead a focused LF setting, or LF-2.0, which places the MLA and sensor relative to each other according to the thin lens equation $\frac{1}{a} + \frac{1}{b} = \frac{1}{F_{ml}}$, where F_{ml} is the focal length of the MLA, a is the distance from the microscope focal plane to the MLA, and b from the MLA to the sensor (see Fig. 2.1). This approach yields an optical configuration where images appear in focus at the sensor plane, which better aligns with the conventional microscopist notion of having the sample in focus, and allows for shifting the focal plane artifacts to less impactful depths. Li et al. [62] propose to instead use a setting such that $\frac{1}{a} + \frac{1}{b} > \frac{1}{F_{ml}}$, so that the induced aliasing could be exploited for 3D reconstruction purposes.

Format and Representation of LF Data

2 Background

This section introduces the basic notation and illustrates the technical details of an LFMic setup. A light field is a 4D function with dimensions $S_x \times S_y \times A_x \times A_y$, where the first two coordinates sample the spatial domain and the last two coordinates sample the angular domain. The 2D spatial coordinates define a position in the object space. The 2D angular coordinates define instead the angle of the ray (in the geometric optics approximation) from which the object is observed. The 2D camera sensor of the LFMic captures an image of $A_x S_x \times A_y S_y$ pixels, which we call the *spatial representation* of an LF. To map the 4D light field (LF), to the 2D sensor, a light field microscope uses a MLA inserted as shown in Fig. 2.1. The MLA has $S_x \times S_y$ micro-lenses and is aligned to the sensor so that each micro-lens covers a region of $A_x \times A_y$ pixels.

For example, in our setup (sec. 4.1.1, we set $S_x = S_y = 39$ micro-lenses (imaged by the whole sensor) and $A_x = A_y = 33$ pixels (per micro-lens). Notice that approximately A_x and A_y can be obtained by dividing the micro-lens diameter by the pixel width, *i.e.*, $112\mu\text{m}/3.45\mu\text{m} \simeq 33$ pixels.

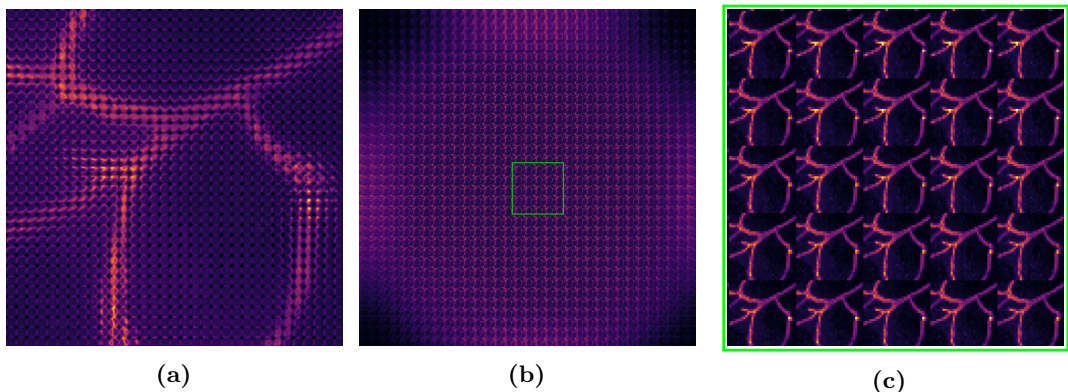


Figure 2.2: Spatial and angular LF representations. **(a)** Spatial representation of the LF (raw LFM image). **(b)** Angular representation of the LF. **(c)** The magnification of the region is shown in (b). The number of micro-lenses is the number of pixels in each sub-image.

The LF 4D information captured at the camera can be mainly visualized as a 2D image in two ways

- **Spatial representation.** This is the format with which the LFM acquires the LF, where each micro-lens samples the object from a different spatial coordinate, as shown in Fig. 2.2 (a), and every pixel inside a micro-lens captures light from the object along different angles.
- **Angular representation or perspective views.** This arrangement is achieved by gathering all the pixels at the same distance relative to each micro-lens center. Because each pixel behind a micro-lens gathers information from a different angle, the angular representation is somehow analogous to a camera array view (which we also call *perspective view*) where each camera captures a small image

from a different angle. These views are tiled as shown in Fig. 2.2 (b) and in the enlargement Fig. 2.2 (c).

Some approaches propose modifying the optics to compensate for the loss of information at the focal plane. For instance:

2.3.2 The Fourier light-field microscope (FLFMic)

The Fourier light field microscope (FLFMic) get its name from having the MLA in a conjugated plane relative to the objective back-focal plane, as in Fig. 1.2 where the emission is organized as the Fourier space, where the low-frequency components are located close to the optical axis and the high frequencies on the borders. This concept was introduced by Cong et al. [63], Scrofani et al. [64], Sung [65], and Cong et al. [66].

In the traditional LFMic, each micro-lens position samples different positions in sample space, and each pixel behind a lenslet a ray with a different angle going through that point. On the contrary, the FLFMic samples the angular space with the micro-lens position and the spatial coordinates with the pixels behind a lenslet.

This provides the FLFMic distinct advantages over the traditional LFM, when imaging fluorescent samples:

- **Space-invariant PSF:** The PSF of the FLFMic is directly capturing the angular space, making them space invariant and allowing to compute or measure a single PSF per depth, and use that PSF for all the voxels on that depth. On the contrary, LFM lenslets sample the angular space directly, making the PSF of a voxel dependent on its position relative to the center of the lenslet. Which translates to computing and storing $A_x \times A_y$ PSF's.
- **High spatial resolution:** In LFM, the spatial resolution is defined by the number of lenslets on the grid. Which is usually limited to a small number (*e.g.* 150×150). In the case of FLFM, the spatial resolution is given by the number of pixels behind a lens that, with larger micro-lenses, makes it highly optimizable to the application's necessities.

As a final remark, the design choice will depend on the priorities of the application. For fluorescent 3D microscopy, FLFM has more advantages, as spatial resolution is a priority. But for applications where angular diversity is of higher importance, such as polarization light field microscopy (PFLFM), the LFM design might be preferred. These designs propose an integral microscope without a tube lens that allows accessing the phase space and with large lenses to recover a high spatial resolution. The system and image comparison of a forward projection with both microscopes is presented in Fig. 2.3.

2.3.3 Alternative optical designs

2 Background

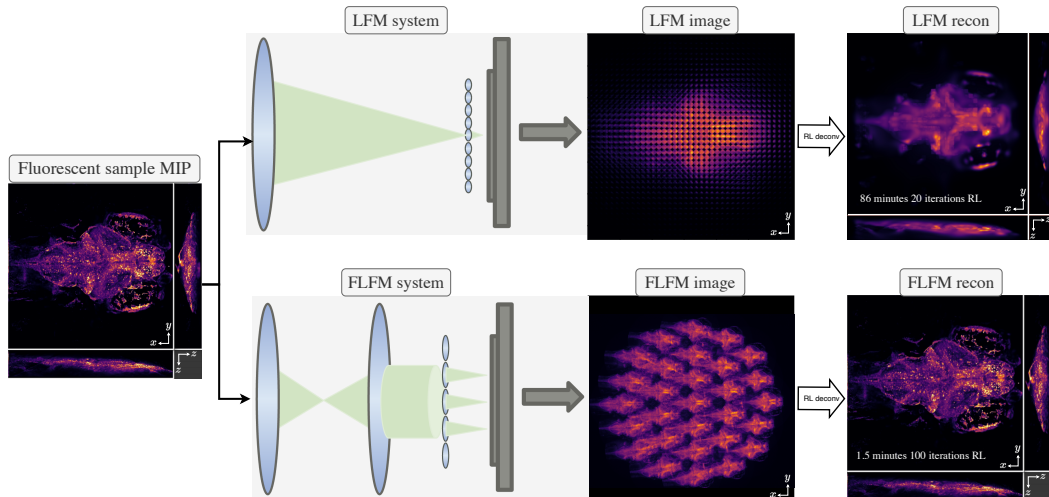


Figure 2.3: 3D reconstructions using a simulated forward projection as the measurement image. **Top:** Light field microscope. **Bottom:** Fourier light field microscope. All volumes are shown as maximum intensity projections.

Variations to LFM The drawbacks of LFM have motivated research to engineering more complex approaches. For example, Cohen et al. [13] and Wu et al. [58] propose introducing phase masks to shift the phase at the focal plane and construct a better behaved PSF.

Furthermore, there are even more complex systems that alleviate most of the above issues. An example is the work of Geng *et al.* [67], where they simultaneously capture an LF image and a bright field image to enhance the resolution at the focal plane. Another possibility is to employ light-sheet microscopy illumination to excite the specimen plane by plane and enhance the contrast and signal-to-noise ratio in the acquired images [9, 10, 56, 57]. Wagner et al. [55] proposed the ISO-LFM, which captures images with two synchronized LFMic’s positioned at 90 degrees to each other. This arrangement reduces the artifact plane to a single line in the volume. It allows a more isotropic resolution of the reconstruction but still relies on individual deconvolutions for each LFM and sacrifices temporal resolution due to the light-sheet scanning. Similarly, Zhu et al. [68] used a dual-view LFMic and DL to alleviate artifacts and slow reconstruction speed. Another approach proposed by Pan et al. [69], is the diffraction-assisted light field microscope. They introduce a diffraction grating between the specimen and the objective, reducing the loss of spatial and angular information received by the sensor. Also, a mathematical model was developed and later used to recover rigid body full-field displacement measurements. Zhang et al. [22] proposed the confocal LFM. Taking the idea from confocal microscopy, their system blocks the out-of-focus light from the specimen, achieving high axial resolution at the expense of increased exposure time. It can track a freely moving zebrafish and capture its full brain at six frames per second. Their reconstructions are performed after the acquisition stage and rely on the Richardson-Lucy deconvolution algorithm, which takes 60 seconds per volume. LFM has also been

successfully used for imaging genetically encoded voltage indicators, as in work from Quicke *et al.* [59].

There are LF microscopy works in the literature specifically aimed to recover neural activity in scattering media, for instance, the seeded iterative demixing [70], the compressive light-field microscopy approach [71], which create a dictionary of activation of individual neurons deep into scattering tissue, and use this dictionary to infer the temporal neural activity from an LF video of the specimen, similar to [72].

Wave-front coding approaches Modifying the optical path with micro-lenses, difusers, SLM etc. has a common goal, creating a PSF or system matrix H that is easier to invert, or in other words, not so ill-posed.

Some works don't utilize a MLA and use different objects and techniques. For example, the work from Kattenborn *et al.* [73] used water droplets instead of micro-lenses. The randomness in size, position, and focal length of the droplets provides a wide spectrum of frequency sampling, helping with the inversion process. Or the work from Yanny *et al.* [74], which introduces a Miniscope3D, which uses the FLFM approach, but with an optimized phase-mask instead of the MLA.

A very flexible approach to wave-front coding uses an SLM, a programmable array of micro-mirrors that shift the light phase. This allows to modify and engineer the PSF easily. The work from Waller and Manley [75] uses an optimized SLM to increase the depth of field, reduce out-of-focus blur, and improve the signal-to-noise ratio, coupled with an iterative algorithm for reconstruction. Finally, the works from Diptodip *et al.* [76] and Muthumbi *et al.* [77], are great examples of computational microscopy, where they optimize an SLM placed on the optical path together with a 3D reconstruction neural network. The former proposed the so-called FourierNets, a neural network architecture that works in Fourier space and thus has access to non-local features. This approach allows them to focus the attention of the network at different places in the volume and optimize the SLM for this case.

2.4 3D reconstruction techniques

2.4.1 Deconvolution

The reconstruction of a volume from a light field image can be cast as a deconvolution problem once the optics' PSF or system matrix H is given or estimated. The main challenge in solving deconvolution problems is that they are very sensitive to small errors in the data (an LF image in this case) and have ambiguities in the solution space. It is common to use a regularized approach to resolve these issues, where information about the solution space is used to constrain the problem better and favor only plausible solutions.

2.4.1.1 The Richardson-Lucy deconvolution

Also known as Maximum-likelihood expectation maximization, RL deconvolution [34,35] aims to maximize the likelihood of a volume given a set of measurements.

Following, a brief derivation is presented. For a full derivation of the algorithm, we refer the reader to the work of Broxton *et al.* [78], from Dey *et al.* [79] or the dissertation from A. Ștefănoiu [36].

If an object V is observed as an image I through an optical system with PSF H and affected by Poisson distributed noise, the likelihood can be written as:

$$p(i|V) = \prod_x \left(\frac{[(H \otimes V)(x)]^{i(x)} e^{-(H \otimes V)(x)}}{i(x)!} \right) \quad (2.1)$$

Where x is a 2D basis function of the image, $i = \mathcal{P}(H \otimes V)$, \mathcal{P} is a Poisson process and \otimes a convolution.

Maximizing the likelihood function in eq. 2.1 is equivalent to minimizing:

$$-\log p(i|V) = \sum_x ((H \otimes V)(x) - i(x) \cdot \log(H \otimes V)(x)) + \log i(x)! \quad (2.2)$$

Hence, we can define the function to minimize as:

$$L(V) = \sum_x ((H \otimes V)(x) - i(x) \cdot \log(H \otimes V)(x)) \quad (2.3)$$

And find a solution where $\nabla L(V) = 0$. Note that $\log i(x)!$ can be dropped, as it's a constant.

This leads to an iterative algorithm with the update rule:

$$V_{k+1} = \left\{ \left[\frac{i}{(H \otimes V_k)} \right] \otimes H^* \right\} \cdot V_K \quad (2.4)$$

Where \otimes is the convolution operator, and H^* is the adjoint (or conjugate transposed) operator of the system matrix H , also known as the backward projection operator, and maps the intensity on pixel x back to the voxels affecting it in V , with the assumption that H is normalized with the column sums equal to 1.

2.4.1.2 Deconvolution of LFM images

Multiple works have explored the PSF modeling and deconvolution in the Fourier domain [80,81] by taking advantage of the Fourier properties of wave optics. Lu *et al.* [82] developed a phase space method that deconvolves an LF image by converting it to up-sampled views and allowing a space-invariant method. This method reconstructs a volume without the zero plane artifacts and enables faster convergence against the LF deconvolution of Broxton *et al.* [6]. Nevertheless, the computational time of their algorithm is still unsuitable for reconstructing a large number of images. The same authors propose the artifact-free deconvolution method [28]. Ștefănoiu *et al.* [4] implemented an

aliasing-aware deconvolution method that adds a filtering step to [6] at every iteration of the deconvolution, which removes artifacts but still incurs a high computational time.

Even though these methods have enhanced the quality of the reconstructions, they still depend on an accurate LF PSF computation, which is difficult to achieve. Furthermore, capturing the microscope’s aberrations, misalignment, and exact parameters within the PSF is extremely challenging and hard to validate. Deconvolution is memory- and time-demanding due to the high dimensionality and complexity of the LFM data.

2.4.1.3 Deconvolution of FLMic images

However, some of the aforementioned issues get alleviated using a FLMic instead of an LFMic. For instance, a measured PSF can be used due to the space-invariance property of the FLMic PSF. Due to the convolution theorem, deconvolution can be performed in Fourier space, allowing for much faster calculations. And a single convolution is needed per depth when forward/backward projecting.

A comparison of a 3D reconstruction with both setups is shown in Fig. 2.3, using a forward projection of a fluorescent zebrafish. The PSF used for the LFMic is simulated, as acquiring it is unfeasible. And the PSF of the FLMic is measured. Note the huge quality and speed increase when using the FLMic, resulting in a $\approx 280\times$ reconstruction time increase.

2.4.2 Deep-learning for 3D reconstruction

DL approaches applied to LFMic [83] aims to learn advanced data priors from the microscope and the observed geometry without requiring an explicit mathematical model of the optics or the sample’s light scattering properties. Instead of providing a hand-crafted approximation of such a model, DL approaches capture the data prior by training a general-purpose neural network with a large data set of input-output examples. The challenges of this approach then lie in how the network is designed and trained.

2.4.2.1 Learning priors from the data

Because the LF image is a 2D rearrangement of a 4D function (two dimensions for the spatial coordinates and two for the angular coordinates), it is useful to consider its structure when designing the neural network.

Based on this principle, Wang et al. [5] proposed a network using a View-to-Channel (V2C) transformation, where the LF image is reshaped into a 1D list of views (see section 2.3.1.1). One of the main advantages of this approach is that the V2C transformation preserves a direct connection between the original structure of the input data and that of the output. Thus the complexity of the reconstruction task for the network is relatively small. However, this approach only works with a fixed LF image size. The 1D mapping of the angular domain destroys the original angular lattice, limiting the network’s learning capabilities. Furthermore, their network was trained on simulated light fields (with the model of Broxton et al. [6]) from confocal stacks and simulated micro-

2 Background

spheres, which limits the possible prior LFM information available to the network, such as noise and optical aberrations of a real microscope.

Hybrid approaches have also been proposed. For example, the DeepLFM by Li et al. [84] is based on first deconvolving the input LF image with a deconvolution method and then using a U-Net [85] to super-resolve the deconvolved volume. This method produces volumes with good accuracy, but the computational workload and time are very high due to using an LF deconvolution step.

The aforementioned drawbacks motivated our work [86] (explained on sec. 4.1.1), which aimed toward a DL approach to reconstruct confocal microscopy stacks from single LF images. To perform the reconstruction, we introduce LFMNet, a fully convolutional neural network architecture inspired by the U-Net design that considers the 4D nature of the LF to make a fully convolutional network, able to reconstruct an arbitrary number of micro-lenses. Furthermore, this work was the first to train on confocal stacks and LF image pairs, contributing with a dataset to the community [87].

2.4.2.2 Including physical model priors

A big advantage of this imaging modality is that the image formation model, although complex, is well understood. This motivated a batch of mixed approaches aiming to incorporate priors from the image formation model into DL models. For example, the work from Verinaz-Jadan et al. [88] substitutes the forward operator of the image formation model of an LFMic by a convolutional network; then, use this operator to train an adversarial network. The same group [72], proposed a neuron localization technique that uses a network to solve a convolutional sparse coding (CSC) problem to map epipolar plane images to corresponding sparse codes, where the sparse codes are individual neurons signatures. Similarly, Zhang et al. [89] proposed an artifact-free and noise-robust approach using a high-resolution dictionary to substitute the artifacted or low-resolution reconstructed zones of a 3D volume.

In terms of Fourier light field microscopy, in our work [90] (presented in sec. 4.1.2), DL has been used for recovering the spatiotemporal sparse neural activity of zebra-fish in real-time with XLFM images. First, by using a dense-sparse decomposition to recover the 2D neural activity of fluorescent zebrafish, then, used a neural network to perform 3D reconstructions.

2.4.2.3 Pros and cons of DL for 3D reconstruction

The main advantage of using DL is reconstruction speed and low memory consumption. However, this comes with strong drawbacks:

- The amount of data needed to train a DL model is quite large, and acquiring pairs of LFMic and 3D ground truth stacks requires either complex imaging systems (*e.g.* simultaneous light-sheet and light field acquisition [57], or alignment of acquired data like in [86]).

- Using DL for biomedical tasks can be risky, as there is no certainty that the reconstructed volumes contain realistic structures and that the network isn't dreaming of biologically unfeasible structures.

This motivates works like the one from Wagner *et al.* [57], where an LFMic and a light-sheet microscope were built in the same setup. This allowed them to acquire data used for training (light-sheet 3D volumes and LF images) and validate whenever the network reconstruction was deviating from the light-sheet stacks. But at the expense of a complex optical system.

2.4.3 Further DL approaches for 3D microscopy

DL is used together with different microscopy modalities to recover 3D information on biological specimens. For example, the work of Ounkomo *et al.* [91] trains networks to recover different fluorescent cellular structures from bright field focal stacks. Furthermore, the work from Huang *et al.* [92], recovers a 3D fluorescent stack from an array of arbitrarily placed focal captures, gaining speed against regular focal stack capturing and confocal microscopy. Although these approaches work well, they are unsuited for real-time imaging due to the requirement of capturing a focal stack. Wu *et al.* [93] with their Deep-Z network, use a single fluorescent image and refocus it to a maximal range of $20\mu m$; this approach is suitable for real-time imaging but with a narrow depth of field.

Another branch of 3D reconstruction and DL is the joint optimization of the microscope and reconstruction algorithm, such as the work from Diptodip *et al.* [76] or Muthumbi *et al.* [77], performing single shot 3D reconstruction using a programmable phase-mask. In the super-resolution and STORM realm, the work from Nehme *et al.* [94] becomes relevant, where they optimize a phase-mask pattern together with a neural network for 3D particle localization, reducing the number of captures needed to reconstruct a volume with an optimized PSF. Kellman *et al.* [95] used a Fourier ptychography microscope and optimized the light patterns projected into the sample together with a neural network, reducing time against conventional Fourier ptychography and similar image quality.

2.4.4 Bayesian networks for 3D reconstruction

As mentioned in the previous chapter, DL models can achieve high 3D reconstruction speeds, but at the expense of a lack of certainty on the reconstructions. The lack of certainty might render DL unsuitable for biomedical applications.

This shifts our attention to Bayesian neural networks (BNNs). These are a type of neural network that uses Bayesian learning to model the uncertainty in the parameters of the network. Traditional neural networks typically use deterministic weights and biases. BNNs extend this to probabilistic models that provide a distribution over the weights and biases of the network. In this case, we are interested in a network that directly computes the uncertainty (or likelihood) of a sample belonging to a given distribution. In other words, out of distribution detection.

Some types of BNNs and their characteristics are:

2 Background

- Generative adversarial networks [96]: Without exact likelihood computation. No OOD capabilities.
- Variational auto-encoders [97]: Optimizes the evidence lower-bound. No OOD capabilities.
- Normalizing flows [98] or invertible neural networks [99]: Access to the exact likelihood. With OOD capabilities.

NFs [98, 100] is a type of Invertible Neural Network (INN) recently used for biomedical imaging [101], inverse problems [102–105] and other applications like image generation [106, 107]. NFs learn a mapping between an arbitrary statistical distribution and a normal distribution through a set of invertible and differentiable functions. Also, a tractable exact likelihood allows for probing the reaction of this mapping for individual samples. This information, in turn, allows for deciding what to do with the new sample, perhaps retraining the network with the new data, if desired.

One disadvantage of NFs is that due to the required invertible mapping, no data bottlenecks are possible (like in an encoder-decoder approach), making it necessary to store all the tensors and gradients in memory during training. This limits the size of the data that can be used due to the limited GPU memory. The conventional solution to this issue is to split the processed tensor after each invertible function [108], feed only one part of it to the next function, and concatenate the other part to the output tensor of the NF. However, when working with large tensors, the gradients during training still overwhelm conventional GPU. Hence, Wavelet-Flow (WF) [106] were introduced, where an invertible down-sampling operation is used (such as the Haar transform) to serially down-sample the input image to the desired size (could be down to a single pixel). This allows training each down-sampling NF individually with commercial GPU, allowing their usage for high data throughput inverse problems for the first time. Generating a new image consists of running the WF backward. The lowest resolution image is sampled from an NF, then up-sampled through all the WF until reaching the original resolution.

2.4.4.1 Mathematical derivation

A normalizing flow (NF), through a sequence of invertible and differentiable functions, transforms an arbitrary distribution $p_{\mathbf{X}}(\mathbf{x})$ into the desired distribution $p_{\mathbf{Z}}(\mathbf{z})$ (usually a normal distribution, hence the name Normalizing Flows). This is possible through the change of variables formula from probability theory, where the density function of the random variable \mathbf{X} is given by:

$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{Z}}(\mathbf{z})|det(J)| \quad (2.5)$$

Where \mathbf{z} is a normally distributed random variable (mean 0 and variance 1), with probability density function $p_{\mathbf{Z}}(\mathbf{z})$. An NF can be trained by setting $\mathbf{z} = f_{\Theta}$, where f_{Θ} is an invertible differentiable function parameterized by Θ , and $J = J_{\Theta} = \frac{\partial f_{\Theta}(\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian of f_{Θ} concerning \mathbf{x} , also known as the volume correction term.

As seen in Eq. 2.5, a tractable and easily computable Jacobian determinant of $f_{\Theta}(x)$ is preferred (for example, where the Jacobian is block-triangular or diagonal). Hence, choosing the functions conforming $f_{\Theta}(x)$ is a crucial step.

A NF can be modified into a CNF and represent a conditional distribution $p_{\mathbf{X}}(\mathbf{X}|\mathbf{C})$, for a set of observations $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$ and conditions $\mathbf{C} = \{\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \dots, \mathbf{c}^{(N)}\}$. The Likelihood from Eq. 2.5 becomes:

$$p_{\mathbf{X}}(\mathbf{X}|\mathbf{C}, \Theta) = p_{\mathbf{Z}}(\mathbf{z}^{(i)} = f_{\Theta}(x, c)) \cdot |\det(J_{\Theta})| \quad (2.6)$$

Due to Bayes theorem, the posterior over the model's parameters are given by $p(\Theta|x, c) \propto p_X(x|c, \Theta) \cdot p_{\Theta}(\Theta)$. And assuming a Gaussian distribution on the model parameters. The negative-log-likelihood (NLL) and the loss function:

$$L = \mathbb{E}^{(i)}[-\log(p_{\mathbf{X}}(\mathbf{x}^{(i)}|\mathbf{c}^{(i)}))] - \log(p_{\Theta}(\Theta)) \quad (2.7)$$

$$L = \mathbb{E}^{(i)} \left[\frac{\|f_{\Theta}(x^{(i)}, c^{(i)})\|_2^2}{2} - \log|J_{\Theta}^{(i)}| \right] + \rho \|\Theta\|_2^2 \quad (2.8)$$

Where $\rho = \frac{1}{2}\sigma_{\Theta}^2$ and $J_{\Theta}^{(i)} = \frac{\partial(f_{\Theta}(x^{(i)}, C^{(i)}))}{\partial(x^{(i)})}$.

2.4.4.2 Training a conditional normalizing flow

Training a CNF involves finding the parameters Θ that maximize the likelihood:

$$(2.9)$$

Or minimize the negative log-likelihood

$$\Theta^* = \arg \min_{\Theta} \sum_{i=1}^N \left[\frac{\|f_{\Theta}(x^{(i)}, c^{(i)})\|_2^2}{2} - \log|\det(J_{\Theta}^{(i)})| \right] \quad (2.10)$$

Inference with a new sample is done by sampling from $\mathbf{z} \sim p_{\mathbf{Z}}(\mathbf{z})$ (in our case $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$) and obtaining \mathbf{x} by applying the inverse transformation of the flow: $x = f_{\Theta}^{-1}(\mathbf{z})$, as shown in Fig. 2.4.

Even though this method is mathematically sound, in practice, its lossless nature limits its usability due to high memory requirements. For 3D reconstruction of large volumes, the amount of memory needed to train a CNF is unpractical for commercially available hardware.

To alleviate this, the WF method was introduced [106], where an image is down-sampled multiple times with an ortho-normal wavelet transform (usually the Haar transform [109] due to its invertibility). This transform splits the input into four half-sized images comprising an average image and three images containing directional detail coefficients. These tensors can recover the original image when applying the inverse transformation. WF are used to learn the distribution of the detail coefficients conditioned by

2 Background

a low-resolution image. And the last down-sampling step comprises an NF that directly learns the probability distribution of the lowest-resolution image.

On the negative side, this architecture is not designed for inverse problems as it ignores the image formation model prior knowledge, such as the raw XLFM image or the point spread function. Furthermore, when training the individual NFs within the WF, the Haar transform operator generates the low-resolution image, hence an exact down-sample of the GT volume. When performing a 3D reconstruction, the low-resolution volume received by the NFs deviates from the GT. These deviations cause errors that propagate upwards through the network affecting the output 3D volume heavily. This means that the quality of the lowest-resolution image greatly impacts the final reconstruction. Given that the 3D reconstruction is performed in a succession of up-sampling operations, the lowest resolution initial guess (V_n) is fundamental.

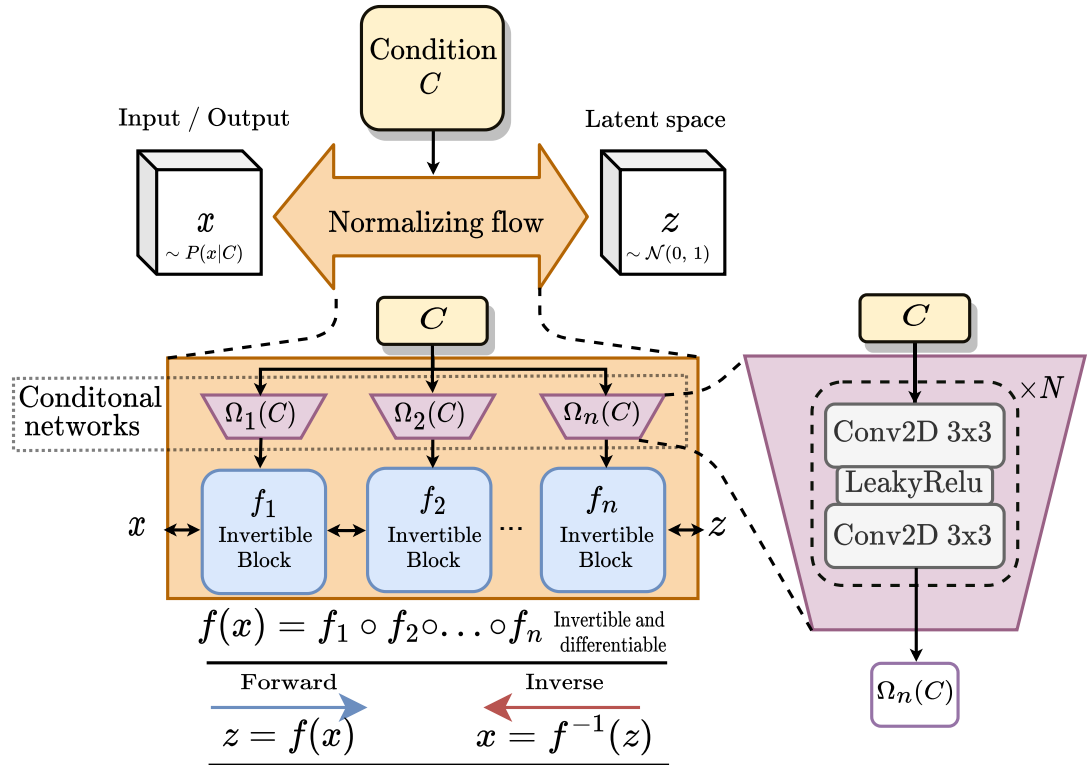


Figure 2.4: Conditional normalizing flow internal functionality.

We modified the WF architecture to account for the mentioned issues. In sec. 4.2, an implementation of a CWF for FLFM is presented, which uses a XLFM image as an input condition. It is suited for the 3D reconstruction of XLFM fluorescent images and domain shift detection (DSD). The CWF reconstruction and DSD capabilities enable a fast, accurate, and robust method for 3D fluorescence microscopy. Fast enough to be applied within close-loop or human-in-the-loop experiments, where reconstructed activity could trigger further steps in an experiment.

3 Traditional 3D reconstruction

For decades, iterative approaches have been the preferred method to solve inverse problems in fluorescence microscopy (described in sec. 2.4.1), as the underlying statistics can be included in a mathematically sound and coherent manner. However, in practice, although the LFM is an application of fluorescence microscopy, its setup is not as simple as the image formation model previously described (see sec. 2.1.3). As a LFMic is characterized by a space-variant PSF, a practical and robust deconvolution implementation is not trivial.

Here we present two approaches used and collaboratively implemented during my project to perform deconvolution in an organized and efficient way.

3.0.1 3D deconvolution with oLaF in Matlab

oLaf is a flexible 3D reconstruction framework for light field microscopy [1] written in Matlab [110], it builds up on previously released source code [54] but offers highly optimized storage and computation. It includes pre-processing routines of LF images from the Light Field Toolbox for Matlab by Donald G. Dansereau [111], point spread function computation, 3D reconstruction methods, and support for multiple LFM system types.

3.0.1.1 LFM images pre-processing

The available pre-processing present is:

- Image calibration: By using a white image (LFM image without a sample and with white light present), the centers of the microlenses are computed.
- LFM image rectification: Rotate the image to account for the rotation of the microlenses concerning the camera. And Scale such that each micro-lens covers an odd number of pixels.

Images from the pre-processing are shown in fig 3.1.

3.0.1.2 Point Spread function simulation

PSF simulation in fluorescence microscopy is commonly used, especially when measuring a real PSF is either unfeasible or the microscope is unavailable. This is particularly useful for the case of LFM, especially the LF 1.0 and LF 2.0 setups, where the PSF is space-variant, which means that the PSF at the sensor changes relative to the emitter's position. *e.g.* the center pixel of a microlens has a different refraction pattern than

3 Traditional 3D reconstruction

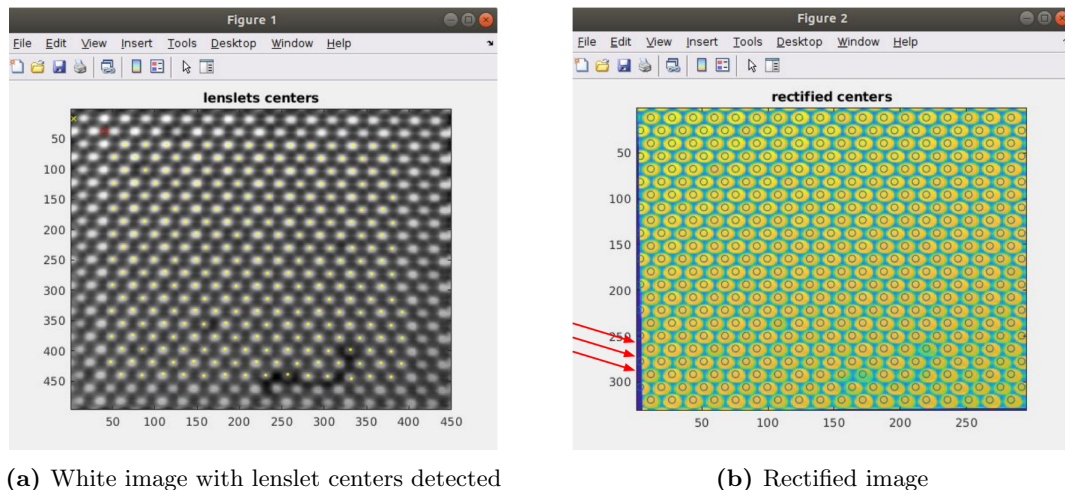


Figure 3.1: Pre and post-rectification of LFM image. The red lines show the rotation correction applied. Images from [1]

a pixel on a border of the microlens. Measuring a PSF with an LFMic would involve having a high-resolution stage that allows moving a fluorescent micro-sphere with sub-micrometer resolution in 3 dimensions. And, somehow, align it with each pixel behind the central microlens.

However, for the FLFMic, due to its space-invariant nature, measuring the PSF involves only moving a micro-sphere in the axial dimension and measuring the different PSFs.

oLaF uses a wave optics approach [6] to compute a simulated PSF. One problematic aspect is the needed storage. Imagine a microscope with a camera with 2000×2000 pixels, 33×33 pixels behind each microlens, and 50 distinct depths. This setup would generate a PSF with shape $2000 \times 2000 \times 33 \times 33 \times 50$. If stored in a *float32* array, this would take $830842\text{Mb} \approx 830\text{Gb}$ for the forward patterns and double that for both forward and backward patterns. Making it very hard to work with.

This inspired a series of optimizations in storage and computation.

3.0.1.3 Memory and computation optimization

The optimizations and changes added to the initial source code [54] can be summarized as:

- Exploit microlens pixel symmetries: Given that the objective + tube lens PSF is radially symmetric, the PSFs from voxels with the same absolute distance to the center of the microlens share a PSF. But depending on their position, they are flipped either vertically or horizontally. Allowing us to compute and store only 1/4th of the PSFs. As seen in Fig. 3.2.
- PSF pattern sparse storage: The fluorescent PSF of a light field microscope is highly sparse (mostly zero values). Hence, instead of storing all the pixels, we can

efficiently store only the non-zero pixel coordinates and their value. As the emitter moves away from the focal plane, the PSF spatial support and the non-zero pixels increases, explaining why the computing and storage load increase together with PSF depth.

- Sparse convolution: We leverage Matlab’s built-in sparse convolution by storing the PSF and the data in sparse format.

When applying these optimizations and considering that around 5% of a PSF contains information, the size of the mentioned PSF goes down to $\sqrt{0.05} \cdot 2000 \times \sqrt{0.05} \cdot 2000 \times \frac{33}{4} \times \frac{33}{4} \times 50 \approx 100 \times 100 \times 8 \times 8 \times 50$ with a storage space of 1024Mb \approx 1Gb. Providing an 800 \times space reduction.

3.0.1.4 3D reconstruction methods

In terms of reconstruction methods, oLaF supports:

- Richardson-Lucy algorithm [34, 35] as described in Sec. 2.1.4.
- Estimate-Maximize-Smooth algorithm [4].
- One-Step-Late algorithm. [112]

3.0.1.5 Optical setups available

And works with a variety of optical setups, such as:

- LFM: regular and hexagonal grid MLA placement.
- MLA focus: single-focus lenslets and mixed multi-focus lenslets.
- MLA optical placement: original 1.0, defocused 2.0, and Fourier LFM designs.

3.0.2 WaveBlocks and 3D deconvolution in Python

The WaveBlocks framework [113] surged as a seamless tool to integrate fluorescence microscopy and machine learning. This is mainly due to the auto-differentiation capabilities in frameworks like PyTorch [114] that are missing in Matlab.

Within WaveBlocks, there are two main ways of performing a 3D reconstruction:

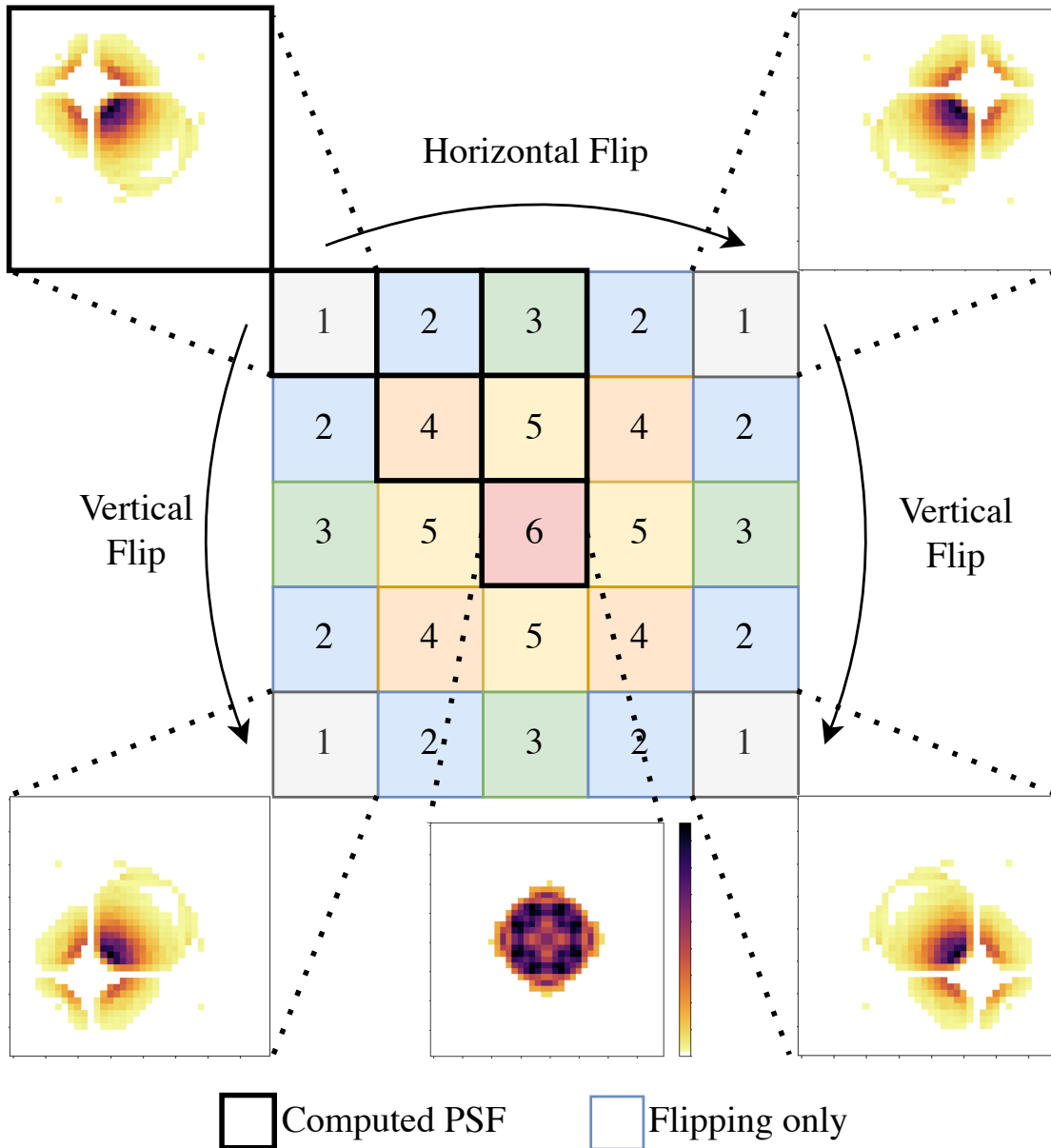
3.0.2.1 Explicitly computing the update step as in the Richardson-Lucy method

This approach follows the logic from oLaF. Where forward and backward operators are known from the PSF and are used to compute an update for the volume iteratively, as described in sec. 2.4.1.

3.0.2.2 Exploiting PyTorch auto-differentiability

This approach is much more flexible, allowing reconstructing by minimizing arbitrary cost functions (log-likelihood as in RL, mean squared error, etc.). This by:

1. Building a forward model (as shown in sec. 2.1.3) in PyTorch.
 2. Forward projecting an initial guess volume (filled with ones, for example).
 3. Computing the loss between the produced LF image and the measurement to reconstruct.
 4. Back propagating gradients of the loss function with respect to the volume.
 5. Update the volume.
 6. Repeat until reaching a stopping criteria.
-



1-6 Space-variant PSFs

Figure 3.2: PSF storage representation of a 5×5 voxels per microlens system, where each cell is a PSF for a position in front of a single microlens. Number 6 is the central voxel. Note how we only need to compute one-fourth of the PSFs, and the rest can be computed by flipping the pre-computed PSF.

4 Deep-learning-based 3D reconstruction

4.1 Traditional deep learning approaches

Traditional reconstruction methods used for 3D LFM (as described in sections 2.4.1 and 3) suffer from slow reconstruction speeds, anisotropic resolution at different depths and artifacts. Which motivated approaching the problem with DL approaches.

In this chapter, you can find implementations for reconstructing 3D volumes using LFM, and FLFM images with two novel architectures based on the U-net [85] together with a dataset of mice brain confocal 3D stacks and LFM pairs used for training. Furthermore, a sparse-low-rank decomposition network is presented, named SLNet, capable of extracting neural activity of fixed living zebrafish images. In some applications, only the neural activity is relevant, so reconstructing the whole fish is unnecessary and only makes the inverse problem harder to solve.

4.1.1 Learning to reconstruct confocal microscopy stacks from single light field images

This is an adapted version of a peer-reviewed manuscript [86] published during my Ph.D.

This work aims toward a deep-learning approach to 3D reconstruct LF images with confocal-like resolution and quality. We introduce LFMNet, fully convolutional neural network architecture inspired by the U-Net design.

LFMNet can reconstruct with near confocal accuracy a $112 \times 112 \times 57.6\mu m^3$ volume ($1287 \times 1287 \times 64$ voxels) in $50ms$ given a single light field image of 1287×1287 pixels, dramatically reducing 720-fold the capture time compared to confocal scanning of assays at the same volumetric resolution and 64-fold the required storage ($72s$ for a confocal stack scan vs. $100ms$ for an LF image capture) and a 30,000-fold decrease in reconstruction time against conventional LF deconvolution methods. On average, for our volume and image resolutions, it takes $1500s$ for a deconvolution-based reconstruction vs. $50ms$ for reconstruction with LFMNet ($30ms$ for the light field rectification, and $20ms$ to run LFMNet).

Our approach is evaluated quantitatively and qualitatively on mouse brain slices with fluorescently labeled blood vessels to prove its applicability in life sciences. Because of the drastic reduction in image acquisition time, our setup and method are suitable for imaging highly dynamic and light-sensitive events and closed-loop systems, where latency due to the reconstruction time is crucial. We provide an empirical analysis of the optical design, the network architecture, and our training procedure to reconstruct confocal-like volumes for a given target depth range. To train our network, we built a

4 Deep-learning-based 3D reconstruction

data set of 362 light field images of the mouse brain sample and the corresponding aligned set of 3D confocal scans, which we use as ground truth, freely available online [87].

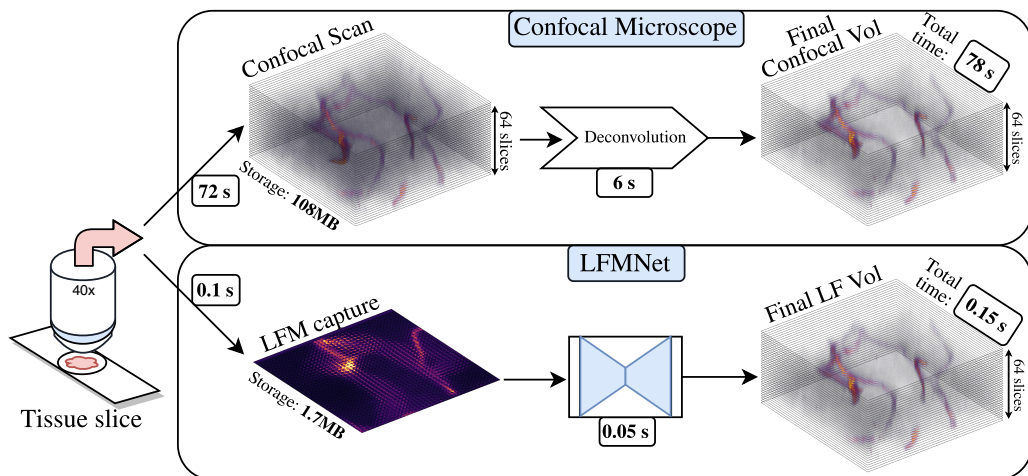


Figure 4.1: Comparison of a confocal stack scan vs. a scan with our LFM. Memory and time measurements refer to a volume with $1287 \times 1287 \times 64$ voxels. Notice how the LFM data acquisition (**bottom row**) is faster at capturing and reconstructing and requires much less storage than the confocal stack scan (**top row**).

In LFM 3D reconstruction, a simulated PSF is commonly used due to the complexity of measuring one on a real microscope. The resemblance between the mathematical model used to simulate the PSF and a real microscope determines the achievable quality of the recovered volumes after deconvolution. Unfortunately, in practice, fitting the model and priors to the microscope and data is a challenging and sometimes impossible task, as the exact aberrations of the microscope are usually unknown.

In this work, we propose to use LFM in combination with a DL approach on a non-simulated data set to address the above shortcomings. As shown in Fig. 4.1, we propose to learn through many examples an effective prior describing both the microscope and the specific data of interest while ensuring high-quality volumetric reconstructions.

We provide three main contributions: 1) a volume reconstruction method from LF images with accuracy similar to that of confocal scans and high computational efficiency compared to deconvolution methods, 2) a data set of non-simulated LF images and corresponding confocal volumetric scans [87], and 3) empirical analysis of the LFM design for confocal-like reconstruction quality. In the first contribution, we introduce LFMNet, a novel CNN architecture with a volume reconstruction accuracy close to that of confocal scans, despite using $64\times$ fewer data, and capable of producing 20 volumes per second from single LF images (for more details see section 4.1.1.1). We present an analysis of the design factors that make the architecture fully convolutional and illustrate the challenges in achieving this with 4D data. LFMNet is thus able to handle light fields with variable spatial resolutions and formats. This flexibility allows us to train LFMNet on large

batches of LF patches and test it on full images, yielding higher computational efficiency than patch-wise inference. This training strategy allows extensive data shuffling and helps prevent overfitting. We introduce a new real data set of LF images and confocal scans in the second contribution. The main advantage of training a network on real instead of synthetic data is that it can learn data prior that also incorporates the optical properties of the microscope. The central challenge in building and exploiting such a data set is that LF images and the corresponding confocal scan volumes must be aligned. We propose an automatic alignment technique that finds the position of an LF image relative to the full confocal scan and extracts the corresponding volume from it. The proposed approach does not require a time-consuming procedure to build the data set, as no manual labeling is required. Only a small number of high-resolution images is needed (362 in our data set). We illustrate the pipeline used to align the LF images to the confocal stacks in section 4.1.1.1.

Our final contribution is the empirical analysis of the LFM optical configuration to determine which settings, among a finite set, yield the best 3D reconstruction with our available hardware. The placement of the MLA and the sensor in the optical path of an LF system plays a crucial role in the amount of angular and spatial information available at the sensor, the amount of aliasing, and the achievable 3D reconstruction accuracy [2, 15, 61, 62]. We carried out an extensive empirical analysis of the possible configurations by using simulated data, and a set of performance metrics as previously done by Broxton et al. [6] and Cohen et al. [13]. This analysis allowed us to find the best configuration for the depth range of interest (see section 4.1.1.1).

4.1.1.1 Methods

In this section, we describe the main steps of our approach. First, our LFMic design criteria, followed by the sample preparation for the created data set and the procedure to acquire and align corresponding light field images and confocal stack scans. Later in the chapter, we describe the architecture of our neural network, its design criteria, and training on the data set. Finally, we describe the steps to make LFMNet a fully convolutional network. All computations were performed on a machine with an Intel Core-i7-6800K processor, two Nvidia Titan-X graphic cards, and 64 GB of RAM.

Designing a Light Field Microscope

Our LFM setup consists of a Zeiss Axio Observer microscope with a $40\times/0.9$ NA air objective. The MLA (from Flexible OKO optical) is placed in the lateral light path of the microscope and built-in regular packing (orthogonal arrangement) with a focal length of 2.5mm and $112\mu\text{m}$ pitch. The MLA is followed by a 1:1 relay lens (Edmund Optics Achromat Pair 100mm focal length) that translates the image formed by the MLA to the camera plane. Our camera is a Baumer VCXG-124M CMOS with $3.45\mu\text{m}$ square pixels. To scan the sample, a motorized stage with universal mounting frame K (Zeiss) is used, and a custom script was written in MicroManager [115], which allows the acquisition of our samples automatically.

The most fundamental element of the design of an LFM is the placement of the MLA relative to the imaging sensor and the object (see a and b in Fig. 2.1 in the appendix). To identify the best settings, we evaluate various parameters against different performance measures on simulated data. As detailed in the Experiments section, this analysis shows that **the LF-1.0 setting provides the best configuration for the depth range of interest**. We evaluate the different configurations via a uniform grid search, where the sensor is placed at 17 positions relative to the MLA (b in the thin lens equation in Fig. 2.1 in the appendix), spanning a range from $1000\mu m$ to $5000\mu m$ with steps of $250\mu m$. We then measure the frequency response of a USAF 1951 resolution target when placed at 65 different depths relative to the front focal plane of the objective, ranging from $-32\mu m$ to $32\mu m$ with steps of $1\mu m$. In this resolution target, the highest frequency is at group 9, element 3 ($645.1\text{line-pairs}/mm$ or $0.78\mu m$ line size).

To evaluate the performance of each configuration, we employ the following metrics:

- a) **Fisher Information (FI) of the Point Spread Function:** The FI matrix F measures the amount of change on the sensor response when varying the location of a source point. This performance metric was previously used by Cohen et al. [13] and measures the amount of information a PSF carries. One of the main advantages of this metric is that it does not depend on a specific deconvolution or reconstruction method.
- b) **USAF 1951 contrast and Pearson correlation coefficient:** This performance metrics are a measure of contrast on a resolution target. The former analyzes how the modulation transfer function (MTF) varies when one places the resolution target at different depths and computes how well the line pairs can be reconstructed. It has been previously used as a metric for lateral resolution [6, 13]. The contrast is given by $C = (I_{max} - I_{min}) / (I_{max} + I_{min})$, where I_{max} and I_{min} are the maximum and minimum intensities of a patch in the observed image. For this experiment, we place the target at the same depths as the computed PSFs from the Fisher Information experiment and compute the contrast and correlation coefficients against a ground truth resolution target as shown in Fig. 4.2.

Sample Preparation

The sample preparation follows three steps:

1. **Fluorescent Labeling of the Vasculature:** To label all blood vessels uniformly, we inject retro-orbitally $30\mu l$ of fluorescently labeled lectin ($1\text{mg}/\text{ml}$, DyLight 594conjugated, DL-1177) into isoflurane-anesthetized C57BL/6J mice. After an awake period of 30 minutes, we sacrifice the mice and intracardially perfuse them with 10ml of Dulbecco's phosphate-buffered saline (PBS,14190-094) followed by perfusion with 2% paraformaldehyde in PBS (PFA,30525-89-4).
2. **Brain Isolation and Processing:** We carefully isolate mice brains from the skull and store them in 2% PFA in PBS for 16 to 20 hours post-fixation. After

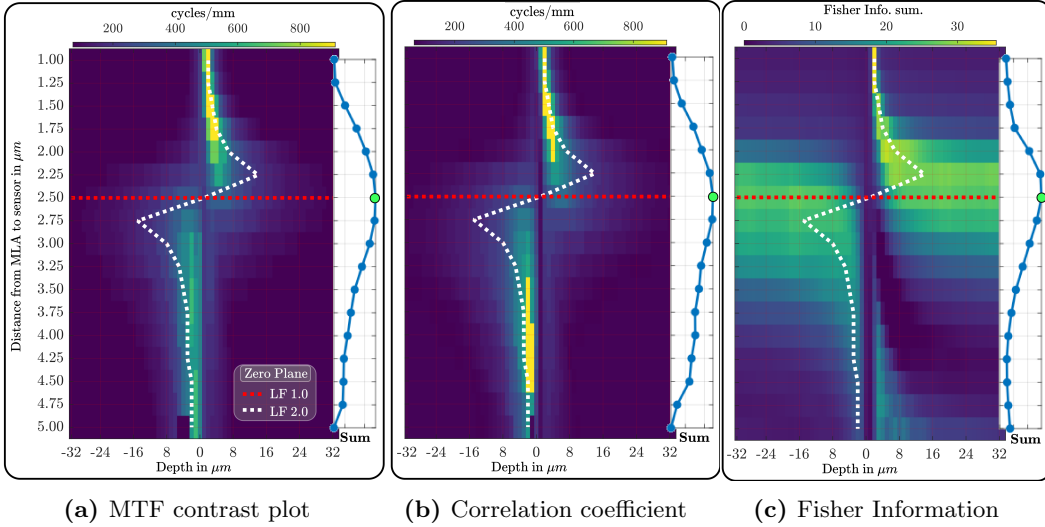


Figure 4.2: Grid evaluation of the MLA-to-sensor distance in the LFM for different depths. The white dotted line is the location where the focal point on an LF-2.0 LFM [2, 3] would be located. The red horizontal line shows our selected setting, equivalent to an LF-1.0 microscope, with the camera sensor placed at the focal plane of the MLA. On the right side of every plot is the depth-wise sum. The green dot in each plot indicates which configuration maximizes the corresponding metric across our selected depth range.

one day, we wash the brains with PBS and then store them for 20 to 24 hours in PBS. Then, we embed the brains in 2% low-temperature gelling agarose (A9414) to provide tissue stability during the sectioning. Finally, we cut $60\mu\text{m}$ thick coronal sections of the brains using a vibrating blade microtome (Leica VT1000S) and collect the slices in PBS.

3. **Sample Mounting:** To create chambers for sample mounting that protect the brain slices from mechanical compression, we glue $120\mu\text{m}$ deep microscopy spacers (S24735) onto microscopy slides. We then position single brain slices in the chamber's center and cover them with $30\mu\text{l}$ Mowiol mounting medium (9002-89-5). We then mount and seal the samples with a cover glass to allow for subsequent imaging. Before imaging, we overnight store the slides in the dark at room temperature.

Data Set Preparation

The next step is to build a data set containing pairs of confocal stacks and corresponding LFM images. One sample consists of a $60\mu\text{m}$ thick slice of the mouse brain, where the blood vessels are fluorescently labeled with tomato lectin, with excitation and emission of 592nm and 617nm , respectively. An essential aspect of our procedure is that the LFM images and the corresponding confocal scans are aligned after capture because the

acquisitions are performed on two separate microscopes. The following sections describe how the imaging and the alignment are performed.

Confocal Microscope Image Acquisition: We acquire the confocal microscope images on a Zeiss LSM 800 microscope. First, we image the full brain slice with a $10\times/0.45$ objective to create a coarse overview of the slice. Then, we select a region of interest (ROI) and scan it with a $40\times/1.3$ oil immersion objective. We image a grid of tiles per ROI, each spanning $1536\times 1536\times 64$ voxels, with a $0.087\mu\text{m}$ lateral sampling and a $0.9\mu\text{m}$ axial sampling. For the confocal data set, we acquire three areas from two brain slices, of 12×8 , 7×17 , and 17×12 tiles, respectively, with a 10% overlap between tiles. We then mark the ROI on the $10\times$ overview image for easier localization of the LFM images. We perform a deconvolution step to improve the acquired volumes' quality further. For this purpose, we run the Classic Maximum Likelihood Estimation algorithm with the Huygens Remote Manager, a web-based implementation of Huygens Core deconvolution software (SVI, the Netherlands), for 25 iterations. Finally, the tiles are stitched together by applying a concentric gradient on the borders of each tile and by adding them to the final image.

Sample scattering and depth of field: Scattering in samples is a hindering factor for fluorescence imaging when trying to image deep into the tissue, mainly when using bright field microscopes. Confocal microscopes dim this effect due to the confocal principle that blocks out-of-focus light. Nevertheless, they are still prone to low signal-to-noise ratios at deep tissue, influencing our decision on the depths to use in this project. To quantify the performance of confocal against the bright field, we compared a deconvolved bright field focal stack against a confocal stack in terms of the PSNR across depths, as seen in Fig. 4.3 and Fig. 4.4. In Fig. 4.3, it is presented how the signal quality of a bright field stack degrades towards deeper depths.

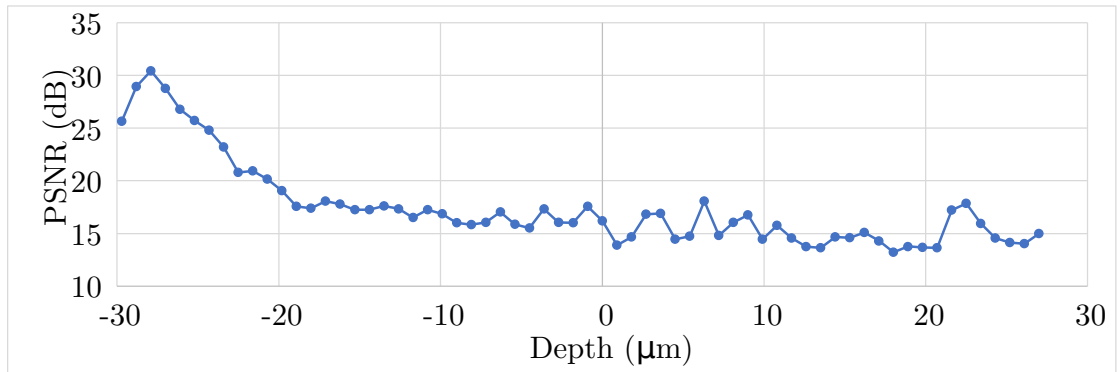


Figure 4.3: PSNR between Confocal and brightfield stacks: By comparing the confocal and brightfield z-stacks, we can observe how even though both get attenuated deeper into the tissue, brightfield stacks suffer from poorer signal due to scattering.

It is essential to mention that this scattering is present in the LF images, as LF microscopes are based on bright field microscopes. Hence, the aim of reconstructing

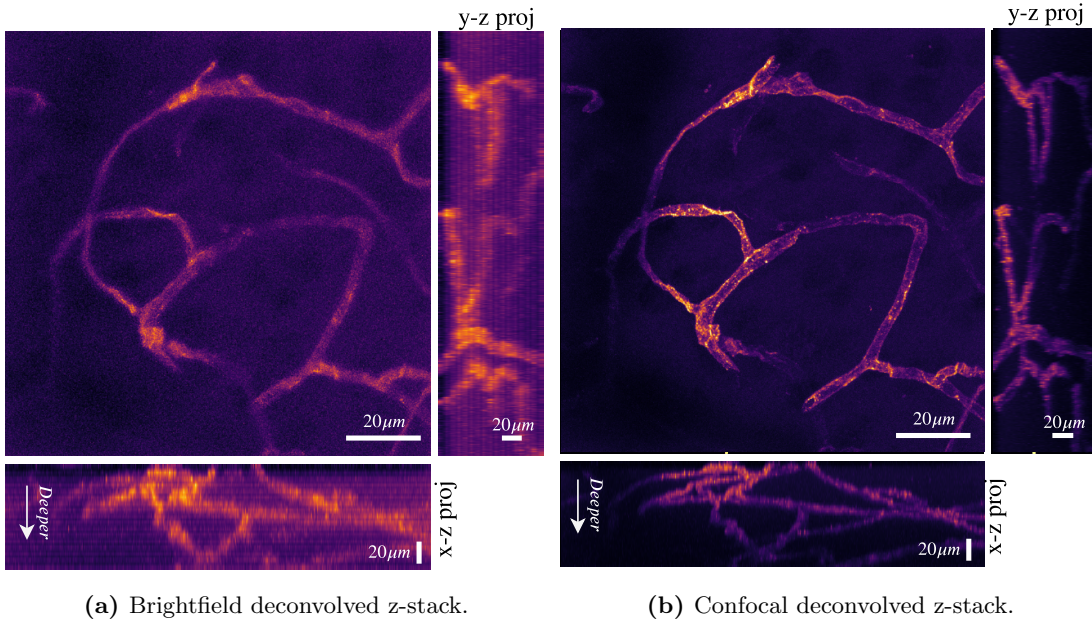


Figure 4.4: Focal stack comparison of a mice brain slice.

confocal volumes from LF images is also a de-scattering task, as the input has scattering and the output has low scattering as in confocal scans.

Acquisition and Alignment of LF Images and Confocal Volumes: In our procedure, we use the confocal stacks as a fixed reference and adjust the capture and alignment of the LFM images instead. To locate the LFM image within the region of interest of the volume scanned with the confocal microscope, we perform the following initial steps:

- Locate the region of interest through visual inspection.
- Manually center the slice in z , such that the center of the sample is focused at the MLA.
- Scan a grid of 25×25 tiles around the detected location (the scanned grid thus covers a larger area than the confocal volume).

Each tile is captured by exposing the camera for $100ms$ and covers an area of $111 \times 111 \mu m$ (with 1287×1287 pixels). The system acquires a 25×25 grid of tiles covering an area of $2.77 \times 2.77 mm$ in ~ 625 seconds (delayed mainly by the motorized stage).

Once the LFM images (with the corresponding confocal stacks) are acquired, the following automated steps are performed to align the data (see Fig. 4.5):

- **Image Rectification (Fig. 4.5 (a)):** LF images are captured with a resolution of 1287×1287 pixels on a grid of 39×39 microlenses (which correspond to the spatial resolution of the LF), each covering 33×33 pixels (which correspond to the angular resolution of the LF). For more details on the structure of LF, images

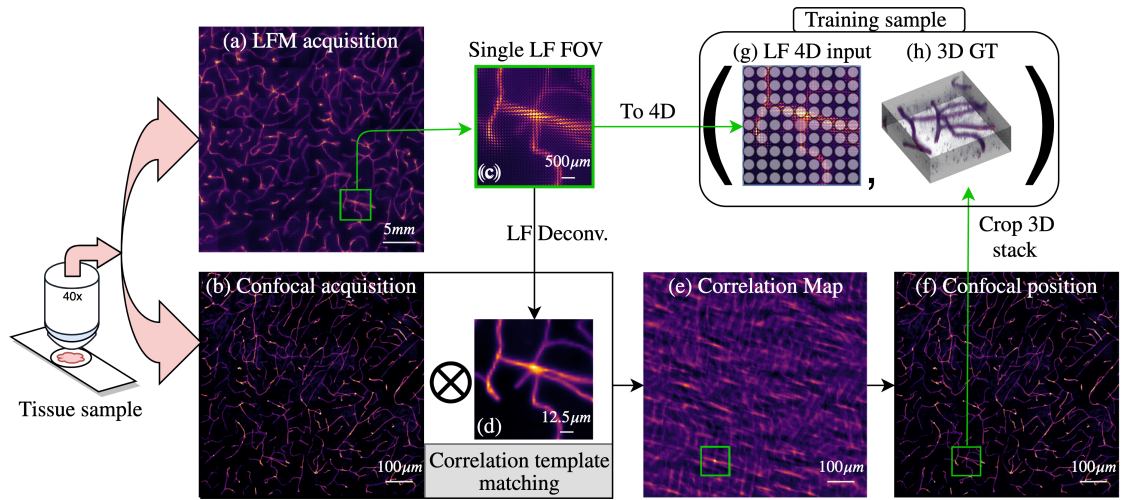


Figure 4.5: Data set sample acquisition and alignment: (a) LF image and (b) average of the confocal stack along the z axis. Both images are obtained by stitching multiple acquisitions of the same brain slice sample. (c) Single LF image tile to be aligned to the confocal volume. (d) Deconvolved [4] volume from the LF image in (c) averaged along the z-axis. (e) Correlation map between (b) and (d). The region in the correlation map with the highest peak is highlighted with a green box. (f) Corresponding position of the tile found in the confocal scan. (g) (h) LFM image crop aligned with 3D Confocal stack crop. The 4D LFM image and the corresponding confocal stack are then stored in the database for training.

see the pioneering work of Ng et al. [61]. The LF images are rectified photometrically using a captured white image and geometrically using the LF Matlab Toolbox [111]. After the geometric rectification, an integer number of pixels fits within exactly one lenslet. In our case, the originally captured LFM images with $\text{lenslet_pitch}/\text{sensor_pitch} = 112\mu\text{m}/3.45\mu\text{m} = 32.46$ pixels per lenslet are rectified to 33 pixels per lenslet. This task takes 30 *ms* per image in our system.

- **3D Deconvolution (Fig. 4.5 (c), (d)):** Each rectified image is deconvolved into a volume that spans the same axial range as the scanned confocal stack ($64 \text{ depths} \cdot 0.9\mu\text{m} = 57.6\mu\text{m}$) by using the aliasing-aware deconvolution algorithm proposed by Stefaniou et al. [4]. The reconstructions are sped up by a modified implementation of the algorithm that reconstructs a smaller number of voxels (7 voxels per lenslet, which corresponds to a lateral resolution of $0.4\mu\text{m}$). Furthermore, because the confocal microscope uses an oil immersion objective and the LFM uses an air objective instead, we compensate for the effective measured depth difference by considering the refractive index of the mouse brain and immersion oil. We use the z-step, adjusted by the ratio $0.9/1.44$, and obtain a compensated depth range of $\sim 40\mu\text{m}$.

- **Alignment (Fig. 4.5 (b)-(e)):** To align an LFM image tile to its corresponding confocal volume, first, we average both the 3D LF deconvolution volume and confocal stack (Fig. 4.5 (b)) along their z-axis. Then, we compute the correlation between these average images [116]. The method returns a 2D correlation coefficient map with a peak at the highest correlated position (Fig. 4.5 (e)). A peak is only selected to avoid false positives if its value surpasses a manually chosen threshold of 0.59 (notice that the correlation coefficient ranges between -1 and 1). The falsely classified samples, which rarely happened, were discarded through a brief visual inspection of the dataset.
- **Storage (Fig. 4.5 (g), (h)):** Finally, the LFM image tile (Fig. 4.5 (h)) is reshaped to a 4D LF and stored as a training sample into the data set together with the corresponding aligned confocal stack patch.

The final data set consists of 362 LF images composed of $33 \times 33 \times 39 \times 39$ elements and their corresponding confocal stacks with $1287 \times 1287 \times 64$ voxels and voxel size of $0.087 \times 0.087 \times 0.9 \mu m$. The data set is split so that 317 images are used for training, 35 for validation, and 10 for testing.

Deep Learning Model

We are interested in mapping LFM images to confocal stacks. This task, however, presents several challenges. Firstly, since LFM images carry less data than confocal stacks, direct mapping is ill-posed. To introduce the missing information, we exploit the fact that the space of confocal stacks has a limited complexity and use neural networks to capture such structure. Secondly, because of the optical arrangement, LFM images do not capture the same amount of information at every depth (*i.e.*, the effective volume slice resolution) [15]. In particular, the angular resolution is the lowest at the depth corresponding to a focused image on the MLA. In fact, aberrations and misalignment can be beneficial because they distort the sampling patterns defined by the ideal system and thus avoid degenerate imaging conditions. These aspects have been exploited, for example, by Li Yi et al. [117]. However, modeling precisely aberrations, misalignment, and other optics parameters is challenging, and errors can result in strong artifacts.

Thus, we address these challenges by taking advantage of the data-driven approach of deep learning (*i.e.*, no explicit modeling is required). We train a neural network with real data and aim at learning a good prior about both the microscope setup and the data domain.

Design Criteria: We call our proposed network LFMNet. It receives as input an LF image in the tensor format $1 \times A_x \times A_y \times S_x \times S_y$, where S_x and S_y are the spatial dimensions and A_x and A_y the angular dimensions (see also section 2.3.1.1 in the 2 chapter). The first layer of LFMNet is a 4D convolutional layer (more details will be presented in section 4.1.1.1). The output of this layer is denoted $T1$ (see Fig. 4.6), with shape $nD \times A_x \times A_y \times O_x \times O_y$ where nD corresponds to the number of depths of the reconstructed volume encoded in the channel dimension and $O_{x,y} = A_{x,y} - \text{fov} + 1$, where the field of view fov is discussed in section 4.1.1.1. $T1$ is then converted to a 2D image

Tensor	Dimensions (ch,dim1,dim2,..)
4D LF in	$1, A_x, A_y, S_x, S_y$
T1	nD, A_x, A_y, O_x, O_y
T2 and Vol Out	$nD, A_x \cdot O_x, A_y \cdot O_y$
D1 and U3	$nD, (A_x \cdot nT)/2, (A_y \cdot O_y)/2$
D2 and U2	$nD, (A_x \cdot O_x)/4, (A_y \cdot O_y)/4$
D3 and U1	$nD, (A_x \cdot O_x)/8, (A_y \cdot O_y)/8$
D4	$nD, (A_x \cdot O_x)/16, (A_y \cdot O_y)/16$

Table 4.1: Tensor sizes for the LFMNet, whose architecture is shown in Fig. 4.6. A_i and S_i are the angular and spatial coordinates, nD the number of depths and $O_i = S_i - \text{fov} + 1$ the output spatial size.

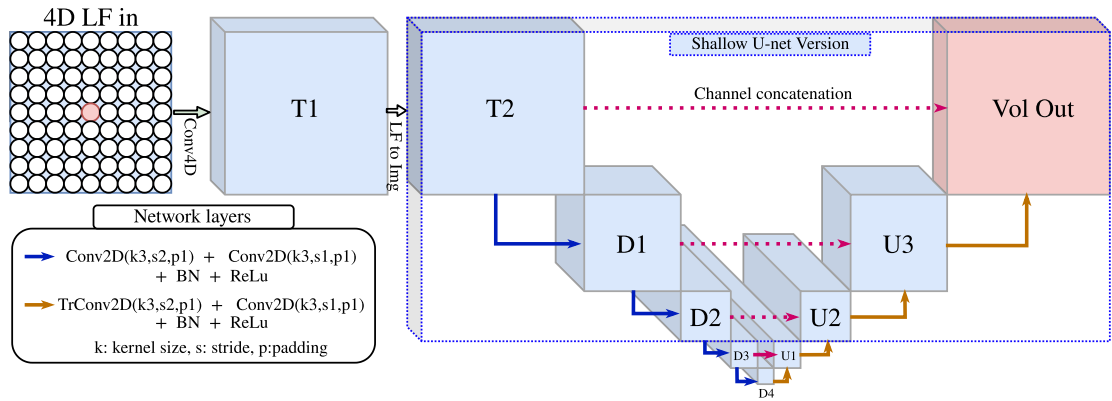


Figure 4.6: Proposed LFMNet architecture. We show two different versions of the U-Net: One is the **full** version, with four down-sampling convolutions, and the other is the **shallow** version, shown in the blue dotted rectangle. The dimensions of the tensors can be found in Table 4.1.

$T2$ through the mapping $nD \times A_x \times A_y \times O_x \times O_y \mapsto nD \times A_x O_x \times A_y O_y$. Then, a modified U-Net [85], which employs convolutions as down-sampling operators, finalizes the feature extraction and 3D reconstruction. The output shape is $nD \times A_x O_x \times A_y O_y$. Furthermore, because of its fully convolutional design, our network can reconstruct the volume behind single or multiple lenslets. The complete model is shown in Fig. 4.6

Tiles vs Full Images and a Fully Convolutional Network: We are interested in designing a fully-convolutional LF network, meaning a network that can process any shape of input LF and reconstruct a volume corresponding to the input. This would allow us to train the network with patches of the LF data, increasing the possibility of data shuffling and augmentation, and use the network with full LF images at inference time.

The LFMNet aims to reconstruct the volume behind a group of lenslets, but 2D lenslet image patterns are hardly exploitable by a 2D convolution, meaning that consecutive pixels in the image don't necessarily correspond to consecutive voxels in the volume;

this is due to the 4D nature of light fields. Hence, we use a 4D convolution as the input layer, which processes the spatial coordinates of the LF image microlens by microlens and has a kernel shape just large enough to fit the neighborhood around our volume of interest. The kernel size of this layer is $3 \times 3 \times \text{fov} \times \text{fov}$, where fov is the size of the lenslet neighborhood used as input (as also discussed in section 4.1.1.1), with a padding equal to $1 \times 1 \times 0 \times 0$, and a stride equal to 1 in every dimension. All other layers are already convolutional and do not require an adjustment for processing any input size.

Receptive Field Analysis: Another important factor for the generalization from patches to full LF images is the receptive field of the network. The receptive field tells us what elements of an input to a layer can potentially affect an output. This information is useful to ensure that the receptive field is large enough to include all the required information to produce the output. In the case of LF microscopy, the image of a voxel is spread in a neighborhood because of the PSF, and its size can then be used to define the minimum receptive field. However, we explain below that special care must be taken at the input boundaries.

A single convolutional layer has a receptive field that depends on its kernel size and the stride (the down- or up-sampling factor). When stacking several convolutional layers, the overall receptive field depends on their connectivity, as shown by Dumoulin and Visin [118]. Our analysis and experiments show that the output might change when LFMNet is fed the same input but with different neighborhood sizes. This effect is undesirable as we want to train the network with the smallest possible size to increase diversity and limit overfitting. Still, we want to use the trained network on full LF images for computational efficiency. Suppose the receptive field of LFMNet is larger than $A_x \times A_y$ of the input LF image region used for training. In that case, the convolutional layers will fill in the missing data at the boundaries with reflecting conditions. However, if at test time, the input LF image is larger than the receptive field of LFMNet, then the reflecting conditions are not applied, and thus the network may produce an unpredictable outcome. We observe that the section of the U-Net closest to the bottleneck (see D3, D4, and U1 in Fig. 4.7) contributes greatly to the broadening of the receptive field.

We experimented with the following settings (see Fig. 4.6 for more details and note that the shallow LFMNet is within the blue dotted box)

1. **Shallow** U-Net with two down/up-sample operations and a receptive field of 19 pixels.
2. **Full** U-Net with four down/up-sample operations and a receptive field of 104 pixels.

In Fig. 4.7, we evaluate both settings (shown on different rows) under two modes of operation (shown on different columns). In the first mode of operation, we reconstruct the full volume by processing LF patches independently (each patch has a size of $33 \times 33 \times 9 \times 9$) and then by stitching the patches together to form the output volumes. In this case, both networks produced similar results. In the second mode, that is, when we reconstruct the full volume by processing a full LF image ($33 \times 33 \times 39 \times 39$ + padding of 4 in the last two dimensions), only the network with the limited receptive field (the

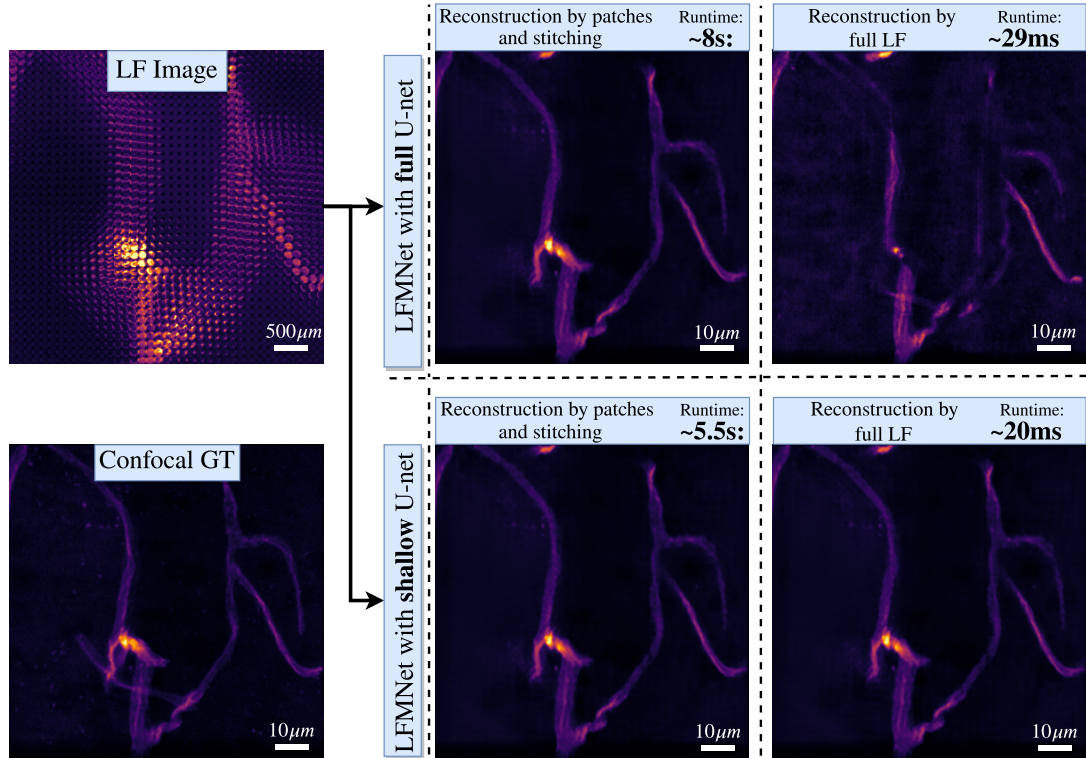


Figure 4.7: Reconstruction comparison when using the **shallow** U-Net vs the **full** U-Net. Left column: input LF image (top) and ground truth volume average z projection from the confocal scan (bottom). Middle column: patch-wise reconstruction with the **full** U-Net (top) and the **shallow** U-Net (bottom). Right column: Fully convolutional reconstruction (20 ms). The reconstruction with the **full** U-Net (top) shows artifacts.

shallow U-Net in the bottom row in Fig. 4.7) maintains the volume behind each lenslet untangled from its neighbors, and avoids artifacts. Moreover, this mode of operation can exploit parallel processing more efficiently than the patch-wise mode and produce the full output in just 20ms. Hence, we use the **shallow** U-Net for our LFMNet.

Field of View of an LF Image:

Our analysis showed that for the desired depth range, a maximum field of view of 21 lenslets is needed (see section 2.3.1.1 in the background chapter) (see Fig. A.1 in Appendix A.1.0.2). In our ablation study, we compare the LFMNet reconstructions when using fields of view with sizes 21×21 , 15×15 , and 9×9 lenslets. We find that the 9×9 fov yields the highest performance (see Table 4.2). Also, it comprises the best trade-off between run time, data overfitting, and minimum field of view requirements.

4.1 Traditional deep learning approaches

Table 4.2: Performance results for different LFMNet configurations. The top part shows ablation results. The bottom part shows the final network compared with previous work. The best results per metric are in boldface. In orange, we highlight the chosen network configuration. The testing is performed on ten full LF images with shape $33 \times 33 \times 39 \times 39$.

Model definition	Train LF input shape	Train volume output shape	Full LF PSNR (db)	Full LF SSIM	Train time ($sec \times img$)	Full LF Test time ($ms \times img$)
LFMNet U-Net receptive field test (trained on 27 images), (section 4.1.1.1)						
Shallow U-Net, no SC	33,33,9,9	33,33,64	28.86	0.69	3.58	20
Full U-Net, no SC	33,33,9,9	33,33,64	22.49	0.55	4.66	29
U-Net with skip connections (SC), (trained on 27 images)						
Shallow U-Net, SC	33,33,9,9	33,33,64	24.03	0.56	3.82	20
Full U-Net, SC	33,33,9,9	33,33,64	26.78	0.66	4.25	29
LFMNet Input field of view (fov), (trained on 27 images) section 4.1.1.1						
Shallow U-Net, no SC, fov = 15	33,33,15,15	33,33,64	25.61	0.60	9.70	29
Shallow U-Net, no SC, fov = 21	33,33,21,21	33,33,64	28.31	0.67	9.59	25
Number of microlenses to reconstruct (nT), with fov=9, (trained on 27 images) , section 4.1.1.2						
Shallow U-Net, no SC, fov = 9, $nT = 3$ MLAs	33,33,11,11	99,99,64	29.56	0.70	10.27	20
Shallow U-Net, no SC, fov = 9, $nT = 5$ MLAs	33,33,13,13	165,165,64	29.12	0.69	23.97	23
Final LFMNet design and comparison to other methods (trained on 317 images)						
Shallow U-Net, no SC, fov = 9, $nT = 3$ MLAs	33,33,11,11	99,99,64	34.45	0.87	5.81	20
Shallow U-Net, no SC, fov = 9, $nT = FullLF$	33,33,39,39	1287,1287,64	27.27	0.58	0.48	20
V2C Net [5]	33,33,39,39	1287,1287,64	31.27	0.82	0.60	160
V2C Net Patch	33,33,11,11	99,99,64	30.76	0.80	5.00	160
Deconvolution [4] 5 iterations	33,33,39,39	1287,1287,64	28.64	0.60	-	1500s

4.1.1.2 Experiments

LFM hardware Design Validation The analysis described in section 4.1.1.1 focuses on the effect of moving the sensor and the MLA within the LFM and proposes several performance metrics to evaluate the different configurations. **PSF validation:** The first step in our analysis is to validate our model of the PSF, which is based on that of Stefanoiu et al. [4], against experimental data by looking at the MTF (MTF). To do so, we first measure the MTF of our LFM setup by using the frequency responses of groups 7-9 from the USAF 1951 resolution target. Then, we compare it to the simulated MTF from our model after inserting all the calibration parameters. Because of space limitations. The measured and simulated MTFs are shown in Fig. 4.8. The simulated MTF (see Fig. 4.8 (a)) has a slightly higher response at all frequencies than that of the measured MTF (see Fig. 4.8 (b)). This mismatch is due to the inaccuracies of the LFM model, mostly caused by the relay optics. The important factor in this analysis is the matching focal plane between the simulated and the real MTF.

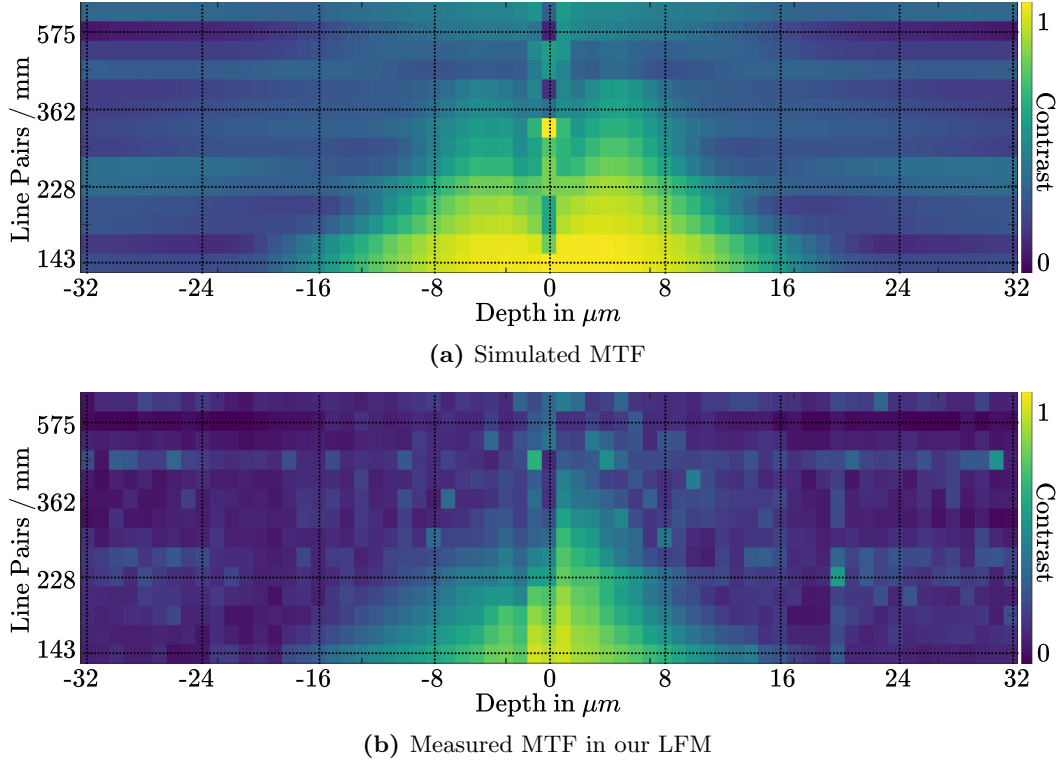


Figure 4.8: Comparison between (a) the simulated MTF and (b) our measured MTF. Measuring the recovered frequencies from reconstructions of an USAF resolution target.

Finding the best MLA placement: After validating our PSF model, we can evaluate the LFM setup. In Fig. 4.2, we show each performance metric for a wide range of settings

in the form of a heat map: For each setting, the heat color corresponds to the highest frequency with at least 80% of (a) contrast, (b) correlation and (c) Fisher Information. Thus, bright (yellow) colors correspond to high frequencies, which should yield high-resolution reconstructions, and dark (blue) colors correspond to low frequencies, which imply a poorer reconstruction quality. We also highlight with a white dotted line the optimal MLA distance for each object depth according to the LF-2.0 design [2, 3]. Note how the LF-2.0 setting follows approximately the profile with the highest peak (*i.e.*, best high-resolution reconstruction) for each MLA distance.

Such a setting would be particularly useful when imaging thin volumes, but this does not apply to us, as our setup acquires a volume spanning $57.6\mu m$. On the right-hand side of each heat map in Fig. 4.2 (a)-(c), we plot the average performance across the depths for each MLA placement. The maximum value across all object depths (marked with a green dot) indicates the best MLA setting for our LFM. This choice matches the conventional LF-1.0 design [2, 3], and thus we used it for our final setup.

Even though the extent of the PSF at the MLA with our setup theoretically covers ~ 20 micro-lenses (as discussed in section A.1.0.2), having such a large input (and 4D convolution kernel) hampers the training time considerably and might incur more easily overfitting.

Network Ablation and Performance Evaluation

In this section, we evaluate the effectiveness of our method on the design criteria introduced in section 4.1.1.1, by comparing side-by-side the two U-Net configurations (**shallow** and **full**), the use of skip connections in the U-Net architecture, the different field of view choices (**fov**) and the volume size used for training, denoted by $A_x nT \times A_y nT \times nD$, where $nT \in \{3, 5\}$. The networks in Table 4.2 are trained on a subset of the data formed by 27 images (41,067 patches), validated on 3 images (4,563 patches), and tested on 10 full LF images ($33 \times 33 \times 39 \times 39$). The quality measures of reconstruction are the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [119]. Also, we show the training and reconstruction times for a single LF image. As a result of our ablation, the design that yielded the highest image PSNR and SSIM (see 4th and 5th columns in Table 4.2) is

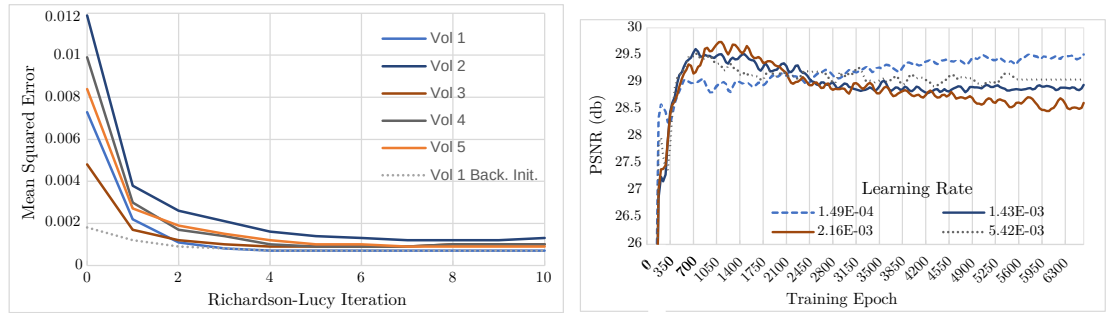
*The **shallow** U-Net without skip connections, trained with $33 \times 33 \times 11 \times 11$ input LF patches and with an output volume of $99 \times 99 \times 64$ voxels. This corresponds to the volume behind 3×3 microlenses (nT), given a $fov = 9$ per microlens.*

Next, we compare the LFMNet with these settings (highlighted in orange at the bottom of Table 4.2) against the V2C approach [5], the V2C trained on patches, the aliasing-aware deconvolution [4] and the LFMNet trained with full LF images. As the original V2CNet employs around 27 million parameters, we use a modified version with 8.5 million parameters, matching the amount present in the LFMNet. All networks are trained with 370 LF images (LFMNet with real LF images and V2C net with simulated LF images as in [5]) and tested on the 10 LF images from the separate test set. The

4 Deep-learning-based 3D reconstruction

LFMNet trained with full images achieves lower performance than when training with patches, showing the advantage of using patches over full images. For details about the deconvolution initialization and the number of iterations, refer to sec. 4.1.1.2 and Fig. 4.9a.

Deconvolution initialization and number of iterations: Algorithmic optimization performance depends heavily on the hyper-parameters used, in the case of Richardson-Lucy deconvolution, the only parameters are the type of initialization and the number of iterations. In this work, we used five iterations and initialized the reconstructed volume with 1’s to be comparable to previous works [4, 6]. However, when initializing with a backward projection, as seen in Fig. 4.9a “Vol 1 Back. Init.” starts with a smaller error and converges faster. Nevertheless, it converges to the same value as the “Vol 1” with initialization equal to 1. We conclude that either of these initialization works for comparison, as at iteration 5, both produce a similar error.



(a) Deconvolution of 5 different volumes: Vol 1-5 where initialized with a volume set to 1 and “Vol 1 Back. Init.” with a backward projection of the LF image. (b) V2CNet PSNR comparison when using different learning rates. The dotted line represents the optimal learning rate [7].

Figure 4.9

LF images reconstructed with the LFMNet, V2C, and deconvolution are qualitatively compared in Fig. 4.10 (zoom-in may be required to see differences). A slice through a single vessel is depicted in Fig. 4.11. Our proposed method provides overall a 75,000 fold improvement in reconstruction time against deconvolution and a higher reconstruction accuracy in terms of PSNR and SSIM.

The networks were trained using Adam optimizer and mean squared error as a loss function. The learning rate of both LFMNet and V2CNet was optimized with the cyclical learning rates method from Smith [7], resulting in a learning rate of 1.08E-3 and 1.49E-4 for LFMNet and V2CNet. For further detail on the learning rate optimization, refer to sec. 4.1.1.2.

Learning rate and number of parameters:

The choice of a learning rate in CNN directly impacts its performance. To use the optimal learning rates for LFMNet and V2CNet, we employed the cyclical learning rates method from Smith [7] implemented in the Python package pytorch-lr-finder.

4.1 Traditional deep learning approaches

	Sample tile 1	Sample tile 2	Sample tile 3
LFM image			
Confocal GT			
LFMNet recon	PSNR: 37.48SSIM: 0.87 	PSNR: 38.26SSIM: 0.87 	PSNR: 38.35SSIM: 0.87
V2C recon.	PSNR: 27.15SSIM: 0.76 	PSNR: 27.08SSIM: 0.75 	PSNR: 29.02SSIM: 0.79
Deconvolution	PSNR: 27.20SSIM: 0.56 	PSNR: 28.79SSIM: 0.56 	PSNR: 29.10SSIM: 0.61

Figure 4.10: Reconstruction comparison between the proposed LFMNet, the V2C network [5] trained on simulated LFs and LF deconvolution [6]. The green line in sample 2, is used for further analysis in Fig. 4.11.

4 Deep-learning-based 3D reconstruction

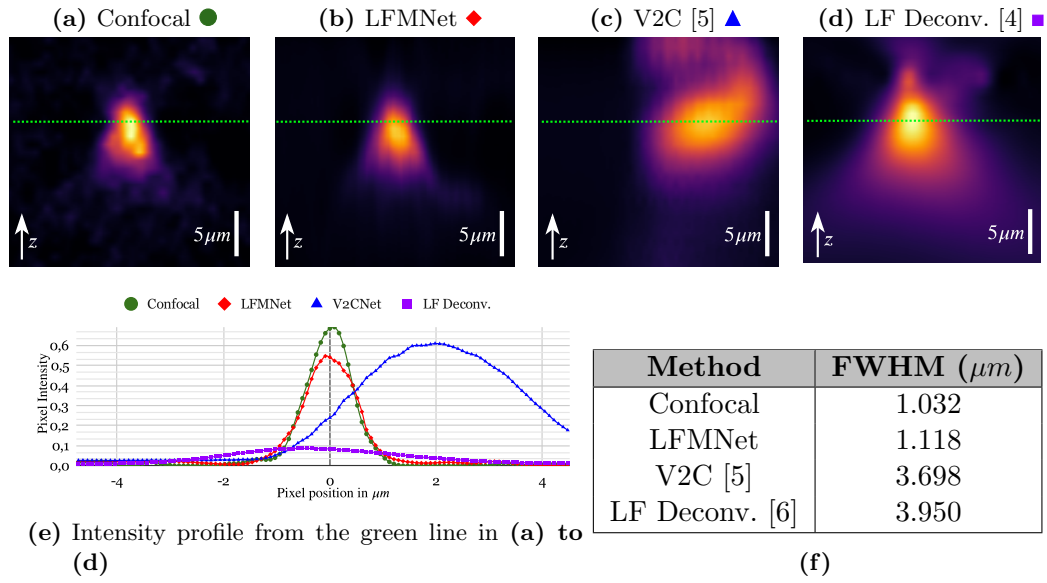


Figure 4.11: Brain vessel axial image comparison. (a)-(d) shows the projection of a blood vessel with different reconstruction methods. The projections are taken from the green line (and indicated by the white arrow) shown in Fig. 4.10, Sample tile 2. (e) shows the intensity profiles through the middle of the blood vessel (green dotted line in (a)-(d)) of different methods. (f) shows the full width at half maximum comparison.

This algorithm searches for a learning rate in a range $R : [lr_{min} : lr_{step} : lr_{max}]$ in the following manner:

```

1 # Define storage for delta = change of error
2 deltas = {}
3 for i in range(max_epochs):
4     # Define the current learning rate
5     lr_curr = lr_min + i*lr_step
6     # Train the network for one iteration
7     net.train()
8     # Infer
9     prediction = net(input)
10    # Measure the mean square error between the predictions and the GT
11    mse_i = MSE(Gt, prediction)
12    # Measure the change in error between
13    delta_mse = mse_i - mse_i1
14    mse_i1 = delta_mse
15    deltas[delta_mse] = lr_curr
16 # Choose the lr producing the largest delta_mse
17 lr = max(deltas)

```

Listing 4.1: Learning rate search

4.1 Traditional deep learning approaches

The optimized learning rate is established by training a network with an incremented learning rate in every epoch, then selecting the learning rate that produces the steepest error between GT and prediction.

The optimization of LFMNet and V2CNet is presented in Fig. 4.12 a) and b) respectively. Note that the MSE is shown in log scale, for visualization purposes.

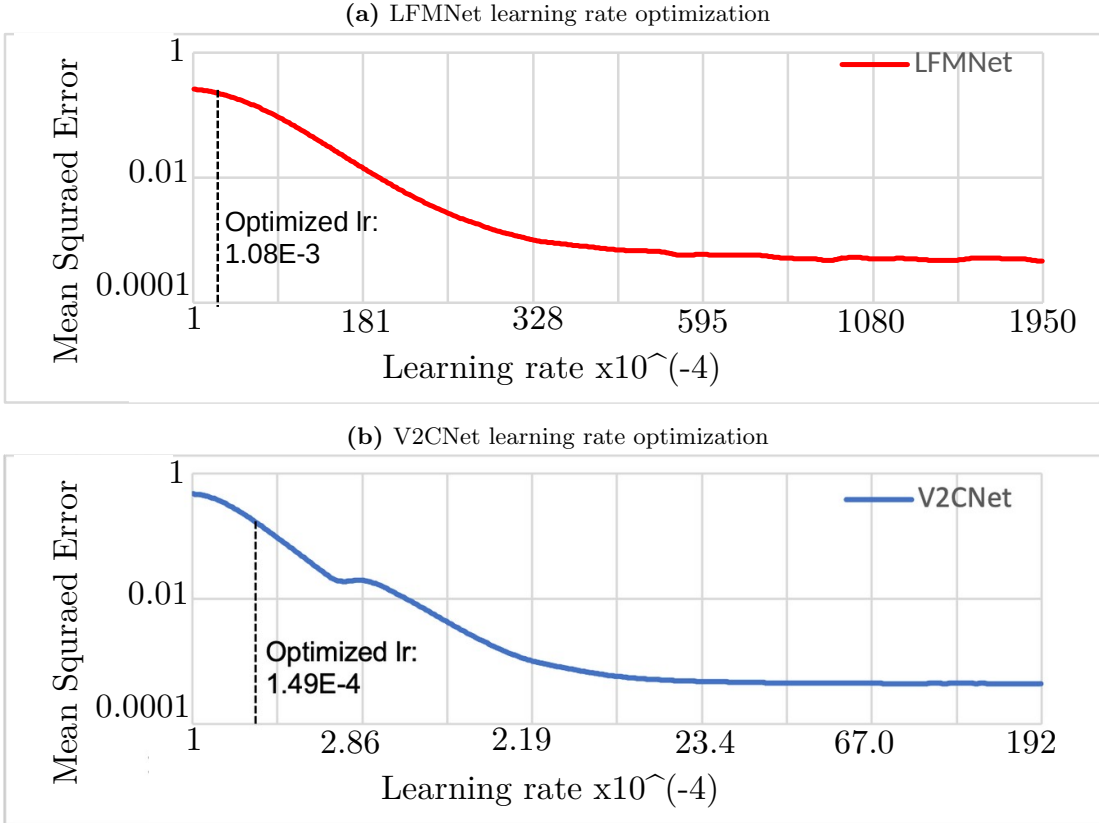


Figure 4.12: Learning rate optimization: learning rate vs. MSE (in log scale for visualization). The dotted line shows the selected learning rate, where the steepest MSE change is present, as in [7].

Furthermore, to corroborate that the optimal learning rate is the one dealing with the smallest error, we compared the optimal V2CNet learning rate against three other learning rates. In Fig. 4.9b, the validation set PSNR or reconstructed volumes is presented, where the dotted line represents the optimized learning rate. Note that when using a higher learning rate *e.g.*, $2.16\text{E-}3$, the PSNR increases but decreases with further training, indicating overfitting. The resulting PSNR of the network trained with the optimal learning rate of $1.49\text{E-}4$ increases slowly, reaching its highest point at the end of the training. Even though higher learning rates reach their highest performance 4x faster than the optimized one, it's essential to consider that these ablations are performed on a subset of the whole dataset. We chose the optimized learning rate to train the V2CNet

with the full dataset due to its stability and constant uptrend and trained it for 8000 epochs.

Resolution Evaluation Measuring the spatial resolution of a microscope is commonly performed by fluorescent imaging objects with sizes similar to the resolution limit [5, 6]. We measure the 3D profile of a blood vessel and compare the reconstructions with different methods. In Fig. 4.11, we show a projection of a 3D slice from a blood vessel indicated by the green arrow in Fig. 4.10, Sample tile 2. Also, the table in Fig. 4.11 (f) compares the Full Width at Half Maximum (FWHM) of an intensity profile across these projections. We show that LFMNet can resolve a blood vessel with $0.086\mu m$ error, in contrast to the V2C and LF deconvolution methods, which obtain $2.666\mu m$ and $2.918\mu m$ errors, respectively.

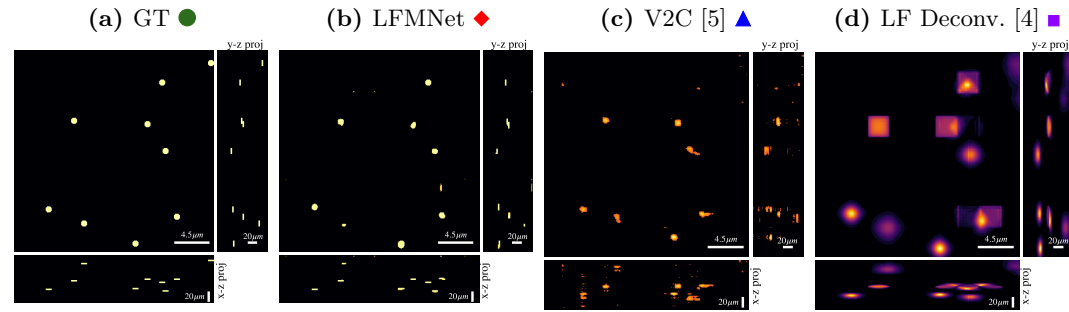
To demonstrate that the LFMNet can handle different types of structures, we retrained the LFMNet with 100 simulated LF images of volumes containing fluorescent beads with a shape of $0.9 \times 0.9 \times 1.8\mu m$. This uses the same simulation setup used for deconvolution in sec. 4.1.1.1. In Fig. 4.13 (a)-(d), we show a detail of the simulated volume and the corresponding reconstructions from LFMNet, V2C, and the deconvolution method. Using this dataset allows measuring the lateral and axial FWHM across all depths, as seen in Fig. 4.13 (e), and to further characterize the proposed method. When using a network with different biological structures, network re-training might be needed.

Moreover, if V2C is trained on the training and test data, it can fit very well. This shows that the network has enough capacity to learn the mapping. However, if it is trained on only the training set (as shown in Fig. 4.13), then it does not generalize well on the test data. As discussed in section 4.1.1.1, we believe this might be caused by the dimensionality of the input used during training. That is why we designed our architecture so that it can be trained with small input images.

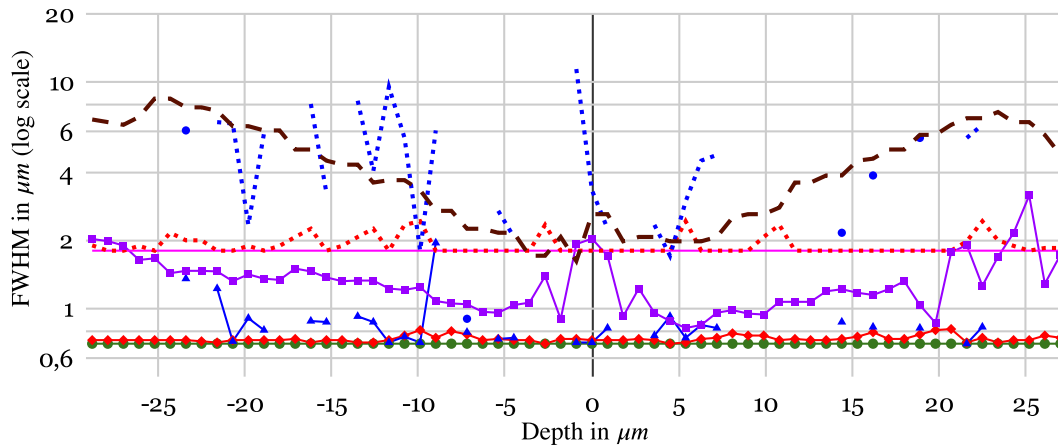
Data Set Evaluation

To better understand the nature and structure of the created data set, we provide some measures of the variability and complexity of the captured samples. Rather than computing statistics of the raw data (*i.e.*, the pixel intensities), we use the first two layers of a pre-trained VGG-16 [120] as feature extractors and analyze the first and second moments of their distribution.

We compare the statistics of these features to those of other data sets. We consider three reference data sets: One is ImageNet [121], which contains images with high texture diversity; a second data set is the C-elegans, which consists of confocal stacks [122]; a third data set is CCDB, which consists of mouse neuronal data [123]. As measures of the data complexity, we use the first ten components of the Principal Component Analysis (PCA) and the coefficient of variation (σ/μ) of the features, where σ is the standard deviation and μ the mean of all the entries of all the features. As one can see in Fig. 4.14, the complexity in terms of both the PCA and the coefficient of variation of our LF brain data set sits between that of ImageNet and that of both CCDB and



● GT lateral ◆ LFMNet lateral ▲ V2C lateral ■ Deconv. lateral — GT axial
◆ LFMNet axial ▲ V2C axial — Deconv. axial



(e) FWHM of beads at different depths with all methods.

Figure 4.13: Simulated beads image comparison. (a)-(d) show zoomed projection of simulated fluorescent beads with different reconstruction methods. (e) shows the mean lateral and axial FWHM, extracted from each depth of the reconstructions generated with different methods. The missing points represent depths where no beads could be found, due to artifacts.

4 Deep-learning-based 3D reconstruction

C-elegans. As expected, ImageNet has a large coefficient of variation, but our data set is quite similar in complexity to other useful data sets in biology.

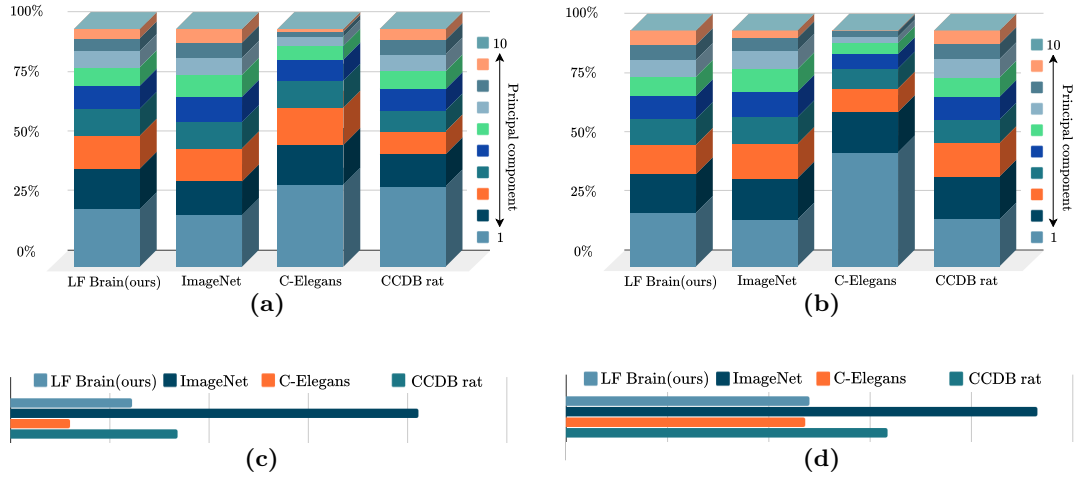


Figure 4.14: Statistical comparison of data sets. (a) and (b): Block histograms showing the relative importance of each PCA coefficient on VGG-16 relu1 (left) and relu2 (right) features. (c) and (d): Coefficients of variation of VGG-16 relu1 (left) and relu2 (right).

4.1.1.3 Discussion

The optical components used in our LFM perform a high-density sampling of the spatial and angular information of the real-world light field due to the $112\mu\text{m}$ lenslet diameter and the $3.45\mu\text{m}$ camera pixel width. These reduced dimensions of the pixels were chosen to enable the reconstruction of the object at a resolution close to that of a confocal scan when using a deep learning approach ($0.086\mu\text{m}$ error when reconstructing a blood vessel). However, small pixels also have a poor signal-to-noise ratio. To compensate for the noise, these sensors require a high exposure time and thus have a limitation in the achievable frame rate when used to capture videos. Ten frames per second is a frame rate that makes scanning large volumes, such as a whole mouse brain, practical (about 30 minutes). We leave the design of an LFM capable of real-time *in vivo* imaging to future work. For example, one could employ a modern sCMOS camera with high quantum efficiency, high frame rate, and larger pixel size (*e.g.*, $6.5\mu\text{m}$ or $4.25\mu\text{m}$) to achieve shorter exposure times while maintaining a high signal-to-noise ratio.

This form of *optical image compression*, as already demonstrated by our LFM, would enable faster scan times and smaller data storage than existing methods (such as light-sheet, confocal or, multi-photon microscopy). The reconstruction network would work as a decompression algorithm that recovers the full-volume scan. We expect that such a system would achieve an even higher performance than what we have demonstrated with our LFMNet.

However, as with most deep-learning approaches, there is no certainty on the biological and optical correctness of the reconstruction. One idea to tackle this, is to use the forward model to evaluate the error in image space, giving a sense of how close or far from the GT the reconstruction is.

4.1.2 Real-Time light field 3D microscopy via sparsity-driven learned deconvolution

This is an adapted version of a peer-reviewed manuscript [124] published during my Ph.D.

XLFM is a scan-less 3D imaging technique similar to FLFM well suited to capture fast biological processes, such as neural activity in zebrafish. However, even with its space-invariant PSF, current methods to recover a 3D volume from the raw data require long reconstruction times (around 0.011Hz), hampering the usability of the microscope in a closed-loop system or with large datasets. Moreover, when high auto-fluorescence is present, neural activity in a fish becomes dim and hard to measure. Thus, the ideal volumetric reconstruction should be sparse to reveal the dominant signals, where methods like the sparse-low-rank decomposition can be used [125]. Unfortunately, these methods are computationally intensive and introduce substantial delays (0.028Hz).

The issues motivated us to introduce a 3D reconstruction method that recovers the sparse spatiotemporal components of an image sequence in real-time. By combining a self-supervised neural network (SLNet) that recovers the sparse components of an LF image sequence and a neural network (XLFMNet) for 3D reconstruction. In particular, XLFMNet can achieve high data fidelity and preserve essential signals, such as neural potentials, even on previously unobserved samples. We demonstrate successful sparse 3D volumetric reconstructions of the neural activity of live zebrafish, with an imaging span covering $800 \times 800 \times 250 \mu\text{m}^3$ at an imaging rate of 24 – 88Hz, which provides a 1500-fold speed increase against prior work [125] and enables real-time reconstructions without sacrificing imaging resolution.

Sparse decomposition algorithms were previously applied to XLFM images to take advantage of the spatiotemporal sparsity of neural activity. For example, in work from Yoon *et al.* [125], where sparse decomposition light field microscopy (SDLFM) was introduced and which improved the resolution and signal-to-noise ratio in immobilized samples.

Several attempts can be found in the literature regarding a fast 3D reconstruction of neural activity. For example, the offline data processing time can be significantly reduced by directly estimating the light field “footprints”, and activity of individual cells and structures without reconstructing volumes [70, 71]. These methods, however, have not been widely used because they are specialized for neural signal extraction in motionless samples and cannot yield actual volumes.

In this work, summarized in Fig. 4.15, we propose SLNet and XLFMNet, two neural networks that can efficiently perform sparse decomposition and volume reconstruction on XLFM raw recordings at high speed. We also evaluate the generalization capabilities of the networks to unseen samples through techniques such as reducing the number of

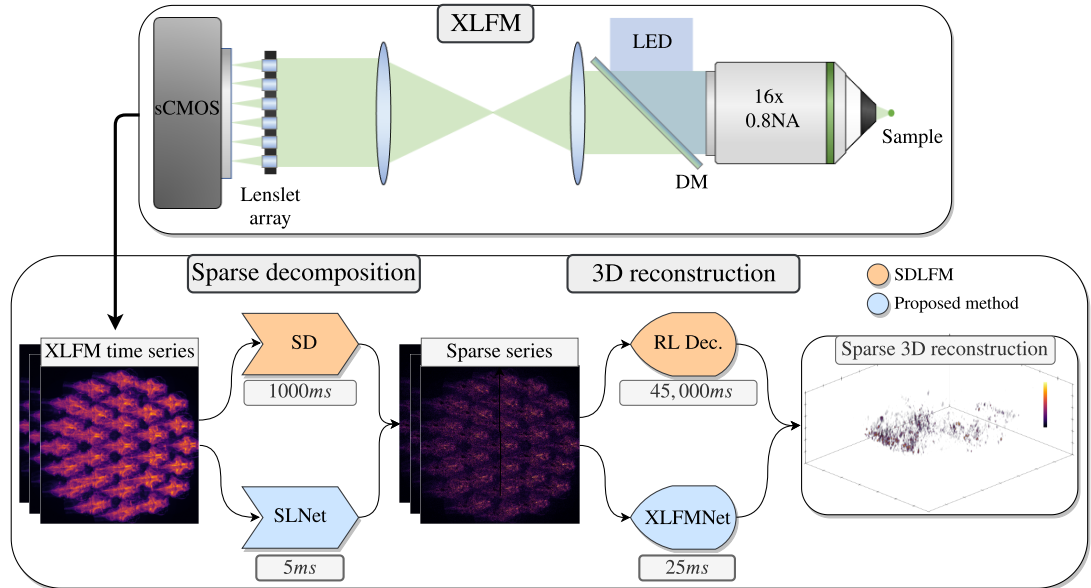


Figure 4.15: Top: Diagram of the XLFM used in this work. The microlens array was conjugated to the back focal plane of the objective lens through a 4-f lens pair. Excitation light from a 470 nm LED was projected on the sample through the objective lens using a dichroic mirror (DM). An sCMOS camera recorded all the sub-images formed behind the microlens array. **Bottom:** A comparison between the state-of-the-art SDLFM reconstruction (in orange) and the proposed method (in blue). Both methods first compute a sparse representation of an XLFM time series stack and later perform a 3D reconstruction of the sparse images.

parameters and augmenting the training data with estimated noise statistics and spatial transformations.

SLNet is trained with an unsupervised approach by minimizing a loss function that aims to approximate the input images with a low-rank representation. The experiments show that a sparse representation can be recovered using this reconstruction. This problem is well suited for the chosen network and allows our method to generalize to new samples. We evaluate a wide range of neural network settings and choose the most robust one when reconstructing seen and unseen samples.

The SLNet network, once trained, can perform a temporally and spatially sparse decomposition using three images of the sample at different time points in an interval of fewer than five milliseconds. The trained XLFMNet reconstructs a 3D volume at 24 – 88Hz. This real-time reconstruction capability would expand not only the applicability of LFM but also unlock novel paradigms of experiments that allow closed-loop feedback control of the experimental parameters, such as instrumental adjustment (*e.g.*, autofocus and tracking), animal stimulus delivery (*e.g.*, visual, auditory, or olfactory), and neuronal activity manipulation (*e.g.*, optogenetics) based on the real-time information of the reconstructed volumes.

In the following sections, we first present the unsupervised training of the SLNet, followed by a description of the XLFMNet, its design criteria, and training strategy, followed by a description of the XLFM microscope and the sample preparation for the experiments.

In the experimental results, sec. 4.1.2.2, we describe our findings on the SLNet training and the effect of the design parameters. Later, we discuss our results on the XLFMNet parameter ablation and an analysis of the neural activity of zebrafish. We present an evaluation of the XLFM generalization capabilities by measuring the full width at half maximum (FWHM) of micro-spheres that are not present in the training set. And finally, an estimation of the achieved resolutions of the different methods is presented using 3D MTF analysis.

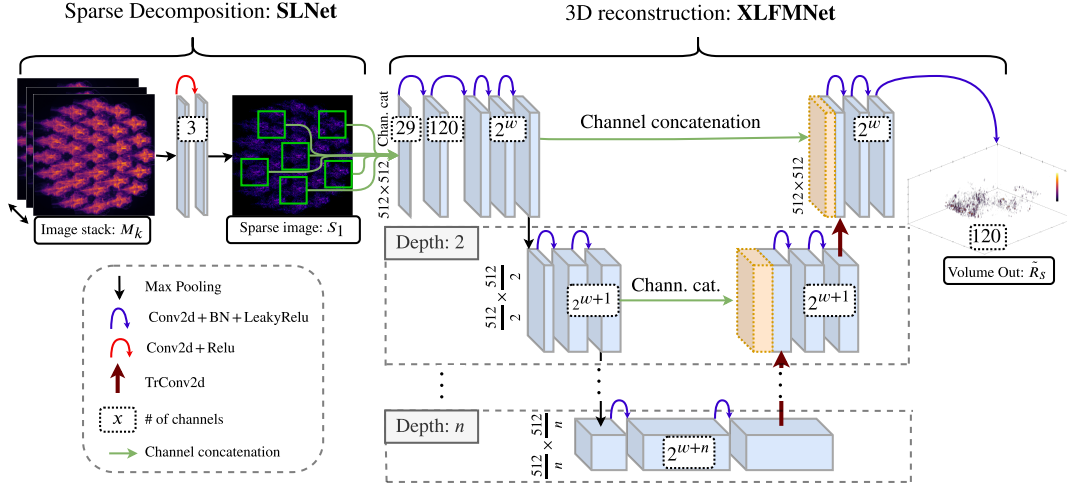


Figure 4.16: Proposed network architecture, where the SLNet performs a sparse decomposition of an image time series and XLFMNet the 3D reconstruction of the sparse component. The number of depths can be controlled by the parameter n , and the amount of channels per convolutional layer per depth is controlled by the parameter w .

4.1.2.1 Methods

Unsupervised sparse decomposition (SD)

The robust principal component analysis or sparse decomposition [126] is a method that decomposes a matrix M into its low-rank (L) and sparse (S) component, such that $M = L + S$.

Let $M_{k,m,r} \in \mathbb{R}_{\geq 0}^{k \times m \times r}$ be a set of images captured at k different times points, with lateral sizes m and r . SD can be used to decompose temporal stacks if we arrange $M_{k,m,r}$ to be $M_{k,mr} \in \mathbb{R}_{\geq 0}^{k \times mr}$ and minimize the optimization problem:

$$\begin{aligned} \min_{L,S} \quad & \|L\|_* + \lambda \|S\|_1 \\ \text{s.t.} \quad & L + S = M_{k,mr} \end{aligned} \quad (4.1)$$

Here $\|L\|_*$ is the nuclear norm of the low-rank component, λ a parameter controlling the degree of sparseness, and $\|S\|_1$ the L1 norm of the sparse component. This constraint optimization is usually solved using Augmented Lagrangian methods, such as the Augmented Lagrangian multiplier [127, 128].

However, as previously shown by [129], the constraint can be implicitly fulfilled if we first compute L and, with it, compute $S = (M - L)_{\geq 0}$, the non-negative result of subtracting the input image and the low-rank representation. Let $\mathcal{N}_{\Theta}^{SL}(M_{k,m,r}) \approx L_{k,mr}$ be a neural network with parameters Θ that generates a low-rank representation of M . We refer to this network as SLNet. Then $S = (M - \mathcal{N}_{\Theta}^{SL}(M))_{\geq 0}$.

The final loss function becomes:

$$\min_{\Theta} \quad \|M - \Gamma_{\mu}(\mathcal{N}_{\Theta}^{SL}(M_{k,m,n}))\|_1 \quad (4.2)$$

$\Gamma_{\mu}(\cdot)$ is a singular value shrinking operator that enforces a low-rank in its output by setting the eigenvalues $\Sigma_{<\mu} = 0$ and shrinking the remaining one. The full operator reads:

$$\begin{aligned} \Gamma_{\mu}(X) &= U [\text{sign}(\Sigma) \cdot \max(|\Sigma| - \mu, 0)] V^* \\ \text{Where: } X &= U \Sigma V^* \end{aligned} \quad (4.3)$$

The work from [129] employed four fully connected layers to perform the decomposition. However, due to the dimensionality of our images ($k = 3, m = n = 2160$), fully connected layers were not plausible due to the memory requirement. Hence, our network is implemented using two convolutional layers followed by a single ReLu as an activation function, as seen in Fig 4.16. The threshold μ dictates the degree of rank shrinkage and the amount of sparseness on S . We explore the effect of varying this parameter in sec. 4.1.2.2.

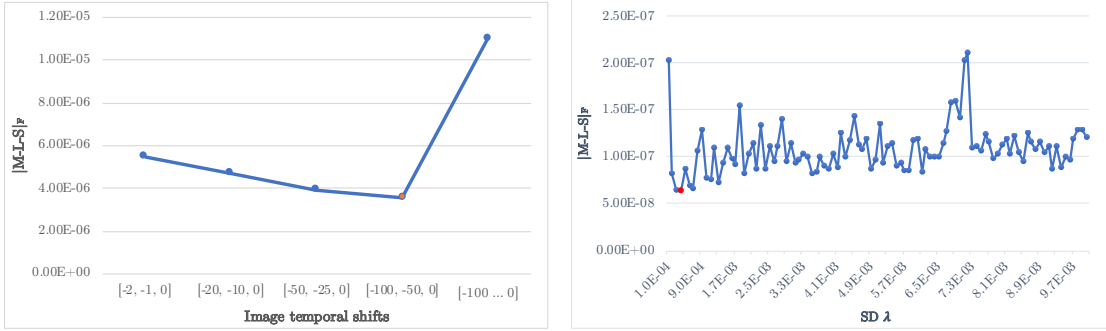
The weight initialization here is crucial, as a higher weight produces an L with entries larger than the entries in M , resulting in zero entries in S and no further training. To ensure this does not happen, we initialize the weights Θ of the SLNet, first using the Kaiming method [130], followed by scaling and a positivity constraint as in $\Theta = 0.1|\Theta|$.

For our decomposition, we chose the frames M_{t-100} , M_{t-50} and, M_t , where k is the time coordinate. We chose these time shifts based on a grid search analysis shown in Fig. 4.17a.

In practice, to use the SLNet, one would start recording frames, and once the images at M_{t-100} and M_{t-50} are available, the network starts to work in real-time by recovering the sparse components of the three images. This could be interpreted as a 100-frame buffer or warm-up.

Something essential to ensure is that the three frames used contain different information or are not inside a neural activation peak. As can be seen in Fig. 4.18, our frame rate (10Hz) is higher than the calcium dynamic of both a single event (decay time >

4.1 Traditional deep learning approaches



(a) Decomposition error for SD decomposition with different choices of time frames. (b) Ablation of λ sparseness parameter for optimal sparse decomposition performance.

Figure 4.17

200ms) and higher than events of sustained activity (*e.g.* bursting neurons) that can appear as a long activation. In red are marked the frames used as input for the SLNet. The difference in the intensities allows the network to find the low-rank component of the neural activity across different frames.

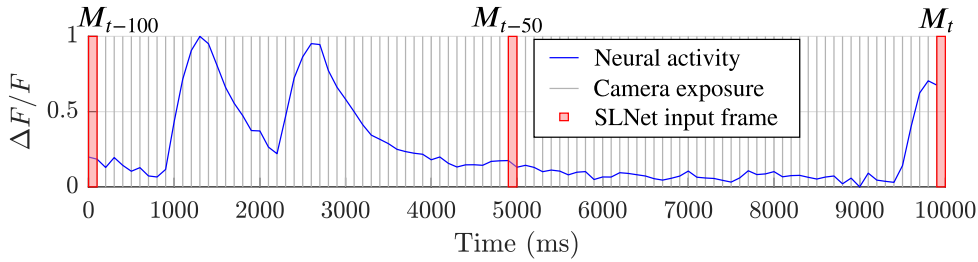


Figure 4.18: Activity of a single neuron and the three frames (M_{t-100} , M_{t-50} , M_t) used as input for SLNet to compute the sparsity of frame M_t .

3D Deconvolution neural network architecture

In the next stage, we employ a 2D U-net [85] for the 3D reconstruction, similar to the one used by Wang *et al.* [5] and Page *et al.* [86], where the depth stacks are stored in the channel dimension. We named this part of our pipeline XLFMNet. Our XLFMNet $\mathcal{N}_{\Theta}^{3D}$ is parametrized by n and w , the number of down/up sample steps, and an exponent to control the number of channels used: the first layer has 2^w channels, and any of the n consecutive layers has 2^{w+n} channels. This parameterization allows the exploration of a wide range of different networks, as seen in Fig 4.16. We explored networks with $n = \{2, 3, 4, 5\}$ and $w = \{4, 5, 6, 7\}$ in a systematic grid-like fashion.

Network selection criteria towards generalization:

4 Deep-learning-based 3D reconstruction

We look for a network that could perform well with unseen images of the same sample (validation set) and on unseen samples (testing set). We employed several performance criteria to achieve this:

- PSNR in the test set reconstruction results,
- SSIM in volume space (compared to a conventional deconvolution algorithm),
- SSIM in image space (see below).

The image space metric is computed by forward projecting the reconstructed volume (\tilde{R}_s) using the image formation model $i_s = \tilde{R}_s \otimes PSF$, and then comparing the generated image against the sparse image (S) used as input to the XLFMNet. This is possible as the PSF of the XLFM setup was measured before the experiments.

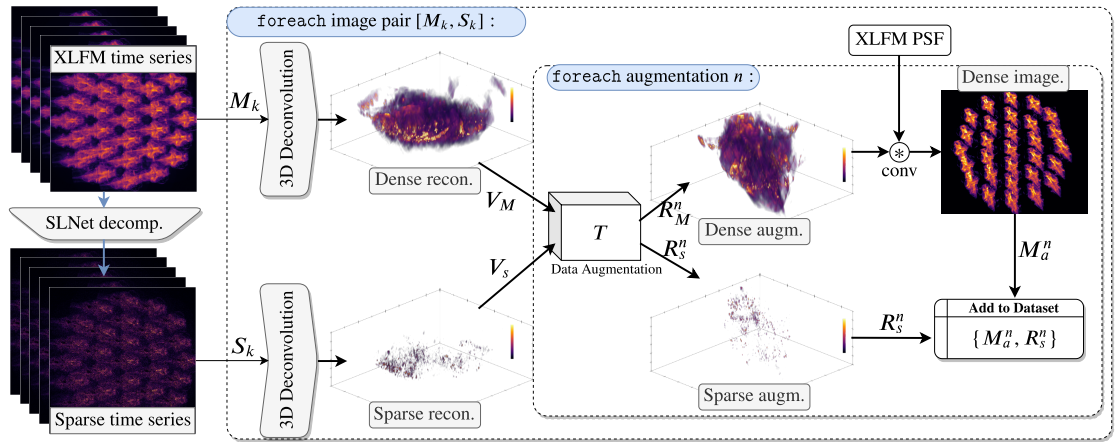


Figure 4.19: XLFMNet training data generation pipeline: From left to right, 3D reconstruction is applied to every image k in a time series. The sparse component of the time series is also extracted with the SLNet and reconstructed. In the middle, for each augmentation n , both the dense and the sparse reconstructions are fed to T , where the same random transformation is applied to both volumes. The dense augmented volume R_M^n is forward projected to image space. In the last step, the dense image M_a^n and the sparse volume R_S^n are stored to train the XLFMNet.

Generalization to non-observed samples: Generalization in neural networks is not trivial, as the networks tend to learn only the observed images' statistics. However, the degree of overfitting of a network to a dataset depends mainly on the dataset, the number of trainable parameters, etc. To evaluate the generalization capability of our networks, we built a testing dataset consisting of several zebrafish and microspheres data acquisitions.

With our architecture parametrization, we perform a systematic grid search of the parameters and evaluate the resulting networks' generalization capabilities using a training data set of 100 images. The results for the grid search and the final training can be found in Tables 4.4 and 4.3, respectively.

4.1 Traditional deep learning approaches

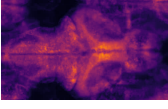
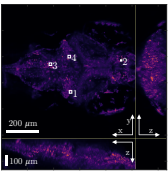
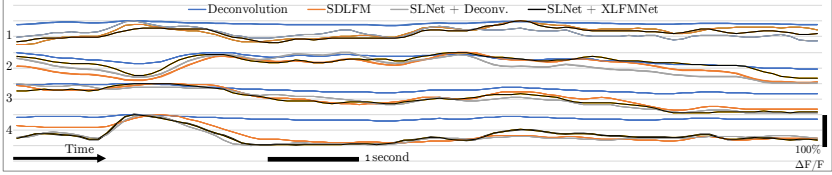
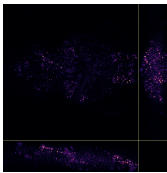
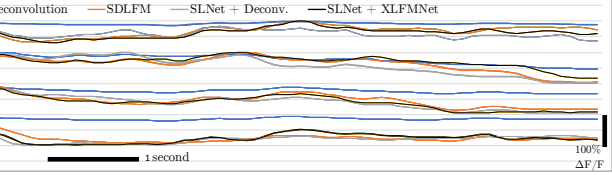
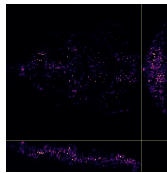
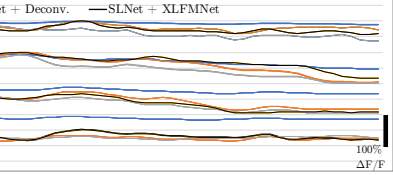
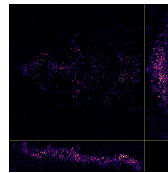
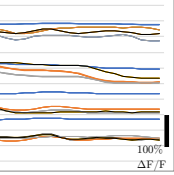
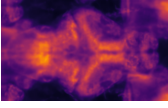
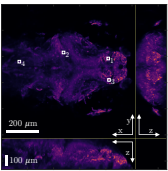
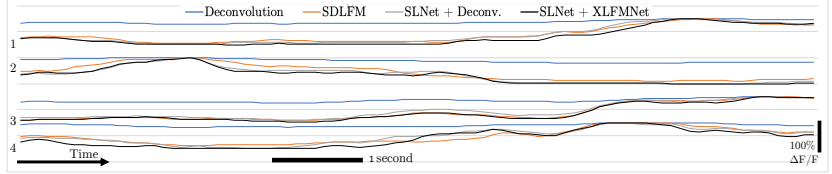
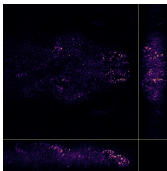
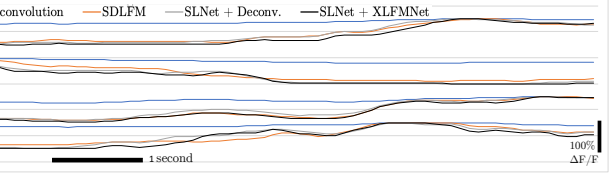
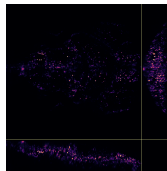
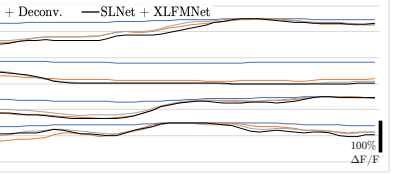
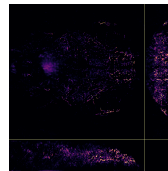
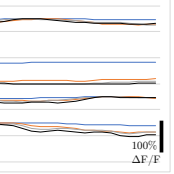
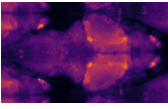
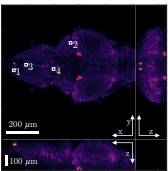
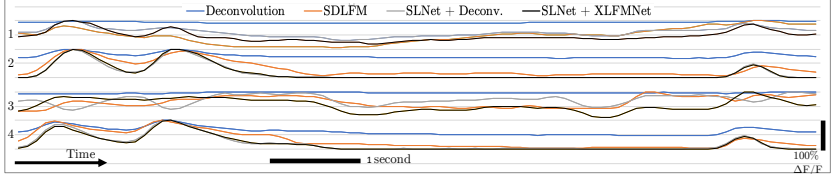
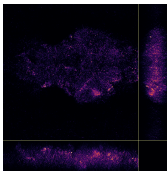
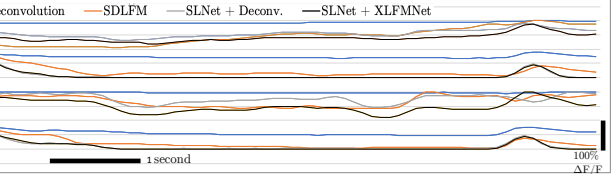
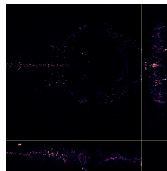
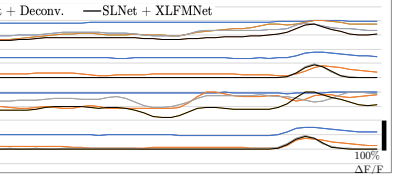
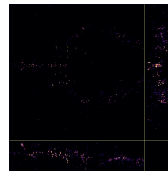
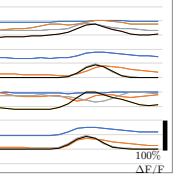
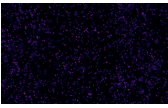
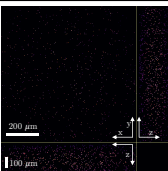
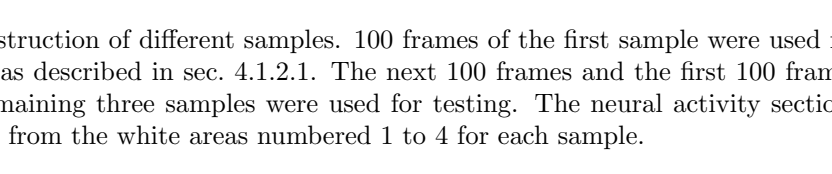
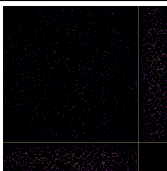
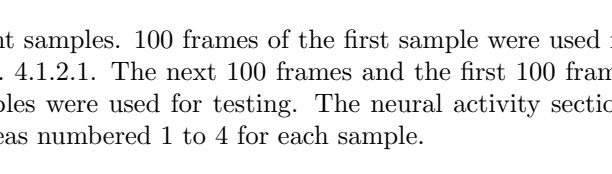
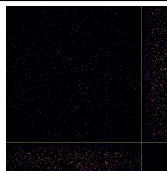
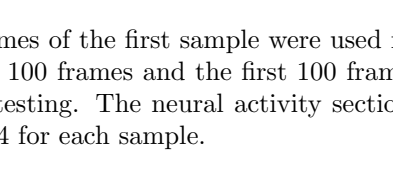
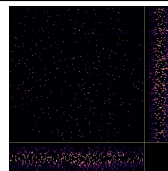
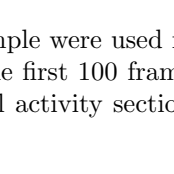
Dataset	Reconstruction Method			
	RL Deconv.	SDLFM [125]	SLNet + Deconv.	SLNet + XLFMNet
Train sample  GCaMP6s	 	 	 	 
Test sample  GCaMP6s	 	 	 	 
Test sample  GCaMP7f	 	 	 	 
Test sample  Fluor. beads	 	 	 	 

Table 4.3: 3D reconstruction of different samples. 100 frames of the first sample were used for training, as described in sec. 4.1.2.1. The next 100 frames and the first 100 frames of the remaining three samples were used for testing. The neural activity sections are taken from the white areas numbered 1 to 4 for each sample.

Dataset generation for XLFMNet training

This section describes the dataset preparation for the XLFMNet, containing pairs of XLFM images and sparse volumes. See Fig. 4.19 for an overview.

An essential aspect for generalization is the close resemblance of the training data and the real-world images. However, capturing a dataset of zebrafish with enough variation is a resource-consuming task that we try to avoid. In this work, we construct the training dataset using a single time sequence of a zebrafish, as explained by the following steps (see also Fig. 4.19):

1. Capture a time series of a fluorescent specimen with the XLFM.
2. Generate a sparse image (S) per frame using a pre-trained SLNet.
3. Apply 3D deconvolution to each time step, using the method from [63].
4. Perform data augmentation to the reconstructed volumes and forward project them to image space.
5. Store the dense images (M) and the sparse volumes (R_s) into the dataset.

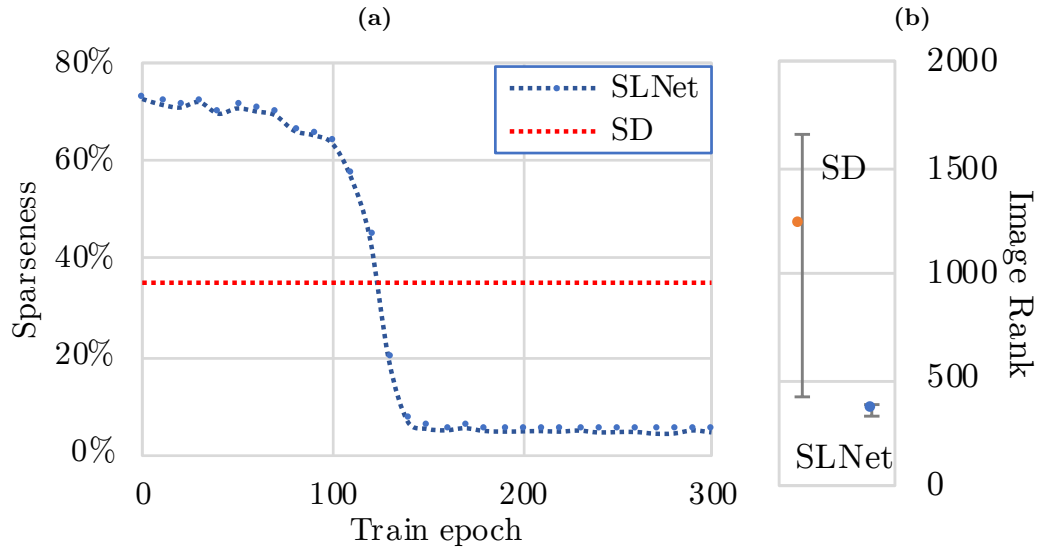


Figure 4.20: (a) Sparseness progression along a SLNet training with $\mu = 2$. Compared against the SD result using the augmented Lagrangian. (b) Mean rank comparison and min/max results between SD method and SLNet with $\mu = 2$ when evaluating 12 images in the test set.

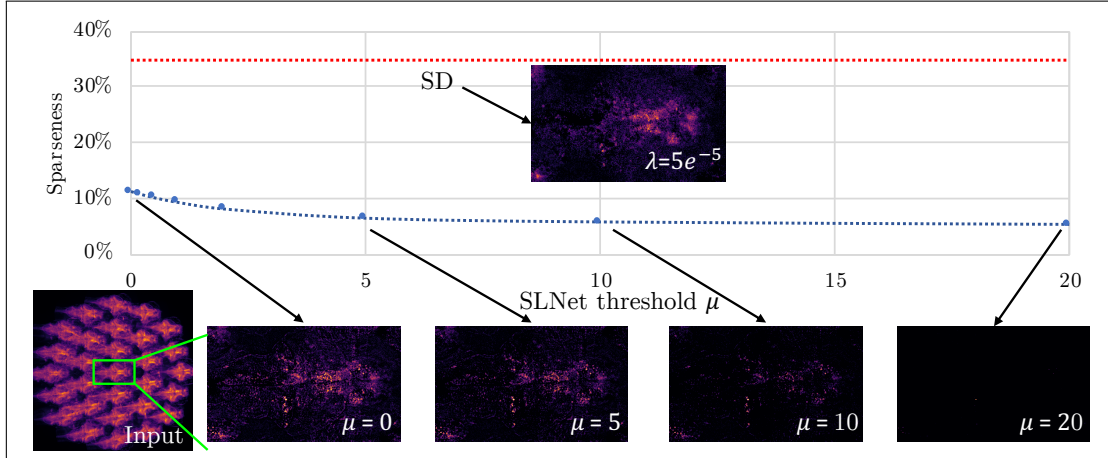


Figure 4.21: Comparison between SLNet trained with different μ values and the SD method based on the augmented Lagrangian.

XLFMNet ablation results						
U-net depth (n)	Channel exponent (w)	# params.	PSNR volume	SSIM volume (%)	SSIM reproj. (%)	Time (Hz)
2	4	76K	23.92	98.69	61.18	88.10
2	5	171K	24.31	98.98	70.29	79.36
2	6	511K	25.36	98.97	71.20	63.37
2	7	1.794M	25.59	99.00	72.64	45.06
3	4	168K	23.56	98.87	70.31	82.65
3	5	537K	25.34	98.87	67.27	71.45
3	6	1.972M	25.40	98.86	70.08	52.5
3	7	7.631M	25.99	98.97	72.43	29.42
4	4	533K	24.49	98.82	67.76	79.38
4	5	1.997M	25.18	98.90	65.54	67.06
4	6	7.809M	25.42	98.92	73.37	46.75
4	7	30.972M	25.91	98.94	72.48	24.00
5	4	1,994K	24.37	98.81	67.73	77.16
5	5	7.835M	25.31	98.96	71.60	62.93
5	6	31.150M	24.92	98.90	70.48	40.04

Table 4.4: Results of XLFMNet ablation study, as described in sec. 4.1.2.1. The row in orange is the setting used for our final tests, achieving the best performance in two out of the four performance metrics.

Zebrafish preparation for imaging: Pan-neuronal nuclear localized GCaMP6s Tg(HuC:H2B:GCaMP6s) and pan-neuronal soma localized GCaMP7f Tg(HuC:somaGCaMP7f) [131] zebrafish larvae were imaged at 4–6 days post fertilization. The transgenic larvae were kept at 28°C

and paralyzed in standard fish water containing 0.25 mg/ml of pancuronium bromide (Sigma-Aldrich) for 2 min before imaging to reduce motion. The paralyzed larvae were then embedded in agar with 0.5% agarose (SeaKem GTG) and 1% low melting point agarose (Sigma-Aldrich) in Petri dishes. The fish water was added to the dishes once the agar solidified.

3D fluorescent bead samples preparation: To better evaluate the performance of our method, we imaged 1- μm -diameter green fluorescent beads (ThermoFisher) randomly distributed in 1% agarose (low melting point agarose, Sigma-Aldrich). The stock beads were serially diluted using melted agarose to 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} of the original concentration. The diluted beads-agar colloid was then transferred to small Petri dishes to gel. The thicknesses of solidified bead samples were approximately 800 μm , which were sufficiently large to cover the full axial field of view of the microscope.

Dataset augmentation: The augmentation consisted of random 3D rotations, translation, and scaling of the volumes R_m and R_s . However, to increase the resemblance of the simulated images to the real microscope, there are two key aspects to take into consideration:

- **Noise augmentation:** Fluorescence microscopes suffer from noise when acquiring an image, mainly shot noise. Due to this, it is important to add the proper shot and background noise to the simulated dense images M . Additionally, as the microscope might be used for imaging samples with very low or very high counts, we augmented M by re-scaling its pixel intensities to a random signal power (between 30^2 and 70^2 ADU). The noise was then added using the camera specifications.
- **Sample axial distribution:** To avoid training bias towards certain depths, we applied z translation of the sample, such that in a subset of the samples, only a few structures are present in depths far from the focus plane. We found that this is of crucial importance due to the nature of the PSF where the pixel intensities decrease for planes far away from the focal plane, biasing the network towards learning the higher-intensity structures near the focal plane.

Extended field-of-view light field microscope (XLFM)

We built an XLFM setup as described in [125] as the imaging hardware (see Fig. 4.15). The microscope used a 16×0.8 NA water dipping objective lens (CFI75 LWD $16\times W$, Nikon) for excitation and detection. The excitation light generated by a blue LED ($\mu = 470$ nm, M470L4, Thorlabs) was collimated and passed a 480 nm short-pass filter before being reflected into the back pupil of the objective lens by a dichroic mirror (FF495-Di03-25 \times 36, Semrock). A customized microlens array (29 lenses, $f = 35.4$ mm or 36.6 mm) was mounted on an sCMOS camera (Zyla 5.5 sCMOS, Andor), with the camera sensor at the focal plane of the microlenses. The microlens array was conjugated with the back pupil plane of the objective lens through a 4f relay lens pair ($f_1 = 180$ mm, AC508-180-A-ML, Thorlabs; $f_2=125$ mm, PAC074, Newport). A 525/50 nm band pass filter (FF03-525/50-25, Semrock) was attached to the microlens array for green

fluorescent imaging. The system point spread function (PSF) was measured by taking a 600 μm thick image z -stack of a 1- μm -diameter green fluorescent bead located at the center of the field of view with an axial step size of 2.5 μm . The total magnification of the system is $M_{\text{total}} = \frac{F_{\text{mfa}} \cdot F_1}{F_{\text{obj}} \cdot F_2} = 4.5312$ as in [132]. Hence, a voxel in the sample space spans 1.4344 μm laterally and 2.5 μm axially.

4.1.2.2 Experiments

Sparseness threshold and the SLNet

Controlling the degree of sparseness in the reconstructions produced by the SLNet is possible through the term μ from eq. (4.2). μ dictates how the eigenvalues of image M get shrunk and thresholded, forming a low-rank representation. This behavior was clearly visible in our experiments, where we evaluated different SLNet networks trained on a subset of 20 temporal 2D stacks and tested on 12 unseen ones. We tested networks with a threshold equal to $\mu = \{0.0, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0\}$. Fig 4.21(a) displays how the different choices of μ 's influence sparseness, which we define as:

$$\text{sparseness} = \frac{\# \text{ non-zero elements}}{\# \text{ total elements}} \% \quad (4.4)$$

Our results corroborate the intuition behind the unsupervised training approach (see sec. 4.1.2.1), as the sparseness increases as μ increases. This makes this parameter a user-friendly way of controlling the sparseness of the network.

An interesting finding is that even with $\mu=0.0$, which corresponds to neither shrinkage nor thresholding, the network produces a low-rank solution in the spatial domain, which is an excellent starting point for the SLNet to focus mostly on refining the temporal sparseness. Our interpretation is that the blurring nature of convolutions in CNN is a good fit for this task. In Fig 4.21(b), a comparison of the rank of the decomposed images with the SD method vs. the SLNet is presented. The mean, min, and max ranks are shown to evaluate the 12 test images previously described. The rank of the images produced by SLNet is distributed in a small region, showing the robustness of the proposed method across sample space.

Fig 4.20 shows how the sparseness of a network (with $\mu=5.0$) decreases across training epochs. We found it helpful that by storing the state of intermediate training steps, the user can decide which level of sparseness is desired for a given application. However, if the SLNet is trained for too long, eventually, the sparseness is too high to be useful, generating very low-contrast images as a result. For our images, a sparseness of 5% suffices for the 3D reconstruction to perform optimally.

Network ablation towards generalization

The network evaluation strategy consists of training on the first 100 frames of a 10Hz capture of a single zebrafish and testing in the following 100 frames of the same fish and two other fish with different fluorescent labeling and age.

4 Deep-learning-based 3D reconstruction

Training the XLFMNet on a workstation with an Nvidia Quadro RTX 6000 graphic card takes around 5 hours for 500 epochs using the Adam optimizer. The possibility of re-training the network for every new sample is at reach. However, in daily microscopy work, one would avoid re-training the network often. Also, the amount of graphic memory required to train it efficiently is too large for regular computers. In our case, we focus on crafting a robust network to work with different samples without compromising the data fidelity. Hence, reducing the network re-training frequency.

The results of the XLFMNet ablation (see Table 4.4) show that the number of trainable network parameters has a direct relation to the achievable reconstruction quality (e.g. SSIM volume). However, the performance decreases when the parameters are spread across deeper networks. Compare, for example, two networks with a similar amount of parameters but different depths (n) in the U-net: the XLFMNet with $n = 2$ and $w = 7$ versus the net with $n = 5$ and $w = 4$, where the first one performs better in all performance metrics.

3D reconstruction of seen and unseen samples

When evaluating neural networks with biological data, it is crucially important that the information of the neural activity is preserved, no matter the method used for 3D reconstruction. In other words, the network should not introduce non-physical artifacts that might hamper the analysis of the recovered signal.

In Fig. 4.3, we evaluated the neural activity across a period of 100 seconds by applying different methods:

- Deconvolution of the raw image,
- SDLFM (SD + Deconvolution),
- SLNet + Deconvolution,
- SLNet + XLFMNet.

The SDLFM algorithm requires parameter tuning for optimal performance. Based on the original work [125], we used the Frobenious norm to find the best settings for the SD method with the training sample. We evaluated which configuration of λ results in the smallest error (e) for use in our experiments. e is measured with the Frobenious norm as in:

$$e = \|M - L - S\|_F \quad (4.5)$$

Where M is the input image, L and S are a low-rank, and the sparse representations from eq. 4.2.

In Fig. 4.17b, a large range of λ was evaluated, and in orange can be found the best configuration with $\lambda = 4E^{-4}$

Reconstructing fluorescent beads

When applying the different sparse decomposition methods to images of beads, we found that the output is quite similar to the input, as seen in the last row of Fig. 4.3. However, we can use the images of the beads to analyze how the XLFMNet infers unseen types of samples. In this experiment, first, we applied conventional deconvolution to XLFM images of beads and computed their FWHM for every detectable bead. The position of the beads was stored for extracting the same information from a volume reconstructed with an XLFMNet trained on a single zebrafish.

In Table. 4.5, we compare the FWHM of beads images when 3D reconstructed with deconvolution against XLMNet, together with the detection histogram per depth — in other words, how many beads were detected per depth — which helps us analyze the information trade-offs of the proposed method. Looking at the FWHM plots, it is evident that XLFM suffers from resolution loss against deconvolution, mainly in the axial direction. However, as seen in the histograms, it keeps a detectability comparable to the deconvolved volume.

Spatial resolution estimation

To further evaluate the image quality from the SLNet and XLFMNet, we estimated the 3D resolutions of the reconstructed larval zebrafish brain images based on 3D MTF analysis. The analysis results are shown in Fig. 4.22. We inferred the 3D resolutions from the spatial frequency support regions in the 3D MTFs, as enclosed in white dotted lines in Fig. 4.22. According to our estimation, XLFM has a lateral resolution of $\sim 4.3\mu m$ and an axial resolution of $\sim 10.0\mu m$. At the same time, SDLFM, SLNet+Deconv., and XLFMNet provided similarly enhanced resolutions of $\sim 3.2\mu m$ laterally and $\sim 7.7\mu m$ axially. These resolution values aligned well with the measured FWHMs of fluorescent beads in Table. 4.5.

RL. Deconvolution		XLFMNet	
FWHM (μm)	Detectability histogram	FWHM (μm)	Detectability histogram
Lateral 		Lateral 	
Axial 		Axial 	

Table 4.5: Generalization analysis with beads. Reconstructed with conventional deconvolution (left) and the proposed XLFMNet (right). The lateral and axial plots show the full width at half maximum for every depth and a detectability histogram, *i.e.* the number of beads found per depth with each method. The beads were not part of the training set of the XLFMNet.

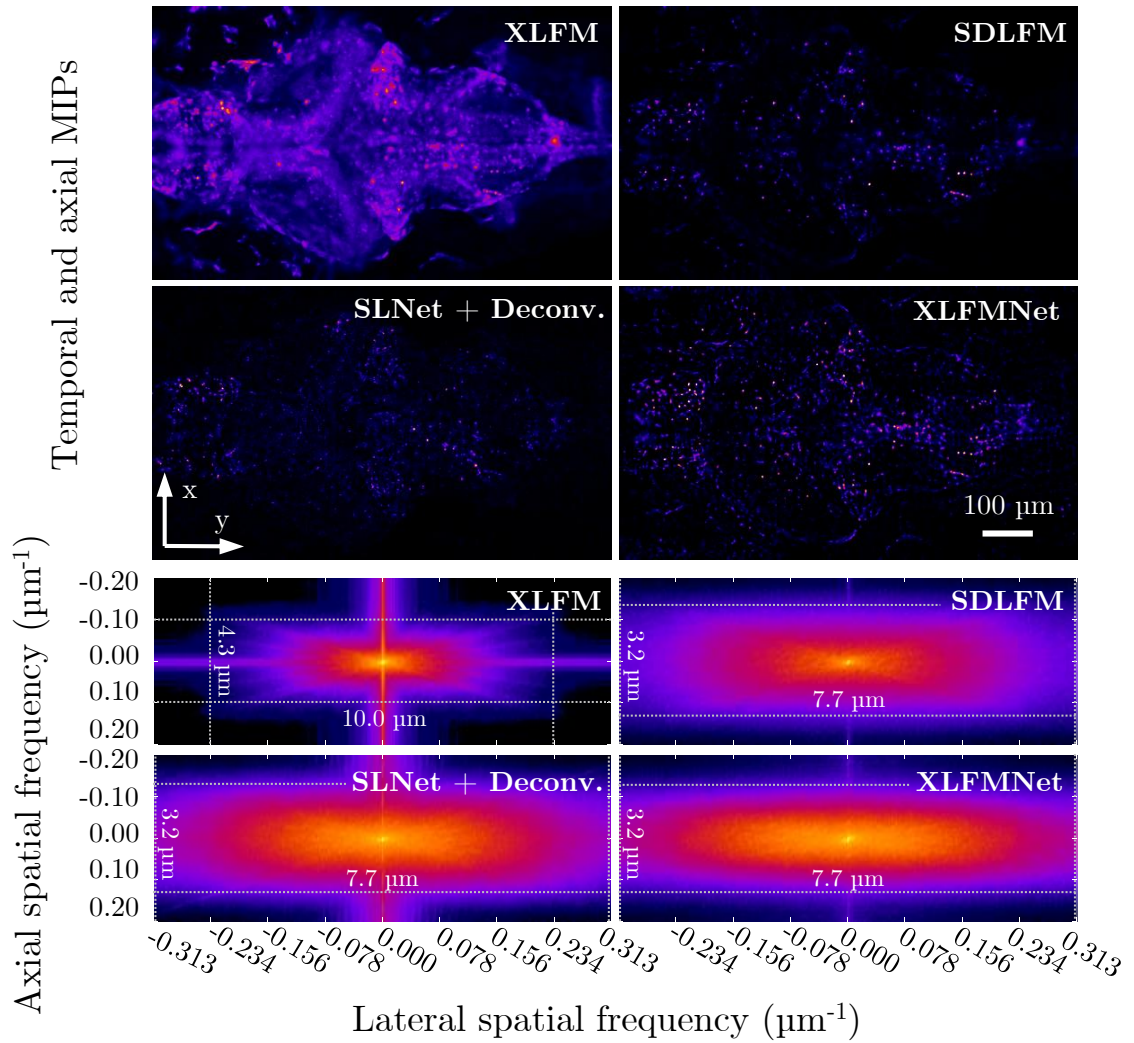


Figure 4.22: Spatial resolution estimation for different reconstruction methods. The top panel shows reconstructed zebrafish brain volumes' temporal and axial max intensity projections. The bottom panel shows 3D MTFs (displayed in log scale) that show the spatial frequency support of each method.

4.1.2.3 Discussion

The first contribution of our work is the SLNet, which performs the sparse decomposition of temporal raw XLFM stacks. During its training and evaluation, we found that CNNs are an excellent pick for this task due to their effect of producing blurry images that are already a low-rank representation of the input in the spatial domain. Then, the SLNet focuses mostly on finding the sparseness in the temporal domain. Also, we found that the sparseness parameter μ used to shrink the principal components of the images during

4.2 Bayesian learning for reconstruction and out-of-distribution detection

training is very user-friendly as it is an intuitive way of controlling the sparseness of the reconstructions. We consider that this approach could be integrated into the microscopy workflow and even have networks trained with different μ stored and let the user decide the level of sparseness desired, for example, as a Micro-Manager or an ImageJ plugin.

The second contribution is the XLFMNet, which reconstructs 3D volumes from the sparse representations produced by the SLNet. Which achieves real-time computations (45Hz) on large volumes ($800 \times 800 \times 250\mu m^3$). The ablation study provided helpful information regarding the network's generalization to unseen samples. A network with a higher number of parameters performs better than one with a lower number. However, these parameters should not be spread across too many down-convolutional steps. The best network we consider performed is one with two down-convolutional steps ($n = 2$) and a channel exponent $w = 7$, which reconstructs volumes at a rate of 45.05Hz.

When evaluating the network with different samples, in Table 4.3, we found out that when trained on a zebrafish, it works well for other unseen zebrafish, preserving the neural potentials at a similar pace as the other reconstruction methods. However, when imaging beads substantially different from fish, the network can reconstruct the central depths with high fidelity but loses contrast when approaching depths farther away from the focal plane. The intuition behind this is that the beads that are far away spread their energy in a larger sensor area, dimming the individual pixels substantially. Nevertheless, the network could still detect the beads even at far away planes, as seen in the detectability histogram comparison from Table 4.5, where the number of detected beads is quite similar to the one from the conventional deconvolution method.

A possible solution would be to retrain a batch normalization block at the network's entrance for every new sample type while keeping the XLFMNet frozen. We leave this for future work.

4.2 Bayesian learning for reconstruction and out-of-distribution detection

After exploring the previously mentioned approaches and not finding a suitable solution for a real-time and reliable 3D reconstruction, I gravitated towards statistically informed reconstruction methods, which have the potential of fast reconstructions and certainty metrics for out-of-distribution detection (OODD), introduced in sec.2.4.4.

In this chapter, you can find an implementation for reconstructing 3D volumes using the XLFM images with a novel architecture based on NF and CWF, trained and tested on six zebrafish image sequences and 3D deconvolutions. Also, once more, the SLNet is employed to extract the neural activity of the fish before reconstruction, putting emphasis on reconstructing the relevant signals only.

4.2.1 Fast Light-field 3D microscopy and out-of-distribution detection enabled by conditional normalizing flows

This is an adapted version of a work-in-progress manuscript, which will be submitted shortly after this dissertation.

This work proposes a workflow including the CWF, a modified WF architecture that uses the XLFM image as an input condition. It is suited for the 3D reconstruction and OODD of XLFM sparse fluorescent images generated with the SLNet, a sparse-low-rank decomposition neural network. The CWF’s reconstruction and OODD capabilities enable a fast, accurate, and robust method for 3D fluorescence microscopy. Fast enough to be applied within close-loop or human-in-the-loop experiments, where reconstructed activity could trigger further steps in an experiment.

OODD is achieved by having access to an exact likelihood computation that allows the proposed CWFA to evaluate the likelihood of a sample belonging to the training distribution at different image scales. Measured in all the down-sampling NFs in the network (see Fig. 4.23). Specifically, in a testing step, the likelihood of a novel sample can be computed and compared with the training set likelihood distribution through the Mahalanobis distance [133]. As described in sec. 4.2.1.2.

We demonstrate the reliability of the proposed CWFA on XLFM acquisitions of living zebrafish larvae full-brain neural activity, processed with the SLNet, which separates the neural activity from the background of the acquisitions. As a gold standard or GT, we used 3D reconstructions (100 iterations with the RL algorithm) of the SLNet output. We compare the CWFA against the XLFMNet [124] and the WF on 3D reconstructions covering a FOV of $734 \times 734 \times 225\mu\text{m}^3$ ($512 \times 512 \times 90$ voxels), the CWFA achieves a MAPE [134] of 0.021 against 0.020 and 0.035 respectively, shown in Fig. 4.26.

Furthermore, when analyzing the neural activity of individual neurons in a time sequence, as seen in Fig. 4.27, the proposed method achieves a Pearson correlation coefficient (PCC) [135] of 0.892, where the XLFMNet and WF achieve 0.541 and 0.408 respectively. In OODD, the CWFA achieved an F1-score [136] of 0.985, as seen in Fig. 4.28. Once an out-of-distribution sample type is detected, we found that 60 minutes of fine-tuning of the proposed CWFA architecture are enough to achieve a substantial increase in quality. Reaching a mean increase of 4.16% and 12.29% of PCC and MAPE. And a final mean PCC of 0.881. And time-wise, CWFA reconstructs a single frame in 0.130s, the XLFMNet in 0.021s, and the WF in 0.100s. This is without applying any network distillation or model compilation.

4.2.1.1 Methods

Proposed Conditional Wavelet Flow architecture The WF architecture uses a multi-scale hierarchical approach that allows training each up/down-scale independently, allowing flexibility during memory management. Each down-sampled operation is a Haar

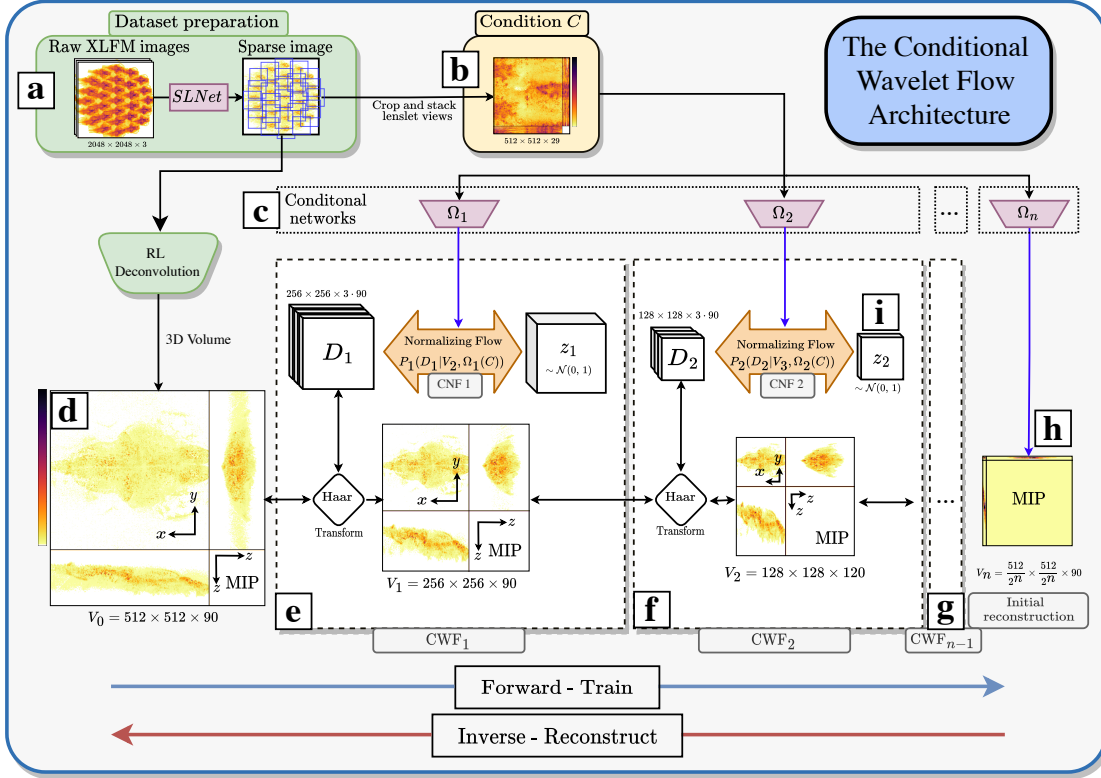


Figure 4.23: The Conditional Wavelet Flow Architecture and workflow: **(a)** Dataset preparation with the SLNet [8] and performing 3D reconstructions using the RL algorithm. The volume **(d)** and image **(b)** pairs are used to train the CWFs. Training is performed in each CWF individually and consists in feeding V_0 and the processed condition **(b)** $\Omega_1(C)$ to the CWF 1, generating outputs V_1 and z_1 . The latter is used in loss function 2.10. V_1 is fed to the next CWF, which is trained similarly. This is repeated **(g)** until reaching the lowest resolution output. This is used to train the deterministic CNN Ω_n **(h)**. 3D reconstruction is performed by running the flow inversely. First, by reconstructing V_n with the CNN **(h)**, feed this to the last flow **(g)**, sample z_{n-1} from a normal distribution **(i)**, and run the CNF_n inversely to generate the Haar detail coefficients D_n used to up-sample the low-resolution volume V_n into V_{n-1} . This process is repeated for each up-sampling step **(g, f, e)** until reaching the desired resolution **(d)**.

transform chosen due to its orthonormality and invertibility. The likelihood function of a WF model is given by:

$$p(I_n) = p(I_0) \prod_{i=0}^{n-1} p(D_i|I_i) \quad (4.6)$$

where each I_n is the final high-resolution image, D_i the Haar transform detail coefficients and I_i the image for down-sampling step i . $p(I_0)$ and all $p(D_i|I_i)$ are normalizing flows.

4 Deep-learning-based 3D reconstruction

In the proposed CWFA, we substituted $p(I_0)$ with a deterministic CNN. We conditioned all the up-sampling NFs on an external condition $\Omega_i(C)$ instead of the low-resolution image I_i (as shown in Fig.4.23). Giving:

$$p(V_0) = p(V_n) \prod_{i=1}^{n-1} p(D_i|\Omega_i(C)) \quad (4.7)$$

And the log-likelihood (LL):

$$\log p(V_0) = \log p(V_n) + \sum_{i=1}^{n-1} \log p(V_i|\Omega_i(C)) \quad (4.8)$$

Note that we changed the indexes order, as in our case, set V_0 as the goal volume with full resolution, and every consecutive CWF runs in lower resolution. The final loss function can be optimized independently as $\log p(V_0)$ is comprised of a sum of probabilities. Showing the advantage of the Wavelet-Flow architecture.

Sampling from the Conditional Wavelet Flow architecture Reconstructing a 3D volume starts by reconstructing $V_n = \Omega_n(C)$. This low-resolution image generation is an easy task for a network, is accurate, and includes information from the image formation model. Then, use this low-resolution volume for reconstructing the detail coefficients of the first flow: $D_i \sim p(D_i|\Omega_i(C))$ and generate an up-sampled volume $V_{i-1} = h^{-1}(V_i, D_i)$, where h^{-1} is the inverse Haar transform. And repeating until reaching the full-resolution volume V_0 .

Conditioning the wavelet flows In the original WF, each NF uses only the next level low-resolution volume as a condition. We propose to append the raw XLFM image pre-processed by a CNN to the conditions. Allowing the NF to extract missing information about the inverse problem from the raw data.

Furthermore, the original Wavelet Flow (WF) network [106] aims to generate human faces from a learned face distribution, where the low-resolution HAAR transformed image is used as a condition on each up-sampling step. In such a case, face distribution is the only known information.

However, when dealing with microscope images, prior information about the system and volume to reconstruct are known, such as the forward process in the form of a point spread function (PSF) and the captured microscope image. Motivating its usage as conditioning in each up-sampling step. In this work, we evaluate the impact on performance when using the low-resolution volume as a condition like [106] plus the post-processed views of the raw XLFM image.

The raw XLFM input for the conditions and the first up-sampling step is prepared first by detecting the center of each micro-lens from the central depth of the measured PSF (using the Python library *findpeaks* [137], then cropping a 512×512 area around

4.2 Bayesian learning for reconstruction and out-of-distribution detection

the 29 centers, then stacking the images in the channel dimension, as seen in Fig 4.23 panel (b). The resulting tensor is passed to a CNN that down-samples and processes the input to the required size. Note that each down-sampling operation of the WF network requires a different condition size.

Domain-shift detection We measured the Mahalanobis distance [133] from the NLL of a test sample to the training data mean, given by: $MD = \frac{\mathcal{L}_{\text{sample}} - \mu_{\mathcal{L}_{\text{train}}}}{\sigma_{\mathcal{L}_{\text{train}}}}$, where $\mu_{\mathcal{L}_{\text{train}}}$ and $\sigma_{\mathcal{L}_{\text{train}}}$ are the mean and standard deviation of the training dataset NLL \mathcal{L} .

The OOD threshold is picked by selecting a small number of samples from the testing set and evaluating their likelihood. Then define ten thresholds linearly distributed between the training and testing Mahalanobis distance mean values, and pick the one achieving the largest F1-score.

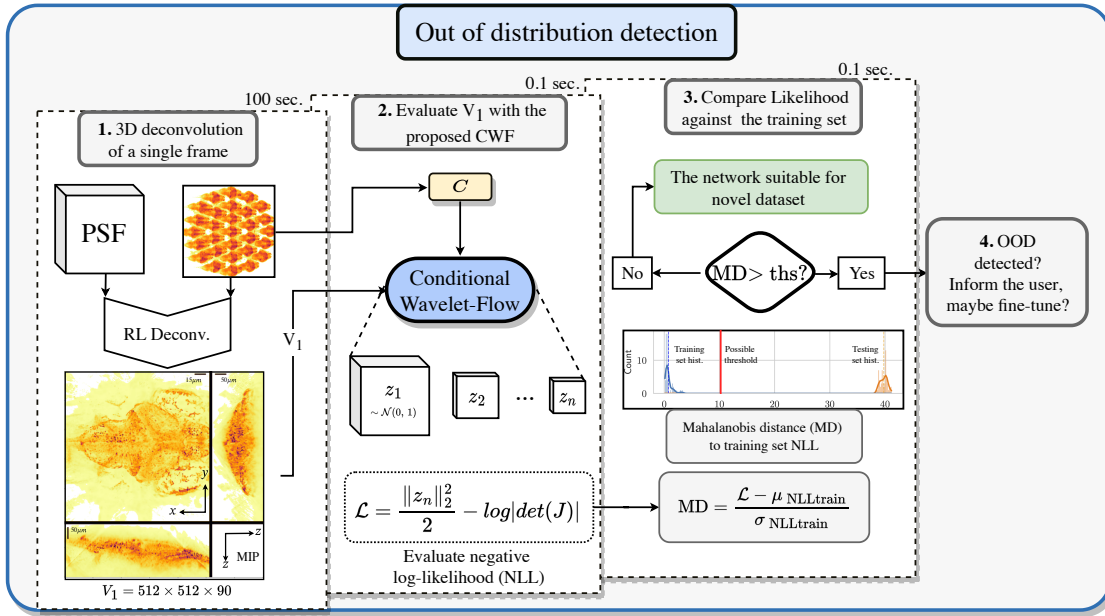


Figure 4.24: Out-of-distribution detection workflow when using the CWFA

The following steps are executed for domain shift detection, also illustrated in Fig. 4.24:

- Train the CWF on a set of fish image-volume pairs.
- Given a novel image I_{novel} , deconvolve via RL and obtain V_{novel} .
- Use the novel reconstruction V_{novel} and the image I_{novel} and feed them to the CWFs in the forward direction.
- Compute the likelihood of the generated normal distributions $z_1 - z_n$ with eq. 2.10
- Measure the Mahalanobis distance [133] between the training set Likelihood distribution and the novel sample as specified in sec. 4.2.1.1.

Sample	Description	Age
Used for training		
GCaMP7f fish 1	NLS GCaMP6s	different age
GCaMP7f fish 2	NLS GCaMP6s	different age
GCaMP6s fish 1	Pan-neuronal nuclear localized GCaMP6s Tg(HuC:H2B:GCaMP6s)	4-6 days
GCaMP6s fish 2	Pan-neuronal nuclear localized GCaMP6s Tg(HuC:H2B:GCaMP6s)	4-6 days
Used for testing		
GCaMP7f fish 3	Soma localized GCaMP7f Tg(HuC:somaGCaMP7f)	different age

Table 4.6: Datasets used: 4 networks were trained, each on a single fish from the ‘Used for training list.’ We re-trained all networks with each fish for the out-of-distribution experiment.

- Decide if a domain shift was detected based on a pre-defined threshold.
- Once detected, the network could be fine-tuned to the new sample type, etc.

The detection threshold is picked by selecting a small subset of samples from the testing set and evaluating their likelihood. Then define ten thresholds linearly distributed between the training and testing Mahalanobis distance mean values, and pick the one achieving the largest F1-score. In this work, we found a threshold of $2.5 \sigma_{train}$, where σ_{train} is one standard deviation from the training set.

The CWFA The proposed architecture uses six CWFs, each comprised of a Haar transform down-sampling operation and an NF, as seen in Fig. 4.23. The internal CNF learns the mapping between the detailed Haar coefficients and a normal distribution. And is built from 7 conditional invertible blocks (see Supplementary Fig. 4.25). The parameters of each block, such as the type and number of invertible blocks (like GLOW [138], RNVP [108], HINT [139], NICE [100], CAT [140] etc.), the number of parameters in each internal convolution, and the training learning rate, were optimized for each block independently in a grid-like fashion. The optimized settings of each NF are presented in Table 4.7 and an example of the grid search visualization is shown in Fig. 4.29. Where we optimized for PSNR and MAPE.

Network implementation The architecture was implemented in PyTorch aided by the Freia framework for invertible neural networks [141]. Adam optimizer was used for training without weight decay or learning rate scheduling. This Hyper-parameters are optimized on a subset of 180 images from 3 fish.

Initial low-resolution reconstruction through a CNN: As mentioned in the proposed architecture section, the first step of reconstruction is performed by a deterministic CNN (Ω_N in Fig. 4.23), that takes an input XLFM sparse image and reconstructs a low-resolution volume ($8 \times 8 \times 90$). Ω_N first multiplies the input image by a global attention module that selects the relevant pixel-wise information by first turning the image into a 1D array, then applying a Conv1D, a ReLU, a Conv1d, and a sigmoid activation which outputs a value from 0 to 1 that gets multiplied or weights the original pixel. This modified input image is then down-sampled by an encoder, formed by two sets of Conv2D (with kernel size eight and stride 8), AlphaDropout (alpha(0.1), SELU activation, and Batch normalization.

CWFs used for up-sampling: Each of the 6 CWF uses 7 CNFs internally, with 16 channels per convolution and CAT invertible blocks. A single block can be visualized in Fig. 4.25. After the 6 CWF

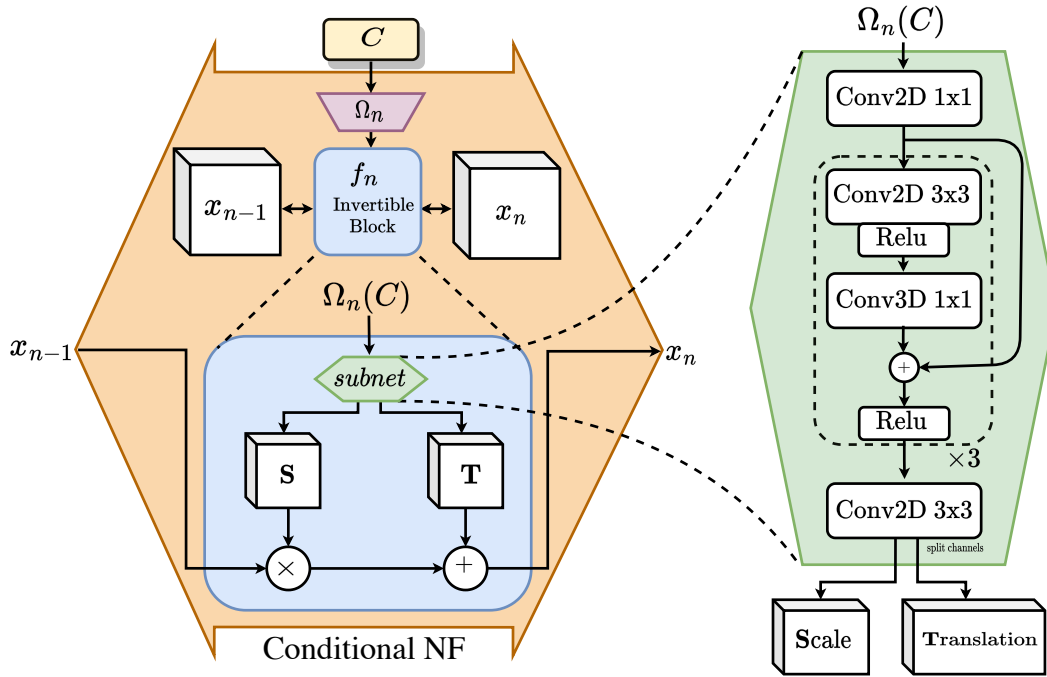


Figure 4.25: Single conditional normalizing flow used within the CWFA, also present in Fig. 4.23 as CNF1,2, etc. In blue, the CAT block is responsible for computing a scaling and translation from the condition and applying it to the input.

Dataset acquisition and pre-processing Pan-neuronal nuclear localized GCaMP6s Tg(HuC:H2B:GCaMP6s) and pan-neuronal soma localized GCaMP7f Tg(HuC:somaGCaMP7f) [131] zebrafish larvae were imaged at 4–6 days post fertilization. The transgenic larvae were kept at 28°C and paralyzed in standard fish water containing 0.25 mg/ml of pan-

4 Deep-learning-based 3D reconstruction

curonium bromide (Sigma-Aldrich) for 2 min before imaging to reduce motion. The paralyzed larvae were then embedded in agar with 0.5% agarose (SeaKem GTG) and 1% low melting point agarose (Sigma-Aldrich) in Petri dishes.

Each fish was imaged for 1000 frames at 10Hz. Neural activity images were extracted using the SLNet [124], as seen in the top part of Fig. 4.23. Later, the resulting images were 3D reconstructed with the RL algorithm for 100 iterations. 500 pairs of XLFM sparse images and volumes were used to train the CWFA, and 500 were used for validation, testing, and domain shift detection.

The beads dataset was created by imaging 1- μ m-diameter green fluorescent beads (Ther-284moFisher) randomly distributed in 1% agarose (low melting285point agarose, Sigma-Aldrich). The stock beads were serially diluted using melted agarose to 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} of the original concentration.

4.2.1.2 Experiments

3D reconstruction of sparse images with the CWFA Fig. 4.23 depicts how to train the CWFA (forward pass) and reconstruct 3D volumes from XLFM images (inverse pass). Reconstruction with a trained CWFA involves, first, reconstructing a low-resolution volume (V_n) using the raw XLFM image with the CNN $\Omega_n(C)$ and using this as an input to the last up-sampling NF (CWF_{n-1}). To up-sample on this CWF: first, sample z_{n-1} from a normal distribution, and input the pre-processed XLFM image $\Omega_{n-1}(C)$ as a condition to the NF. Then, generate the Haar coefficients (D_{n-1}), later used for up-sampling V_{n-1} by a factor of 2 with the Haar transform and generate V_{n-2} . We repeat this process until reaching V_0 at full resolution.

We compared the proposed method against the XLFMNet [8] and a modified version of the Wavelet Flow [106] (WF). The XLFMNet is a U-net-based architecture [85] that achieves high reconstruction speeds but lacks any certainty metrics (as in conventional deep learning techniques). The WF modifications are necessary, as the original model was used as a face generator without external conditions, which doesn't comply with inverse problems modus-operandi, where a measurement is required to reconstruct the variable in question. Also, in the original WF, the lowest resolution was reconstructed from an unconditioned normalizing flow based purely on the training data distribution.

The modified version follows the original design in which only the low-resolution image is used as a condition in each flow; however, we used a CNN Ω_n instead of the lowest-resolution NF, informing the system about the measurement and making a fair comparison. A grid search of learning rates was performed on both methods, and the model with the highest validation performance was selected.

4.2 Bayesian learning for reconstruction and out-of-distribution detection

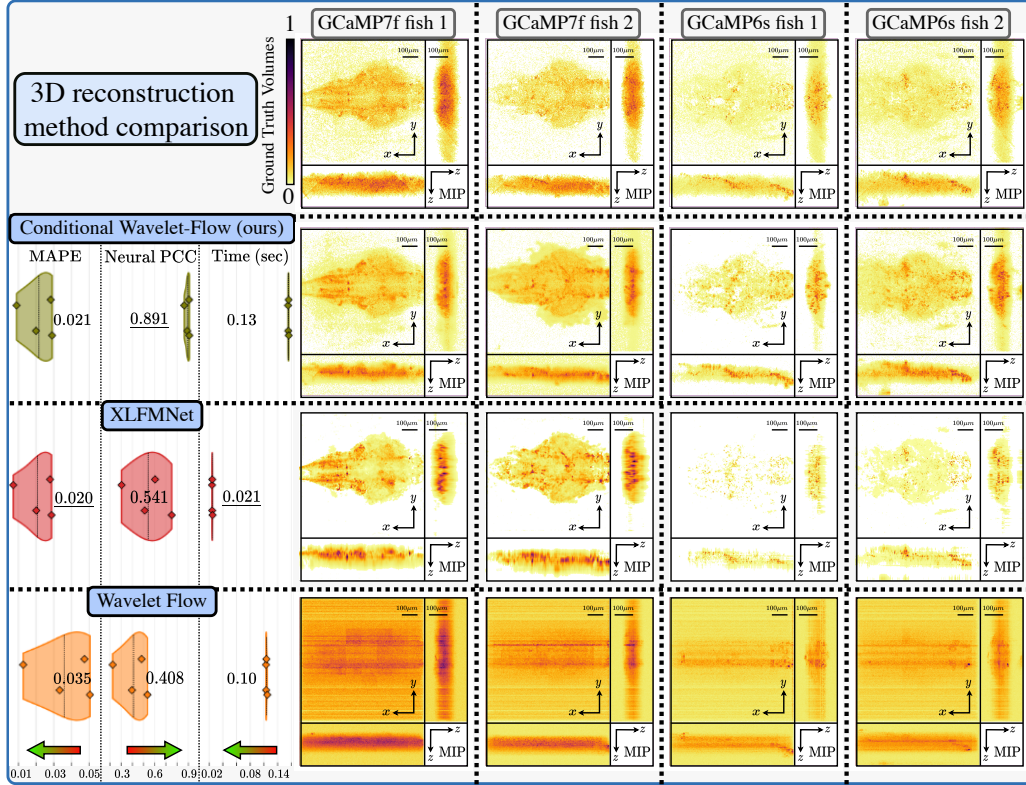


Figure 4.26: 3D reconstruction comparison of sparse zebrafish images with different methods. On the top row, the volumes considered GT generated with 100 iterations of the RL algorithm using a measured PSF. On each column, a different zebrafish acquisition. The following rows show reconstructions with different methods. The left-most column shows the performance metrics used for comparison: MAPE, followed by the mean Pearson correlation coefficient (PCC) of 100 frames from the same fish acquisition measured on the top 50 most active neurons per fish. The bottom arrows show which direction the metrics are considered better.

Comparison metrics Conventionally, metrics like peak signal-to-ratio and structured similarity are used for image quality assessment. However, these might be deceiving when working with highly sparse data, as most of the pixel’s intensity is close to zero. We consider the neural activity similarity to the GT most relevant. Hence, the PCC is used on the activity of single neurons across time, and the MAPE for single frame quality assessment.

As seen in Fig. 4.26, the proposed method outperforms the other two approaches regarding the neural PCC. And has a similar MAPE performance to the XLFMNet. However, XLFMNet provides faster inference capabilities but lacks any certainty metrics.

Domain shift or out-of-distribution-detection (OODD) We evaluated this method by training the CWFA on four zebrafish fluorescent activity datasets. Then presented, the domain shift detection pipeline with different sample types, such as previously unseen fish sparse images (processed with the SLNet), raw XLFM fish images (not pre-processed), and fluorescent bead images with different concentrations. As shown in Fig. 4.28. Our algorithm achieves an F1-score of 0.9848 on CWF₁. The achieved F1-scores for all the down-sampling CWFs are presented in Fig. 4.28 panel d.

4.2.1.3 Discussion

In this work, I explored for the first time Bayesian Networks, which promised a robust workflow for inverse problems, particularly when false positives should be minimized, as in the case of bio-medical data. Most of the works using generative models are oriented towards either art or applications where accuracy is not essential, which pushed me toward exploring new designs. Implementing a new architecture was a challenge, as designing, justifying, and ablating a new and large one, as shown here, is a lengthy task where experiments must be performed and documented meticulously.

A possible enhancement to the CWFA is to, instead of up-sampling the x-y space with the individual CWFs, up-sample the depth dimension. This makes much more sense, as the shape of the fluorescent fish is present at high definition since the beginning, coming from the raw XLFM images used as conditions. Furthermore, the conditional networks act as a bottleneck that might be unnecessary, making feature extraction harder for the networks. However, I realized this late in the project when most of the experiments and results were ready, and making such a large change to the architecture wouldn't make sense. I'll leave this for a future project or for you, the reader, to implement.

Step	Block Type	# channels	Learning rate	Volume size
1	CAT	16	5×10^{-5}	$512 \times 512 \times 90$
2	CAT	16	5×10^{-5}	$256 \times 256 \times 90$
3	CAT	16	1×10^{-4}	$128 \times 128 \times 90$
4	CAT	16	1×10^{-4}	$64 \times 64 \times 90$
5	CAT	16	5×10^{-5}	$32 \times 32 \times 90$
6	CAT	16	1×10^{-5}	$16 \times 16 \times 90$

Table 4.7: Result of hiper-parameter optimization on each CNF step.

4.2 Bayesian learning for reconstruction and out-of-distribution detection

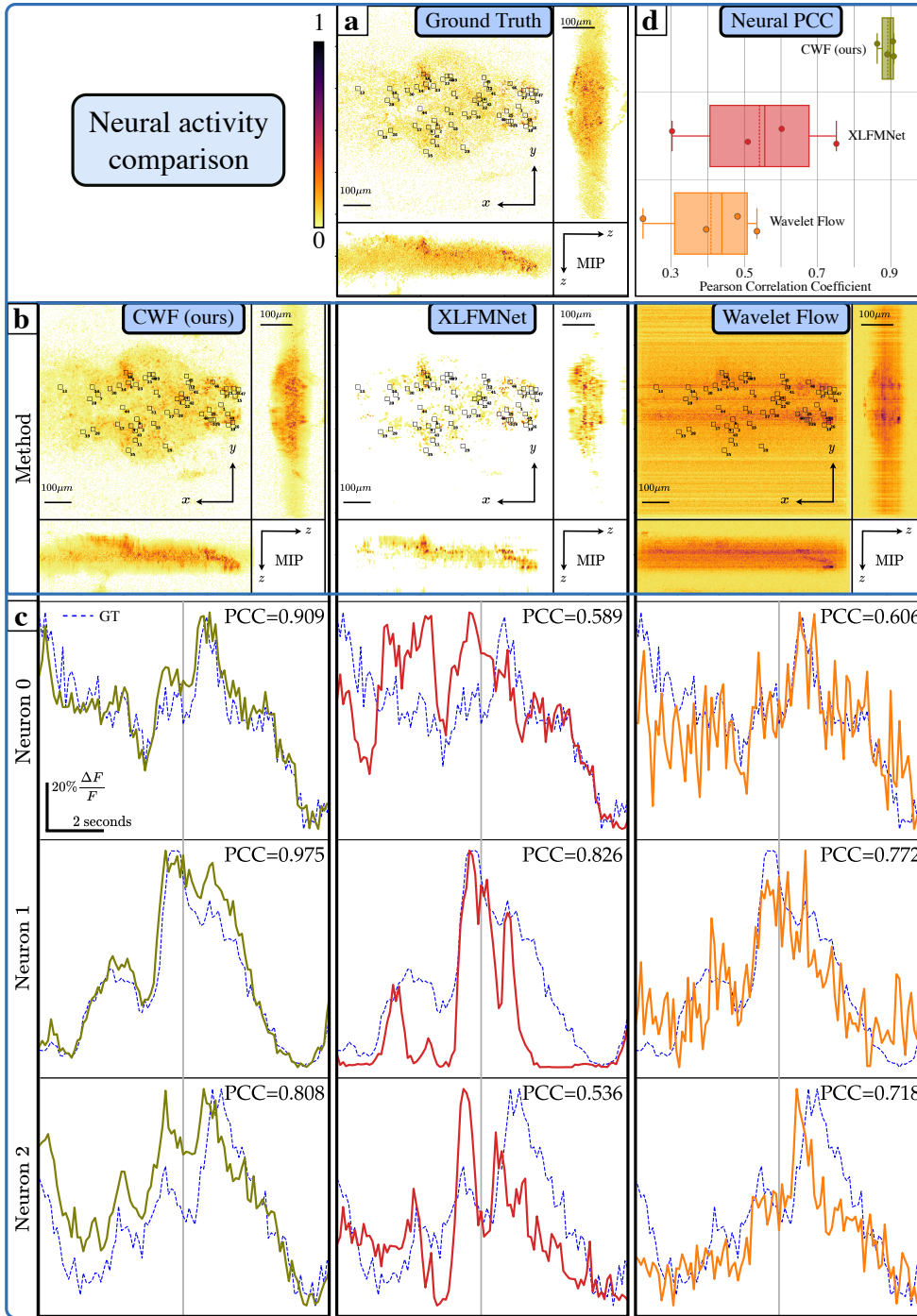


Figure 4.27: Neural activity comparison with different methods on a single fish acquisition. In (a), the MIP of the GT volume, with the top 50 most active neurons highlighted. A single reconstructed frame with different methods is in the (b) row. In (c), the neural potentials of 3 neurons in 100 frames (10 seconds). (d) shows the mean PCC with four different fish and the three methods.

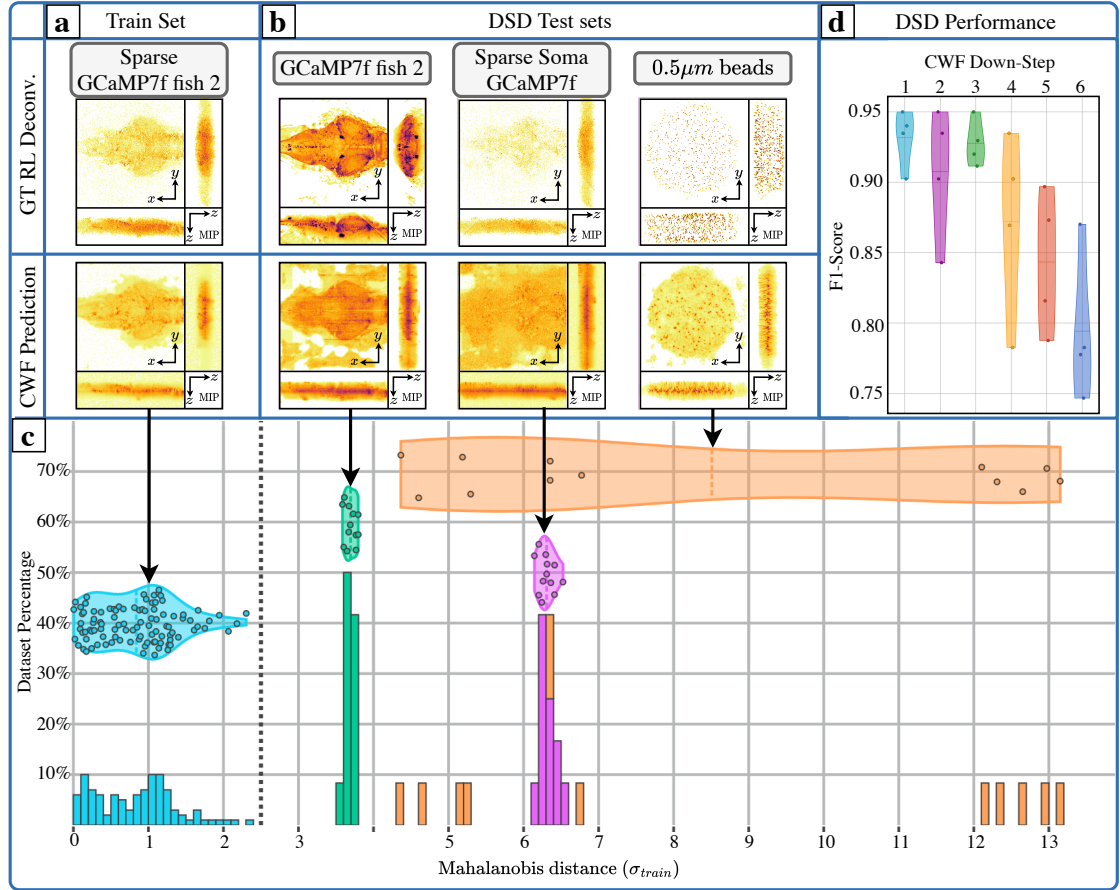


Figure 4.28: Domain Shift Detection (DSD) performance of the CWF proposed method. In (a) the MIP of the GT and predicted CWF volumes. In (b), 3 types of datasets to test the DSD. In (c), the Mahalanobis distance from a sample negative Log-Likelihood training mean to different samples per dataset, measured in training standard deviations. A threshold of $2.5 \sigma_{train}$ was used to discern out-of-distribution samples. Each CWF in the architecture has access to the NLL. Hence the Mahalanobis approach can be applied to discern OOD samples, and in (c), the output distributions of step 1 are shown. In (d), the performance of all the down-sampling is presented, where each dot represents the performance when training the CWF in a different fish.

4.2 Bayesian learning for reconstruction and out-of-distribution detection

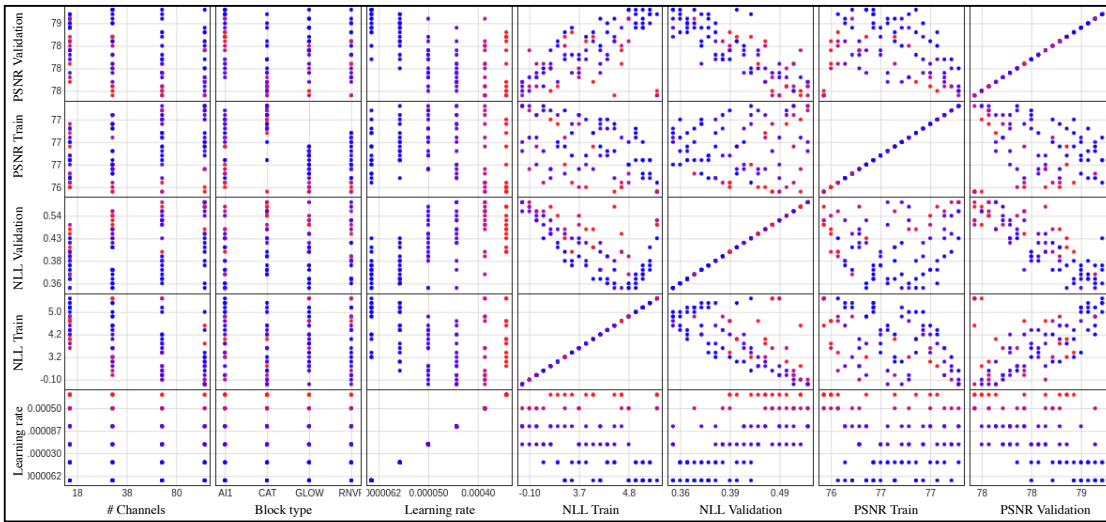


Figure 4.29: Grid search for hyper-parameter tuning for up-sample step-5,, conducted with guild.ai and visualized in tensorboard. In our experiments, we optimized for the highest PSNR value. The optimized settings for each step can be found in Table. 4.7

5 Mix of both worlds: Joint optimization

5.1 Using auto-differentiability for inverse problems

Reconstruction algorithms in medical imaging rely on a deep understanding of measuring devices' forward process or image formation model. For example, when designing a LFMic the trade-off between angular and spatial resolution is dictated by the size and placement of the micro-lenses. Which might be optimally chosen for one task but might not be optimal for different sample types or situations.

Why not let the reconstruction algorithm choose which optical setup provides the best features to solve a problem? (3D reconstruction, segmentation, background removal, etc.) Furthermore, as in most optimization problems, the derivatives of an objective function with respect to the parameters to optimize are required. Having a flexible system where the derivatives are automatically computed becomes essential. This is possible in machine learning frameworks, such as PyTorch [114] and JAX [142].

In this chapter, WaveBlocks is introduced, a flexible wave-optics fluorescence simulator useful for applications like forward modeling, microscope calibration, joint optimization of optical systems and algorithms, etc.

5.1.1 Wave-blocks: Learning to model and calibrate optics via a differentiable wave optics simulator

This is an adapted version of a peer-reviewed manuscript [113] published during my Ph.D.

We present a novel learning-based method to build a differentiable computational model of a real fluorescence microscope. Our architecture can calibrate an actual optical setup directly from data samples, and engineer point spread functions by specifying the desired input-output data. This approach is poised to drastically improve the design of microscopes because the parameters of current models of optical setups cannot be easily fit to real data. Inspired by the recent progress in deep learning, our solution is to build a differentiable wave optics simulator composed of trainable modules, each computing light wave-front (WF) propagation due to a specific optical element. We call our differentiable modules WaveBlocks and show reconstruction results in the case of lenses, wave propagation in air, camera sensors, and diffractive elements (*e.g.*, phase masks).

Microscopes have a fundamental impact on biology, life science, and engineering. Because of their essential role in imaging, their design has constantly evolved to improve the imaging quality (*e.g.*, the lateral and axial resolutions and the speed). One approach toward this goal is to improve the quality of the captured data through better sensors,

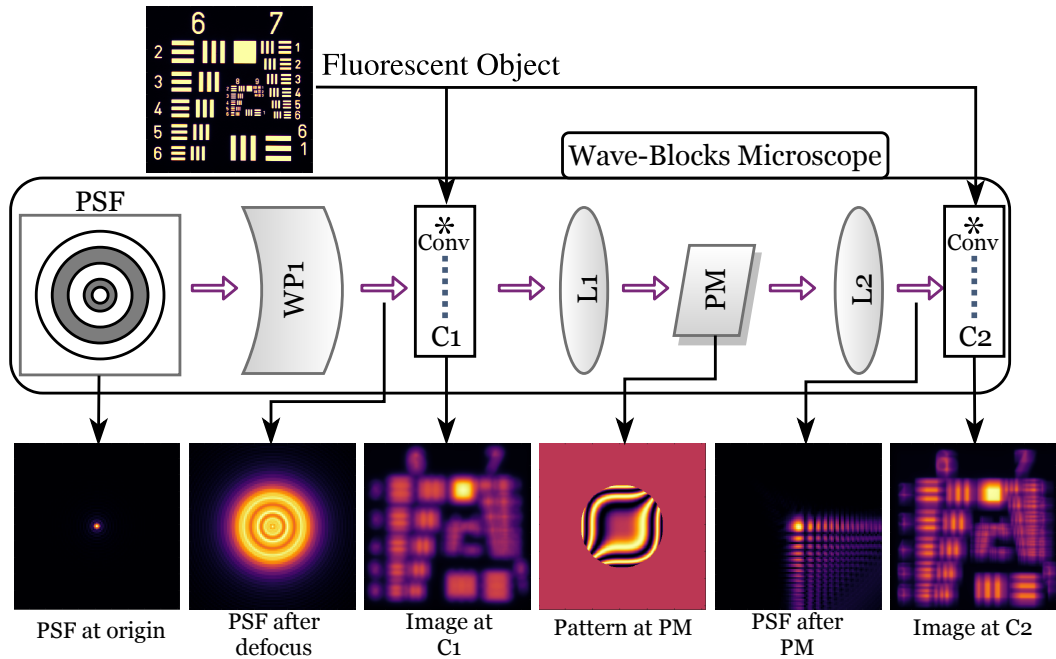


Figure 5.1: Fluorescent microscope recreated with WaveBlocks. A bright-field PSF is propagated through the air by WP1, then imaged by the first camera (C1). Alternatively, the wavefront continues to the 4-f system (L1-2) with a phase mask (PM) placed at its Fourier plane. Later, the second Camera (C2) convolves the object and the PSF from the back focal plane of L2. Each camera (C1-C2) is used for a separate experiment in sec. 5.1.1.2.

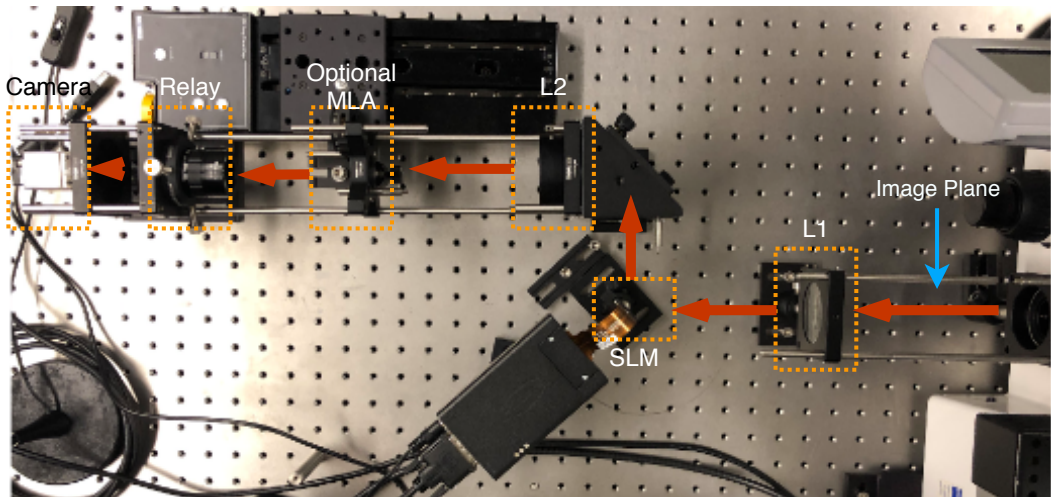


Figure 5.2: Microscope used for the experiments. See Fig. 5.1 for a definition of the optical elements involved.

optics, and configuration. Such an approach is currently based mostly on manual design and analysis and is often referred to as *point spread function engineering* [143–145].

A second approach is instead to improve the captured data via image processing. The task is posed as an inverse problem, where the latent high-quality image is reconstructed by combining a generative model of observed data (the data fidelity term) with a model of the patterns typically observed in the data (prior). In this domain, recent deep learning methods have demonstrated impressive performance, both in terms of image reconstruction accuracy and speed, thanks to their ability to efficiently and accurately describe complex data priors and generative models [146].

Traditional PSF engineering is sub-optimal as it does not consider the subsequent image-processing step. In fact, an optimal data capture system should be designed around the patterns of typically observed data, and this is a complicated problem to solve analytically due to the complex iterative nature of image reconstruction algorithms. Therefore, we argue that a better approach is to optimize both steps numerically through real data jointly. Towards this goal, in this paper, we introduce and study a model of optics that can be readily integrated with deep learning models for image processing. Our solution is to first accurately describe light propagation through optical elements (*e.g.*, lenses, air, phase-masks) with computational modules, which we call *WaveBlocks* (see Fig. 5.1). This way, it can relate each module to a physical counterpart. Secondly, to enable the automatic optimization of the optical elements, we ensure that the modules are differentiable with respect to their parameters. We can then fit real data to our compositional model through optimization (*i.e.*, via stochastic gradient descent), a procedure that we call *calibration*. In this paper, we illustrate our approach with several calibration tasks. First, we show how to accurately fit the PSF of a real microscope by using real data. Later, we show how to recover the diffracting pattern caused by the lack of a 100% fill factor in an SLM within a 4-f setup.

PSF engineering depends on the goal of the system. For example, one could aim to improve imaging for super-resolution [143, 144, 147–149], 3D reconstruction [144, 146, 150, 151], extended depth of field [145, 151, 152], even light-field microscopy [150], general phase-mask optimization [145], and calibration [143]. One approach to solve calibration is to pose the task as a blind deconvolution problem [153–155]. Also, phase retrieval and the removal of the zero-order diffraction from an SLM have been previously studied [156, 157]. A microscope simulator for generating synthetic biological images has also been explored [158]. However, such a computational model is unsuitable for the optical components’ data-driven automatic optimization. Thus, our proposed *WaveBlocks*, a differentiable and scalable optimization framework, provides a fundamental automatic data-driven PSF engineering tool. In future work, we aim to demonstrate our framework on various image processing tasks and imaging devices.

5.1.1.1 Methods

Building a Wave Optics Simulator We are interested in building a differentiable

model that can accurately approximate real optics with modules associated with each optical component in a microscope, following the image formation model defined in sec. 2.1.3. The PSF or system matrix H can be acquired in several ways: By measuring it directly with the microscope or by modeling it computationally. In our modules, the light emitted by a fluorescent particle is propagated as a complex WF through each of the optical elements in the microscope until it hits the sensor, where the irradiance of the WF is computed and stored as the PSF for later use.

The linear operators that model each optical element forming the PSF are based on complex diffraction integrals, which are difficult to optimize when stacked together. However, their linearity enables the usage of each operator as a block that receives an input WF and produces an output WF. Moreover, it is possible to stack as many blocks as desired easily. We implement these operators in the Pytorch framework [114], which performs auto-differentiation. The architecture is user-friendly and allows the optimization of various parameters in an optical system. The rationale behind WaveBlocks is that the user provides a bright-field PSF (for example, generated in Fiji [159]). This PSF propagates through the user-defined optical blocks until it reaches a camera module. In this module, the image irradiance is computed and convolved with a user-provided object, as shown in Fig. 5.1.

Wave Blocks Implementation The following sections describe the computational models of the key optical elements implemented.

Wave-front propagation (WP): To compute the monochromatic WF propagation through a medium (the WP block), we use the Rayleigh-Sommerfeld integral (see [160], page 52). Our implementation of the Rayleigh-Sommerfeld propagation is limited to a minimum distance of $200\mu m$ to ensure that the required sampling remains computationally practical. According to the Fourier convolution theorem (see [160], page 39), the WP block can be obtained via $U_2(x, y) = \mathbf{F}^{-1}\{\mathbf{F}\{U_1(x, y)\}\mathbf{F}\{h(x, y)\}\}$, with $h(x, y) = \frac{z}{j\lambda} \frac{\exp(jkr)}{r^2}$, where \mathbf{F} denotes the Fourier transform, \mathbf{F}^{-1} denotes its inverse, (x, y, z) are 3D image spatial coordinates, j denotes the imaginary component in complex numbers, $h(x, y)$ is the Rayleigh-Sommerfeld impulse response, λ is the light wavelength, k is the wave number and $r = \sqrt{z^2 + x^2 + y^2}$ is the distance from a point in U_1 to a point in U_2 .

Lens (L): The lens block (the L block) describes a convex lens that propagates a WF from a plane at the front of the lens to the back of it. According to the Fraunhofer approximation (see [160], page 97), the lens propagation from the front plane to the back focal plane is given by the following equation

$$U_2(x_2, y_2) = c \iint U_1(x_1, y_1) P(x_1 + x_2, y_1 + y_2) \exp[-j \frac{2\pi}{\lambda f_l} (x_1 x_2 + y_1 y_2)] dx_1 dy_1 \quad (5.1)$$

where $P(x, y)$ is the pupil function and c is the scaling factor given by $c = \frac{\exp(jk f_l)}{j\lambda f_l}$, where f_l is the focal length of the lens.

Camera (C):

The camera block (the C block) performs two tasks: 1) It determines the PSF H by computing the irradiance of the incoming WF, which is the time-averaged square magnitude of the field U , given by $H(x, y) = U(x, y)U^*(x, y) = |U(x, y)|^2$. This time averaging occurs due to the inability of current detectors to follow the high-frequency oscillations of the electric field ($> 10^{14}Hz$) (see [160], page 49); 2) It convolves the computed PSF H with an object V placed in front of the microscope, *i.e.*, $i = H * V$.

Phase-mask (PM) or spatial light modulator (SLM): The phase-mask block (the PM block) describes a phase mask that distorts the phase of the WF in a space-variant way. The incoming WF U_1 is modified by the modulation function ϕ , defined by the PM block, as in $U_2(x, y) = U_1(x, y) \exp(j\phi(x, y))$. A particular case applies when a phase-only SLM is used, where only the imaginary part of the exponential is multiplied by U_1 , and the real part becomes zero. In WaveBlocks, ϕ is a parameter that can be changed freely or even optimized for.

Data-driven calibration

To calibrate an optical system, we capture images of several known objects. If we denote the real system matrix with H^{gt} , captured real images i of known objects V are modeled as $i = H^{\text{gt}} * V$. Thus, WaveBlocks allows us to describe this data via a system matrix H_Θ parametrized via Θ , which collects all the settings of the optical elements (*e.g.*, the main objective PSF, the phase-mask phase change). Then, this optimization problem can be written as the minimization of the expectation over all objects and corresponding observed images (which can be approximated by using a finite number of samples)

$$\hat{\Theta} = \arg \min_{\Theta} E_{i,o} [\ell(H_\Theta * o, i)] \quad (5.2)$$

where ℓ is the cost function measuring the discrepancy between the observed images i and the synthesized ones $H_\Theta * o$. The successive stack of blocks can form H_Θ . For example, the configuration shown in Fig. 5.1 yields $H_\Theta = C2(L2(PM(L1(WP1(PSF))))))$, where Θ has the parameters of the PM and/or the PSF. This configuration is also used in the experiments in sec. 5.1.1.2.

5.1.1.2 Experiments

To demonstrate the capabilities of our proposed calibration approach, we aim to reconstruct two sets of parameters: 1) the initial PSF of the microscope and 2) the distortion diffraction pattern generated by the inactive space between the pixels of an SLM, shown in Fig. 5.2. We employ a USAF 1951 resolution target, a $20\times 0.45\text{NA}$ objective, a 165mm focal length tube-lens, a camera (C1-C2) with $6.9\mu\text{m}$ pixel size, two lenses (L1-L2) with a focal length of 150mm and aperture of 50.8mm and the PM used is an SLM Holoeye Pluto-vision. All optimizations use the Adam optimizer.

Bright-field PSF estimation In this section, we focus on reconstructing the PSF of

5 Mix of both worlds: Joint optimization

	NMSE(i^{gt}, i)		NMSE($\Theta^{\text{gt}}, \Theta$)	
	Mean	Std-Dev	Mean	Std-Dev
PSF	$9.34 \cdot 10^{-3}$	$6.95 \cdot 10^{-3}$	1.5	0.43
PM	$6.30 \cdot 10^{-2}$	$2.14 \cdot 10^{-3}$	0.51	0.42

Table 5.1: Mean and standard deviation of NMSE error between the image stack (i) captured by $C1$ and $C2$ and the ground truth stack (i^{gt}). The first row is the PSF in experiment one, and the second row is the PM in experiment two.

a real microscope in the simplest case, where only a PSF block representing the PSF generated by the objective and tube lens, a wave-propagation block, and a camera block is used (PSF, WP1, and C1 in Fig. 5.1). As V we choose the USAF 1951 target object (only one image in this case), and the loss is the normalized mean square error (NMSE) loss given by $\text{NMSE}(i, g) = \frac{\|i-g\|_2^2}{\|i\|_2^2}$.

Recovery of the PSF from synthetic data: To measure the robustness of the PSF reconstruction algorithm, we randomly transform and reconstruct a PSF computed via the scalar Debye theory [78, 161]. First, we translate the object randomly in a range of ± 20 pixels, then re-scale it in x and y directions independently in the range 1 ± 0.3 and defocus it using WP1 by a random distance between ± 200 and $\pm 1000 \mu\text{m}$ in image space. H^{gt} is built by setting the parameters Θ^{gt} . A total of 500 different PSFs were estimated using defocused images at $-50, 0,$ and $50 \mu\text{m}$ relative to the focal plane of the objective. Table 5.1 shows that even though the error in the PSF parameters estimation is not small, the error in the image reconstruction is very low, a sign that there exist ambiguities in solution space. This is not an issue since we can use any solution to reproduce the observed images.

Recovery of the PSF from real data: In this experiment, we aim to recover the PSF of a real microscope. We acquire a stack of images of the USAF 1951 target placed at depths spanning -50 to $50 \mu\text{m}$ in steps of $2 \mu\text{m}$ relative to the focal plane of the objective. For this experiment, only the images at depths $-50, 50,$ and $0 \mu\text{m}$ were used in training, and the rest were stored for later testing. The recovered PSF can be seen in Fig. 5.3 next to the PSF of an ideal microscope. Notice how the recovered PSF exhibits aberrations not present in the ideal case.

To verify that the recovered PSF is optimal, we use the optical configuration in Fig. 5.1 with the optimized PSF to predict the depth at which an image with the real microscope was taken. We compare each image in the full depth range against a stack generated through our estimated model and plot which depth yields the smallest error. As shown at the top of Fig. 5.4 the recovered PSF achieves the highest accuracy, with a mean NMSE error of $2.8 \mu\text{m}$, against the initial PSF, with an error of $19.21 \mu\text{m}$. Note how the ideal PSF sometimes predicts the wrong direction of defocus.

Phase-mask distortion pattern recovery

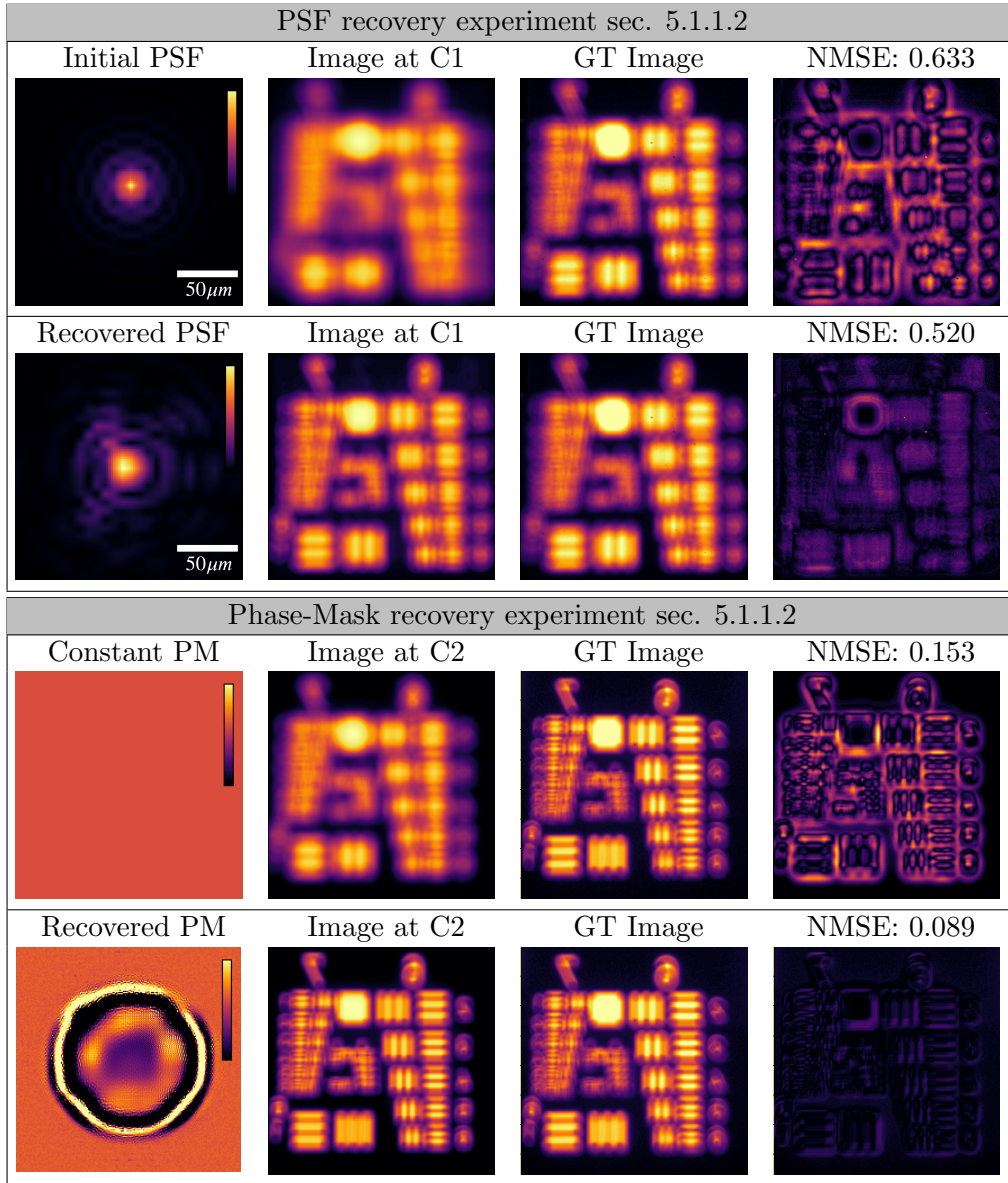


Figure 5.3: First row: comparison between the PSF (at $0\mu m$) of an ideal microscope and the recovered PSF obtained through our approach. Second row: comparison between a constant PM pattern and the recovered one.

5 Mix of both worlds: Joint optimization

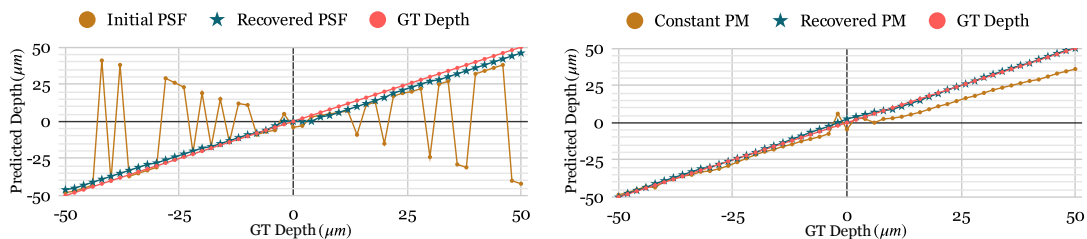


Figure 5.4: Top: Comparison of depth prediction using the initial PSF or the optimized one. Bottom: Comparison of depth prediction using the recovered PSF and either a constant PM as displayed in the SLM or the recovered PM.

In this section, we aim to recover the diffraction pattern created by the lack of a 100% pixel fill factor in an SLM [156, 157] placed at the Fourier plane of a 4-f system. The zero-order diffraction lobe hits exactly the center of the output image, thus hampering the usability of an SLM.

With Synthetic data: Consistently with sec. 5.1.1.2, a test on synthetic data was first performed, where a PM was selected randomly between a cubic mask due to its simplicity, and stable behavior in the Fourier domain [152] and a circular mask with a circular gradient towards the center, simulating the zero diffraction mode produced by a real SLM. Then, random scaling and translation were performed. A total of 500 examples were run, for which the results are presented in Table 5.1.

Recovering the phase-mask distortion pattern: We use the recovered PSF from sec. 5.1.1.2 in a 4-f system and propagate the WF until C2. As discussed in sec. 5.1.1.2, the diffraction pattern created by the SLM distorts the PSF in the frequency domain, and by recovering this pattern, the distortion can be corrected. In this experiment, we show that by using our synthesis model with the correction pattern displayed at the SLM, an accurate depth of a stack of defocused images can be inferred (see Fig. 5.4 bottom). We observe a mean NMSE error of $0.67\mu\text{m}$ against $5.70\mu\text{m}$ without PM correction. The recovered phase range matches the vendor’s description (0 to 5π max shift).

5.1.1.3 Discussion

We have introduced a novel learning-based method to build a differentiable computational model of a real microscope. Experiments with synthetic and real experiments demonstrated that the proposed method allows the recovery of latent parameters of an optical setup (*e.g.*, a PSF or a PM) with high accuracy. We encourage the reader to explore the project <https://github.com/pvjosue/WaveBlocks> repository for working examples and constant updates.

6 Toolkit for research project management

Through this Ph.D., a variety of projects took place. Regarding timelines, some lasted months and other years, and, regarding collaboration, most of them were individual projects, but some also interdisciplinary teams and supervision of student projects.

In general, tools for managing research projects were essential for organized advancement and reduced procrastination during work-from-home periods.

In this chapter, I present a set of tools for different parts of project management that could aid any Ph.D. student and perhaps me of the past.

6.1 Time management

Time management became a key necessity, especially during the COVID-19 pandemic, where working from home was necessary, and going to the office stayed optional even after it was over. The Pomodoro technique is based on having working cycles of 25 minutes, followed by 5-minute breaks, and after four cycles, a 25-minute break. Although this sounds trivial, I don't know anyone who uses it consistently apart from me.

Pomodoro allowed me to manage my working time; for example, sometimes working 8 hours straight (16 cycles) becomes heavy and tiring. In the summer days, working at midday is unfeasible due to the heat. So, I would split my day accordingly: do eight cycles in the morning (8 AM till midday) and 6-8 cycles in the evening when the sun goes down (4 PM - 8 PM). This, of course, changed around the year, but it was a consistent way of measuring my amount of work.

And interestingly, with some tools, you can even get some statistics of cycles, as seen in Fig. 6.1

6.2 Progress tracking

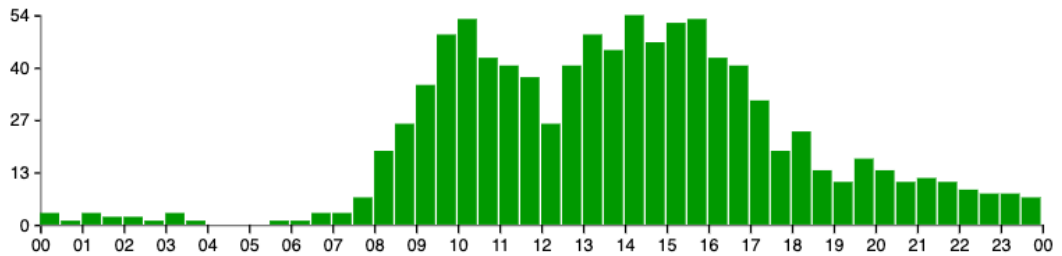
In terms of progress tracking and planning, gathering information was required, for example:

1. Listing all or a subset of tasks to do.
2. Mention the estimated task duration (perhaps given in Pomodoro cycles, as in 6.1).
3. Details of each task.
4. Assign them to the correct person.
5. Define task hierarchy and order.

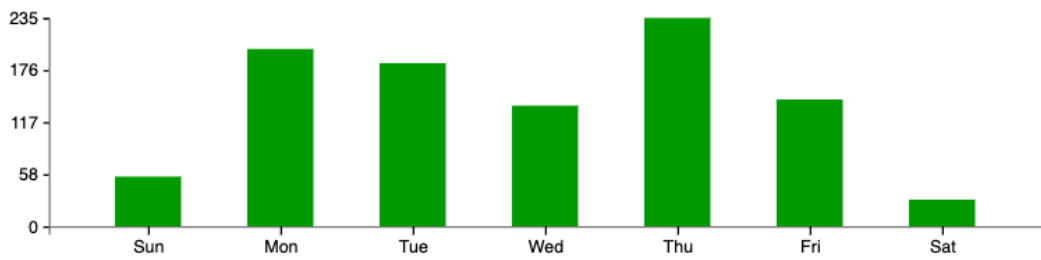
6 Toolkit for research project management

Daily Distribution

15 MIN 30 MIN 1 HR 2 HR



Weekly Distribution



871 Pomodoros in the Last 9 Months

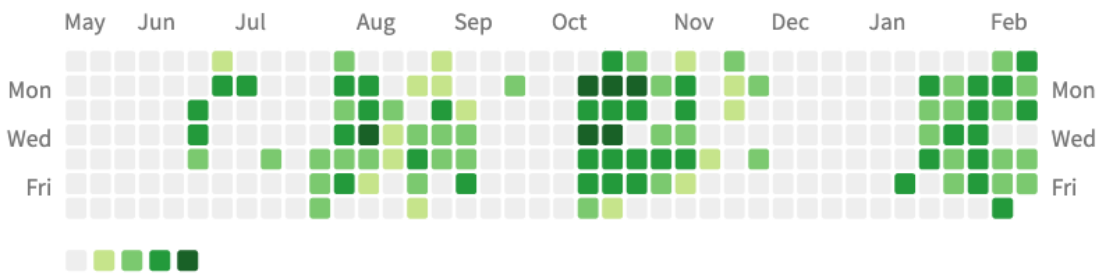
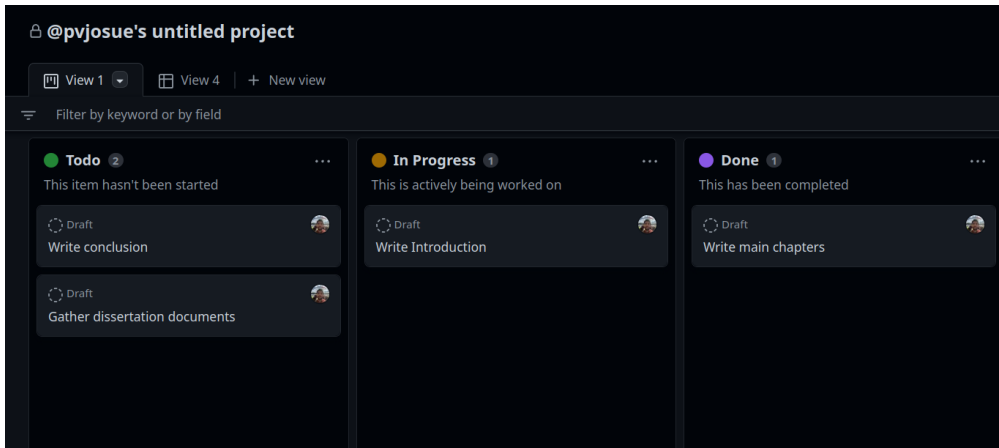


Figure 6.1: Pomodoro cycle statistics from the last 9 months



(a) Board view

The screenshot shows a list view with columns: Title, Assignees, and Status. The items are:

ID	Title	Assignees	Status
1	Write Introduction	pvjosue	In Progress
2	Write conclusion	pvjosue	Todo
3	Gather dissertation documents	pvjosue	Todo
4	Write main chapters	pvjosue	Done

(b) List view

Figure 6.2: Github project planning views

Then, when working on the project, this list can be multiply iterated through the project, as in Sprints of an Agile approach.

Some tools aid in this, for example:

6.2.1 Github project

Github (github.com) can handle project boards where planning, progress, and tracking can be easily applied, together with commit information, issues, etc. It allows representation of the tasks, their state, and information with different views, for example, board and list view, as seen in Fig. 6.2.

6.2.2 Notion

A very flexible approach to achieve the same and more is Notion (notion.so). Which is a powerful editor where you can embed images, videos, calendars, diagrams, etc., and create pages, sub-pages, and full Wikis. Notion is great for project planning, as it allows you to include all the information you want and still has a clean interface. It is beneficial for student projects, where planning and task tracking are possible in a seamless manner.

Like Github, you can arrange tasks and visualize them with different views, as seen in Fig. 6.3.

6.3 Experiment repeatability

Returning to previous configurations and results is crucial for long research projects. One advantageous approach is to use Git and constantly commit the changes. However, it is easy to lose track of the changes or comeback to specific results, especially when the source code grows in size and complexity.

As soon as I noticed that this was crucial, I took two main approaches:

6.3.1 Copy and store the used code and arguments manually

For this, I reused a code snippet across many projects, presented in the code snippet 6.1

```
1 # Create tensorboard summary
2 save_folder = '.'
3 writer = SummaryWriter(log_dir=save_folder)
4 # Store all the arguments used in the tensorboard log file
5 writer.add_text('arguments', str(vars(args)), 0)
6
7 # Store files for backup
8 import zipfile
9 zf = zipfile.ZipFile(save_folder + "/files.zip", "w")
10 # Iterate the list of files to store, for example, '*.py*'
11 for ff in args.files_to_store:
12     zf.write(str(file_path) + '/' + ff, ff)
13 zf.close()
```

Listing 6.1: Storing source code

6.3.2 The guild.ai framework

A more complete solution toward repeatability is (guild.ai), a tool 100% open source, designed for tracking experiments, pipeline automation, hyper-parameter tuning, etc.

A program run through guild will store in the 'runs' directory the following:

- A copy of the files or file type specified.
- The arguments associated with that run.
- Scalars, images, etc. logged through a tensorboard summary writer.
- The run name, date, tags, etc.

6.3.2.1 Running a script

For example, let's run the script `helloworld.py` by typing in the console:

```
1 guild run helloworld.py arg1=123 arg2=456
```

Listing 6.2: Running a program with guild

This will run and block the console until it finishes. If we want to add a tag to this run, we can do the following:

```
1 guild run helloworld.py arg1=123 arg2=456 --tag=first_run
```

Listing 6.3: Tagging runs

6.3.2.2 Listing runs status

To visualize in the console what we just ran, we can do the following:

```
1 # For all runs
2 guild runs
3 # For all runs with a given tag
4 guild runs -Ft first_run
5 # For all runs in a given time frame
6 guild runs --started "today"
```

Listing 6.4: Checking guild runs

The list of available filters is truly complete. For further detail, check the guild.ai webpage.

6.3.2.3 Visualizing in the browser

Guild has a pretty good user interface, which can be run by using the following:

```
1 # For all runs
2 guild view
3 # And support the same filtering operations just mentioned.
4 # Also view the runs in tensorboard is possible, by typing:
5 guild tensorboard
```

Listing 6.5: Visualization in the browser

6.4 Hardware management in shared computed servers

Sharing resources in a research group is quite common. Everyone had a working station and a laptop, and we shared two large computing servers. As a group that does research in machine learning, large GPUs were a must.

A very useful work paradigm is how to split the resources and how to allocate these such that we don't interfere with the research from the others.

6.4.1 Guild queues

One feature worth mentioning from the guild framework is the queues, which allow the creation of a queue of jobs, and executing them in 'runners'. These queues can have specific GPUs associated with them, making them great for multiple experiments where each run requires a full GPU, for example, hyper-parameter optimization for deep learning. You can start a queue and enqueue a job by running:

```
1 guild run queue --gpus 0
2 # Or a non-blocking queue
3 guild run queue --gpus 0 --background -y
4 # And to add a job to a queue
5 guild run helloworld.py arg1=123 arg2=456 --tag=first_run --stage
```

Listing 6.6: Starting queues with guild

Stopping a queue or any job can be done by first identifying the job id with `guild runs`, then calling `guild stop id` with the corresponding id.

6.4.2 SLURM

We also had SLURM (slurm.schedmd.com) installed in the servers, which was the main job scheduling software that we used. However, I won't go into detail, as it is well documented online, and we didn't have anything particular within our setup.

6.4 Hardware management in shared computed servers

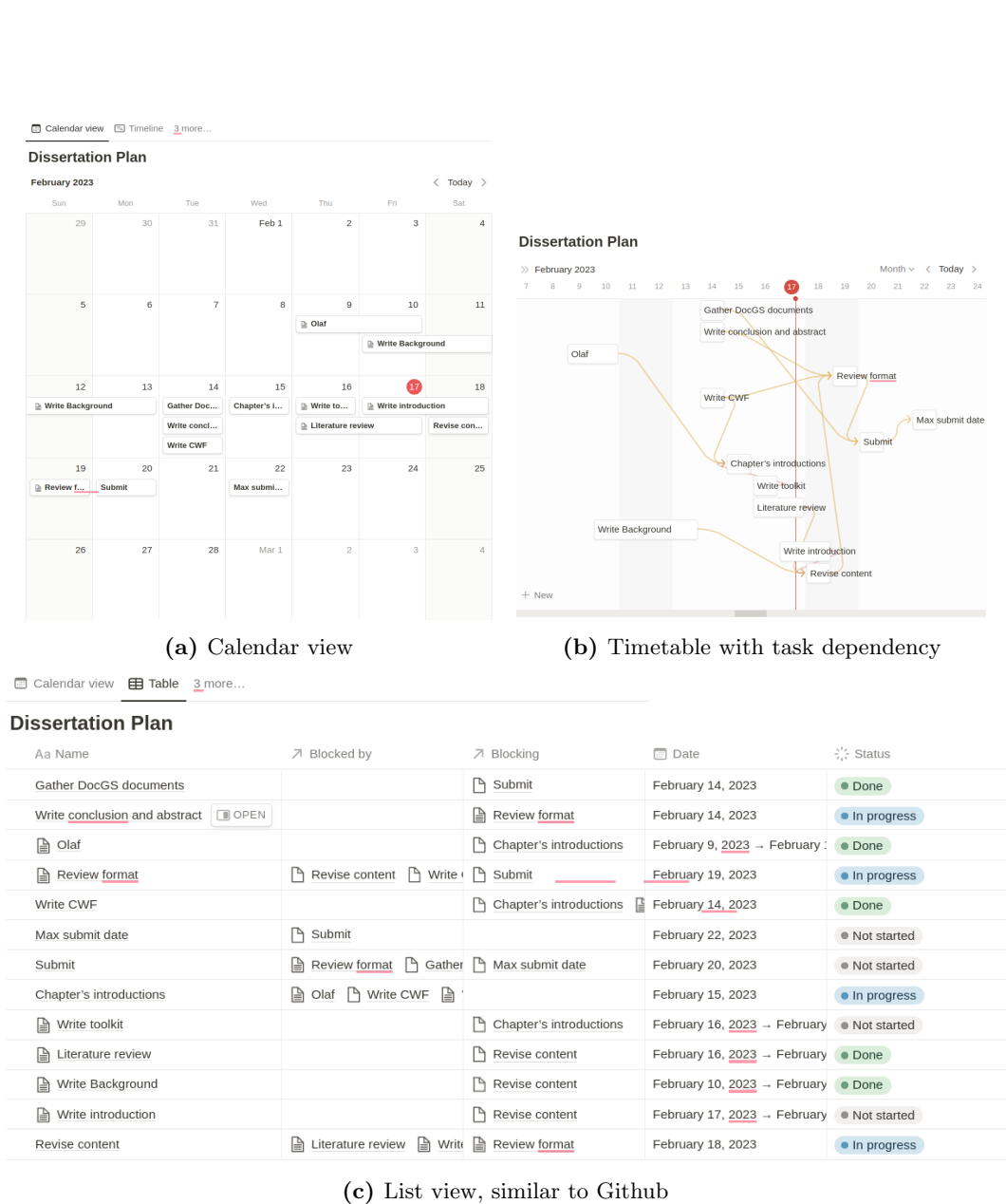


Figure 6.3: Notion project organization views.

7 Conclusion & closing thoughts

In this dissertation, I first explored the fluorescence microscopy world by clarifying why it is so essential for humanity and how important it is to have access to 3D real-time imaging. Different methods and approaches towards this goal were presented, trying to share with the reader the thought process on the decisions made in the optical and algorithm architectures.

DL for problem-solving has become an often picked approach; however, it's essential to take some things into account: Certainty metrics and ways to evaluate how sure a network is are necessary for biomedical imaging. Relying on a deep-learning-only approach might be powerful, but a physics-informed approach, where the image formation model is taken into consideration, or a jointly optimized algorithm + optics is done, may offer unprecedented quality and powerful insights.

It became clear that microscopy has evolved to be computational microscopy and that joint optimization of physical devices and reconstruction algorithms will be highly utilized in the future. And for a good reason, the features extracted from a hand-designed microscope aren't necessarily the most useful for all applications. And by relying on the algorithm to choose a design and hence the features used, we can aim to extract the most helpful information and reach unprecedented performance.

I finish this five years of work with an even greater excitement towards nature, physics, and their interaction. And looking forward to new and exciting projects to come.

Bibliography

- [1] A. Stefanoiu, J. Page Vizcaino, and T. Lasser. olaf: A flexible 3d reconstruction framework for light field microscopy. *TUM Media*, TUMI-1978, 2020.
- [2] T. G. Georgiev and Lumsdaine. Focused plenoptic camera and rendering. *Journal of Electronic Imaging*, 19(2), 2010.
- [3] A. Lumsdaine and T. Georgiev. The focused plenoptic camera. *2009 IEEE International Conference on Computational Photography, ICCP 09*, 2009. doi:10.1109/ICCPHOT.2009.5559008.
- [4] A. Stefanoiu, J. Page, P. Symvoulidis, G. G. Westmeyer, and T. Lasser. Artifact-free deconvolution in light field microscopy. *Opt. Express*, 27(22), 2019. doi:10.1364/OE.27.031644.
- [5] Z. Wang, L. Zhu, H. Zhang, G. Li, C. Yi, Y. Li, Y. Yang, Y. Ding, M. Zhen, S. Gao, T. K. Hsiai, and P. Fei. Real-time volumetric reconstruction of biological dynamics with light-field microscopy and deep learning. *Nature Methods*, 2021. doi:10.1038/s41592-021-01058-x.
- [6] M. Broxton, L. Grosenick, S. Yang, N. Cohen, A. Andalman, K. Deisseroth, and M. Levoy. Wave optics theory and 3-d deconvolution for the light field microscope. *Opt. Express*, 21(21):25418–25439, Oct 2013. doi:10.1364/OE.21.025418.
- [7] L. N. Smith. Cyclical learning rates for training neural networks. *arXiv*, 2017.
- [8] J. Page Vizcaino, Z. Wang, P. Symvoulidis, P. Favaro, B. Guner-Ataman, E. S. Boyden, and T. Lasser. Real-time light field 3d microscopy via sparsity-driven learned deconvolution. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, 2021. doi:10.1109/ICCP51581.2021.9466256.
- [9] J. Madrid-Wolff and M. Forero-Shelton. Protocol for the Design and Assembly of a Light Sheet Light Field Microscope. *Methods and Protocols*, 2(3), 2019. doi:10.3390/mps2030056.
- [10] D. Wang, S. Xu, P. Pant, E. Redington, S. Soltanian-Zadeh, S. Farsiu, and Y. Gong. Hybrid light-sheet and light-field microscope for high resolution and large volume neuroimaging. *Biomedical Optics Express*, 10(12), 2019. doi:10.1364/boe.10.006595.

BIBLIOGRAPHY

- [11] T. Schrödel, R. Prevedel, K. Aumayr, M. Zimmer, and A. Vaziri. Brain-wide 3d imaging of neuronal activity in *Caenorhabditis elegans* with sculpted light. *Nature methods*, 10(10):1013–1020, 2013.
- [12] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. *ACM Transactions on Graphics*, 25(3), 2006. [arXiv:1508.03590](https://arxiv.org/abs/1508.03590), [doi:10.1145/1141911.1141976](https://doi.org/10.1145/1141911.1141976).
- [13] N. Cohen, S. Yang, A. Andalman, M. Broxton, L. Grosenick, K. Deisseroth, M. Horowitz, and M. Levoy. Enhancing the performance of the light field microscope using wavefront coding. *Optics Express*, 22(20), 2014. [doi:10.1364/OE.22.024817](https://doi.org/10.1364/OE.22.024817).
- [14] M. Levoy, Z. Zhang, and I. McDowall. Recording and controlling the 4 D light field in a microscope using microlens arrays. *Journal of Microscopy*, 235(2), 2009. [doi:10.1111/j.1365-2818.2009.03195.x](https://doi.org/10.1111/j.1365-2818.2009.03195.x).
- [15] T. E. Bishop and P. Favaro. The light field camera: Extended depth of field, aliasing, and superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5), 2012. [doi:10.1109/TPAMI.2011.168](https://doi.org/10.1109/TPAMI.2011.168).
- [16] L. Grosenick, T. Anderson, and S. J. Smith. Elastic source selection for in vivo imaging of neuronal ensembles. *Proceedings - 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2009*, 2009. [doi:10.1109/ISBI.2009.5193292](https://doi.org/10.1109/ISBI.2009.5193292).
- [17] L. M. Grosenick, M. Broxton, C. K. Kim, C. Liston, B. Poole, S. Yang, A. S. Andalman, E. Scharff, N. Cohen, O. Yizhar, C. Ramakrishnan, S. Ganguli, P. Suppes, M. Levoy, and K. Deisseroth. Identification Of Cellular-Activity Dynamics Across Large Tissue Volumes In The Mammalian Brain. *bioRxiv*, 2017. [arXiv:132688](https://arxiv.org/abs/132688), [doi:10.1101/132688](https://doi.org/10.1101/132688).
- [18] T. Nöbauer, O. Skocek, A. J. Pernía-Andrade, L. Weilguny, F. Martínez Traub, M. I. Molodtsov, and A. Vaziri. Video rate volumetric Ca²⁺ imaging across cortex using seeded iterative demixing (SID) microscopy. *Nature Methods*, 14(8), 2017. [doi:10.1038/nmeth.4341](https://doi.org/10.1038/nmeth.4341).
- [19] N. C. Pégard, H.-Y. Liu, N. Antipa, M. Gerlock, H. Adesnik, and L. Waller. Compressive light-field microscopy for 3D neural activity recording. *Optica*, 3(5), 2016. [doi:10.1364/optica.3.000517](https://doi.org/10.1364/optica.3.000517).
- [20] R. Prevedel, Y.-g. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, and A. Vaziri. Simultaneous whole- animal 3D imaging of neuronal activity using light-field microscopy. *Nature Methods*, 11(7), 2014. [doi:10.1038/nmeth.2964](https://doi.org/10.1038/nmeth.2964).

- [21] C. Cruz Perez, A. Lauri, P. Symvoulidis, M. Cappetta, A. Erdmann, and G. G. Westmeyer. Calcium neuroimaging in behaving zebrafish larvae using a turn-key light field camera. *Journal of Biomedical Optics*, 20(9), 2015. doi:10.1117/1.jbo.20.9.096009.
- [22] Z. Zhang, L. Bai, L. Cong, P. Yu, T. Zhang, W. Shi, F. Li, J. Du, and K. Wang. Capturing volumetric dynamics at high speed in the brain by confocal light field microscopy. *bioRxiv*, 2020. doi:10.1101/2020.01.04.890624.
- [23] O. Skocek, T. Nöbauer, L. Weilguny, F. Martínez Traub, C. N. Xia, M. I. Molodtsov, A. Grama, M. Yamagata, D. Aharoni, D. D. Cox, P. Golshani, and A. Vaziri. High-speed volumetric imaging of neuronal activity in freely moving rodents. *Nature Methods*, 15(6), 2018. doi:10.1038/s41592-018-0008-0.
- [24] Q. W. Lin Cong, Zeguan Wang, Yuming Chai, Wei Hang, Chunfeng Shang, Wenbin Yang, Lu Bai, Jiulin Du, Kai Wang Is a corresponding author. Rapid whole brain imaging of neural activity in freely behaving larval zebrafish (*Danio rerio*). *eLIFE*, 6, 2017.
- [25] S. Aimon, T. Katsuki, T. Jia, L. Grosenick, M. Broxton, K. Deisseroth, T. J. Sejnowski, and R. J. Greenspan. Fast near-whole-brain imaging in adult drosophila during responses to stimuli and behavior. *PLoS Biology*, 17(2), 2019. doi:10.1371/journal.pbio.2006732.
- [26] M. Shaw, H. Zhan, M. Elmi, V. Pawar, C. Essmann, and M. A. Srinivasan. Three-dimensional behavioural phenotyping of freely moving *C. Elegans* using quantitative light field microscopy. *PLoS ONE*, 13(7), 2018. doi:10.1371/journal.pone.0200108.
- [27] N. C. Pégard, H.-Y. Liu, N. Antipa, M. Gerlock, H. Adesnik, and L. Waller. Compressive light-field microscopy for 3D neural activity recording. *Optica*, 3(5), 2016. doi:10.1364/optica.3.000517.
- [28] Z. Lu, J. Wu, H. Qiao, T. Yan, Z. Zhou, X. Zhang, J. Fan, and Q. Dai. Artifact-free 3d deconvolution for light field microscopy. In *Biophotonics Congress: Optics in the Life Sciences Congress 2019*. OSA, 2019. doi:10.1364/NTM.2019.NS1B.2.
- [29] F. A. Haight. Handbook of the poisson distribution. In *John Wiley and Sons*. John Wiley and Sons, 1967. doi:ISBN978-0-471-33932-8.
- [30] H. Verinaz-Jadan, P. Song, C. L. Howe, A. J. Foust, and P. L. Dragotti. Volume reconstruction for light field microscopy. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1459–1463, 2020.
- [31] I. C. Ghiran. Introduction to fluorescence microscopy. *Methods Mol Biol*, 689:93–136, 2011.

BIBLIOGRAPHY

- [32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [33] R. A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925.
- [34] L. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745–754, 1974.
- [35] W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.
- [36] A. E. Ștefănoiu. *Fluorescence light field microscopy: wave optics modeling and 3D image reconstruction*. PhD thesis, TUM School of Computation, Information and Technology, 2022.
- [37] A. D. Smith, C.-M. Chuong, P. D. Tran, Y. Kim, and D.-S. Kim. Three-dimensional imaging of fluorescent proteins in live cells using brightfield microscopy. *Nature methods*, 7(5):339–341, 2010.
- [38] M. O. Scully and M. S. Zubairy. Confocal scanning laser microscopy and its biological applications. *Science*, 278(5338):2075–2080, 1997.
- [39] E. A. Stettner, R. Fan, M. E. Sykes, R. F. Tyo, and E. D. Young. Light-sheet fluorescence microscopy: a whole-animal imager. *Nature methods*, 7(6):527–529, 2010.
- [40] K. Lee, T. Harris, and A. Woolley. Nonlinear magic: multiphoton microscopy in the biosciences. *Nature methods*, 5(5):410–413, 1998.
- [41] Y. Zhu, S. Wang, and S. J. Quirk. Fourier ptychographic microscopy. *Nature photonics*, 9(6):430–434, 2015.
- [42] R. Rigler, E. Stelzer, and M. Fernandez. Fluorescence correlation spectroscopy: a new tool for studies of membrane transport and protein-protein interaction. *Nature*, 369(6479):548–550, 1994.
- [43] T. Harris and A. Woolley. Lifetime imaging in fluorescence microscopy. *Nature methods*, 4(2):119–126, 1997.
- [44] S. Xia and X. Zhuang. Superresolution fluorescence microscopy. *Annual review of biochemistry*, 78:993–1016, 2009.
- [45] S. W. Hell and J. Wichmann. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Optics letters*, 19(11):780–782, 1994.

- [46] C. M. Li, D. R. Williams, and P. H. O’Toole. Structured illumination microscopy: a simple method for breaking the diffraction resolution limit. *Nature methods*, 5(7):635–641, 2008.
- [47] E. Betzig, X. Chen, C. R. Baldwin, S. Wang, Y. Wei, D. B. Bartel, J. S. Kirchhausen, R. D. Kirchhausen, and X. Zhuang. Stochastic optical reconstruction microscopy (storm) and single-molecule imaging. *Science*, 313(5793):1642–1645, 2006.
- [48] X. Zhuang, E. Betzig, J. S. Weisshaar, J. R. HEuser, D. B. Arnold, and R. M. Ryan. 3d super-resolution imaging by stochastic optical reconstruction microscopy (storm). *Nature*, 455(7216):487–491, 2008.
- [49] A. Keller, S. L. Rickelt, J. Börcsök, and K. L. Briggman. Fast, three-dimensional, volumetric imaging of fluorescent molecules in large samples. *Nature methods*, 7(8):599–605, 2010.
- [50] T. Eichele, J. L. Nielsen, M. K. Callaway, and E. Y. Jones. Real-time imaging of neural activity in the awake mouse brain with multi-photon microscopy. *Nature protocols*, 5(6):1051–1059, 2010.
- [51] R. Gordon and J. Sedat. Real-time, 3d imaging of live cells using two-photon excited fluorescence and structured illumination microscopy. *Nature methods*, 3(7):555–560, 2006.
- [52] H. Sun, J. Min, J. Lee, and P. T. So. Simultaneous multiplane imaging with reverberation two-photon microscopy. *Nature communications*, 6:8763, 2015.
- [53] T. Wilson, C. Davey, and S. Allan. Spinning disk confocal microscopy: design, performance, and applications. *Journal of Microscopy*, 217(1):17–28, 2005.
- [54] R. Prevedel, Y.-G. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, et al. Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy. *Nature methods*, 11(7):727–730, 2014.
- [55] N. Wagner, N. Norlin, J. Gierten, and G. D. Medeiros. Instantaneous isotropic volumetric imaging of fast biological processes. *Nature Methods*, 16, 2019.
- [56] Z. Wang, Y. Ding, S. Satta, M. Roustaei, P. Fei, and T. K. Hsiai. A hybrid of light-field and light-sheet imaging to decouple myocardial biomechanics from intracardiac flow dynamics. *bioRxiv*, 2020.
- [57] N. Wagner, F. Beuttenmueller, N. Norlin, J. Gierten, J. C. Boffi, J. Wittbrodt, M. Weigert, L. Hufnagel, R. Prevedel, and A. Kreshuk. Deep learning-enhanced light-field imaging with continuous validation. *Nature Methods*, 18(5):557–563, May 2021. doi:10.1038/s41592-021-01136-0.

BIBLIOGRAPHY

- [58] J. Wu, Z. Lu, H. Qiao, X. Zhang, K. Zhanghao, H. Xie, T. Yan, G. Zhang, X. Li, Z. Jiang, X. Lin, L. Fang, B. Zhou, J. Fan, P. Xi, and Q. Dai. 3D observation of large-scale subcellular dynamics in vivo at the millisecond scale. *bioRxiv*, 2019. doi:10.1101/672584.
- [59] P. Quicke, C. L. Howe, P. Song, H. V. Jadan, C. Song, T. Knöpfel, M. Neil, P. L. Dragotti, S. R. Schultz, and A. J. Foust. Subcellular resolution three-dimensional light-field imaging with genetically encoded voltage indicators. *Neurophotonics*, 7(03), 2020. doi:10.1117/1.nph.7.3.035006.
- [60] Q. Lin, J. Manley, M. Helmreich, F. Schlumm, J. M. Li, D. N. Robson, F. Engert, A. Schier, T. Nöbauer, and A. Vaziri. Cerebellar neurodynamics predict decision timing and outcome on the single-trial level. *Cell*, 180(3):536–551, 2020.
- [61] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. *Main*, 2005.
- [62] L. Haoyu, G. Changliang, K.-H. Deborah, L. Weiyi, A. Yelena, S. Bryce, L. Wenhao, M. Yizhi, B. F. Jarrod, T. Ken-Ichi, A. F. Michael, and J. Shu. Fast, volumetric live-cell imaging using high-resolution light-field microscopy. *Biomedical Optics Express*, 10(1), 2019.
- [63] L. Cong, Z. Wang, Y. Chai, W. Hang, C. Shang, W. Yang, L. Bai, J. Du, K. Wang, and Q. Wen. Rapid whole brain imaging of neural activity in freely behaving larval zebrafish (*Danio rerio*). *eLife*, 6:e28158, sep 2017. doi:10.7554/eLife.28158.
- [64] G. Scrofani, J. Sola-Pikabea, A. Llavador, E. Sanchez-Ortiga, J. C. Barreiro, G. Saavedra, J. Garcia-Sucerquia, and M. Martínez-Corral. FIMic: design for ultimate 3D-integral microscopy of in-vivo biological samples. *Biomedical Optics Express*, 9(1), 2018. doi:10.1364/boe.9.000335.
- [65] Y. Sung. Snapshot projection optical tomography. *arXiv*, 2019. arXiv:1906.04720.
- [66] L. Cong, Z. Wang, Y. Chai, W. Hang, C. Shang, W. Yang, L. Bai, J. Du, K. Wang, and Q. Wen. Rapid whole brain imaging of neural activity in freely behaving larval zebrafish (*danio rerio*). *eLife*, 6, Sep 2017. 28930070[pmid]. doi:10.7554/eLife.28158.
- [67] Q. Geng, Z. Fu, and S.-C. Chen. High-resolution 3D light-field imaging. *Journal of Biomedical Optics*, 25(10):106502, 2020. doi:10.1117/1.JBO.25.10.106502.
- [68] L. Zhu, C. Yi, G. Li, Y. Zhao, and P. Fei. Deep-learning based dual-view light-field microscopy enabling high-resolution 3d imaging of dense signals. In *Biophotonics Congress 2021*, page DTh2A.3. Optica Publishing Group, 2021. doi:10.1364/BODA.2021.DTh2A.3.

- [69] Z. Pan, M. Lu, and S. Xia. Diffraction-Assisted Light Field Microscopy for Microtomography and Digital Volume Correlation with Improved Spatial Resolution. *Experimental Mechanics*, C, 2019. doi:10.1007/s11340-019-00522-2.
- [70] T. Nöbauer, O. Skocek, A. J. Pernía-Andrade, L. Weilguny, F. M. Traub, M. I. Molodtsov, and A. Vaziri. Video rate volumetric ca 2+ imaging across cortex using seeded iterative demixing (sid) microscopy. *Nature methods*, 14(8):811, 2017.
- [71] N. C. Pégard, H.-Y. Liu, N. Antipa, M. Gerlock, H. Adesnik, and L. Waller. Compressive light-field microscopy for 3d neural activity recording. *Optica*, 3(5):517–524, 2016.
- [72] P. Song, H. V. Jadan, C. L. Howe, P. Quicke, A. J. Foust, and P. Luigi Dragotti. Model-inspired deep learning for light-field microscopy with application to neuron localization. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8087–8091, 2021. doi:10.1109/ICASSP39728.2021.9414236.
- [73] T. Kattenborn, G. Wetzstein, J. Böhme, M. Voit, U. Wallrabe, G. Pitruzzello, F. Rückriem, J. Wittbrodt, and V. Willenbockel. Microfluidic generation of water droplet lenses for digital light field microscopy. *Nature communications*, 7:11346, 2016.
- [74] K. Yanny, N. Antipa, W. Liberti, S. Dehaeck, K. Monakhova, F. L. Liu, K. Shen, R. Ng, and L. Waller. Miniscope3d: optimized single-shot miniature 3d fluorescence microscopy. *Light: Science & Applications*, 9(1):171, 2020. doi:10.1038/s41377-020-00403-7.
- [75] L. Waller and S. Manley. Wavefront coding for 3d fluorescence microscopy. *Nature Methods*, 12(4):361–369, 2015.
- [76] D. Deb, Z. Jiao, R. Sims, A. B. Chen, M. Broxton, M. B. Ahrens, K. Podgorski, and S. C. Turaga. Fourieriennets enable the design of highly non-local optical encoders for computational imaging, 2021. doi:10.48550/ARXIV.2104.10611.
- [77] A. Muthumbi, A. Chaware, K. Kim, K. C. Zhou, P. C. Konda, R. Chen, B. Judkewitz, A. Erdmann, B. Kappes, and R. Horstmeyer. Learned sensing: jointly optimized microscope hardware for accurate image classification. *Biomed. Opt. Express*, 10(12):6351, 2019. doi:10.1364/boe.10.006351.
- [78] M. Broxton, L. Grosenick, S. Yang, N. Cohen, A. Andalman, K. Deisseroth, and M. Levoy. Wave optics theory and 3-d deconvolution for the light field microscope. *Opt. Express*, 21(21):25418–25439, Oct 2013. doi:10.1364/OE.21.025418.
- [79] L. B.-F. N. Dey, P. R. Zvi Kam eraud, Christophe Zimmer, and J. Z. J.-C. Olivo-Marin. 3d microscopy deconvolution using richardson-lucy algorithm. *HAL*, 2004.

BIBLIOGRAPHY

- [80] W. Liu, C. Guo, X. Hua, and S. Jia. Fourier light-field microscopy: An integral model and experimental verification. In *Biophotonics Congress: Optics in the Life Sciences Congress 2019 (BODA,BRAIN,NTM,OMA,OMP)*, page DT1B.4. Optica Publishing Group, 2019. doi:10.1364/BODA.2019.DT1B.4.
- [81] C. Guo, W. Liu, and S. Jia. Fourier-domain light-field microscopy. In *Biophotonics Congress: Optics in the Life Sciences Congress 2019 (BODA,BRAIN,NTM,OMA,OMP)*, page NS1B.3. Optica Publishing Group, 2019. doi:10.1364/NTM.2019.NS1B.3.
- [82] Z. Lu, J. Wu, H. Qiao, Y. Zhou, T. Yan, Z. Zhou, X. Zhang, J. Fan, and Q. Dai. Phase-space deconvolution for light field microscopy. *Optics Express*, 27(13), 2019. doi:10.1364/oe.27.018131.
- [83] K. Wang. Deep-learning-enhanced light-field microscopy. *Nature Methods*, 18(5):459–460, May 2021. doi:10.1038/s41592-021-01151-1.
- [84] X. Li, H. Qiao, J. Wu, Z. Lu, T. Yan, R. Zhang, X. Zhang, and Q. Dai. DeepLFM : Deep Learning-based 3D Reconstruction for Light Field Microscopy. *Biophotonics Congress*, 2019(1), 2019.
- [85] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. arXiv:1505.04597.
- [86] J. P. Vizcaíno, F. Saltarin, Y. Belyaev, R. Lyck, T. Lasser, and P. Favaro. Learning to reconstruct confocal microscopy stacks from single light field images. *IEEE Transactions on Computational Imaging*, 7:775–788, 2021. doi:10.1109/TCI.2021.3097611.
- [87] J. Page, F. Saltarin, Y. Belyaev, R. Lyck, and P. Favaro. Mouse brain lightfield-confocal stack dataset. <http://cvg.unibe.ch/media/project/page/LFMNet/index.html>, March 2020.
- [88] H. Verinaz-Jadan, P. Song, C. L. Howe, P. Quicke, A. J. Foust, and P. L. Dragotti. Deep learning for light field microscopy using physics-based models. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1091–1094, 2021. doi:10.1109/ISBI48211.2021.9434004.
- [89] Y. Zhang, B. Xiong, Y. Zhang, Z. Lu, J. Wu, and Q. Dai. Dilfm: an artifact-suppressed and noise-robust light-field microscopy through dictionary learning. *Light: Science & Applications*, 10(1):152, Jul 2021. doi:10.1038/s41377-021-00587-6.
- [90] J. Page Vizcaino, Z. Wang, P. Symvoulidis, P. Favaro, B. Guner-Ataman, E. S. Boyden, and T. Lasser. Real-time light field 3d microscopy via sparsity-driven learned deconvolution. *2021 IEEE International Conference on Computational Photography, ICCP 21*, 2021.

- [91] C. Ounkomol, S. Seshamani, M. M. Maleekar, F. Collman, and G. R. Johnson. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature Methods*, 15(11):917–920, 2018. doi:10.1038/s41592-018-0111-2.
- [92] L. Huang, H. Chen, Y. Luo, Y. Rivenson, and A. Ozcan. Recurrent neural network-based volumetric fluorescence microscopy. *Light: Science and Applications*, 10(1), 2021. arXiv:2010.10781, doi:10.1038/s41377-021-00506-9.
- [93] Y. Wu, Y. Rivenson, H. Wang, Y. Luo, E. Ben-David, L. A. Bentolila, C. Pritz, and A. Ozcan. Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning. *Nature Methods*, 16(12), 2019. doi:10.1038/s41592-019-0622-5.
- [94] E. Nehme, D. Freedman, R. Gordon, B. Ferdman, L. E. Weiss, O. Alalouf, T. Naor, R. Orange, T. Michaeli, and Y. Shechtman. Deepstorm3d: dense 3d localization microscopy and psf design by deep learning. *Nature Methods*, 17(7):734–740, Jul 2020. doi:10.1038/s41592-020-0853-5.
- [95] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller. Memory-Efficient Learning for Large-Scale Computational Imaging. *IEEE Transactions on Computational Imaging*, 6:1403–1414, 2020. arXiv:2003.05551, doi:10.1109/TCI.2020.3025735.
- [96] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [97] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [98] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows, 2015.
- [99] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.
- [100] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.
- [101] A. Denker, M. Schmidt, J. Leuschner, and P. Maass. Conditional invertible neural networks for medical imaging. *Journal of Imaging*, 7, 2021. doi:10.3390/jimaging7110243.
- [102] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

BIBLIOGRAPHY

- [103] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe. Analyzing inverse problems with invertible neural networks. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–20, 2019.
- [104] G. Anantha Padmanabha and N. Zabaras. Solving inverse problems using conditional invertible neural networks. *Journal of Computational Physics*, 433:110194, 2021. doi:<https://doi.org/10.1016/j.jcp.2021.110194>.
- [105] S. Kousha, A. Maleky, M. S. Brown, and M. A. Brubaker. Modeling srgb camera noise with normalizing flows. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17442–17450, 2022. doi:10.1109/CVPR52688.2022.01694.
- [106] J. J. Yu, K. G. Derpanis, and M. A. Brubaker. Wavelet flow: Fast training of high resolution normalizing flows. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6184–6196. Curran Associates, Inc., 2020.
- [107] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe. Guided image generation with conditional invertible neural networks. *arXiv:1907.02392*, 2019.
- [108] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- [109] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [110] MATLAB. *R2018a*. The MathWorks Inc., Natick, Massachusetts, 2018.
- [111] D. Dansereau. Light field toolbox, December 9, 2019 v0.4.
- [112] A. Stefanoiu, G. Scrofani, G. Saavedra, M. Martínez-Corral, and T. Lasser. 3d deconvolution in fourier integral microscopy. In L. Tian, J. C. Petrucci, and C. Preza, editors, *Computational Imaging V*, volume 11396, page 113960I. International Society for Optics and Photonics, SPIE, 2020. URL: <https://doi.org/10.1117/12.2558516>, doi:10.1117/12.2558516.
- [113] J. Page and P. Favaro. Learning to model and calibrate optics via a differentiable wave optics simulator. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2995–2999, 2020. doi:10.1109/ICIP40778.2020.9190870.
- [114] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.
- [115] A. Edelstein, N. Amodaj, K. Hoover, R. Vale, and N. Stuurman. Computer control of microscopes using µmanager. *Current protocols in molecular biology*, Chapter 14, Oct 2010. 20890901[pmid]. doi:10.1002/0471142727.mb1420s92.

- [116] W. Kirch. *Pearson's Correlation Coefficient*. Springer Netherlands, 2008.
- [117] L. Y. Wei, C. K. Liang, G. Myhre, C. Pitts, and K. Akeley. Improving light field camera sample design with irregularity and aberration. *ACM Transactions on Graphics*, 34(4), 2015. doi:10.1145/2766885.
- [118] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning, 2016. arXiv:1603.07285.
- [119] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), April 2004. doi:10.1109/TIP.2003.819861.
- [120] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. arXiv:1603.08155.
- [121] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [122] EPFL. Epfl caenorhabditis elegans embryo dataset, 2016. data retrieved from: <http://bigwww.epfl.ch/deconvolution/bio/>.
- [123] M. Martone, D. Price, and A. Thor. Ccdb, rattus norvegicus, protoplasmic astrocyte. cil. dataset, 2001. data retrieved from: http://www.cellimagelibrary.org/images/CCDB_5.
- [124] J. Page Vizcaino, Z. Wang, P. Symvoulidis, P. Favaro, B. Guner-Ataman, E. S. Boyden, and T. Lasser. Real-time light field 3d microscopy via sparsity-driven learned deconvolution. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, 2021. doi:10.1109/ICCP51581.2021.9466256.
- [125] Y.-G. Yoon, Z. Wang, N. Pak, D. Park, P. Dai, J. S. Kang, H.-J. Suk, P. Symvoulidis, B. Guner-Ataman, K. Wang, and E. S. Boyden. Sparse decomposition light-field microscopy for high speed imaging of neuronal activity. *Optica*, 7(10):1457, 2020. doi:10.1364/optica.392805.
- [126] Q. Wang, Q. X. Gao, G. Sun, and C. Ding. Double robust principal component analysis. *Neurocomputing*, 391:119–128, 2020. arXiv:arXiv:0912.3599v1, doi:10.1016/j.neucom.2020.01.097.
- [127] L. Z., C. M., W. L., and Y. Ma. The augmented lagrange multiplier method for exact recovery of a corrupted low-rank matrices. *Mathematical Programming*, submitted, 2009.
- [128] Y. Xiaming and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *preprint*, 2009.

BIBLIOGRAPHY

- [129] C. Herrera, F. Krach, A. Kratsios, P. Ruysen, and J. Teichmann. Denise: Deep learning based robust pca for positive semidefinite matrices, 2020. [arXiv:2004.13612](https://arxiv.org/abs/2004.13612).
- [130] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. doi:[10.1109/ICCV.2015.123](https://doi.org/10.1109/ICCV.2015.123).
- [131] O. A. Shemesh, C. Linghu, K. D. Piatkevich, D. Goodwin, O. T. Celiker, H. J. Gritton, M. F. Romano, R. Gao, C.-C. J. Yu, H.-A. Tseng, et al. Precision calcium imaging of dense neural populations via a cell-body-targeted calcium indicator. *Neuron*, 107(3):470–486, 2020.
- [132] N. Incardona, A. Tolosa, G. Saavedra, M. Martinez-Corral, and E. Sanchez-Ortiga. Fast and robust wave optics-based reconstruction protocol for fourier lightfield microscopy. *Optics and Lasers in Engineering*, 161:107336, 2023. doi:<https://doi.org/10.1016/j.optlaseng.2022.107336>.
- [133] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.
- [134] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016. doi:<https://doi.org/10.1016/j.neucom.2015.12.114>.
- [135] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.
- [136] N. Chinchor. Image quality metrics: Psnr vs. ssim. In *Proc. of the Fourth Message Understanding Conference, (MUC-4)*, pages 22–29, 1992.
- [137] E. Taskesen. findpeaks is for the detection of peaks and valleys in a 1D vector and 2D array (image)., 10 2020. URL: <https://erdogant.github.io/findpeaks>.
- [138] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.
- [139] J. Kruse, G. Detommaso, U. Köthe, and R. Scheichl. Hint: Hierarchical invertible neural transport for density estimation and bayesian inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8191–8199, 2021.
- [140] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [141] L. Ardizzone, T. Bungert, F. Draxler, U. Köthe, J. Kruse, R. Schmier, and P. Sorrenson. Framework for Easily Invertible Architectures (FrEIA), 2018-2022. URL: <https://github.com/vislearn/FrEIA>.

- [142] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL: <http://github.com/google/jax>.
- [143] A. Klauss, F. Conrad, and C. Hille. Binary phase masks for easy system alignment and basic aberration sensing with spatial light modulators in STED microscopy. *Sci. Rep.*, 7(1):1–11, 2017. doi:10.1038/s41598-017-15967-5.
- [144] B. Shuang, W. Wang, H. Shen, L. J. Tauzin, C. Flatebo, J. Chen, N. A. Moringo, L. D. Bishop, K. F. Kelly, and C. F. Landes. Generalized recovery algorithm for 3D super-resolution microscopy using rotating point spread functions. *Sci. Rep.*, 6(August):1–9, 2016. doi:10.1038/srep30826.
- [145] T. Zhao, T. Mauger, and G. Li. Optimization of wavefront-coded infinity-corrected microscope systems with extended depth of field. *Biomed. Opt. Express*, 4(8):1464, 2013. doi:10.1364/boe.4.001464.
- [146] Y. Wu, Y. Rivenson, H. Wang, Y. Luo, E. Ben-David, L. A. Bentolila, C. Pritz, and A. Ozcan. Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning. *Nat. Methods*, 16(12):1323–1331, 2019. doi:10.1038/s41592-019-0622-5.
- [147] E. Nehme, D. Freedman, R. Gordon, B. Ferdman, L. E. Weiss, O. Alalouf, T. Naor, R. Orange, T. Michaeli, and Y. Shechtman. Deepstorm3d: dense 3d localization microscopy and psf design by deep learning. *Nature Methods*, 17(7):734–740, Jul 2020. doi:10.1038/s41592-020-0853-5.
- [148] A. Durand, T. Wiesner, M. A. Gardner, L. É. Robitaille, A. Bilodeau, C. Gagné, P. De Koninck, and F. Lavoie-Cardinal. A machine learning approach for online automated optimization of super-resolution optical microscopy. *Nat. Commun.*, 9(1), 2018. doi:10.1038/s41467-018-07668-y.
- [149] G. Grover, K. DeLuca, S. Quirin, J. DeLuca, and R. Piestun. Super-resolution photon-efficient imaging by nanometric double-helix point spread function localization of emitters (spindle). *Opt. Express*, 20(24):26681–26695, Nov 2012. doi:10.1364/OE.20.026681.
- [150] N. Cohen, S. Yang, A. Andalman, M. Broxton, L. Grosenick, K. Deisseroth, M. Horowitz, and M. Levoy. Enhancing the performance of the light field microscope using wavefront coding. *Opt. Express*, 22(20):24817, 2014. doi:10.1364/OE.22.024817.
- [151] S. Quirin, D. S. Peterka, and R. Yuste. Instantaneous three-dimensional sensing using spatial light modulator illumination with extended depth of field imaging. *Opt. Express*, 21(13):16007, 2013. doi:10.1364/oe.21.016007.

BIBLIOGRAPHY

- [152] T. Zhao and F. Yu. Design of wavefront coding microscope objective based on 4f system. *OPEE 2010 - 2010 Int. Conf. Opt. Photonics Energy Eng.*, 1:76–79, 2010. doi:10.1109/OPEE.2010.5508099.
- [153] F. Soulez, L. Denis, Y. Tourneur, and É. Thiébaud. Blind deconvolution of 3D data in wide field fluorescence microscopy. *Proc. - Int. Symp. Biomed. Imaging*, pages 1735–1738, 2012. doi:10.1109/ISBI.2012.6235915.
- [154] S. Lim and J. C. Ye. Blind Deconvolution Microscopy Using Cycle Consistent CNN with Explicit PSF Layer. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11905 LNCS:173–180, 2019. arXiv:1904.02910, doi:10.1007/978-3-030-33843-5_16.
- [155] B. Kim and T. Naemura. Blind depth-variant deconvolution of 3d data in wide-field fluorescence microscopy. *Scientific Reports*, 5, 2015. doi:10.1038/srep09894.
- [156] H. Zhang, J. Xie, J. Liu, and Y. Wang. Elimination of a zero-order beam induced by a pixelated spatial light modulator for holographic projection. *Appl. Opt.*, 48(30):5834–5841, 2009. doi:10.1364/AO.48.005834.
- [157] J. Liang, S. Y. Wu, F. K. Fatemi, and M. F. Becker. Suppression of the zero-order diffracted beam from a pixelated spatial light modulator by phase compression. *Appl. Opt.*, 51(16):3294–3304, 2012. doi:10.1364/AO.51.003294.
- [158] S. Rajaram, B. Pavie, N. E. F. Hac, S. J. Altschuler, and L. F. Wu. SimuCell: a flexible framework for creating synthetic microscopy images. *Nature Methods*, 9(7):634–635, 2012. doi:10.1038/nmeth.2096.
- [159] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7):671–675, 2012. doi:10.1038/nmeth.2089.
- [160] D. G. Voelz. *Computational fourier optics: a MATLAB tutorial*, volume 534. SPIE press Bellingham, Washington, 2011.
- [161] M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th Edition)*. Cambridge University Press, 7th edition, 1999.

A Appendix

A.1 LFM geometry

A.1.0.1 Field of view of an LF image

The PSF of an optical system changes its support depending on the depth of the point light source. The LFMNet considers this aspect by using a neighborhood of lenslets to reconstruct the volume behind the central lenslets. Given this specification, we compute analytically the blur at the MLA plane generated by a point-light source placed in front of the microscope at different depths (see Fig. 2.1 in the supplementary material) according to the model used by Bishop and Favaro [15]. The derivation of the blur-depth relationship is also reported in section A.1.0.2 in the supplementary material.

A.1.0.2 Blur of an object at the MLA plane

As described in section A.1.0.1 the number of MLAs that gather information about an object in front of the microscope depends on the MLA blur equation given by

$$ML_b = \frac{TL_r \cdot |c - i_2|}{i_2}, \quad (\text{A.1})$$

where TL_r is the blur radius size at the tube-lens, c the distance from the tube-lens to the MLA and i_2 the position where a point in object space forms the image. We take into account that the objective back aperture (with radius $Obj_r = F_{obj} \cdot NA$) works as a telecentric stop and that the distance between the objective and the tube-lens is equal to the sum of their focal lengths (as in a 4-F system), which is equal to $(M + 1) \cdot F_{obj}$. From similar triangles, we find that

$$TL_r = \frac{Obj_r [i_1 - (M + 1)F_{obj}]}{i_1}. \quad (\text{A.2})$$

When imaging an object of size Os , its blur diameter is $ML_{tb} = 2 \cdot ML_b + M \cdot Os$. By using the thin lens equation $1/i + 1/o = 1/F$, we can express i_1 and i_2 in terms of the point emitter and the position in object space o_1 such that

$$i_1 = \frac{F_{obj} o_1}{o_1 - F_{obj}} i_2 = M(F_{obj}(1 + M) - M \cdot o_1). \quad (\text{A.3})$$

This relationship between object depth and MLA blur can be better observed in Fig. A.1.

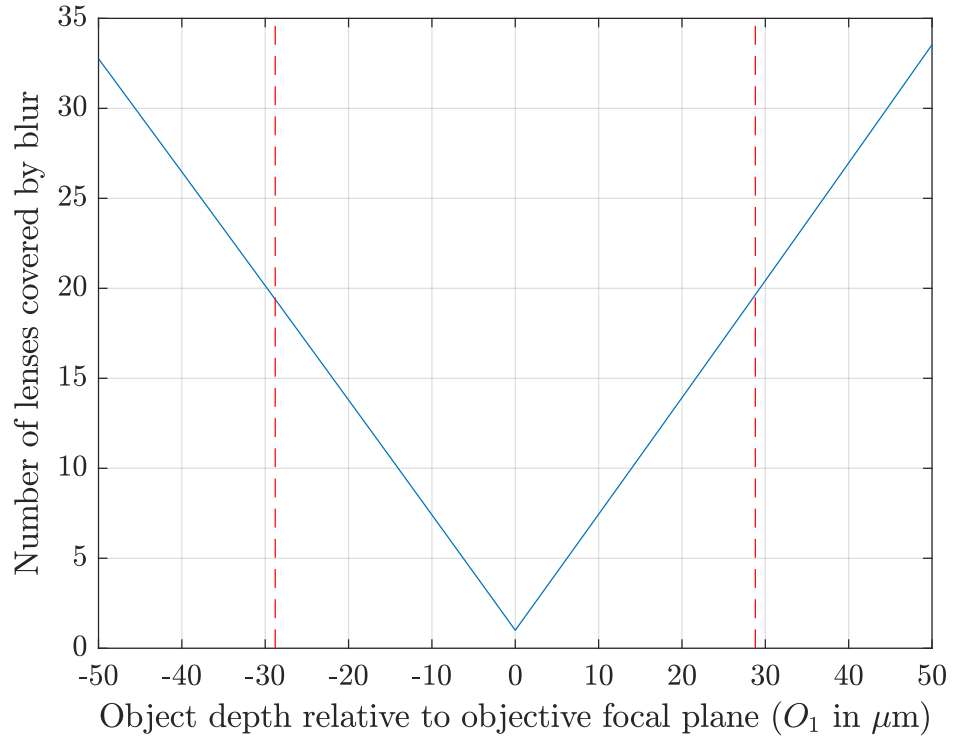


Figure A.1: Blur at MLA vs. the number of lenslets. The number of micro-lenses overlapping with the blur from an object with size $O_s = \frac{MLpitch}{M} = \frac{112}{40} \mu\text{m} = 2.8 \mu\text{m}$ placed at different depths (O_1) in front of the microscope. The red lines show the depths used in our setup (-28.8 to $28.8 \mu\text{m}$). In this range, the blur covers approximately 20 micro-lenses.

A.2 List of publications

A.2.1 Peer-reviewed conference submissions

- "Real-Time Light Field 3D Microscopy via Sparsity-Driven Learned Deconvolution". **J. Page Vizcaino**, Z. Wang, P. Symvoulidis, P. Favaro, B. Guner-Ataman, E. Boyden, T. Lasser IEEE International Conference on Computational Photography 2021
- "Learning to model and calibrate optics via a differentiable wave optics simulator". **J. Page Vizcaino**, P. Favaro. IEEE International Conference on Image Processing 2020

A.2.2 Peer-reviewed journal submissions

- "Learning to Reconstruct confocal Microscopy Stacks From Single Light Field Images". **J. Page Vizcaino**, F. Saltarin, Z. Belzaev, R. Lyck, T. Lasser P. Favaro IEEE Transactions on Computational Imaging 2020
- "Artifact-free deconvolution in light field microscopy". A. Stefanoiu, **J. Page Vizcaíno**, P. Symvoulidis, G. Westmeyer, T. Lasser. Optics Express 27 (22), 31644-31666, October 2019

A.2.3 Peer-reviewed abstract submissions and posters

- "Fast light-field 3D microscopy and domain shift adaptation through optics-aware invertible neural networks". **J. Page Vizcaíno**, Z. Wang, P. Symvoulidis, P. Favaro, E. S. Boyden, T. Lasser 21st International European Light Microscopy Initiative Meeting, 2022
- "Investigation and implementation of Lighd Field forwards models". **J. Page Vizcaíno**, A. Stefanoiu, T. Lasser. 3DTV Conference 2018