



**BERGISCHE
UNIVERSITÄT
WUPPERTAL**



JÜLICH
FORSCHUNGSZENTRUM

**Parallelization of a Data-Driven Independent
Component Analysis to Analyze Large
3D-Polarized Light Imaging Data Sets**

Dissertation zur Erlangung des Doktorgrades:

Dr. rer. nat.

im Fachbereich C: Mathematik und Naturwissenschaften
an der Bergische Universität Wuppertal

vorgelegt von

Dipl.-Ing. Giuseppe Teodoro Maria Tabbi
aus Dortmund

April 2016

Prof. Dr. Dr. Thomas Lippert

Dr. Markus Axer

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20160725-085219-8

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20160725-085219-8>]

Contents

Kurzfassung	V
Abstract	VII
1. Introduction	1
2. Principles of 3D-Polarized Light Imaging	3
2.1. Histological Preparation of Brain Tissue	5
2.2. Polarimetric Measurement	5
2.3. Jones Calculus applied on 3D-PLI	8
2.4. Fourier Analysis of the 3D-PLI Signal	9
2.5. Workflow	10
3. Principles of the Independent Component Analysis	19
3.1. Blind Source Separation	22
3.2. Information Based Maximization	23
3.3. Ambiguities of Independent Component Analysis	29
3.4. Pre-processing	29
3.5. Constrained Independent Component Analysis for 3D-PLI	35
4. High Performance Computing	41
4.1. System Configuration of JuDGE	43
4.2. Parallel Programming Paradigms	43
4.3. Parallel Programming Strategies	46
4.4. Parallel Programming Models	50
5. Development of an Adapted Constrained Independent Component Analysis on White and Gray Matter	53
5.1. Experimental Setup	55
5.2. Methods	56
5.3. Results	60
5.4. Discussion	67
6. Parameter Optimization of the Adapted Constrained Independent Component Analysis	69
6.1. Methods	70
6.2. Results	72

6.3. Discussion	74
7. A New Parallelization Concept for Adapted Constrained Independent Component Analysis	77
7.1. Concept Development, Implementation and Procedure	79
7.1.1. Development of a Parallelized ICA Concept	80
7.1.2. Implementation	85
7.1.3. Experimental Procedure	89
7.2. Results	90
7.2.1. Scalability	90
7.2.2. GPU Acceleration	92
7.2.3. Validation and Consistency	94
7.3. Summary of Achievements	98
8. Discussion	101
9. Conclusion and Outlook	105
A. Derivation of the 3D-PLI Signal	107
A.1. Jones Calculus for the LAP	108
A.2. Jones Calculus for the PM	109
A.3. Derivation of the Phase Retardation	110
B. Derivation of the Natural-gradient Version of Infomax	113
C. Nongaussianity for Source Signals in Independent Component Analysis	117
D. Basis vectors	119
List of Figures	121
List of Tables	125
List of Algorithms	127
Bibliography	129
Acknowledgements	

Kurzfassung

3D-Polarized Light Imaging (3D-PLI) ist eine Methode, welche die optischen Eigenschaften von myelinisierten Nervenfasern nutzt, um die anatomische Konnektivität in post-mortem Gehirnen zu erforschen. Für die dargelegte Studie wurden zwei unterschiedliche optische Systeme verwendet: das *Large Area Polarimeter* (LAP) und das *Polarizing Microscope* (PM). In beiden Systemen werden histologische Schnitte untersucht, indem diese mit polarisiertem Licht durchleuchtet werden. Aus der gemessenen Änderung des Polarisationszustandes des Lichtes, die durch das doppelbrechende Gewebe verursacht wird, können die räumlichen Verläufe von Nervenfasern bestimmt werden. Im Falle von menschlichen Gehirnschnitten fällt ein Datenvolumen von bis zu einem Terabyte an. Da das gemessene Signal durch Rauschen, Streuung und Filterinhomogenitäten beeinflusst wird, wurde die *Independent Component Analysis* (ICA) eingeführt, die bisher nur für das LAP eine Wiederherstellung des ursprünglichen 3D-PLI Signals ermöglicht. Die Signalstärke, die mit der doppelbrechenden Myelinscheide skaliert, variiert von der grauen zur weißen Substanz. So weisen schwache Signale, die in der grauen und an Grenzen zwischen der grauen und weißen Substanz liegen, ein geringeres Signal-Rausch-Verhältnis auf als stärkere Signale in der weißen Substanz.

Aus diesem Grund führt diese Arbeit ein neues datenorientiertes ICA-Verfahren ein, welches spezifisch für die graue und die angrenzende weiße Substanz der vorhandenen Messdaten optimiert wurde. Die Methode basiert auf der constrained ICA, worin apriori Informationen der unterliegenden Signale genutzt werden, um die Signaltrennung zu optimieren und zu beschleunigen. Die verwendeten apriori Informationen bestehen aus den Verteilungsfunktionen der grauen und weißen Substanz, was zu einer gewebespezifischen Signaltrennung führt. Das neue Verfahren weist im Vergleich zu der gegenwärtigen ICA für 3D-PLI eine schnellere Signaltrennung und erhöhte Signalverbesserung auf. Zusätzlich ist ein neues Konzept der ICA, das auf großen Datensätzen des PMs basiert, entwickelt worden. Die Nutzung von *High Performance Computing* (HPC) und des datenorientierten ICA-Verfahrens ist in der neuen parallelisierten ICA-Methode einbezogen worden.

Abstract

3D-Polarized Light Imaging (3D-PLI) is a method which uses the optical properties of myelinated fiber tracts to investigate the anatomical connectivity in post-mortem human brains. For the presented study, two different optical systems were used to map connectivity: the *Large Area Polarimeter* (LAP) and the *Polarizing Microscope* (PM). In both systems the histological sections are studied by passing linear polarized light. From the measured changes of polarized light caused by passing through a birefringent tissue, the 3D-information of the nerve fibers are extracted. These optical systems provide for human histological sections image sizes up to one Terabyte. Since the measured polarized light signal is deteriorated by noise, light scatter and filter inhomogeneities to name a few, *Independent Component Analysis* (ICA) was introduced only for the LAP to recover the original PLI signal on a whole histological section. The signal strength, which scales with the multiple layers of the birefringent myelin sheaths, varies from the gray matter to the white matter. Thus, weaker signals located in the gray and at boundaries between gray and white matter are more afflicted with noise than stronger signals in the white matter. This thesis introduces a new data-driven ICA approach specifically developed for the gray and boundaries between gray and white matter of histological sections. The method is based on constrained ICA, where a priori information of the underlying sources is used to optimize and accelerate signal decomposition. Thereby, prior information is incorporated by using the density distribution of the gray and white matter, which leads to a tissue specific signal decomposition algorithm. The new approach reveals a faster signal separation and increased signal enhancement compared to the current standard ICA approach in 3D-PLI. Additionally, a new concept for applying ICA on large high-resolution data sets of the PM is introduced. The exploitation of *High Performance Computing* (HPC) and the data-driven ICA approach are included in the new parallelized ICA method.

1. Introduction

In the 21st century a major challenge in neuroscience is the decoding of the human brain. To understand the human brain it is fundamental to gain profound insights in structural and functional brain connectivity. For the structural connectivity an essential aspect are the linkages of the morphological entities such as neurons. Since the beginning of the 20th century, it is well known that myelinated nerve fibers of neurons in the brain exhibit optical birefringence [45, 99, 100]. Birefringence is induced by the regular arrangement of proteins and lipids in the myelin sheath, which surrounds most nerve fibers [9, 81]. During the late 1990s a polarizing imaging technique was developed based on the optical birefringence of the myelin sheaths [5–7]. This new technique, which is referred to as *3D-Polarized Light Imaging* (3D-PLI), visualizes the fiber architecture in post-mortem human brains at macroscopic and microscopic scales [6, 8, 9]. 3D-PLI is able to map the courses of single fibers up to thick fiber bundles within a brain. Therefore, two different optical systems are available: the *Large Area Polarimeter* (LAP) and the *Polarizing Microscope* (PM). For both optical systems the post-mortem brain is cut into thin histological sections of about $50\ \mu\text{m} - 70\ \mu\text{m}$ yielding up to 3500 histological sections. The LAP and PM generate raw images of different data volumes ranging from approximately a few hundred Megabyte (MB) up to one Terabyte (TB) for each histological brain section, which results in TBs of data for a whole brain. Thus, image processing and analysis methods that are scalable to handle Big Data problems are of crucial importance.

One of these image processing methods is the *Independent Component Analysis* (ICA), which has been successfully used in a variety of neuroscience applications, both for analyzing brain data recorded in the time domain [31, 33, 67, 97] as well as in the spatial domain [11, 13, 14, 36]. Dammers et al. [30, 32] introduced the first application of ICA on 3D-PLI and showed that the measured birefringence signals which are afflicted with noise and artifact components can be restored to their original state. Motivated by these results a new algorithm, referred to as *constrained ICA for Polarized light imaging* (cICAP), has been developed by Breuer

et al. [21]. In cICAP, an expected birefringence signal is incorporated to improve the signal separation quality with a directly included component selection. So far, the application of the cICAP was only developed for the LAP setup and tested on whole histological brain sections yielding a general decomposition of deteriorated measured signals in both gray and white matter. Since noise and artifacts have a signal strength that can deteriorate the measured signal of the birefringence signal different in gray and white matter, a specific cICAP application adapted on these different brain tissue types is essential. Moreover, due to the challenge of processing data in the MB to TB range, which includes parallel I/O, analysis, storage, transfer and efficient usage of the provided hardware, a parallelized cICAP for LAP and PM is required that can exploit *High Performance Computing* (HPC).

This thesis focuses on adapting the cICAP for different brain tissue types and on developing a novel and fast parallelized cICAP for both the LAP and PM setup. The thesis is composed of eight chapters. The principles of 3D-PLI including the current work flow and signal interpretation, for which the cICAP is used and has been optimized are described in the second chapter. In Chapter 3 the concept of cICAP is explained. The cICAP is based on *Information based maximization* (Infomax) [15], a deep-rooted principle in information theory, and Chapter 4 provides relevant aspects in HPC. Chapter 5 describes the development and validation of the newly adapted cICAP on different tissue types. In Chapter 6 the intrinsic parameters in cICAP, i.e., threshold and confidence parameter and stopping criterion, are optimized with the help of the *Downhill Simplex Algorithm* (DSA). Chapter 7 presents a new parallelized concept for cICAP, which is developed specifically for processing ICA on Big Data in 3D-PLI. The results of the newly developed parallelized cICAP are then compared with the sequential and newly adapted cICAP. Furthermore, a hybrid implementation using *Compute Unified Device Architecture* (CUDA) and the *Message Passing Interface* (MPI) is proposed. The presented results are critically discussed in Chapter 8. Based on the discussion, Chapter 9 concludes and indicates new paths for future research.

2. Principles of 3D-Polarized Light Imaging

3D-PLI determines the orientations of nerve fibers in histological sections of post-mortem brains down to the micrometer level by measuring local changes in the polarization state of light [6, 8, 9]. Due to the birefringence of the myelin sheaths of the nerve fibers the linearly polarized light is refracted in two rays experiencing a phase shift (retardation) when passing the brain tissue. The change of the incoming linearly polarized light caused by the brain tissue can be quantified by the “Jones Calculus” [64, 65] (Sec. 2.3), enabling the determination of the 3D fiber orientation. The following sections describe the 3D-PLI workflow, starting from the preparation procedure of the brain tissue (Sec. 2.1), followed by the polarimetric measurements in the LAP and the PM (Sec. 2.2). The theoretical background for analyzing polarimetric signals is described in Section 2.4. In Section 2.5 an outline of the post-processing of the acquired polarimetric images is given.

For the sake of clarity and comprehension, the definition of variables and abbreviations for the following sections are summarized in Table 2.1.

Table 2.1.: Definition of variables and abbreviations in this chapter.

	definition
A	Ideal linear vertical polarizer
α	Out-of-plane/Inclination angle
a_0, a_1, b_1	Fourier coefficients
β	Absolute positions of the PM tiles
$\hat{\beta}$	Estimated absolute positions of the PM tiles
δ	Phase shift
d	Thickness of the brain section

Some parts of this chapter have been adapted from my diploma thesis [105]

2. Principles of 3D-Polarized Light Imaging

\vec{E}'	Outgoing electric field
\vec{E}	Incoming electric field
$f(u_\rho)$	Theoretically expected function
$g(\rho)$	Gain factor
λ	Wavelength
M_{tissue}	Assumed model of birefringent tissue as a single retarder
$M_{\lambda/4}$	Quarter-wave retarder
Δn	Local birefringence
ω	Weighting factor
P	Ideal linear horizontal polarizer
φ	In-plane/Direction angle
R	Rotation matrix
ρ	Rotation angle
$\hat{\mathbf{t}}$	Translation vector
$wrGOF$	Weighted relative goodness-of-fit
\mathbf{X}	Regressor matrix

CCD	Charge-Coupled Device
EPA	Efficient PLI Analysis
FOM	Fiber Orientation Map
LAP	Large Area Polarimeter
LED	Light Emitting Diode
PM	Polarizing Microscope
sICA	Spatial Independent Component Analysis
3D-PLI	3D-Polarized Light Imaging

2.1. Histological Preparation of Brain Tissue

In order to investigate the nerve fibers, the brain has to be cut into sections. Since 3D-PLI is restricted to post-mortem brains, the current workflow starts with this histological preparation of the brain tissue and its subsequent slicing.

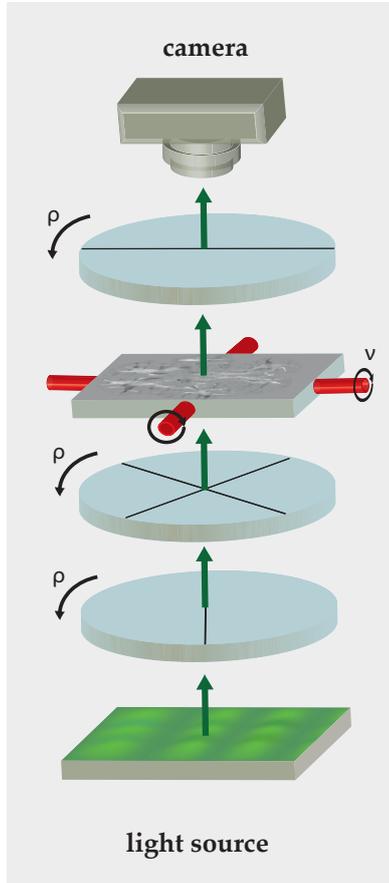
The first step after removing the brain from the skull is the fixation of the brain. The brain is immersed in a 4% buffered formaldehyde solution [8], which conserves the brain and prevents the decay of myelin. This step is followed by a cryoprotection step preventing the development of ice crystals, for which the brain is immersed in a 20% solution of glycerin for several days. After the cryoprotection the brain is frozen to -80°C and cut into sections with a thickness between $50\ \mu\text{m}$ and $70\ \mu\text{m}$ using a large scale cryostat microtome. Finally, the brain sections are mounted onto glass slides, embedded in glycerin, covered by a glass sheet and sealed with lacquer.

2.2. Polarimetric Measurement

3D-PLI enables the investigation of morphological entities such as nerve fibers at different spatial scales. In general, passing linearly polarized light through a birefringent tissue results in local changes of the polarization state of light, which is then captured by a camera. This principle is referred to as polarimetry. In 3D-PLI two fully automatized rotating polarimeter are employed: the LAP and the PM (Fig. 2.1). The LAP has an in-plane sampling resolution of $64\ \mu\text{m} \times 64\ \mu\text{m}$ and allows the mapping of small fiber bundles, whereas the PM has an in-plane sampling resolution of $1.33\ \mu\text{m} \times 1.33\ \mu\text{m}$, which enables to visualize a high level of detail such as single fibers. Due to the field of view of 24 cm in diameter in the LAP, single-shot imaging of a whole histological section is performed. In contrast, the PM enables a tessellated scan of a whole histological section covering an area of $2.7\ \text{mm} \times 2.7\ \text{mm}$ [9].

Both setups consist of a light source, two linear polarizers (Fig. 2.1 (I, IV)), a specimen stage (Fig. 2.1 (II)), a quarter-wave retarder (Fig. 2.1 (III)) and a high-resolution, light-sensitive camera (*Charge-Coupled Device* (CCD)). For the LAP setup, a green *Light Emitting Diode* (LED) grid is used that emits partial unpolarized light. The first linear polarization filter (polarizer, (IV)) converts the incoming unpolarized light into linearly polarized light. The quarter-wave retarder (III) is employed to transform the linearly polarized light from the polarizer into circularly

Large-area Polarimeter (LAP)



I. linear polarizer

II. specimen stage
(with brain tissue)

III. quarter-wave
retarder

IV. linear polarizer

Polarizing Microscope (PM)

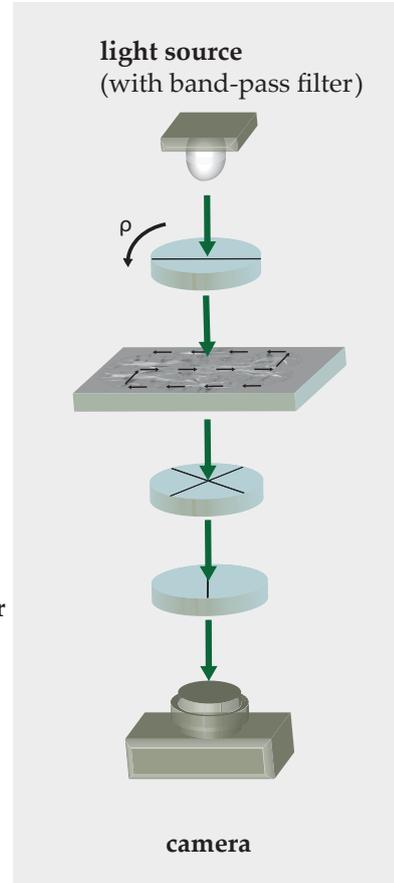


Figure 2.1.: Measurement setup for the LAP and the PM in 3D-PLI. The light path is illustrated (green arrow) going through all optical elements in the polarimeter of the LAP and PM. During the measurement the labeled optical elements are rotated simultaneously at 18 different rotation angles ρ ranging from 0° to 170° . Only the specimen stage in the LAP can be tilted by an angle ν in the range of 0° to 8° .

polarized light. With respect to the birefringence of the myelin sheaths in the histological section, the mounted brain tissue in the specimen stage (II) induces an additional phase shift to the refracted rays of the incoming light so that the outgoing light is elliptically polarized. The transmission axis of the second linear polarizer (analyzer, (I)) is set perpendicular to the transmission axis of the first one. At the end of the light path a CCD camera captures the transmitted light (Fig. 2.1).

There are a few differences between the two systems LAP and PM. In the LAP, the light is passed through the optical system from below and the specimen stage can be tilted from 0° to 8° . The tilting of the specimen allows a more accurate determination of the out-of-plane angles of the nerve fibers [9, 109]. Furthermore, the polarizer, the quarter-wave retarder and the analyzer are rotated simultaneously. In the PM, the light comes from above and a white LED with a band-pass filter is used. After each acquisition of the PM tile the specimen stage is moved in meandering pattern with an overlap of 30% between the tiles.

A standard 3D-PLI measurement produces an image for each rotation angle in the range of $\rho = 0^\circ, 10^\circ, \dots, 170^\circ$, so that a series of 18 images is acquired. The signal profiles of two pixels located on a birefringent myelinated and a non-birefringent nerve fiber, respectively, are exemplary illustrated in Figure 2.2. A non-birefringent nerve fiber is characterized by a flat curve, whereas birefringent myelinated nerve fibers show sinusoidal profiles.

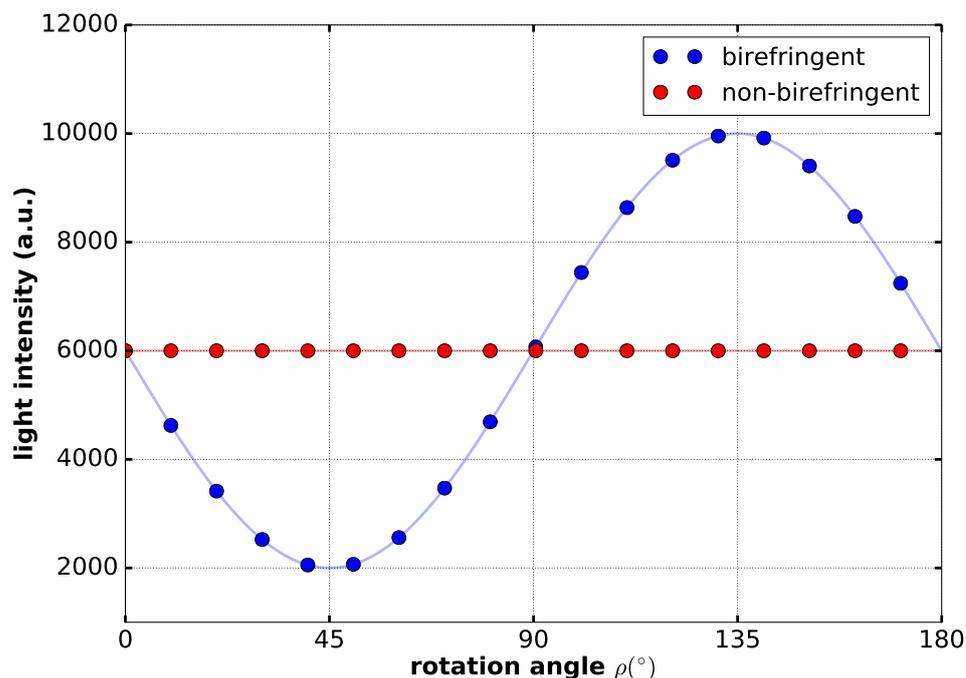


Figure 2.2.: Example of light intensities of two pixels at different rotation angles ρ . Each data point (circle) corresponds to one measured intensity value of two pixels located on a birefringent myelinated nerve fiber (blue) and on a non-birefringent nerve fiber (red), respectively.

2.3. Jones Calculus applied on 3D-PLI

In optics, a mathematical description of polarized light in an optical system was introduced by R. C. Jones in 1941 [65] and 1942 [64]. Mathematically, the effect of a linear polarization filter on a polarized light beam can always be represented as a linear transformation of the components of the electric field vector. This algorithm is referred to as ‘‘Jones Calculus’’. The ‘‘Jones Calculus’’ can only describe completely polarized light. With respect to the two transverse components of the electric field vector of the light, linear polarizing filters and retardation plates can be expressed as two-by-two matrices.

In the analysis of 3D-PLI, it is assumed that each volume element of the brain tissue, which contributes to one image pixel, acts as a single retarder due to its birefringent properties. Hence, the 3D-PLI setup for the LAP at every pixel location px and py may be mathematically expressed as (the mathematical expression for the PM is described in Appendix A.2)

$$\vec{E}'_{px,py} = A_{px,py} \cdot M_{\text{tissue},px,py} \cdot M_{\lambda/4,px,py} \cdot P_{px,py} \cdot \vec{E}_{px,py}, \quad (2.1)$$

where $\vec{E}'_{px,py}$ is the outgoing electric field vector, $\vec{E}_{px,py}$ is the incoming electric field vector of the LAP setup, $P_{px,py}$ and $A_{px,py}$ correspond to two orthogonal linear polarizers, $M_{\lambda/4,px,py}$ describes a quarter-wave retarder and $M_{\text{tissue},px,py}$ denotes the assumed model of birefringent tissue as a single retarder (see Appendix A). $M_{\text{tissue},px,py}$ is expressed as

$$M_{\text{tissue},px,py} = R_{px,py} [-(\rho - \varphi)] \cdot \begin{pmatrix} e^{\frac{i}{2}\delta} & 0 \\ 0 & e^{-\frac{i}{2}\delta} \end{pmatrix} \cdot R_{px,py} (\rho - \varphi), \quad (2.2)$$

where i is the imaginary unit, $R_{px,py}$ is a rotation matrix, δ describes the phase retardation and φ denotes the in-plane fiber direction. It is noted that the embodied polarization filters are rotated simultaneously in counter-clockwise direction with an angle ρ . For simplicity, the mathematical equivalence is considered in which only the specimen stage is rotated in counter-clockwise direction with an angle $-\rho$ while the polarization filters are fixed. Furthermore, the rotation matrix $R_{px,py}$ is given by

$$R_{px,py}(\rho - \varphi) = \begin{pmatrix} \cos(\rho - \varphi) & \sin(\rho - \varphi) \\ -\sin(\rho - \varphi) & \cos(\rho - \varphi) \end{pmatrix}. \quad (2.3)$$

The retardation δ is the phase shift induced to the electric vector of the incident light through the birefringent brain tissue. This phase shift between the components of the electric vector depends on the thickness d of the brain section, the wavelength λ , the local birefringence Δn and the out-of-plane angle α of the fiber (see Appendix A.3). An approximation of δ [76] is given by

$$\delta \approx 2\pi \frac{d\Delta n}{\lambda} \cos^2 \alpha. \quad (2.4)$$

Thus, the Jones calculus permits to describe the transmitted light intensity through the following expression (see Appendix A.1):

$$I_{px,py}(\rho) = \frac{I_0}{2} [1 + \sin(2(\rho - \varphi)) \sin(\delta)], \quad (2.5)$$

where I_0 is referred to as transmittance and $\sin(\delta)$ is referred to as retardation. The in-plane fiber direction φ referred to as direction can also be derived from Equation 2.5, whereas the out-of-plane angle α , referred to as inclination, can be derived from the retardation and from the Equations 2.4 and 2.5.

2.4. Fourier Analysis of the 3D-PLI Signal

The light transmittance through the polarimeter, which is described in Equation 2.5, can be fitted by means of discrete harmonic Fourier analysis [8]:

$$I_{px,py}(\rho) = \frac{I_0}{2} + \frac{I_0}{2} |\sin(\delta)| \cos(2\varphi) \sin(2\rho) \quad (2.6)$$

$$- \frac{I_0}{2} |\sin(\delta)| \sin(2\varphi) \cos(2\rho) \quad (2.7)$$

$$\equiv a_0 + a_1 \sin(2\rho) + b_1 \cos(2\rho), \quad (2.8)$$

$$\text{with the Fourier coefficients } a_0 = \frac{I_0}{2}, \quad (2.9)$$

$$a_1 = \frac{I_0}{2} |\sin(\delta)| \cos(2\varphi) \text{ and} \quad (2.10)$$

$$b_1 = -\frac{I_0}{2} |\sin(\delta)| \sin(2\varphi), \quad (2.11)$$

where $|\sin(\delta)|$ is the absolute value of the retardation. The convention of the Equation 2.8 and its Fourier coefficients conform to the used nomenclature introduced in Axer et al. [8]. The Equations 2.9, 2.10 and 2.11 can be discretized accordingly to the standard 3D-PLI measurement at rotation angles $(\rho_1, \rho_2, \dots, \rho_N)$ with $N = 18$. By using a discrete approximation the Fourier coefficients can be expressed as

$$a_0 = \frac{1}{N} \sum_{i=1}^N I_{px,py}(\rho_i), \quad (2.12)$$

$$a_1 = \frac{2}{N} \sum_{i=1}^N \sin(2\rho_i) I_{px,py}(\rho_i) \text{ and} \quad (2.13)$$

$$b_1 = \frac{2}{N} \sum_{i=1}^N \cos(2\rho_i) I_{px,py}(\rho_i). \quad (2.14)$$

2.5. Workflow

In order to reconstruct and analyze the orientation of the nerve fibers with 3D-PLI, different image processing steps are required after the image acquisition. The post-processing consists of the following steps:

Calibration Due to electronic noise and thermal effects in the CCD camera, intensity variations can occur at each pixel. Additionally, inhomogeneities in the polarizers can lead to non-linear signal variations across the pixels of the image (Fig. 2.3 (a)). Thus, a calibration is needed to compensate for the latter effect (Fig. 2.3 (b)). For calibration, a set of 60 flat images (i.e., measurements without a brain section in the polarimeter) are acquired for each angle ρ . From the 60 flat images an average intensity value $\bar{I}(\rho)$ is calculated at each pixel and rotation angle ρ . I_{ref} denoting the peak intensity is calculated from the joint density of all 1080 (18×60) flat images, so that the calibration can be described as

$$I(\rho) = g(\rho)I'(\rho), \quad (2.15)$$

where $I'(\rho)$ is the intensity before calibration and $I(\rho)$ is the intensity after calibration, accordingly. The gain factor $g(\rho)$ can be calculated as $g(\rho) = \frac{I_{\text{ref}}}{\bar{I}(\rho)}$ and describes the factor that compensates the fluctuation in the image [32].

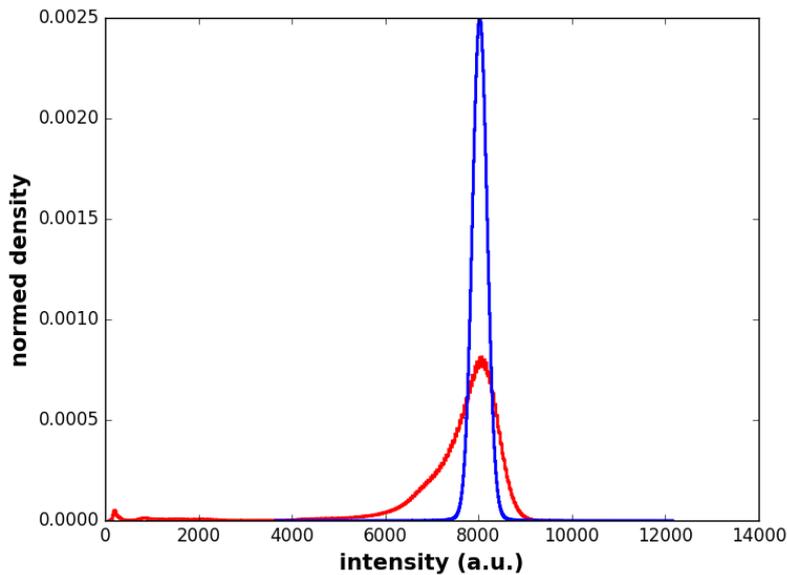
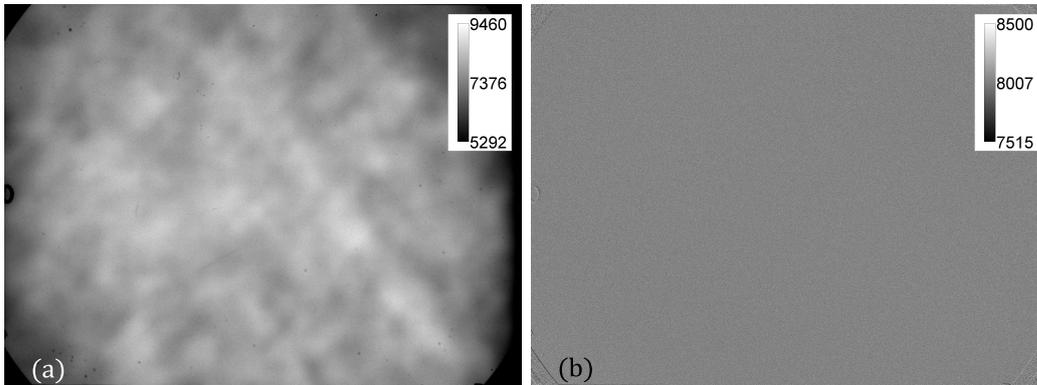


Figure 2.3.: Contrast enhanced flat images before (a) and after (b) calibration of the LAP. Here only the flat images of ρ_0 are shown. (c) shows the intensity distributions before (red) and after (blue) the image calibration. It is evident that the image inhomogeneity is reduced after calibration.

Segmentation In order to ensure that only brain tissue is processed, the background is removed from the acquired images. Different approaches for the LAP and PM are currently employed. However, for this thesis a manual segmentation approach for the LAP was used since an automated algorithm was lacking. For the PM images, on the contrary, an automatic in-house segmentation based on region growing algorithm was available.

Spatial Independent Component Analysis An important part of 3D-PLI analysis is the processing of images with the *spatial Independent Component Analysis* (sICA). The sICA decomposes the acquired images into spatial independent components. Noise and artifacts, which are superimposed on the optical signal, are effectively rejected using sICA. By taking the components of interest, the birefringent 3D-PLI signal, referred to as original signal, is reconstructed. Components of interest are identified by the sinusoidal nature of the 3D-PLI signal. sICA in 3D-PLI is an integral part of this thesis, which will be addressed in detail later on (see Sec. 3.5).

Estimation of Signal Enhancement After applying sICA and reconstructing the images freed from artifacts, i.e., dust particles, and noise components, the signal enhancement is estimated. This estimation evaluates how well the sICA filtered data, fits the theoretical function predicted by the ‘‘Jones Calculus’’. Therefore, a *Goodness-Of-Fit* (GOF) test is performed using the Pearson’s chi-squared test before and after the sICA application [32]. The simplified chi-squared statistic χ^2 includes the variance σ_ρ^2 obtained from 60 flat images in the calibration process:

$$\chi^2 = \frac{1}{\nu} \sum_{i=1}^N \frac{(I_{px,py}(\rho_i) - f_{px,py}(u_{\rho_i}))^2}{\sigma_{\rho_i}^2}, \quad (2.16)$$

with $I_{px,py}(\rho_i)$ being the measured intensity at angle ρ and $f_{px,py}(u_\rho)$ denoting the theoretically expected function, which is fitted based on Equation 2.8. ν describes the degrees of freedom and normalizes the sum over N angles. It has a size of $(N - \#\text{parameters} - 1)$. The parameters which reduce the degrees of freedom are a_0 , a_1 and b_1 . A good fit is achieved when the squared difference between the measurement and the expected function is equal to the variance of the measurement. This results in a chi-square value equal to one. In order to obtain a measure that indicates a fit improvement after sICA, the ratio of χ_{raw}^2 before and χ_{ICA}^2 after sICA is calculated:

$$rGOF = \frac{\chi_{\text{raw}}^2}{\chi_{\text{ICA}}^2}, \quad (2.17)$$

where $rGOF$ describes the relative goodness-of-fit value. A fit improvement is indicated by $rGOF > 1$. However, the $rGOF$ cannot take into account whether any component of interest after sICA is missing. Therefore, to consider missing gray or white matter components, a weighting factor ω is included, which penalizes

the goodness of fit value, if any component of interest is missing [30]. The *weighted rGOF* (*wrGOF*) can be described as follows:

$$wrGOF = \frac{1}{\omega} \frac{\chi_{\text{raw}}^2}{\chi_{\text{ICA}}^2}, \quad (2.18)$$

with

$$\omega = \frac{1}{\nu} \sum_{i=1}^N \frac{(f_{\text{raw}}(u_{\rho_i}) - f_{\text{ICA}}(u_{\rho_i}))^2}{\sigma_{\rho_i}^2},$$

and $\omega \geq 1$.

Using the included weighting factor ω , a penalty term is computed whenever the squared difference between the two expectation functions $f_{\text{raw}}(u_{\rho})$ and $f_{\text{ICA}}(u_{\rho})$ is large. When a component of interest is missing, the signal strength of the sICA filtered data is reduced at the corresponding pixel locations. This yields a difference in the amplitude of the expectation function $f_{\text{ICA}}(u_{\rho})$. It is assumed that the signal power (across the rotation angles) of the signal of interest is larger than the noise level [30].

Efficient 3D-PLI Analysis The *Efficient PLI Analysis* (EPA) calculates the parameters called transmittance, retardation and direction from calibrated and sICA corrected images according to Axer et al. [8]. These parameters are generated by the appropriate combinations of the fit parameters a_0 , a_1 and b_1 (Eq. 2.8). The transmittance maps, which are sensitive to absorption and scattering processes, are calculated as

$$I_0 = 2a_0. \quad (2.19)$$

The retardation maps include information about the elevation of fibers out of the section plane. The retardation is defined as

$$|\sin(\delta)| = \frac{\sqrt{a_1^2 + b_1^2}}{a_0}. \quad (2.20)$$

It can also be interpreted as the relative amplitude ($|\sin(\delta)| = \frac{\Delta I}{I_0}$) of the sinusoidal intensity function in Figure 2.4. Furthermore, the direction maps are obtained by

$$\varphi = \frac{1}{2} \arctan2(-b_1, a_1) + \frac{\pi}{2} = \begin{cases} \frac{1}{2} \arctan(-b_1/a_1) + \frac{\pi}{2}, & a_1 > 0 \\ \frac{1}{2} \arctan(-b_1/a_1) + \frac{3\pi}{2}, & b_1 \geq 0, a_1 < 0 \\ \frac{1}{2} \arctan(-b_1/a_1) - \frac{\pi}{2}, & b_1 < 0, a_1 < 0 \\ \pi, & b_1 > 0, a_1 = 0 \\ 0, & b_1 < 0, a_1 = 0 \end{cases} \quad (2.21)$$

The direction can also be obtained from the phase shift relative to the minimum of the fitted intensity profile (Fig. 2.4).

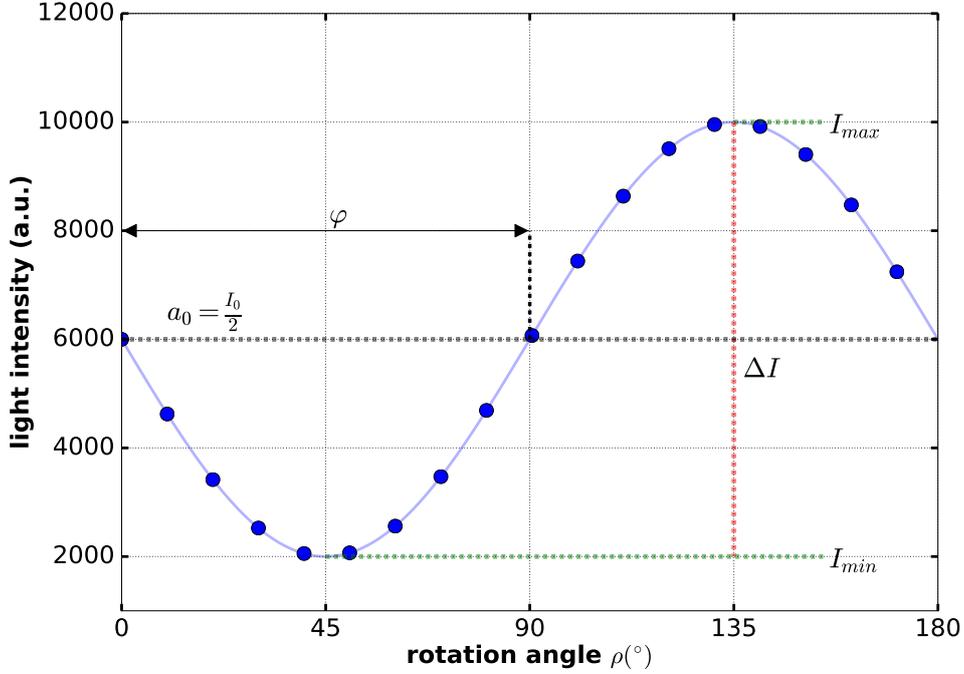


Figure 2.4.: Measured light intensities of one pixel at different rotation angles ρ . Each data point (blue circle) on the sinusoidal profile corresponds to one measured intensity value. The continuous line illustrates the derived transmitted light intensity from the "Jones Calculus" for one pixel.

Stitching Stitching is a processing step in 3D-PLI, where several PM tiles are combined together into one continuous PM image without the measured overlap (Fig. 2.5). The first step for stitching is computing EPA on each PM tile. By using

Scale-Invariant Feature Transform (SIFT, [78, 79]) on the resulting transmittance or direction tile maps, keypoint descriptors are extracted. The correspondence of each keypoint descriptor in one transmittance or direction map to its neighboring tile is found by matching the keypoint descriptors via brute-force approach. The displacements of all PM tiles can be written as

$$\hat{\mathbf{t}} = \mathbf{X} \cdot \beta, \quad (2.22)$$

where $\hat{\mathbf{t}}$ corresponds to the translation vector of the matching keypoint descriptors, \mathbf{X} denotes a regressor matrix of the affected PM tiles and β describes the unknown absolute positions of the PM tiles. A unique global minimum at β can be estimated with

$$\hat{\beta} = \arg \min_{\beta} \left\| \hat{\mathbf{t}} - \mathbf{X} \cdot \beta \right\| \quad (2.23)$$

$$= (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \hat{\mathbf{t}}, \quad (2.24)$$

where $\|\cdot\|$ describes the L^2 -norm and $\hat{\beta}$ denotes the estimated absolute positions of the PM tiles.

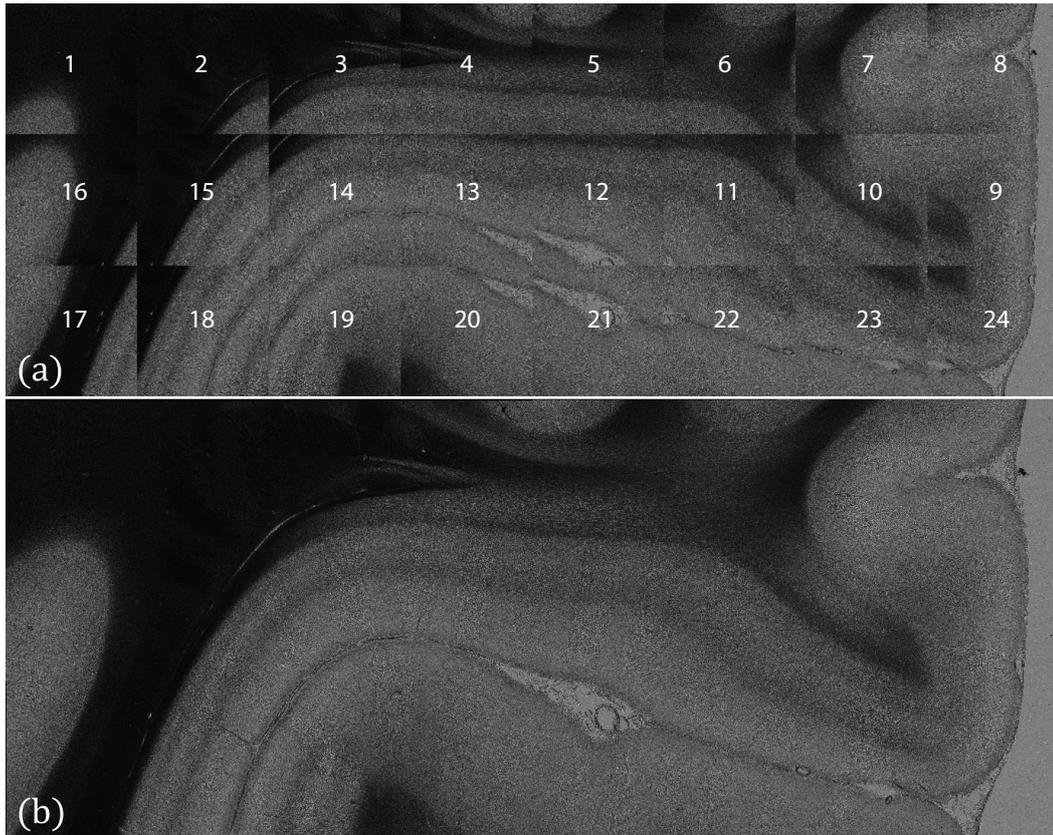


Figure 2.5.: The sticking procedure combines all tiles together with the best correspondences. In (a) the transmittance maps of the acquired PM tiles (numbered in meandering pattern) with an overlap of 30% were calculated. In (b) a combined transmittance map after stitching without the overlap is shown.

Fiber Orientation Map By means of the direction φ and the inclination α , which is derived from the retardation $|\sin(\delta)|$, a fiber orientation is calculated per pixel. Thus, a *Fiber Orientation Map* (FOM) is created (Fig. 2.6). In a FOM each image pixel is characterized by a unit vector given by

$$\begin{aligned}x &= \cos \alpha \cos \varphi \\y &= \cos \alpha \sin \varphi \\z &= \sin \alpha.\end{aligned}\tag{2.25}$$

Figure 2.6 shows a FOM in an HSV (H: Hue, S: Saturation, V: Value) color scheme, where the direction φ and the inclination α is transformed as

$$\begin{aligned}H &= 2\varphi, \\S &= 1 - \frac{\alpha}{90^\circ} \text{ and} \\V &= 1 - \frac{\alpha}{90^\circ}.\end{aligned}\tag{2.26}$$

The stacking of FOMs results in a volume of fiber orientations, which represents the basis for high-resolution structural fiber mapping using 3D-PLI data sets [8].

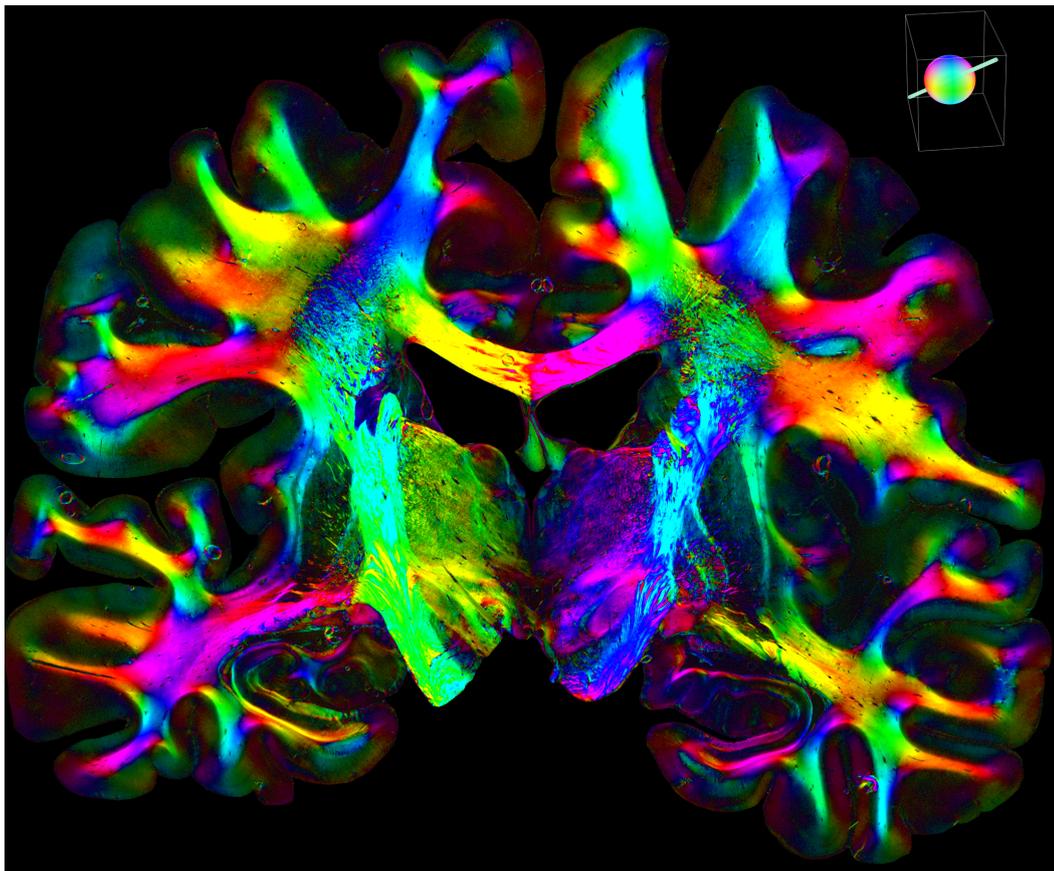


Figure 2.6.: A color coded FOM in HSV color space of a transversal brain section acquired in the LAP.

3. Principles of the Independent Component Analysis

In neural network research, a challenge is to find a representation of data that depends on multiple variables, referred to as multivariate data. The representation is often simplified as a linear transformation of the original data that finds independent sources of activity. *Independent Component Analysis* (ICA) is one of many linear transformation methods, such as principal component analysis [51], factor analysis [25] and projection pursuit [43]. Generally, the linear transformation of the original data and the independent sources are latent since they are not directly observable. The goal of ICA is to find a linear transformation of independent sources, where the independent sources, referred to as independent components, have non-gaussian distributions [55]. The statistically independent components can represent the signal of interest, noise and artifact components. By rejecting noise and artifact components, only the original signal is retained.

ICA was first introduced by Jutten et al. [68] and incorporates two different aspects [15]: first, the information-theoretic unsupervised learning rule for neural networks [4, 12, 77], and second, the use of higher-order statistics for separating out mixtures of independent sources [27]. Moreover, “standard” ICA can be subdivided into either spatial (sICA) or temporal (tICA) ICA [103]. In the case of 3D-PLI, sICA is used. It is assumed that light scattering in the investigated object, different light sources, reflections and dust may affect the original 3D-PLI signal in each pixel of the acquired image (Fig. 3.1). Thus, the measured light intensity is a linear mixture from different contributing sources. Dammers et al. [30, 32] and Breuer et al. [21] have previously shown that sICA applied to polarized light images can be used to effectively remove the contribution from impairing light sources and to restore the original sinusoidal intensity profile.

A detailed description of the general model of blind source separation is given in Section 3.1. Section 3.2 provides a common theoretical framework for solving the problem of blind source separation through the use of information-theoretic func-

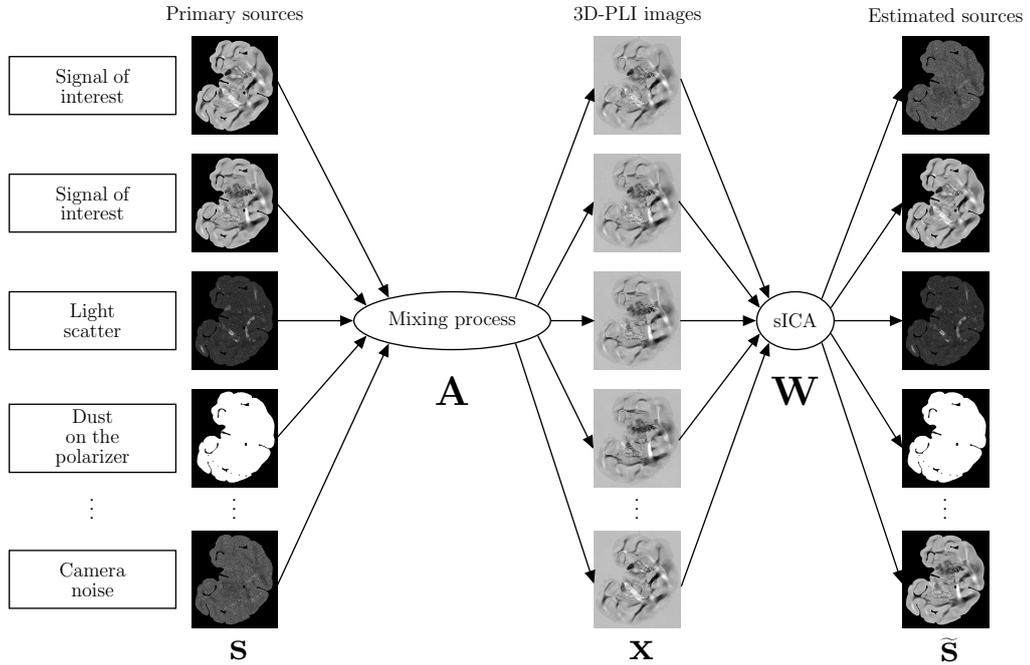


Figure 3.1.: Schematic illustration of the mixing process in 3D-PLI and the decomposition in sICA (adapted from [21]), where \mathbf{s} denotes the independent sources, \mathbf{A} describes the unknown mixing process, \mathbf{x} refers to the 18 acquired 3D-PLI images, \mathbf{W} denotes the estimated mixing process and $\tilde{\mathbf{s}}$ describes the estimated independent sources.

tions applied to neural networks. The ambiguities of sICA (and ICA in general) are described in Section 3.3. Section 3.4 provides an insight into the pre-processing steps of sICA. The chapter is concluded with a new concept of sICA applied specifically to 3D-PLI (Section 3.4).

For the sake of clarity and comprehension, the definition of variables and abbreviations for the following sections are summarized in Table 3.1.

Table 3.1.: Definition of variables and abbreviations in this chapter.

	definition	reference
\mathbf{A}	Mixing matrix	
$\hat{\mathbf{A}}$	Orthogonal mixing matrix	
\mathcal{A}	Event of an experiment	

$\hat{a}(\rho)$	Basis vector of $\hat{\mathbf{A}}$	
$\hat{a}^*(\rho)$	Basis vector with prior information	
\mathcal{B}	Event of an experiment	
\mathbf{D}_x	Whitening matrix	
\mathbf{D}_w	Random whitening matrix	
ϵ	threshold value	
η	Confidence value	
$f(u_\rho)$	Theoretically expected function	see Chapter 2
g	Cost-function	
\mathbf{I}	Identity matrix	
k	Sequence of signal	
m	Number of receivers	
n	Number of primary or source signals	
\mathbf{P}	Permutation matrix	
ς	Constant	
\mathbf{s}	Vector of source signals	
$\tilde{\mathbf{s}}$	Vector of estimated source signals passed through the cost-function g	
$\tilde{\mathbf{s}}'$	Vector of centered source signals	
$\tilde{\mathbf{S}}$	Resulting vector after passing $\tilde{\mathbf{s}}$ through the cost-function g	
t	tolerance value	
τ	Learning rate	
\mathbf{v}	Vector of random noise	
\mathbf{W}	Unmixing matrix	
$\widehat{\mathbf{W}}$	Orthogonal unmixing matrix	
\mathbf{W}_0	Bias weight	
\mathbf{x}	Vector of recorded sensor signals	
$\tilde{\mathbf{x}}'$	Vector of centered sensor signals	
$\hat{\mathbf{x}}$	Vector of centered sensor signals after the whitening process	

BSS	Blind Source Separation	
cICAP	Constrained Independent Component Analysis for 3D-PLI	
ICA	Independent Component Analysis	
Infomax	Information based maximization	
MIMO	Multiple-Input/Multiple-Output	
sICA	Spatial Independent Component Analysis	
tICA	Temporal Independent Component Analysis	
3D-PLI	3D-Polarized Light Imaging	see Chapter 2

3.1. Blind Source Separation

Blind Source Separation (BSS) addresses the problem of decomposing a set of mixed signals into its underlying sources, without the aid of a priori knowledge (or with exploiting some a priori information about the mixing system). In case of speech decomposition, a demonstration tackling this problem is, e.g., the “cocktail party problem” [24]. The “cocktail party problem” describes a source separation problem, where a listener tries to focus on one discussion in a room of simultaneously talking people. The human brain is able to solve this problem effortlessly. The general problem can be formulated as follows:

Let $\mathbf{x}(k) = [x_1(k), \dots, x_m(k)]^T$ be the vector of sensor signals received from a *Multiple-Input/Multiple-Output* (MIMO) nonlinear dynamical system, where m refers to the number of receivers and k denotes the sequence of the signal. Now the aim is to estimate n primary or original source signals $\mathbf{s}(k) = [s_1(k), \dots, s_n(k)]^T$ with $m \geq n$ by finding the inverse system without knowing the underlying mixing characteristics. The inverse system can be modeled as a neural network which was firstly introduced in [47, 68]. Jutten et al. [68] firstly described a recursive linear adaptive filter mimicking the architecture of a neural network (Fig. 3.2).

Since the mixing and the decomposition process are unknown, the models are assumed linearly composed (Fig. 3.2). This simplification (usually with $m \geq n$) for the mixing process can be described as

$$\begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_m(k) \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_n(k) \end{pmatrix} + \begin{pmatrix} v_1(k) \\ v_2(k) \\ \vdots \\ v_m(k) \end{pmatrix} \quad (3.1)$$

where \mathbf{A} describes an unknown $m \times n$ mixing matrix and $\mathbf{v}(k) = [v_1(k), \dots, v_m(k)]^T$ is an additive random noise vector influencing the records of sensor signals $\mathbf{x}(k)$. Since the distribution of $\mathbf{v}(k)$ is unknown it is generally impossible to estimate the exact source signals $\mathbf{s}(k)$ [27]. Hence, the mixing process is expressed as

$$\mathbf{x}(k) = \mathbf{A} \cdot \mathbf{s}(k). \quad (3.2)$$

The corresponding decomposition process reads as follows

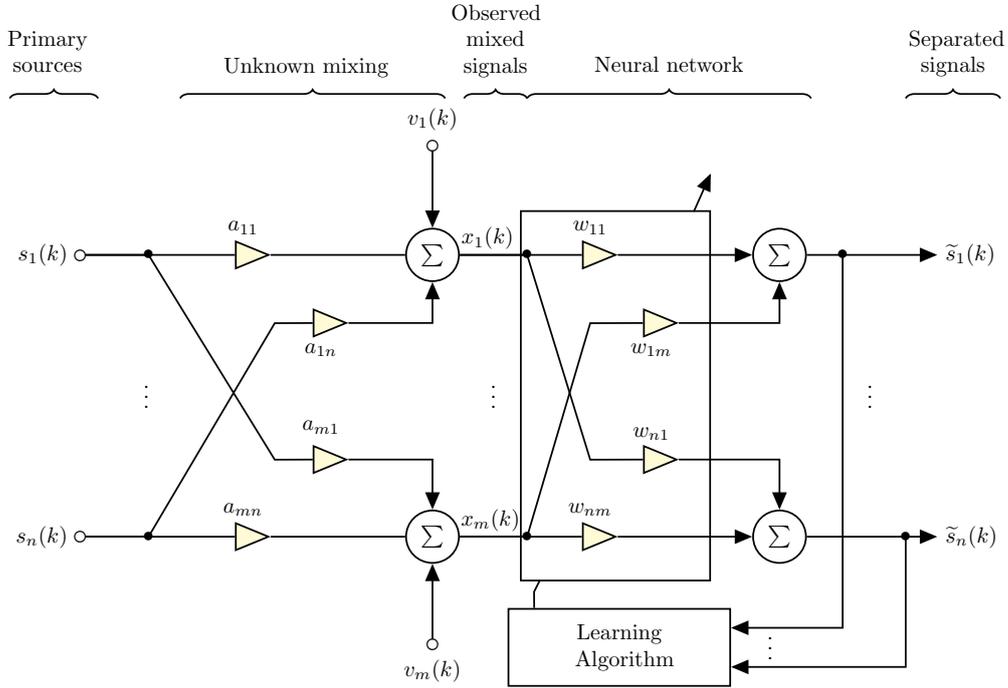


Figure 3.2.: Block diagram showing the linearized BSS problem. The number of noise and sensors m can be greater, equal or even less than the number of sources n (adapted from [26]).

$$\tilde{\mathbf{s}}(k) = \mathbf{W} \cdot \mathbf{x}(k), \quad (3.3)$$

with $\tilde{\mathbf{s}}(k) = [\tilde{s}_1(k), \dots, \tilde{s}_n(k)]^T$ being a vector of the estimated source signals. \mathbf{W} denotes an unknown $n \times m$ unmixing matrix.

3.2. Information Based Maximization

The basic problem addressed by the *Information based maximization* (Infomax) [15] is to find an unmixing matrix \mathbf{W} that decomposes a set of mixed signals into statistically independent sources $\tilde{\mathbf{s}}(k)$ and at the same time maximizes the information flow of the corresponding neural network. For fully understanding how to obtain statistical independence and what “information” entails, an explanation according to information theory is given in the next paragraph, followed by the presentation of the Infomax principle.

Some Theoretical Background in Information Theory Measuring the degree of statistical independence is of importance in Infomax and is one of the elementary problems in information theory. The concept of statistical independence can be formulated as follows. In an experiment two events \mathcal{A} and \mathcal{B} are described as independent only if

$$P(\mathcal{AB}) = P(\mathcal{A})P(\mathcal{B}), \quad (3.4)$$

with P being the probability that the event occurs. The conditional probability $P(\mathcal{B}|\mathcal{A})$ is described as

$$P(\mathcal{B}|\mathcal{A}) = \frac{P(\mathcal{AB})}{P(\mathcal{A})}. \quad (3.5)$$

If the probability $P(\mathcal{A}) \neq 0$, independence will be implied by $P(\mathcal{B}|\mathcal{A}) = P(\mathcal{B})$. This implication holds for Equation 3.5 if \mathcal{A} and \mathcal{B} are interchanged. Analogously, independence of two random variables x and y is given if

$$P(x \in A, y \in B) = P(x \in A)P(y \in B), \quad (3.6)$$

for $A, B \subset \mathbb{R}$. In case of existing densities, the Equation 3.6 will be equivalent to

$$f_{xy}(x, y) = f_x(x) f_y(y), \quad (3.7)$$

where $f_{xy}(x, y)$ is the joint probability density function (joint pdf) of the marginal probability density functions (marginal pdf) $f_x(x)$ and $f_y(y)$.

Thus, the conditional pdf $f_y(y|x)$ becomes for independent marginal densities [91]:

$$f_y(y|x) = \frac{f_{xy}(x, y)}{f_x(x)} \quad (3.8)$$

$$= f_y(y). \quad (3.9)$$

In this case, Equation 3.7 implies that if the marginal pdfs $f_x(x)$ and $f_y(y)$ are known, the joint pdf $f_{xy}(x, y)$ will be also known. Furthermore, Equation 3.8 implies that a value of one random variable gives no information about a value of the other random variable.

In order to measure the information content in a given variable, the entropy is used

(Eq. 3.10). Assumed are the possible outcomes \mathcal{A}_i of the partition $\mathcal{A} = [\mathcal{A}_1, \dots, \mathcal{A}_n]$ prior to the performance of an experiment, e.g., a fair-die experiment. The less likely an event \mathcal{A}_i occurs in an experiment, the more information it provides when it occurs [91]. This information can be measured by the entropy

$$H(\mathcal{A}) = - \sum_i p_i \log p_i, \quad (3.10)$$

where the probabilities $p_i = P(\mathcal{A} = \mathcal{A}_i)$ and i is an index ranging over all partitions of \mathcal{A} . To define the entropy of a discrete random variable x_D , a suitable partition has to be defined. Let $p_i = P(x_D = x_i)$, where the events $\{x_D = x_i\}$ are mutually exclusive. Unified they define a certain event. This partition is denoted by \mathcal{A}_x . Hence, the entropy $H(x_D)$ is equal to the entropy $H(\mathcal{A}_x)$: $H(x_D) = H(\mathcal{A}_x)$ [91]. Furthermore, the entropy as a measure of information (Eq. 3.10) has a maximum value of $\log p_i$ if all p_i values are uniformly distributed. Thus, a uniform probability distribution corresponds to a maximal entropy distribution [103]. It is important to note that a uniform probability distribution implies bounded signals. For the case of signals with the constrain of unit variance, gaussian distributed signals have maximum entropy [56].

To measure only changes in information, the differential entropy is regarded. The differential entropy for a continuous random variable x is expressed as [44]

$$H(x) = - \int_{-\infty}^{\infty} f_x(x) \log f_x(x) dx. \quad (3.11)$$

When changes in information about x are measured in which y occurs, the conditional differential entropy will be considered. The conditional differential entropy is described by

$$H(x|y) = - \int_{-\infty}^{\infty} f_y(y) \int_{-\infty}^{\infty} f_x(x|y) \log f_x(x|y) dx dy. \quad (3.12)$$

Thus, relating the information between x and y , the average mutual information is defined as [44]

$$I(x, y) = H(x) + H(y) - H(x, y) \quad (3.13)$$

$$= H(x) - H(x|y) = H(y) - H(y|x), \quad (3.14)$$

with

$$H(x, y) = H(y) + H(x|y) \quad (3.15)$$

$$= H(x) + H(y|x). \quad (3.16)$$

The joint entropy can also be denoted by

$$H(x, y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{xy}(x, y) \log f_{xy}(x, y) dx dy. \quad (3.17)$$

In terms of distributions Equation 3.14 is formed to

$$I(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{xy}(x, y) \log \left(\frac{f_{xy}(x, y)}{f_x(x) f_y(y)} \right) dx dy, \quad (3.18)$$

which is also defined as the Kullback-Leibler distance between the joint pdf $f_{xy}(x, y)$ and the product of $f_x(x) f_y(y)$ [73]. The Kullback-Leibler distance is zero if $f_{xy}(x, y) = f_x(x) f_y(y)$. From this definition one can infer that the mutual information of random variables is zero if the random variables are statistically independent. This criterion is fundamental for Infomax.

The Principle of Infomax Infomax is a method of ICA rooted in information-theoretic learning, which aim is to calculate statistical independent source signals by maximizing the entropy distribution. Therefore, it is important to find an unmixing matrix \mathbf{W} that maximizes the joint entropy $H(\tilde{\mathbf{s}}(k))$ (Eq. 3.17) yielding statistical independence between $\tilde{s}_1(k), \dots$ and $\tilde{s}_n(k)$. The idea is to map $\tilde{\mathbf{s}}(k)$ to an alternate set of signals (Fig. 3.3)

$$\tilde{\mathbf{S}}(k) = g(\tilde{\mathbf{s}}(k)) \quad (3.19)$$

that are uniformly distributed and, thus, have a maximal entropy distribution. In order to achieve the information transfer between $\tilde{\mathbf{s}}(k)$ and $\tilde{\mathbf{S}}(k)$, an invertible monotonic increasing nonlinear function g (referred to as cost-function, Eq. 3.25) is used. For simplicity, the cost-function is assumed to be bounded between 0 and 1. The minimization of the mutual information (Kullback-Leibler distance) between the derivative of the cost-function

$$g'(\tilde{\mathbf{s}}(k)) = \left| \frac{d\tilde{\mathbf{S}}(k)}{d\tilde{\mathbf{s}}(k)} \right| \quad (3.20)$$

and the distribution $f_{\tilde{\mathbf{s}}}$ will be equivalent to the maximization of the entropy of $H(\tilde{\mathbf{S}}(k))$ ([86], see Appendix B). The Kullback-Leibler distance is minimal if $f_{\tilde{s}_i} = g'(\tilde{s}_i)$. When the entropy $H(\tilde{\mathbf{S}}(k))$ is maximal it becomes evident that the cumulative density function (cdf) of bounded $\tilde{\mathbf{s}}(k)$ matches the cost-function g . Therefore, the components of $\tilde{\mathbf{S}}(k)$ become statistically independent by finding the matching cost-function g . The retrieved signals

$$\tilde{\mathbf{s}}(k) = g^{-1}(\tilde{\mathbf{S}}(k)) \quad (3.21)$$

are also independent because g is an invertible function. In general it can be said that any invertible function of maximum entropy signals yields signals that are also mutually independent [103]. This approach is known as ‘‘cdf-matching’’ and can be seen as one of the principles of Infomax.

The ‘‘cdf-matching’’ is performed by iteratively estimating the optimal unmixing matrix \mathbf{W} using the natural-gradient [1, 3, 22] version of Infomax (see Appendix B) which is described by

$$\tilde{\mathbf{s}}(k) = \mathbf{W} \cdot \mathbf{x}(k) + \mathbf{W}_0, \quad (3.22)$$

$$\Delta \mathbf{W} = \tau \left[\mathbf{I} + (1 - 2g(\tilde{\mathbf{s}}(k))) \tilde{\mathbf{s}}(k)^T \right] \mathbf{W} \text{ and} \quad (3.23)$$

$$\Delta \mathbf{W}_0 = \mathbf{I}(1 - 2g(\tilde{\mathbf{s}}(k))), \quad (3.24)$$

where \mathbf{I} describes the identity matrix, τ is the learning rate, \mathbf{W}_0 refers to the bias weight which is initialized at the beginning of the iteration as $\mathbf{W}_0 = \mathbf{I}$ and $\Delta \mathbf{W}$ denotes the estimated differential unmixing matrix and is also initialized as $\mathbf{W} = \mathbf{I}$. The learning rate τ is consistently decreasing for every iteration step which

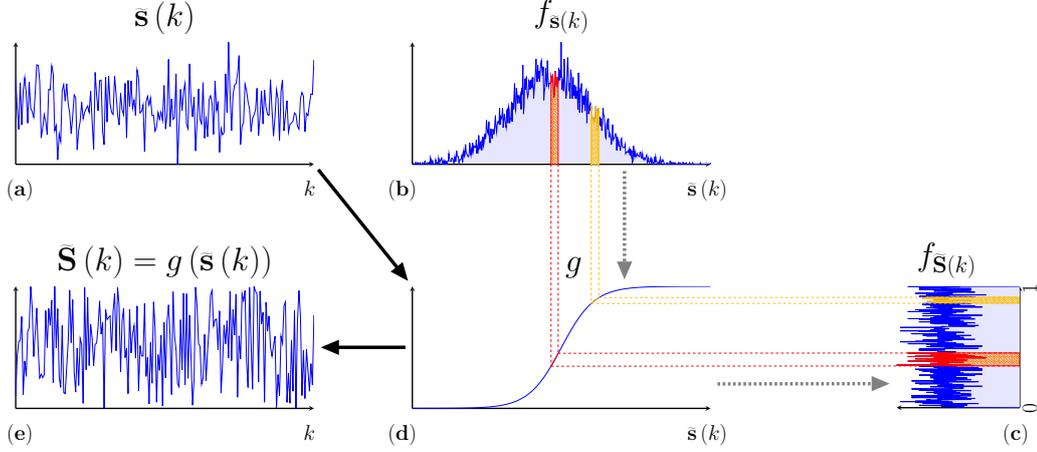


Figure 3.3.: Schematic diagram of the principle of Infomax where a uniform distribution is obtained if the transformation of the signal matches its own cdf (adapted from [15, 103]). A normally distributed signal $\tilde{\mathbf{s}}(k) = [\tilde{s}(1), \dots, \tilde{s}(10000)]^T$ was generated where in (a) only 200 of the 10000 signal values are shown. (b) shows the approximated pdf $f_{\tilde{\mathbf{s}}(k)}$ of all signal values. By integrating the pdf $f_{\tilde{\mathbf{s}}(k)}$, an approximation to the cost-function g of $\tilde{\mathbf{s}}(k)$ is calculated (dashed arrow to (d)). If the obtained cdf matches the cost-function g , a uniformly distributed signal $\tilde{\mathbf{S}}(k)$ will be received (dashed arrow to (c)). The corresponding pdf $f_{\tilde{\mathbf{S}}(k)}$ is shown in (c) rotated by 90 degrees. The signal $\tilde{\mathbf{S}}(k)$ is uniformly distributed because $f_{\tilde{\mathbf{S}}(k)} \Delta\tilde{\mathbf{S}}(k)$ (shaded areas in (d)) must correspond to $f_{\tilde{\mathbf{s}}(k)} \Delta\tilde{\mathbf{s}}(k)$ (shaded areas in (b)). Thus, the intervals $\Delta\tilde{\mathbf{s}}(k)$ are either compressed or expanded by mapping them through the cost-function g to the corresponding intervals $\Delta\tilde{\mathbf{S}}(k)$ [103]. (e) shows 200 signal values of $\tilde{\mathbf{S}}(k)$ which are calculated by applying the cost-function g on $\tilde{\mathbf{s}}(k)$.

becomes noticeable in smaller changes of $\Delta\mathbf{W}$. A measure for the changes in $\Delta\mathbf{W}$ is the euclidean norm. If the euclidean norm is under a specified threshold then the optimal unmixing matrix \mathbf{W}_{opt} is found and the algorithm stops [22]. Furthermore, the natural-gradient version provides isotropic convergence properties for any local minimum [2]. In Infomax the density in $f_{\tilde{\mathbf{s}}(k)}$ depends on matching the slope of g with the bias weight \mathbf{W}_0 on the mean of the pdf $f_{\tilde{\mathbf{s}}(k)}$ where g is assumed to be a sigmoidal function with

$$g(\tilde{\mathbf{s}}(k)) = \frac{1}{1 + e^{-\tilde{\mathbf{s}}(k)}}, \quad (3.25)$$

since many underlying natural processes follow a sigmoidal distribution [15].

3.3. Ambiguities of Independent Component Analysis

As ICA belongs to the class of BSS, its goal is to estimate the unknown unmixing matrix \mathbf{W} in order to receive the original source signals $\mathbf{s}(k)$. However, the source signals $\mathbf{s}(k)$, which are unknown, can be estimated accepting a few restrictions. Features that cannot be recovered are the scaling and the signs of the components in $\mathbf{s}(k)$. In case of no additive noise, any multiplication of a constant ς_i with $s_i(k)$, $i = 1, \dots, n$, can be canceled by dividing the corresponding column $a_j(i)$, $j = 1, \dots, m$ of the mixing matrix \mathbf{A} by the same constant ς_i :

$$x_j(k) = \sum_{i=1}^n \left(\frac{1}{\varsigma_i} a_j(i) \right) (s_i(k) \varsigma_i). \quad (3.26)$$

Hence, any component of $\mathbf{s}(k)$ can be recovered up to a multiplicative constant, where the components have unit variance. Furthermore, for any component $s_i(k)$ the negative signed component $-s_i(k)$ can as well be considered, which leads to a sign ambiguity.

Another restriction is the undetermined order of the components $s_i(k)$. The order in \mathbf{A} and in $\mathbf{s}(k)$ can jointly vary without effecting the observed signals $\mathbf{x}(k) = \mathbf{A} \cdot \mathbf{s}(k)$. In terms of a permutation matrix \mathbf{P} and its inverse \mathbf{P}^{-1} the observed signals $\mathbf{x}(k)$ can be written as

$$\mathbf{x}(k) = \mathbf{A} \cdot \mathbf{P}^{-1} \cdot \mathbf{P} \cdot \mathbf{s}(k) \quad (3.27)$$

without taking any effect on the result [55–57].

3.4. Pre-processing

The pre-processing is the first step before applying ICA on the records of sensor signals. There are mainly two pre-processing steps, namely the “centering” and the “whitening”. In case of 3D-PLI, the recorded sensor signals are the 18 acquired 3D-PLI images, in the following referred to as PLI image series, which are calibrated and segmented. The calibrated and segmented PLI image series are then pre-processed. These two pre-processing steps are formulated as follows:

The PLI image series \mathbf{x} are vectorized for each angle ρ taking the form

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_{18}^T \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} x_{1,1} & \cdots & x_{1,K} \end{pmatrix} \\ \begin{pmatrix} x_{2,1} & \cdots & x_{2,K} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} x_{18,1} & \cdots & x_{18,K} \end{pmatrix} \end{pmatrix}, \quad (3.28)$$

where K describes the total number of pixels per image.

The vectorized PLI image series are then centered (Fig. 3.4) by subtracting their sample mean $\bar{\mathbf{x}} = E\{\mathbf{x}\}$, thus, the centered PLI image series \mathbf{x}' are described by

$$\mathbf{x}' = \mathbf{x} - \bar{\mathbf{x}}. \quad (3.29)$$

Accordingly, the estimated source images $\tilde{\mathbf{s}}'$ are zero averaged as well, since

$$\tilde{\mathbf{s}} = \mathbf{W} \cdot \mathbf{x} \iff \tilde{\mathbf{s}} - \mathbf{W} \cdot \bar{\mathbf{x}} = \mathbf{W} \cdot \mathbf{x} - \mathbf{W} \cdot \bar{\mathbf{x}} \iff \tilde{\mathbf{s}} - \mathbf{W} \cdot \bar{\mathbf{x}} = \mathbf{W} (\mathbf{x} - \bar{\mathbf{x}}) \iff \tilde{\mathbf{s}}' = \mathbf{W} \cdot \mathbf{x}'. \quad (3.30)$$

After estimating the unmixing matrix \mathbf{W} , the estimated source images $\tilde{\mathbf{s}}$ can be reconstructed by adding $\mathbf{W} \cdot \bar{\mathbf{x}}$ to the estimated source images $\tilde{\mathbf{s}}'$. For simplicity, \mathbf{x} and $\tilde{\mathbf{s}}$ are denoted in the following as images that are already centered.

As a second of the sICA's pre-processing step, the centered PLI image series \mathbf{x} are whitened (Fig. 3.5), i.e., images \mathbf{x}_j of the 18 3D-PLI images are uncorrelated and their variances are made equal to unity [56]:

$$\text{Cov}(\mathbf{x}, \mathbf{x}^T) = E\{\mathbf{x} \cdot \mathbf{x}^T\} = \mathbf{I}, \quad (3.31)$$

where $\text{Cov}(\mathbf{x}, \mathbf{x}^T)$ is the covariance matrix of the outer product between \mathbf{x} and \mathbf{x}^T . Note, whitening is a term coming from white noise, which has a constant power spectrum over all frequencies [56]. One possible way to achieve whitening (Eq. 3.31) is to apply *eigenvalue decomposition* (EVD) to the covariance matrix

$$E\{\mathbf{x} \cdot \mathbf{x}^T\} = \mathbf{E} \cdot \mathbf{\Lambda} \cdot \mathbf{E}^T, \quad (3.32)$$

where \mathbf{E} is an orthogonal matrix containing the eigenvectors of $E\{\mathbf{x} \cdot \mathbf{x}^T\}$ and $\mathbf{\Lambda}$ is a diagonal matrix of the corresponding eigenvalues λ_i , $i = 1, \dots, 18$ with

$$\lambda_1 \geq \dots \geq \lambda_{18}.$$

Whitening is applied by:

$$\hat{\mathbf{x}} = \mathbf{E} \cdot \mathbf{\Lambda}^{-1/2} \cdot \mathbf{E}^T \cdot \mathbf{x}, \quad (3.33)$$

$$= \mathbf{D}_{\mathbf{x}} \cdot \mathbf{x}, \quad (3.34)$$

where $\mathbf{\Lambda}^{-1/2} = \text{diag}(\lambda_1^{-1/2}, \dots, \lambda_{18}^{-1/2})$, $\mathbf{D}_{\mathbf{x}}$ is the whitening matrix and $\hat{\mathbf{x}}$ are the whitened PLI image series (Fig. 3.6). Thus, whitening transforms the mixing matrix \mathbf{A} into a new matrix referred to as $\hat{\mathbf{A}}$ by:

$$\hat{\mathbf{x}} = \mathbf{E} \cdot \mathbf{\Lambda}^{-1/2} \cdot \mathbf{E}^T \cdot \mathbf{x} = \mathbf{E} \cdot \mathbf{\Lambda}^{-1/2} \cdot \mathbf{E}^T \cdot \mathbf{A} \cdot \mathbf{s} = \hat{\mathbf{A}} \cdot \mathbf{s}. \quad (3.35)$$

Furthermore, $\hat{\mathbf{A}}$ is orthogonal:

$$E \{ \hat{\mathbf{x}} \cdot \hat{\mathbf{x}}^T \} = \hat{\mathbf{A}} \cdot E \{ \mathbf{s} \cdot \mathbf{s}^T \} \cdot \hat{\mathbf{A}}^T = \hat{\mathbf{A}} \cdot \mathbf{I} \cdot \hat{\mathbf{A}}^T = \hat{\mathbf{A}} \cdot \hat{\mathbf{A}}^T = \mathbf{I}. \quad (3.36)$$

The new orthogonal mixing matrix $\hat{\mathbf{A}}$ contains $n(n-1)/2$ degrees of freedom instead of n^2 parameters (for $n = m$) that correspond to the elements of the original mixing matrix \mathbf{A} . Thus, whitening reduces the complexity of the problem and solves almost half of the parameters to be estimated. The remaining parameters are estimated by the following sICA, which task is now well-posed due to the whitening procedure [26, 55].

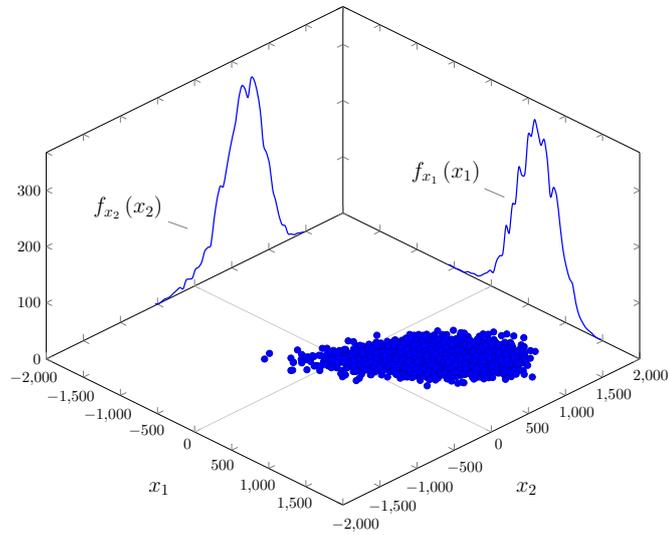


Figure 3.4.: Joint distribution of the first two centered images of one PLI image series \mathbf{x}_1 and \mathbf{x}_2 where only the first 5000 values are shown. The corresponding distributions $f_{\mathbf{x}_1}(\mathbf{x}_1)$ and $f_{\mathbf{x}_2}(\mathbf{x}_2)$ are depicted at the \mathbf{x}_1 and \mathbf{x}_2 axis.

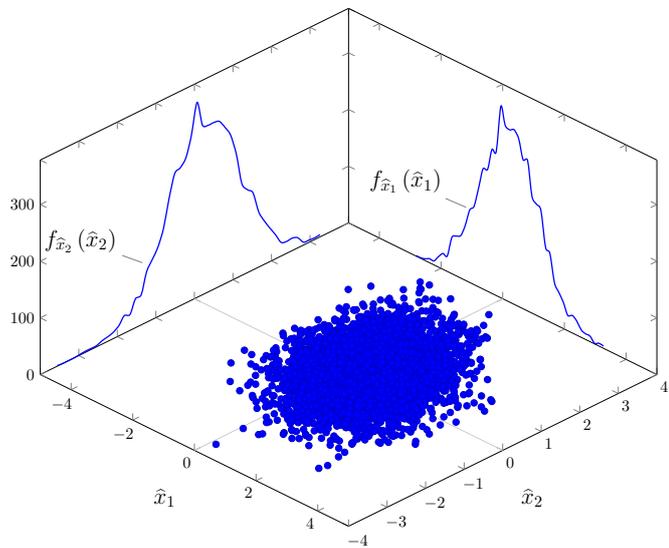
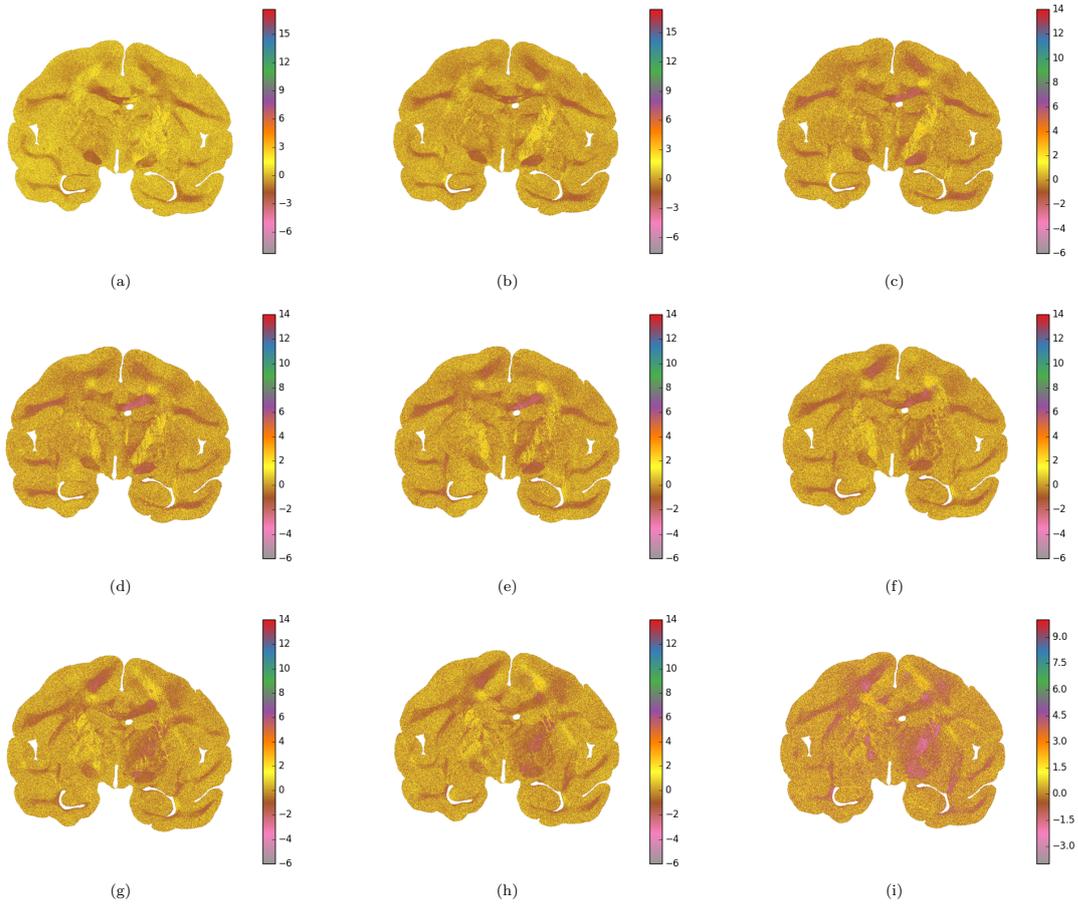


Figure 3.5.: Joint distribution of the first two centered images of one PLI image series $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ after whitening. Only two first 5000 pixel values are shown in the joint distribution. After whitening the joint distribution is more spreaded across the $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ plane yielding to a set of uncorrelated signals $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ with unit variance.

3. Principles of the Independent Component Analysis



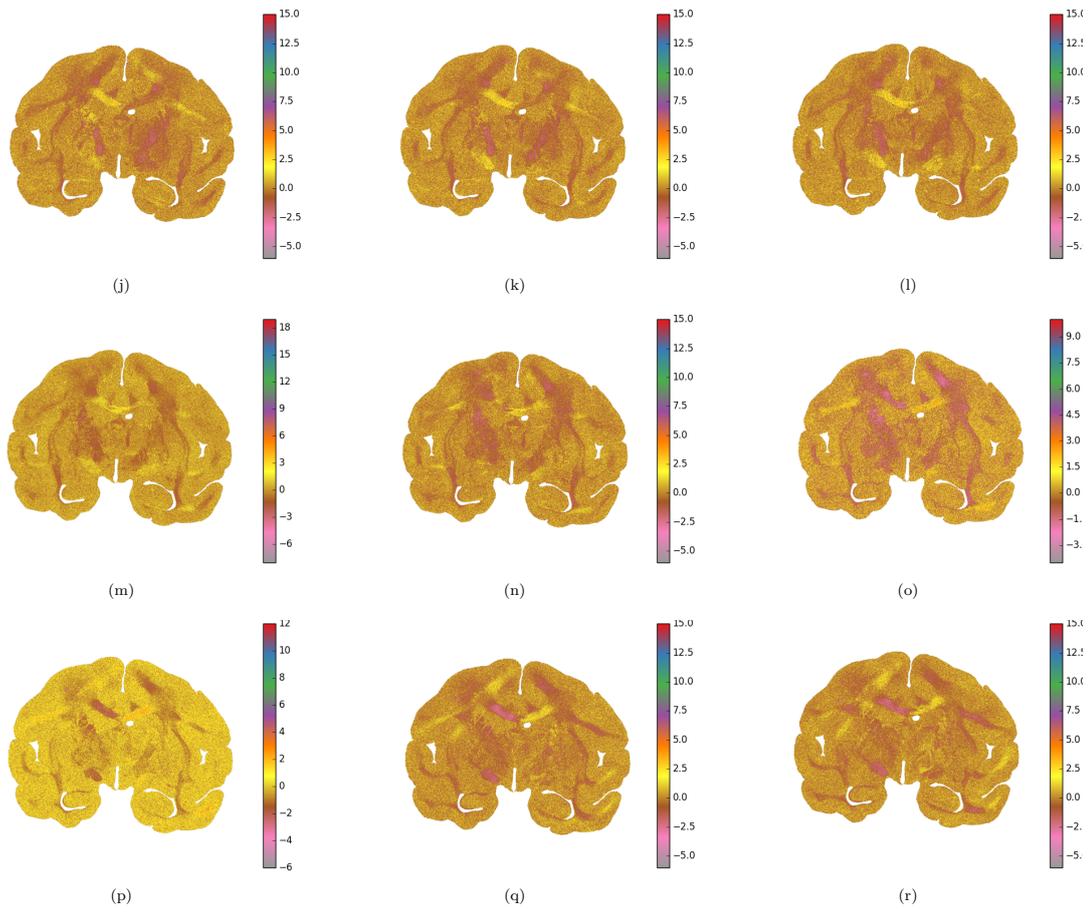


Figure 3.6.: (a) - (r) show the decorrelated PLI image series $[\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{18}]$ after whitening. Note, that the intensities in the whitened PLI image series are arbitrary. Since all components of $\hat{\mathbf{x}}$ can be attributed to white and gray matter structure, an identification of noise and artifact components is not feasible.

3.5. Constrained Independent Component Analysis for 3D-PLI

Constrained Independent Component Analysis for 3D-PLI (cICAP, [21]) is a new derived approach from sICA applications. In cICAP, Infomax is the basic implementation with the additional incorporation of prior information to the cost-function. The addition of prior information enables a faster extraction of the original sinusoidal profile in the set of 3D-PLI images and the automatic identification of components of interest [21].

For applying cICAP on 3D-PLI images, three assumptions for a valid sICA application are regarded [26, 57]:

- ▷ The source signals are statistically independent, since different physical processes generate outputs that are independent of each other [103]. In 3D-PLI, spatial independent components at different rotation angles ρ are expected, where the mixing process occurs across space.
- ▷ The source signals must be nongaussian distributed in order to estimate the mixing matrix \mathbf{A} . In case of gaussian distributed source signals sICA is not feasible, since uncorrelated gaussian variables are already independent (see Appendix C).
- ▷ The number of source signals n may not exceed the number of observed signals m . Note, in 3D-PLI the number of source signals are assumed to be equal to the number of observed signals yielding an estimation of a squared 18×18 unmixing matrix \mathbf{W} and 18 source signals.

Application of Constrained Independent Component Analysis for 3D-PLI In cICAP, the centered and whitened PLI image series are decomposed in spatially independent components, referred to as basis vectors (columns of the mixing matrix \mathbf{A}), yielding maximal independent source images $\tilde{\mathbf{s}}$. Figure 3.8 shows noise sources and the typical sources attributed to white and gray matter structure for one histological section. By removing noise and artifact components, the underlying sinusoidal profile in the PLI image series can be restored. The application of cICAP in pseudocode reads as

Algorithm 1 Constrained Independent Component Analysis for 3D-PLI

Precondition: \mathbf{x} is centered and whitened

```

1:  $\mathbf{W}_0 \leftarrow \mathbf{D}_W \cdot \mathbf{W}_\Pi$ 
2:  $\mathbf{W}_0^* \leftarrow \mathbf{I}$ 
3:  $c \leftarrow 0$ 
4: while  $\Delta \mathbf{W}_c > 10^{-10}$  AND  $c \leq 500$  do
5:    $\tilde{\mathbf{s}}_c \leftarrow \mathbf{W}_c \cdot \mathbf{x} + \mathbf{W}_0^*$ 
6:    $g(\tilde{\mathbf{s}}_c) \leftarrow \frac{1}{1+e^{-\tilde{\mathbf{s}}_c}}$ 
7:    $\Delta \mathbf{W}_c \leftarrow \tau [\mathbf{I} + (1 - 2g(\tilde{\mathbf{s}}_c)) \tilde{\mathbf{s}}_c^T] \mathbf{W}_c$ 
8:    $\Delta \mathbf{W}_0^* \leftarrow \mathbf{I} (1 - 2g(\tilde{\mathbf{s}}_c))$ 
9:    $\mathbf{W}_c \leftarrow \Delta \mathbf{W}_c + \mathbf{W}_c$ 
10:   $\mathbf{W}_0^* \leftarrow \Delta \mathbf{W}_0^* + \mathbf{W}_0^*$ 
11:  Incorporation of prior information
12:   $c \leftarrow c + 1$ 
13: end while

```

with \mathbf{W}_0^* being a bias weight matrix, $\Delta \mathbf{W}_0^*$ denoting the differential bias weight matrix, \mathbf{I} representing the identity matrix, $\Delta \mathbf{W}_c$ being the differential unmixing matrix, and \mathbf{W}_c being the unmixing matrix [21].

For $c = 0$, \mathbf{W}_c is initialized with a random orthogonal matrix

$$\mathbf{D}_W = \mathbf{E}_W \cdot \Lambda_W^{-1/2} \cdot \mathbf{E}_W^T \quad (3.37)$$

multiplied by a uniformly distributed matrix \mathbf{W}_Π in order to avoid local minima and computational bias [55]. During the learning algorithm the differential unmixing matrix $\Delta \mathbf{W}_c$ and the differential bias weight matrix $\Delta \mathbf{W}_0^*$ are estimated (step 7, 8) and added to the estimated matrices of the previous iteration step $c - 1$ (step 9, 10).

In the last step the prior information is incorporated as follows:

The weight matrix \mathbf{W}_c has to be adjusted by

$$\widehat{\mathbf{W}}_c = \mathbf{D}_x \cdot \mathbf{W}_c \quad (3.38)$$

with the whitening matrix \mathbf{D}_x of the decorrelation process of the centered PLI image series. The adjustment in $\widehat{\mathbf{W}}_c$ is mandatory for retrieving the temporal signal in the columns of $\widehat{\mathbf{A}}_c$ for the subsequent automatic identification of gray and white matter [21]. Then, $\widehat{\mathbf{A}}_c$ and n theoretically expected functions $f_j(u_\rho)$, $j = 1, \dots, n$

Algorithm 2 Incorporation of prior information

- 1: $\widehat{\mathbf{W}}_c \leftarrow \mathbf{D}_x \cdot \mathbf{W}_c$
 - 2: $\widehat{\mathbf{A}}_c \leftarrow \widehat{\mathbf{W}}_c^{-1}$
 - 3: Calculation of N theoretically expected functions
 - 4: Identification of basis vectors, whose entries are most similar to the corresponding expectation function (Fig. 3.7)
 - 5: Modification of basis vectors
 - 6: Check if quadratic difference between the modified basis vectors and the corresponding expectation function are in a predefined tolerance
 - 7: $\mathbf{W}_c^* \leftarrow (\widehat{\mathbf{A}}_c^{*-1} \cdot \mathbf{D}_x)^{-1}$
-

are calculated. The theoretically expected functions $f_j(u_\rho)$, which are fitted from the n basis vectors of $\widehat{\mathbf{A}}_c$ (Fig. 3.7, red curves), consists of the parameters a_0 , a_1 and b_1 as described by Equation 2.8. In step 3, a comparison between the n basis vectors $\widehat{a}_j(\rho)$, $j = 1, \dots, n$ and the expected functions $f_j(u_\rho)$ is performed (Fig. 3.7). The set of $\widehat{a}_j(\rho)$ that is most similar to $f_j(u_\rho)$ is determined. The similarity is measured by the kurtosis of

$$d_j = \widehat{a}_j(\rho) - f_j(u_\rho). \quad (3.39)$$

Therefore, the kurtosis can be expressed as:

$$\text{kurt}(d_j) = \frac{1}{N} \sum_{\rho} \left(\frac{d_j - \bar{d}_j}{\sigma_j} \right)^4 - 3, \quad (3.40)$$

with \bar{d}_j being the mean of the deviation function of the j^{th} basis vector and σ_j denoting the variance. The kurtosis is the only known measure providing a 100% positive rate for the automatic identification of gray and white matter as has been demonstrated in [30].

Moreover, the selection of each $\widehat{a}_j(\rho)$ is performed in two steps. First, the minimum of d_j is taken. Second, prior information is included. Step 4 describes the incorporation of this prior information as

$$\widehat{a}_j^*(\rho) = (1 - \eta) \widehat{a}_j + \eta f_j(u_\rho), \quad (3.41)$$

where η denotes the confidence value between 0 and 1. To decide whether the difference of the modified basis vector $\widehat{a}_j^*(\rho)$ and $f_j(u_\rho)$ is within a predefined

3. Principles of the Independent Component Analysis

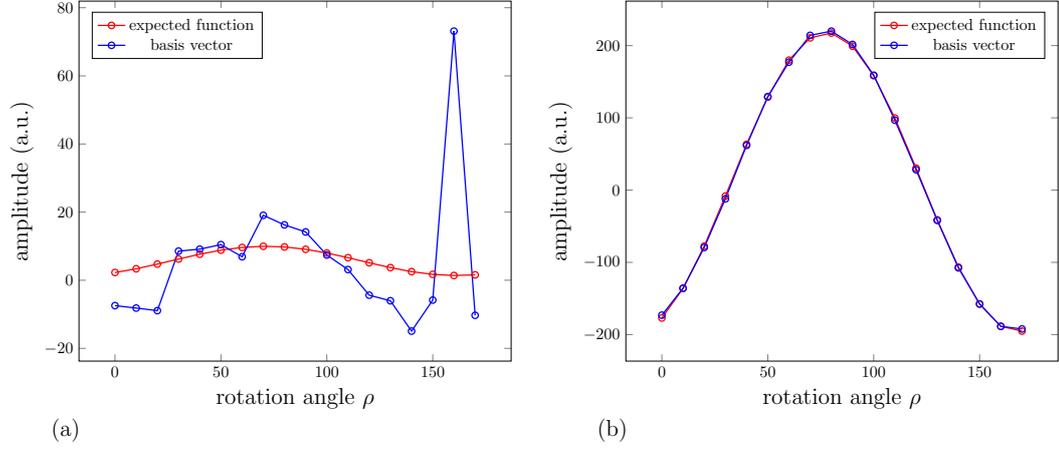


Figure 3.7.: Two exemplary basis vectors (blue) with their theoretically expected functions $f(u_\rho)$ (red). (a) shows a basis vector that differs across all rotation angles ρ from its $f(u_\rho)$. Hence, this basis vector is identified as noise component. (b) depicts an identified component with signal of interest which fits its $f(u_\rho)$. The identified basis vector representing signal of interest will be modified further by including prior information.

tolerance value t , the squared difference is computed with

$$\frac{1}{N} \sum_{\rho} \left[\hat{a}_j^*(\rho) - f_j(u_\rho) \right]^2 \quad (\text{Step 5}). \quad (3.42)$$

If

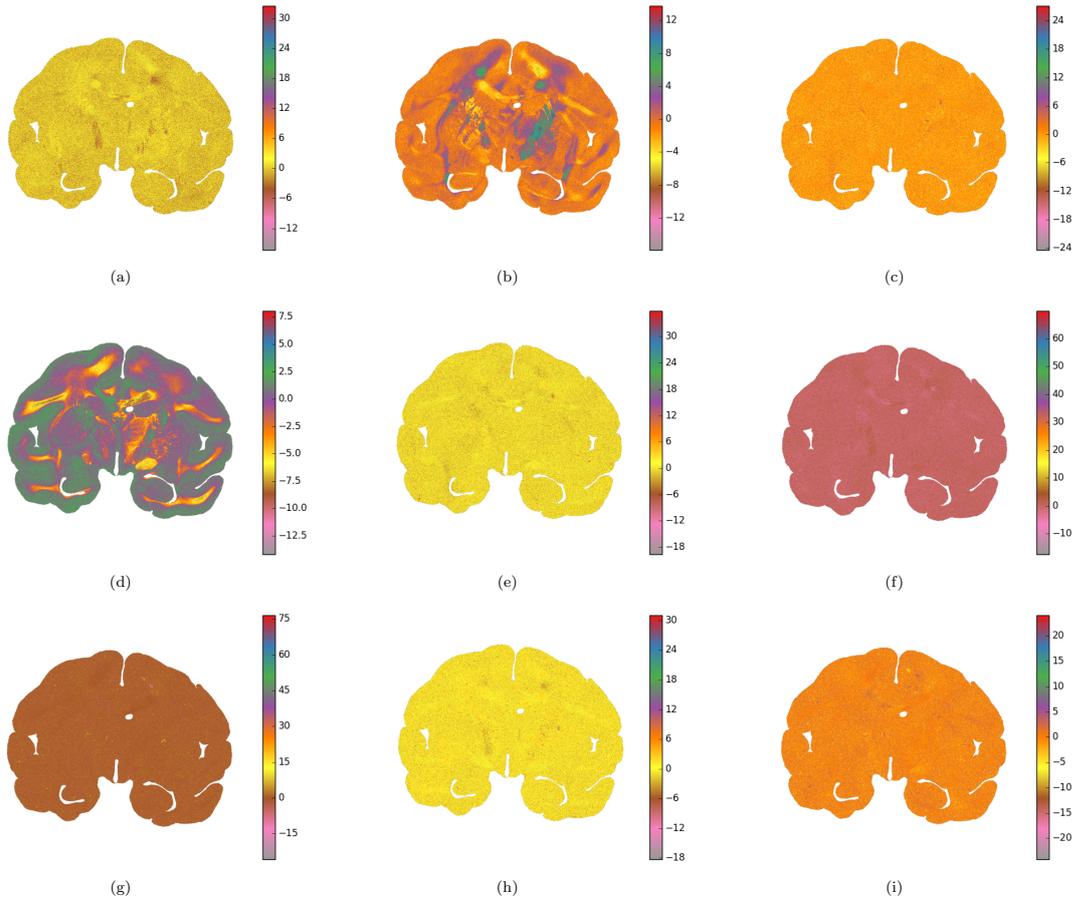
$$t > \frac{1}{N} \sum_{\rho} \left[\hat{a}_j^*(\rho) - f_j(u_\rho) \right]^2 \quad (3.43)$$

the entries of $\hat{a}_j^*(\rho)$ are fixed, because they represent a part of the signal. The fixed entries of $\hat{a}_j^*(\rho)$ are therefore approximately identical with $f_j(u_\rho)$. The last step is the calculation of $\mathbf{W}_c^* \leftarrow (\hat{\mathbf{A}}_c^{*-1} \cdot \mathbf{D}_x)^{-1}$. All steps are repeated until convergence. The selection of components of interest and differentiation of updating $\hat{a}_j^*(\rho)$ is repeated for all remaining columns in $\hat{\mathbf{A}}_c^*$. A stopping criterion for the iteration is a predefined threshold ϵ for the squared difference (with $\epsilon \gg t$) and until no further components of interest are found in $\hat{\mathbf{A}}_c^*$ [21].

An important factor for an accurate application of cICAP is the right choice for the parameters η , t and ϵ . The selection of these parameters was performed by

3. Principles of the Independent Component Analysis

Breuer et al. [21] in an independent data set of 100 brain sections, where the best parameters were found to be $\eta = 0.16$, $t = 2.2 \cdot 10^{-7}$ and $\epsilon = 0.01$.



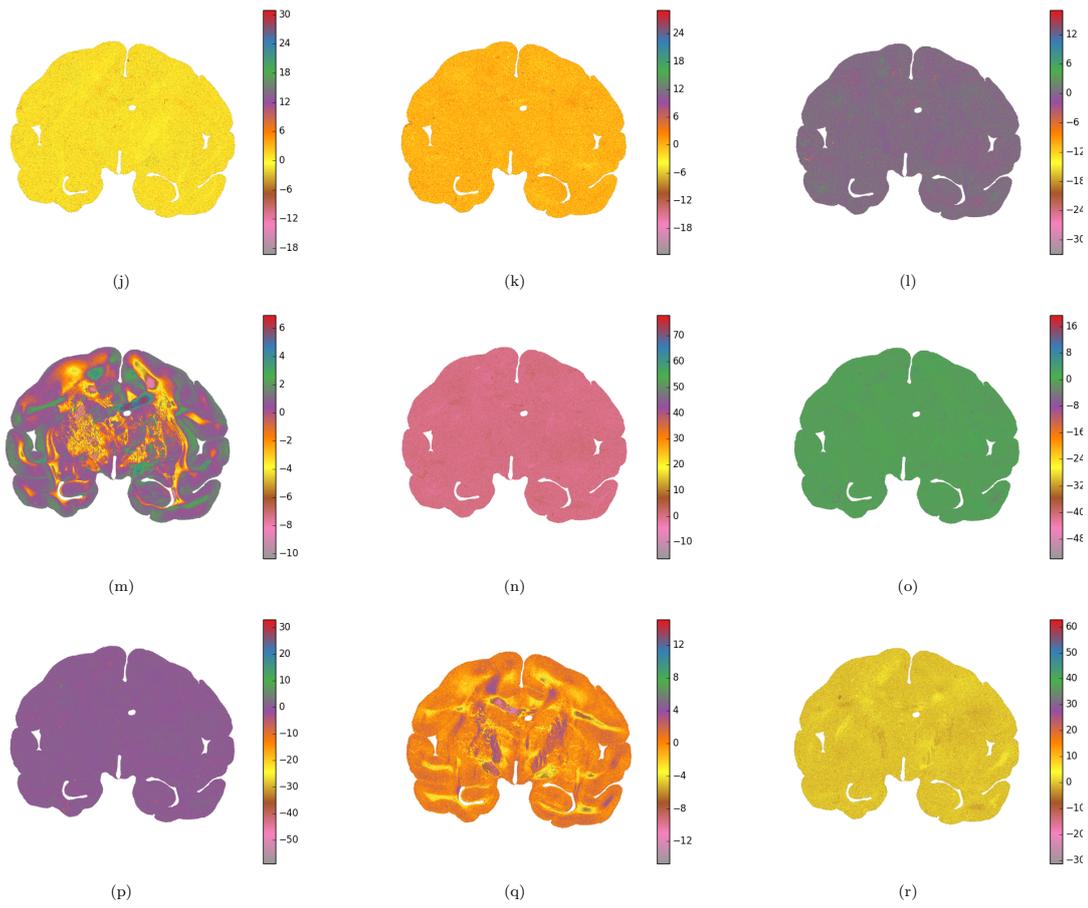


Figure 3.8.: (a) - (r) show the independent source images $[\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{18}]$ after cICAP. Note, that the intensities in the components of $\tilde{\mathbf{s}}$ are arbitrary. The white and gray matter structure in (b), (d), (m) and (q) are clearly accentuated by their contrast edges and hue difference and can be easily separated from noise and artifact components.

4. High Performance Computing

In the LAP and PM a great amount of data, ranging from a few hundred MB up to one TB for one histological brain section, are accumulated during the acquisition procedure. The acquired PLI image series are processed independently for both optical systems following the workflow procedure of 3D-PLI. In case of the PM, the acquired tiles for one histological section are passed through the workflow, where the intermediate results of similar size are stored. Furthermore, parts of the workflow, e.g., the segmentation and the ICA procedure, require to handle an entire histological section at once. Consequently, each image processing software used in the workflow is confronted with the problems of processing and properly managing Big Data. So far the traditional sequential method was used which has reached its limitations. However, processing in a sequential fashion means discretizing a computational task into a series of instructions, where each instruction is executed on a single processor sequentially. In contrast, parallel computing solves the discretized tasks in a concurrent manner by using multiple compute resources, for example a desktop computer with multiple cores or cluster of single computers connected by a network. Such an approach is able to speed up the compute process and to provide sufficient memory in a scalable manner. Hence, a high performance computing system is required to provide efficient analysis, concurrency and to process Big Data.

The Forschungszentrum Jülich hosts a variety of *High Performance Computing* (HPC) platforms, e.g., the *Jülich Blue Gene/Q* (JuQUEEN), the *Jülich Research on Exascale Cluster Architectures* (JuRECA), the *Dynamical Exascale Entry Platform* (DEEP) and the *Jülich Dedicated GPU Environment* (JuDGE), to name a few [62]. For this work, the *Graphics Processing Unit* (GPU) cluster JuDGE is used, providing suitable conditions for running cICAP in parallel.

The system configuration of JuDGE is described in Section 4.1. Section 4.2 emphasizes the parallel programming paradigms. In Section 4.3, parallelization strategies are explained. The last section focuses on the models for parallel programming used in this work.

For the sake of clarity and comprehension, the definition of abbreviations for the following sections are summarized in Table 4.1.

Table 4.1.: Definition of abbreviations in this chapter.

	definition	reference
API	Application Programming Interface	
cICAP	Constrained Independent Component Analysis for 3D-PLI	see Chapter 3
COMA	Cache Only Memory Access	
CPU	Central Processing Unit	
CUDA	Compute Unified Device Architecture	
DEEP	Dynamical Exascale Entry Platform	
GB	Gigabyte	
GHz	Gigahertz	
GPU	Graphics Processing Unit	
GPGPU	General Purpose GPU programming	
HPC	High Performance Computing	
ICA	Independent Component Analysis	see Chapter 3
INM-1	Institute of Neuroscience and Medicine 1	
JuQUEEN	Jülich Blue Gene/Q	
JuDGE	Jülich Dedicated GPU Environment	
JuRECA	Jülich Research on Exascale Cluster Architectures	
JSC	Jülich Supercomputing Center	
LAP	Large Area Polarimeter	see Chapter 2
MB	Megabyte	
MIMD	Multiple Instruction - Multiple Data	
MISD	Multiple Instruction - Single Data	
MPI	Message Passing Interface	
NUMA	Nonuniform Memory Access	
PE	Processor Element	
PM	Polarizing Microscope	see Chapter 2
SIMD	Single Instruction - Multiple Data	
SISD	Single Instruction - Single Data	
TB	Terabyte	
UMA	Uniform Memory Access	
3D-PLI	3D-Polarized Light Imaging	see Chapter 2

4.1. System Configuration of JuDGE

JuDGE is a supercomputer hosted at the *Jülich Supercomputing Center* (JSC). During the research period of this thesis, JuDGE was the only supercomputer in the JSC providing GPUs together with *Central Processing Units* (CPUs). A description of the specific system configuration is given in Table 4.2.

122 compute nodes are dedicated for tasks of the *Institute of Neuroscience and Medicine 1* (INM-1) including tasks of 3D-PLI.

Table 4.2.: System configuration of JUDGE (adapted from [59])

Complete System	
• 206 compute nodes	
• Compute node:	<ul style="list-style-type: none"> • 2 CPUs Intel Xeon X5650 (Westmere) 6-core processor 2.66 GHz • 2 GPUs NVIDIA Tesla M2050/M2070 • 1.15 GHz 3GB/6GB memory • 96 GB main memory
• 239 TeraFLOPS* peak performance	
• Infiniband QDR** network (40 GB/s)	

* Floating Operation Per Second (FLOPS)

** Quad Data Rate (QDR)

4.2. Parallel Programming Paradigms

In order to develop parallelized software on a supercomputer, it is necessary to understand the underlying computer architecture and the programming paradigms¹. The computer architecture, which defines the granularity supported on the supercomputer, together with the type of parallelism determine the paradigms that can be applied [102]. In 1966, Michael Flynn [39] distinguished four different categories of computer architectures, known as “Flynn taxonomy”. Flynn classified four categories based on the data stream and number of concurrent instructions (Table 4.3, Fig. 4.2). In addition, another programming paradigm exists which defines thread-

¹The interaction between application implementation and architectural development defines a "co-design" process.

level parallelism at various granularity levels within a single die [92]. The “Flynn taxonomy” and the core multi-threading are described in the following.

Table 4.3.: Flynn’s taxonomy classifying the four categories: Single Instruction - Single Data (SISD), Single Instruction - Multiple Data (SIMD), Multiple Instruction - Single Data (MISD) and Multiple Instruction - Multiple Data (MIMD) architecture.

Single Instruction	Multiple Instruction	
SISD	MISD	Single Data
SIMD	MIMD	Multiple Data

Single Instruction - Single Data (SISD) SISD describes a single core computer that performs no parallel computation in either the data stream or instructions. SISD generally refers to a *Personal Computer* (PC) based on the Von Neumann architecture [88].

Single Instruction - Multiple Data (SIMD) SIMD refers to a computer which consists of array or vector processors. The array or vector processors compute the same instructions synchronously on multiple incoming data streams. The first operational machine using an SIMD array architecture was the ILLIAC IV in 1972 [60]. Some applications of SIMD include 3D rendering, image and video processing. For instance, the SIMD architecture of GPUs exhibits data level parallelism with a high arithmetic throughput for general purpose programming [98, 110]. Compared to GPUs the Intel Xeon Phi coprocessor offers a better programmability and a broader applicability and demonstrated that exploiting SIMD parallelism achieves significant speed ups [63, 107].

Multiple Instruction - Single Data (MISD) MISD is an architecture where pipelining is performed to achieve parallelism. Fault tolerance computers that executes consistently the same instructions for error detection belong in the category of MISD [37].

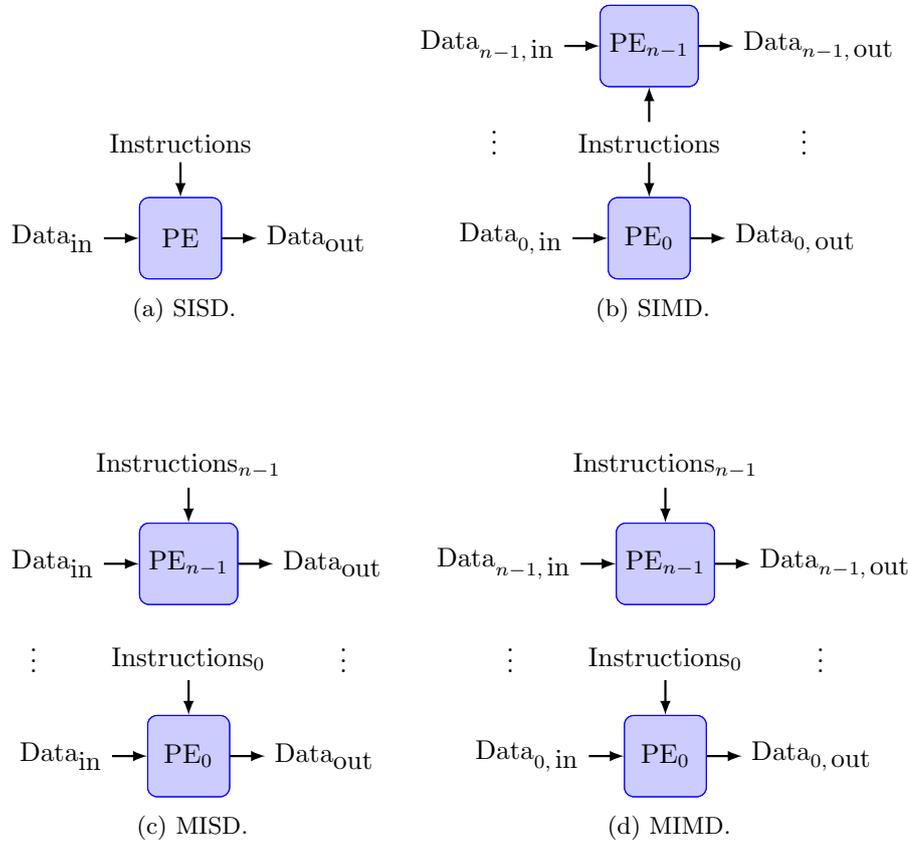


Figure 4.1.: A schematic illustration of SISD, SIMD, MISD and MIMD. SISD is characterized by one Processor Element (PE), one data and instruction stream. SIMD and MISD consist of n PEs with either one instruction stream or one data stream. MIMD describes an architecture with n instruction streams, n data streams and n PEs (adapted from [38]).

Multiple Instruction - Multiple Data (MIMD) MIMD consists of multiple interconnected processors performing synchronously different instructions on different data streams. A distinction in MIMD is drawn between a tightly coupled system and a loosely coupled system. A tightly coupled system refers to a multiprocessor system, whereas the loosely coupled system describes a multicomputer system. A multiprocessor system is also a shared memory system where each processor unit shares main memory and peripherals. Every shared memory system can be subdivided into: *Uniform Memory Access* (UMA), *Nonuniform Memory Access* (NUMA)

and a *Cache Only Memory Access* (COMA). In case of a multicomputer system each computer has its individual memory, referred to as distributed memory system, connected via a local network. If a processor in a distributed memory system needs data from another processor's memory, messages are passed between those two [37, 53].

Core Multi-Threading Core multi-threading provides a concurrent execution of multiple threads within one core, where a thread is defined as the execution of computational steps in an algorithm at a time. In general, three different granularity levels are distinguished in core multi-threading, namely block, interleaved, and simultaneous multi-threading.

In a block multi-threading each thread runs until a long latency event is encountered. At this point, the thread with the long latency event is suspended and the next thread starts running. In case of an interleaved multi-threading, all threads run simultaneously, where the processing of instructions from different threads are performed in consecutive cycles. The simultaneous multi-threading is an extension of interleaved version, where the instructions from simultaneous running threads are executed and processed in each and every CPU cycle [34].

4.3. Parallel Programming Strategies

In order to achieve a performance scaling on different computer architectures, specialized parallel programming strategies have been introduced. In the following, parallel programming strategies are described which vary from dividing a data set onto multiple processors to partitioning an algorithm in parallel.

Cloning/Farming Cloning/Farming is a strategy which focuses on performing the same task on multiple processors independently. All processors use their own parameter space for the calculations and no communication between the processors is needed. By covering a wide range of parameters, different analysis can be performed, e.g., achieving an optimal parameter combination for biological results [16] or establish a large statistical summary.

Master - Slave The master-slave programming strategy consists of one master entity and multiple slave entities. The master divides a big computing task into

partial tasks and distributes these small tasks to all slave processes. After computation the master gathers the partial results to form a total result. In contrast, the slave processes execute the given tasks and deliver them back to the master process. Thus, the only communication processes that are performed are between master and slaves. This strategy achieves a load-balancing and scales with the number of slave processes. However, a bottleneck arises if a large number of slaves are used and the entire communication is performed through one master process only [102] (Fig. 4.2).

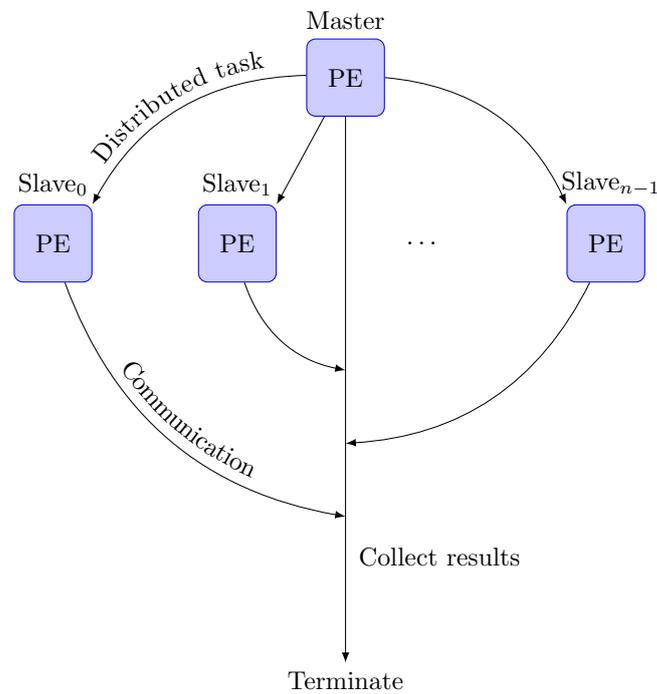


Figure 4.2.: A master-slave setup with a task and communication workflow (adapted from [102]).

Data Decomposition Another strategy for parallel programming is data decomposition, where concurrency is represented by the decomposition of its core data structure. A core data structure can be recursive, linear or represented as arrays. In all cases the algorithms need to take the composition into account. If a recursive data structure is given, divide-and-conquer patterns are feasible in which a problem is split into subproblems and solved concurrently. In case of a linear or

array representation, the data structure is divided into subregions with separated tasks processing each subregion. This decomposition is referred to as geometric decomposition. The resulting subregions with one or more dimension, referred to as chunks, are processed concurrently. Moreover, a geometric decomposition can be represented by overlapping chunks or by disjoint chunks. However, a consequence of this strategy is the choice of the sizes of the chunks. The communication of information between tasks should not exceed the processing time of each chunk. Hence, larger chunks are feasible with the constrain that too large chunks may antagonize a good load-balancing [82] (Fig. 4.3).

Functional Decomposition In functional decomposition, an algorithm is divided in tasks, which are insured to be disjoint. For computation, the disjoint tasks may require either disjoint data or overlapping data, where the computation of a task is dependent on the result of another task. The partition is complete if the disjoint tasks hold disjoint data. Thus, a divide-and-conquer approach is used. In case of data dependencies, the calculations can be viewed as a workflow through all tasks, referred to as pipelining [82] (Fig. 4.4).

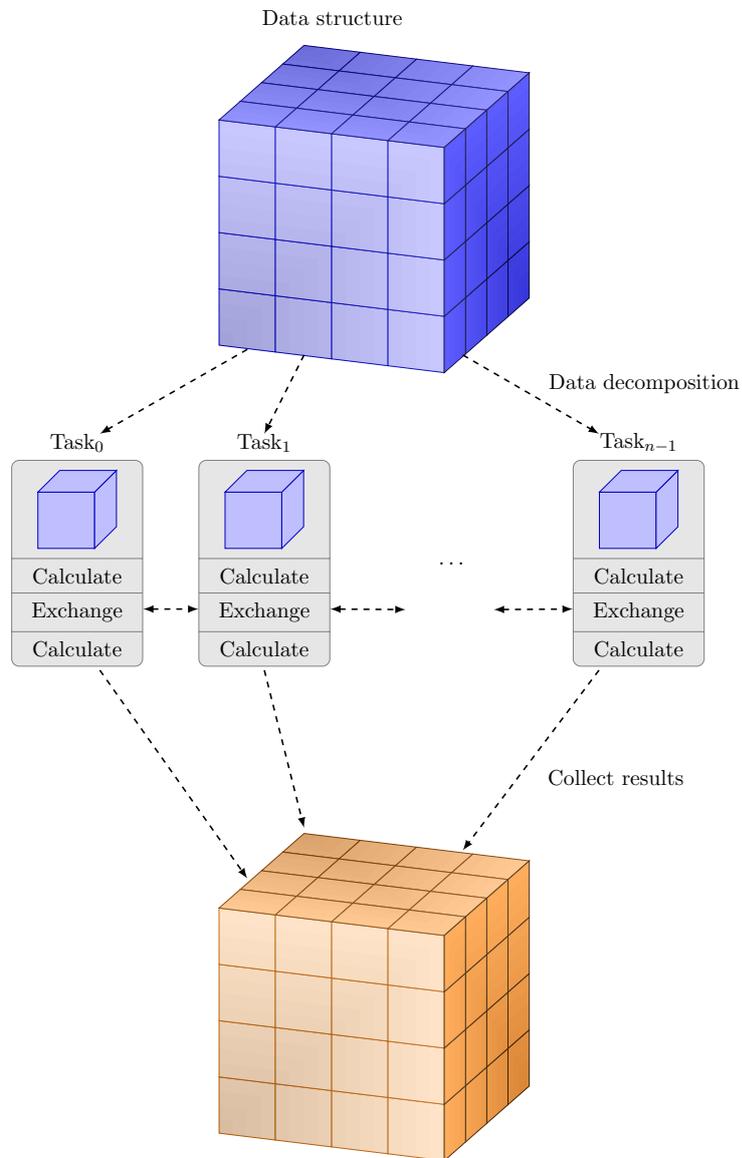


Figure 4.3.: A data decomposition strategy in which the core data structure is divided into chunks. Each processor load its prescribed chunk. After calculating and exchanging information with the other processors the results are collected (adapted from [102]).

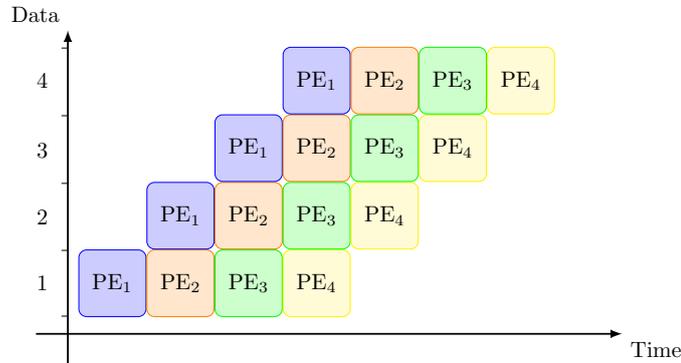


Figure 4.4.: Workflow of a pipeline strategy. The first PE₁ processes the first data element and sends the result to PE₂. After that, PE₂ continues processing the received results while PE₁ processes the second data element. This is performed until completion of all tasks (adapted from [82]).

4.4. Parallel Programming Models

Parallel programming models provide different implementation approaches for different architectures. A parallel programming model which can be applied and executed is valued by concurrency, scalability, locality and modularity [41]. The ability to perform many instructions simultaneously is referred to as concurrency, whereas scalability shows the ability to accommodate loads on increasing number of processors. Furthermore, locality indicates the access ratio between local and remote memory. Modularity means to simplify complex entities into simpler modules. Hence, many models with differing support regarding these four attributes have been proposed. In this work, the *Compute Unified Device Architecture* (CUDA) and *Message Passing Interface* (MPI) are used and described in the following.

Compute Unified Device Architecture An alternative to CPUs is using hardware accelerators, more precisely, GPUs. Due to the high arithmetic throughput with programmable pipelines, GPUs have attracted many researchers. Hence, the graphics hardware and graphics *Application Programming Interface* (API) were redirected to perform general purpose programming, referred to as *General Purpose GPU programming* (GPGPU). Since the introduction of CUDA in 2007 the general purpose programming on NVIDIA graphic cards was simplified rendering parallel programming on GPUs available to a broader range of developers [98, 110].

Due to the fact that the supercomputer JuDGE at the JSC is equipped with NVIDIA graphic cards, CUDA with the available third party wrapper in Python, referred to as PyCUDA [72], was chosen for the implementation of this work. CUDA and PyCUDA offer an extensive documentation with tools for debugging and optimizing the performance of applications. Moreover, PyCUDA takes advantage of the interpreter language Python to access the complete CUDA API. Due to the abstraction in Python, the performance demands and efforts on the part of the programmer are reduced while nearly retaining the same full performance compared to C based GPU-code [72].

A simplified view of the components in CUDA-capable GPUs is given in the following list:

- ▷ Host interface
- ▷ Copy engine(s)
- ▷ Device memory interface
- ▷ Streaming multiprocessors

The host interface synchronizes the GPU and CPU, different engines on the GPU and multiple GPUs. Additionally, the host interface dispatches the GPU commands to the hardware units. The copy engine(s) performs a memory transfer between host and device and vice versa while computations are performed by the streaming multiprocessors. Furthermore, the device memory interface joins memory requests and supports bandwidths of more than 100 GB per second. The streaming multiprocessors contain processors that perform 32-bit integer and single- and double-precision floating-point arithmetics. Each streaming multiprocessor contains a register file and shared memory and connections to the global, constant and texture memory in the device memory. Moreover, the streaming multiprocessors are allowed to perform arbitrary read and write operations to their registers and shared memory [98, 110].

Message Passing Interface MPI specifies a variety of library interfaces to provide collective and remote-memory operations, parallel I/O and dynamic process creation to function mainly on MIMD architectures. If data is moved from one process to another through arbitrarily cooperative operations on each process, a communication environment for message-passing is mandatory, which is addressed by MPI [40]. MPI features thread safety, point-to-point and collective communications for all MPI-processes. If a point-to-point communication is performed between processes, different sending modes are therefore available:

- ▷ Standard: Minimal transfer time but no guarantee that the corresponding receive instruction has started.
- ▷ Synchronous: The send instruction only completes when the corresponding receive instruction has started.
- ▷ Ready: User has to guarantee that the receive instruction is already called before sending.
- ▷ Buffered: The buffer space needs to be declared. Irrespective of the receiver instruction the communication always completes.

In contrast, the receive instructions in MPI operate on all modes. Moreover, all routines are either blocking or non-blocking. A blocking communication returns when the sending and receive instructions are locally complete. In case of a non-blocking communication, deadlocks and idle times are avoided. Non-blocking communication returns immediately and the resulting overlap can be used for further communications or computations. The completion of a non-blocking communication is performed by waiting.

On the other hand, collective operations involve a group of processes where an action is performed on each process. For instance, collective actions are broadcast and reduce operation. In a broadcast operation a root process sends its buffer content to all processes in a group, whereas a global operation (sum or product) is performed in case of a reduce operation. Furthermore, all collective actions are defined as blocking communications [40].

Due to MPI's maturity of standardization and programming model it allows to write portable programs for any hardware architecture in the compiled programming language (C, Fortran) and also in an interpreter programming language (Python). The third party wrapper, MPI for Python [28], which was used in this work, is based on the standard MPI-2 C++ bindings. MPI for Python allows two different ways of communication with either generic Python objects or memory buffers on the C side. Thus, advantages provided from the C interface in MPI are combined with the convenience of Python.

5. Development of an Adapted Constrained Independent Component Analysis on White and Gray Matter

In the community of neuroscience several ICA variants have been proposed in order to reveal features or hints of neuronal activities in the acquired data. As an example, signals of interest such as neuronal oscillations may be extracted from recordings of *Magnetoencephalography* (MEG) and *Electroencephalography* (EEG) [29, 84, 89]. Also source localization in EEG and MEG applications are performed using ICA approaches [49]. In addition to seeking neuronal activities, identifying artifact signals, e.g., ocular artifacts, cardiac artifacts and muscle activity, using ICA has become a promising way to separate noise/artifacts from signals of interest in EEG and MEG [20, 31, 35, 66, 70, 80]. Furthermore, ICA is effectually used in *functional Magnetic Resonance Imaging* (fMRI) to denoise the measurements from random noise, pulsation and breathing artifacts. ICA in fMRI also allows to detect unexpected responses to stimuli [83].

For 3D-PLI, studies [21, 30, 32] showed that ICA is capable of restoring deteriorated polarized light signals by identifying the signal of interest, more precisely the sinusoidal nature of 3D-PLI, throughout the histological section. However, in contrast to MEG, EEG and fMRI the underlying sources in *gray* (GM) and *white matter* (WM) are still not understood sufficiently. The generalized source separation method (Eq. 3.25) proposed by Dammers et al. [30, 32] and Breuer et al. [21] elicits a global improvement of the *Signal-to-Noise Ratio* (SNR), but disregards the fact that the birefringence signal in gray and white matter varies in signal strength, whereas the distortion of gray matter signals due to noise is in magnitude higher than in white matter signals. As a consequence, the analysis of PLI signals requires a source separation method adapted to the signal dynamics across the histological

section, taking into account the birefringence property.

This chapter focuses on the development and reproducibility of a cICAP algorithm adapted on white and gray matter. The new method introduced here is specifically designed for the PLI image series and is based on cICAP, in which prior information of the underlying source signals of the gray and white matter is incorporated. This leads to a faster signal decomposition and an increased signal enhancement. In this new technique, referred to as adapted constrained ICA (acICA), different separation functions are used depending on the underlying source signals of the histological section.

For the sake of clarity and comprehension, the definition of variables and abbreviations for the following sections are summarized in Table 5.1.

Table 5.1.: Definition of variables and abbreviations in this chapter.

	definition	reference
b	Parameter for varying the growth rate	
G	Gray level of intensity	
$h(G)$	Gray level histogram	
$g(\tilde{\mathbf{s}})$	(Sigmoidal) cost-function	see Chapter 3
$g_a(\tilde{\mathbf{s}})$	Adapted cost-function	
$J(T)$	Criterion function of the minimum error thresholding	
μ	Mean	
$P(T)$	Priori probability	
q	Parameter for varying the point of inflection	
r	Parameter for varying the characteristic of the sigmoidal curve	
σ	Standard deviation	
$\tilde{\mathbf{s}}$	Vector of estimated source signals passed through the cost-function g	see Chapter 3
s_{jack}	Sample standard deviation	
SE_{jack}	Standard error	
T	Intensity threshold	
$\hat{\theta}$	Jackknife estimator	
\mathbf{W}	Unmixing matrix	see Chapter 3
\mathbf{W}_0	Bias weight	see Chapter 3
$wrGOF$	Weighted relative goodness-of-fit	see Chapter 2
\mathbf{x}	Vector of recorded sensor signals	see Chapter 3

\tilde{x}	Median of the 50 <i>wrGOF</i> images	
acICA	Adapted constrained Independent Component Analysis	
cICAP	Constrained Independent Component Analysis for 3D-PLI	see Chapter 3
cdf_{GM}	Cumulative Density Function of Gray Matter	
cdf_{pcWM}	Cumulative Density Function of Percentiles of White Matter	
EEG	Electroencephalography	
EPA	Efficient PLI Analysis	see Chapter 2
fMRI	Functional Magnetic Resonance Imaging	
GM	Gray Matter	
ICA	Independent Component Analysis	see Chapter 3
Infomax	Information based maximization	see Chapter 3
IQR	Interquantile Range	
LAP	Large Area Polarimeter	see Chapter 2
MEG	Magnetoencephalography	
pcWM	Percentiles of White Matter	
PM	Polarizing Microscope	see Chapter 2
SNR	Signal-to-Noise Ratio	
WM	White Matter	
3D-PLI	3D-Polarized Light Imaging	see Chapter 2

5.1. Experimental Setup

To understand the underlying source signals in 3D-PLI, the analysis of gray and white matter was evaluated on the acquired PLI image series of the LAP and the PM. For this purpose, a series of histological sections of one post-mortem vervet monkey brain was acquired in the LAP and the PM. The frozen vervet brain (male; age 2.4 years) was cut into coronar sections with a thickness of 60 μm using a cryotome. The number of histological sections amounts to 1182. The acquisition of the PLI image series of the 1182 sections in both optical setups is not finished yet. Thus, 140 sections were used for this work. These sections range from the frontal to the occipital pole and show a symmetric distribution across the brain (Fig. 5.1). For the evaluation of the reproducibility and performance of the acICA approach a section in the center of the 140 sections was chosen which comprised both gray and white matter. This section was measured 50 times consecutively in the LAP result-



Figure 5.1.: A schematic of 1182 coronar sections from a post-mortem vervet brain, where 140 samples were taken and acquired in the LAP and PM.

ing in 50 PLI image series which is in the following referred to as 50 “repro series”. For the PM, 20 PLI image series out of the 140 sections ranging equidistantly from the frontal to the occipital pole were chosen due to the limitation of working storage and time-span needed to fulfill a prompt analysis. All the 140 PLI image series and 50 repro series of the LAP and the 20 PLI image series of the PM were acquired with an exposure time of 70 ms and calibrated afterwards for further processing.

5.2. Methods

The analysis of gray and white matter relies upon accurate segmentation of brain regions from the PLI image series. At present, the accepted standard in the LAP for removing the background from a histological section is manual segmentation. In case of the PM, the removal of the background is performed by an in-house segmentation tool based on region growing algorithm. However, relying on manual segmentation to separate brain regions such as the gray and white matter results in a labor intensive and longsome process. In addition, the fact that manual segmentation among different observer may lead to different criteria renders an automated segmentation approach necessary.

The automated segmentation of gray and white matter evaluated here uses thresholding for segmenting gray level images of the LAP and PM. The method uses minimum error thresholding [71], which is based on the Bayesian classification rule [90], to determine an optimum threshold in order to separate the gray from the white matter in an image. This method supposes normal distributed components in the bi-modal gray level histogram $h(G)$ with G gray levels. The unknown standard

deviations σ_i , the unknown means μ_i and the unknown a priori probabilities P_i of the normal distributed components need to be estimated from $h(G)$ via fitting techniques. For the two classes ($i = 1, 2$), white and gray matter, the parameters σ_i , μ_i and P_i can be estimated with

$$P_i(T) = \sum_{G=a}^b h(G), \quad (5.1)$$

$$\mu_i(T) = \frac{1}{P_i(T)} \sum_{G=a}^b h(G) G \text{ and} \quad (5.2)$$

$$\sigma_i^2(T) = \frac{1}{P_i(T)} \sum_{G=a}^b (G - \mu_i(T))^2 h(G), \text{ where} \quad (5.3)$$

$$a = \begin{cases} 0 & i = 1 \\ T + 1 & i = 2 \end{cases} \text{ and} \quad (5.4)$$

$$b = \begin{cases} T & i = 1 \\ n & i = 2 \end{cases}. \quad (5.5)$$

By varying the threshold T the normal distributed components change. The more the estimated model fits the data, the smaller the overlap of the two normal distributed components resulting to a smaller classification error [71]. The criterion function incorporating the classification error is formulated as

$$J(T) = 1 + 2 [P_1(T) \log(\sigma_1(T)) + P_2(T) \log(\sigma_2(T))] - 2 [P_1(T) \log(P_1(T)) + P_2(T) \log(P_2(T))]. \quad (5.6)$$

The minimum error threshold is then calculated by minimizing $J(T)$.

For determining an optimal histogram bin width, the Freedman-Diaconis rule was used. The Freedman-Diaconis rule gives a simple and robust rule, namely, bin width = $2(\text{IQR})n^{-1/3}$, where IQR is the interquartile range of the data and n is the number of observations [42, 61].

In order to segment white and gray matter, all 140 PLI image series of the LAP and 20 PLI image series of the PM were processed first via EPA (see Section 2.5). Thus, direction, transmittance and retardation maps were generated for each PLI image

series. According to Axer et al. [9] the transmittance map roughly provides a general separation between white and gray matter due to the different light extinctions. By taking advantage of the transmittance map and applying minimum error thresholding, 140 LAP masks for gray and white matter were generated, respectively. For the 50 repro series one mask for gray and white matter was computed. In case of the PM, an in-house segmentation tool produced 20 masks separating tissue from background. Subsequently, minimum error thresholding and manual correction yielded 20 gray and white matter masks each.

With the intention of understanding how the different masks affect the signal separation and signal enhancement (*wrGOF*, Eq. 2.18) in cICAP, the 50 repro series with the corresponding masks for white and gray matter were processed with cICAP. Due to fact that cICAP, which was applied to the entire white matter, identified only components of interest no noise/artifact component could be removed. Thus, cICAP applied on partitions containing percentiles of the white matter density was carried out. By applying the corresponding gray matter and percentiles of the white matter masks to the 140 and 20 PLI image series respectively, two cdfs each were generated for the LAP and PM. For this purpose, the Freedman-Diaconis rule was also used to determine the same bin width for both cdfs. These two cdfs, which represent mostly the density of gray and percentiles of white matter, were used to optimize the cost-function (Eq. 3.25).

The here newly introduced ICA-based method for gray and white matter in 3D-PLI is slightly in line with the approach of Breuer et al. [20]. As a starting point the natural-gradient version of Infomax (see Section 3.2) was taken into account. The presented approach uses a modified cost-function $g(\tilde{\mathbf{s}}(k))$, where the generated cdfs of the individual gray (cdf_{GM}) and percentiles of white matter (cdf_{pcWM}) are embedded. By fitting $g(\tilde{\mathbf{s}}(k))$ to the cdf_{GM} and to the cdf_{pcWM} , two optimal analytical cost-functions for each optical setup are computed with

$$g_a(\tilde{\mathbf{s}}(k)) = \frac{1}{(1 + q \cdot e^{-b \cdot \tilde{\mathbf{s}}(k)})^{\frac{1}{r}}}, \quad (5.7)$$

where q affects the point of inflection, b describes the growth rate and r allows the characteristic of the sigmoidal curve to be varied. The cost-function $g_a(\tilde{\mathbf{s}}(k))$ is an adapted version of the Richards curve [96]. It should be noted that if $q = b = r = 1$, the cost-function $g_a(\tilde{\mathbf{s}}(k))$ will match the standard sigmoidal function $g(\tilde{\mathbf{s}}(k))$. All three parameters q , b and r are fitted in terms of the minimum square error to the

curves of the cdf_{GM} and of the cdf_{pcWM} .

By modifying the cost-function $g_a(\tilde{\mathbf{s}}(k))$ the learning rule described in Equation B.17 and B.18 is newly evaluated. First, the cost-function $g_a(\tilde{\mathbf{s}}(k))$ with $\tilde{\mathbf{s}}(k) = \mathbf{W} \cdot \mathbf{x}(k) + \mathbf{W}_0$ is formed to

$$g_a(\tilde{\mathbf{s}}(k)) = \left(1 + q \cdot e^{-b \cdot \tilde{\mathbf{s}}(k)}\right)^{-\frac{1}{r}} \text{ with the first derivative} \quad (5.8)$$

$$g'_a(\tilde{\mathbf{s}}(k)) = \frac{\mathbf{W} \cdot g_a(\tilde{\mathbf{s}}(k)) \cdot b}{r} (1 - g_a^r(\tilde{\mathbf{s}}(k))). \quad (5.9)$$

Second, we newly derive $\frac{d}{d\mathbf{W}}(g'_a(\tilde{\mathbf{s}}(k)))$ analogous to Equation B.15 as

$$\begin{aligned} \frac{d}{d\mathbf{W}}(g'_a(\tilde{\mathbf{s}}(k))) &= \frac{d}{d\mathbf{W}} \left(\frac{\mathbf{W} \cdot g_a(\tilde{\mathbf{s}}(k)) \cdot b}{r} (1 - g_a^r(\tilde{\mathbf{s}}(k))) \right) \\ &= \frac{b \cdot g_a(\tilde{\mathbf{s}}(k))}{r} \left[\mathbf{I} + \frac{\mathbf{W} \cdot \tilde{\mathbf{s}}(k) \cdot b}{r} (1 - g_a^r(\tilde{\mathbf{s}}(k)))^2 - \right. \\ &\quad \left. \frac{\mathbf{W} \cdot \tilde{\mathbf{s}}(k) \cdot b}{r} (1 - g_a^r(\tilde{\mathbf{s}}(k))) g_a^r(\tilde{\mathbf{s}}(k)) - g_a^r(\tilde{\mathbf{s}}(k)) \right]. \end{aligned} \quad (5.10)$$

By dividing Equation 5.10 by Equation 5.9 a new learning rule is calculated for $g_a(\tilde{\mathbf{s}}(k))$, which is formulated as

$$\Delta \mathbf{W} = \frac{d}{d\mathbf{W}} H(\tilde{\mathbf{S}}(k)) = \tau (g'_a(\tilde{\mathbf{s}}(k)))^{-1} \frac{d}{d\mathbf{W}}(g'_a(\tilde{\mathbf{s}}(k))) \mathbf{W}^T \mathbf{W} \quad (5.11)$$

$$= \tau \left[\mathbf{I} + \frac{b}{r} (1 - 2g_a^r(\tilde{\mathbf{s}}(k))) \tilde{\mathbf{s}}(k)^T \right] \mathbf{W}, \quad (5.12)$$

$$\Delta \mathbf{W}_0 = \mathbf{I} (1 - 2g_a^r(\tilde{\mathbf{s}}(k))). \quad (5.13)$$

The resulting $\Delta \mathbf{W}$, $\Delta \mathbf{W}_0$ and $g_a(\tilde{\mathbf{s}}(k))$ are integrated in the acICA algorithm for estimating the optimal unmixing matrix \mathbf{W}_{opt} individually for gray and percentiles of white matter.

In order to test the reproducibility and the signal enhancement of acICA, all 50 repro series were processed by acICA. Afterwards, 50 *wrGOF* images were generated. By means of a Jackknife test [85, 101] the standard error and confidence interval was estimated [17]. For this, the median \tilde{x} of all *wrGOF* images was computed, where the estimator in the Jackknife test is supposed to be $\hat{\theta} = \tilde{x}$. By leaving out one

wrGOF image at the time, one can compute $\bar{\theta} = \frac{1}{n} \sum_{i=1}^n \hat{\theta}_{(i)}$, where $\hat{\theta}_{(i)}$ denotes the estimate calculated from the *wrGOF* images with the i^{th} *wrGOF* image removed and n is the total number of images. Thus, the Jackknife estimator $\hat{\theta}_{\text{jack}}$ is generated as

$$\hat{\theta}_{\text{jack}} = \frac{1}{n} \sum_{i=1}^n \tilde{\theta}_i = n\hat{\theta} - (n-1)\bar{\theta}. \quad (5.14)$$

Hence, the standard error SE_{jack} is calculated with

$$\text{SE}_{\text{jack}} = \frac{s_{\text{jack}}}{\sqrt{n}}, \quad (5.15)$$

where s_{jack} describes the sample standard deviation. The confidence interval with a confidence level of 95% ($\alpha = 0.05$) was used and constructed as: $\hat{\theta}_{\text{jack}} \pm t_{1-\frac{\alpha}{2}; n-1} \text{SE}_{\text{jack}}$, where t is the Student's t -distribution with $n - 1$ degrees of freedom [104].

5.3. Results

The transmittance maps, which were first removed from background, are processed via minimum error thresholding yielding white and gray matter masks for the LAP and PM. For simplicity, the results of the segmentation of the 50 repro series is described. As the 50 repro series were acquired by measuring 50 times consecutively the same histological section in the LAP, only one white and gray matter mask is necessary. Therefore, by constructing a histogram with a bin width of 57.92 and 130 bins, resulting from the Freedman-Diaconis rule, a bimodal distribution is observed (Fig. 5.2(a)). The minimizing criterion $J(T)$ for this histogram has its local minimum at a gray value of 11917.64 (arbitrary unit) corresponding to the threshold value for segmenting mostly the white from the gray matter (Fig. 5.2(b)). Note that the peak of the larger mode (13448.6) corresponds mostly to the gray matter, whereas the mode at 10582.72 represents the white matter (Fig. 5.2(a)). Thus, a white and gray matter mask for the 50 repro series was generated. Similar results for the white and gray matter distributions were observed for 140 PLI image series of the LAP. The image series acquired with the PM showed no clear bimodal distribution, making it difficult to locate a local minimum of $J(T)$ which clearly separates white from gray matter. Due to this reason, the gray and white matter masks for the PM image series were afterwards manually corrected.

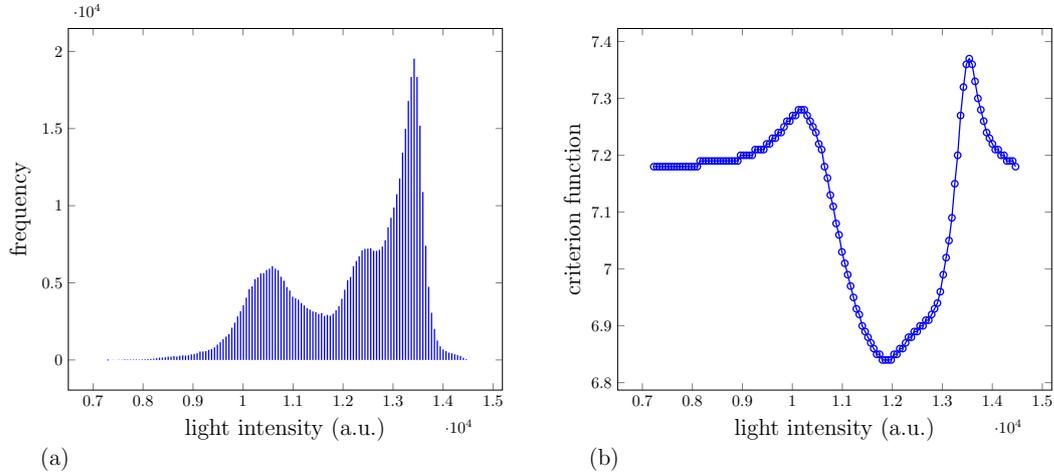


Figure 5.2.: A histogram of one transmittance map of the 50 repro series with 130 bins (a) and its criterion function $J(T)$ (b).

To evaluate signal enhancement on white and gray matter, cICAP was applied on white and gray matter of the 50 repro series. For the cICAP application on the whole white matter no signal enhancement ($wrGOF = 1$) was observed. Thus, a brute-force approach on percentiles of the white matter density of the retardation and transmittance map was carried out. Figure 5.3 provides a summary of these results. The $wrGOF$ minima of all percentiles of the transmittance map are below 1 (Fig. 5.3 (a)), whereas cICAP on white matter of the retardation map is significant at 4 percentiles with a $wrGOF$ of 10.34 (Fig. 5.3 (b)).

Consequently, all 140 image series of the LAP were masked and one cdf_{GM} and cdf_{pcWM} was generated. The resulting cdf_{GM} and cdf_{pcWM} , illustrated in Figure 5.4, show major differences. The cdf_{GM} is steeper compared to the cost function $g(\tilde{\mathbf{s}}(k))$ and the cdf_{pcWM} , whereas the cdf_{pcWM} has a higher progression at the lower asymptote than the $g(\tilde{\mathbf{s}}(k))$.

Then, $g_a(\tilde{\mathbf{s}}(k))$ is fitted to both cdfs resulting to the parameters q , b and r . Analogously, one cdf_{GM} and cdf_{pcWM} of the 20 PLI image series of the PM were produced with the same number of bins. The resulting parameters for both optical setups are shown in Table 5.2.

The cdf_{GM} and cdf_{pcWM} of the 20 PLI image series of the PM are depicted in Figure 5.5, where both cdfs show a slight resemblance to the cost function $g(\tilde{\mathbf{s}}(k))$.

To compare acICA_* with cICAP and Infomax_* , all three methods were tested on the 50 repro series of the LAP. In case of acICA_* and cICAP, the optimal parameters

5. Development of an Adapted cICAP on White and Gray Matter

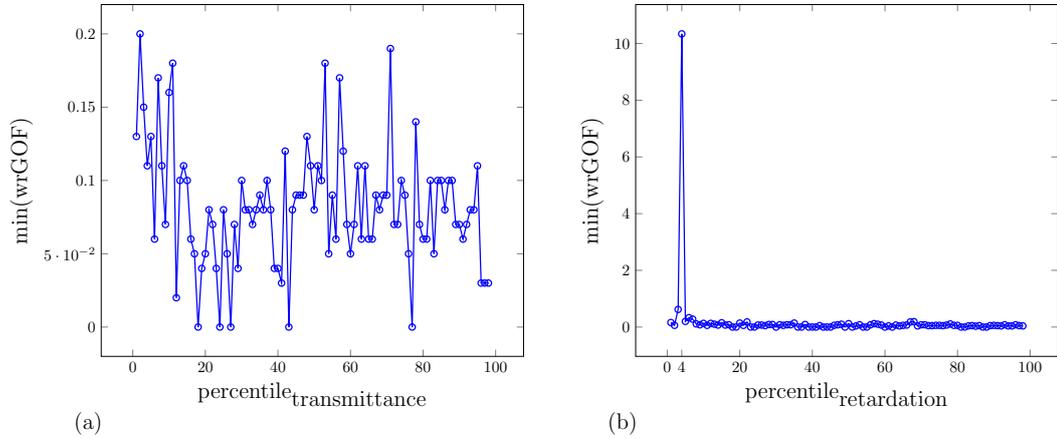


Figure 5.3.: Percentiles of the white matter distribution of the transmittance map ($\text{percentile}_{\text{transmittance}}$) and retardation map ($\text{percentile}_{\text{retardation}}$) were processed by cICAP. Then the minimum of all wrGOF values was calculated for $\text{percentile}_{\text{transmittance}}$ (a) and $\text{percentile}_{\text{retardation}}$ (b).

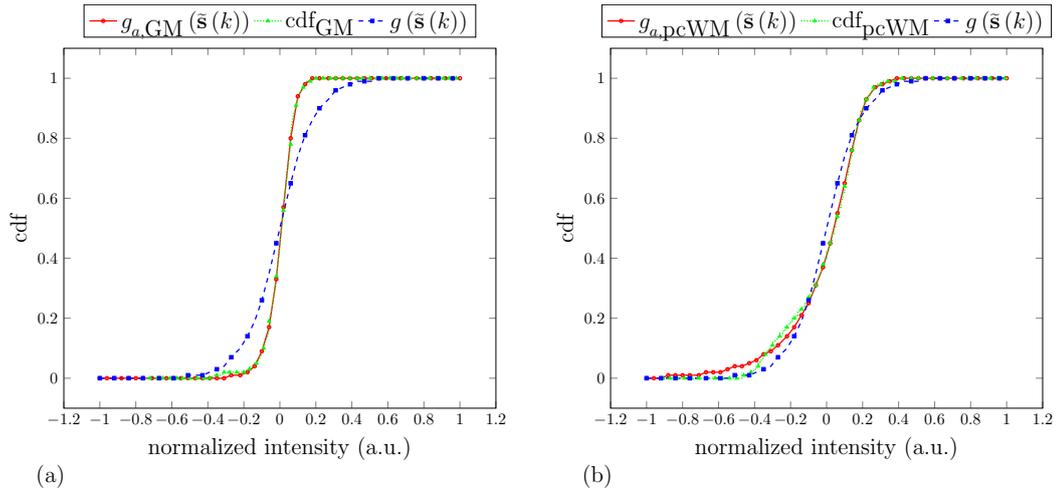


Figure 5.4.: Comparison of cdfs of GM (a) and pcWM (b) generated from the 140 PLI image series of the LAP. Illustrated are the default sigmoidal function $g(\tilde{s}(k))$ (blue, dashed) used in the cICAP, the $\text{cdf}_{\text{GM}}/\text{cdf}_{\text{pcWM}}$ (green, dot) and $g_a(\tilde{s}(k))$ (red, solid) of this work.

for GM and for pcWM were first determined. Table 5.3 shows the comparison of the three methods in terms of number of iterations. The fitted $g_a(\tilde{s}(k))$ in $\text{Infomax}_{\text{GM}}$ and in acICA_{GM} converged approximately 25% and 19% faster than using $g(\tilde{s}(k))$ in Infomax and in cICAP, respectively. For the pcWM, $\text{Infomax}_{\text{pcWM}}$ and Infomax were not able to converge. In contrast, $\text{acICA}_{\text{pcWM}}$ performed approximately 6%

Table 5.2.: Parameters for the GM and pcWM of the LAP and PM.

	q	b	r
GM of the LAP	3.8599	32.7346	1.9251
pcWM of the LAP	50.0522	21.4897	4.4419
GM of the PM	0.7761	7.1339	0.9109
pcWM of the PM	$2.9898 \cdot 10^{-6}$	6.1503	$1.682 \cdot 10^{-4}$

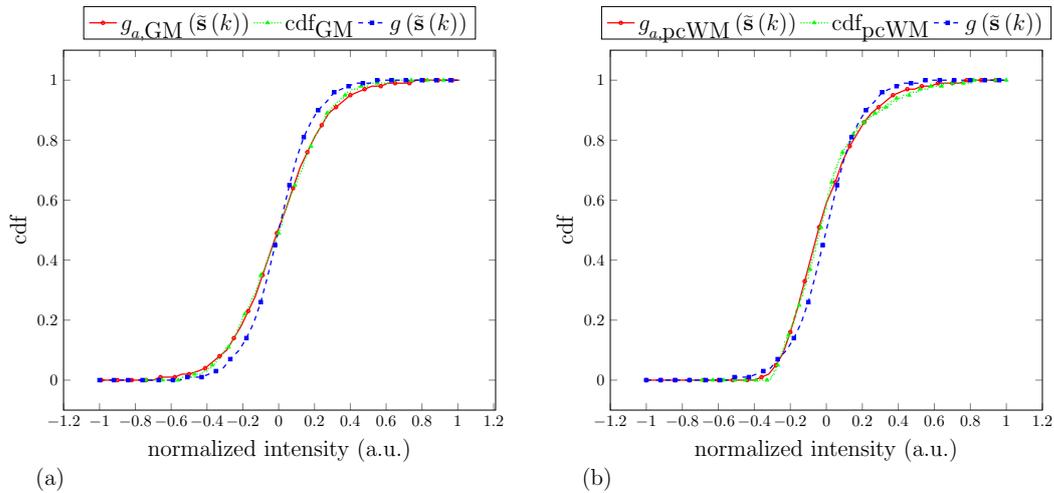


Figure 5.5.: Comparison of cdfs of GM (a) and pcWM (b) generated from the 20 PLI image series of the PM. Illustrated are the default sigmoidal function $g(\tilde{s}(k))$ (blue, dashed) used in the cICAP, the $\text{cdf}_{\text{GM}}/\text{cdf}_{\text{pcWM}}$ (green, dot) and $g_a(\tilde{s}(k))$ (red, solid) of this work.

faster than cICAP on pcWM. Additionally, the standard deviation of the number of iterations decreased by about 73% for acICA_{GM} and by about 13% for $\text{acICA}_{\text{pcWM}}$. A second comparison including the wrGOF of each method is illustrated in Table 5.4. acICA_* , cICAP and Infomax_* applied either on GM or pcWM show a significant higher mean of wrGOF s than the standard Infomax and cICAP applied on the whole histological section. Minor variations between the means and the standard errors for GM and pcWM of acICA_* , cICAP and Infomax_* are found. By comparing $\text{Infomax}_{\text{GM}}$ with acICA_{GM} , a 17% increased signal enhancement is observed. As $\text{Infomax}_{\text{pcWM}}$ and Infomax on pcWM did not converge, $\text{acICA}_{\text{pcWM}}$ and cICAP on pcWM were able to increase the signal enhancement where the mean wrGOF values were found to be at 149.1 and 155.27, respectively.

Table 5.3.: A comparison of number of iterations from acICA, cICAP and Infomax.

Application	mean (min, max)	median	standard deviation
Infomax on GM	103.16 (102, 114)	103	1.66
Infomax _{GM} on GM	76.78 (74, 96)	76	3.62
cICAP* on GM	99.14 (76, 132)	89	17.07
acICA _{GM} [†] on GM	80.04 (71, 89)	79.5	4.57
Infomax on pcWM	—	—	—
Infomax _{pcWM} on pcWM	—	—	—
cICAP** on pcWM	479.88 (299, 500)	500	52.51
acICA _{pcWM} ^{††} on pcWM	451.08 (346, 500)	449.5	45.41

* with optimized parameter: $\eta = 0.15597$, $t = 1.17482 \cdot 10^{-6}$ and $\varepsilon = 0.01$

** with optimized parameter: $\eta = 0.137$, $t = 1.19814 \cdot 10^{-6}$ and $\varepsilon = 0.004$

† with optimized parameter: $\eta = 0.164$, $t = 9.61183 \cdot 10^{-7}$ and $\varepsilon = 0.015$

†† with optimized parameter: $\eta = 0.178$, $t = 1.02125 \cdot 10^{-6}$ and $\varepsilon = 0.01$

Table 5.4.: A comparison of wrGOFs from acICA, cICAP and Infomax.

Application	mean	standard error	confidence interval lower/upper endpoint
Infomax on GM	276.31	3.87	268.52/284.11
Infomax _{GM} on GM	270.36	2.82	264.69/276.04
cICAP* on GM	325.71	2.53	320.61/330.81
acICA _{GM} [†] on GM	326.67	2.57	321.49/331.85
Infomax on pcWM	—	—	—
Infomax _{pcWM} on pcWM	—	—	—
cICAP** on pcWM	155.27	4.54	146.14/164.41
acICA _{pcWM} ^{††} on pcWM	149.1	5.39	138.26/159.93
Infomax on the whole section	19.51	0.08	19.35/19.68
cICAP on the whole section	70.05	24.1	21.62/118.48

Looking at each *wrGOF* value per pixel of one of the 50 repro series for the standard cICAP and the new acICA application (Fig. 5.6), strikingly increased signal enhancement at almost all pixels across the processed tissue is observed. The magnified region of interest in Figure 5.6 provides a closer view to pixels in the GM for

cICAP and acICA.

In order to visualize the effect of acICA on the PLI image series, one pixel of GM, pcWM and WM was chosen. The profile of these pixels are compared with the pixels, which were not processed via acICA (Fig. 5.7). The results of this comparison show that the sinusoidal nature of the PLI signals has been restored for the case of the GM and pcWM (Fig. 5.7(a) and (b)). For the WM, the measured light intensity profile shows a pronounced and undeteriorated sinusoidal characteristic.

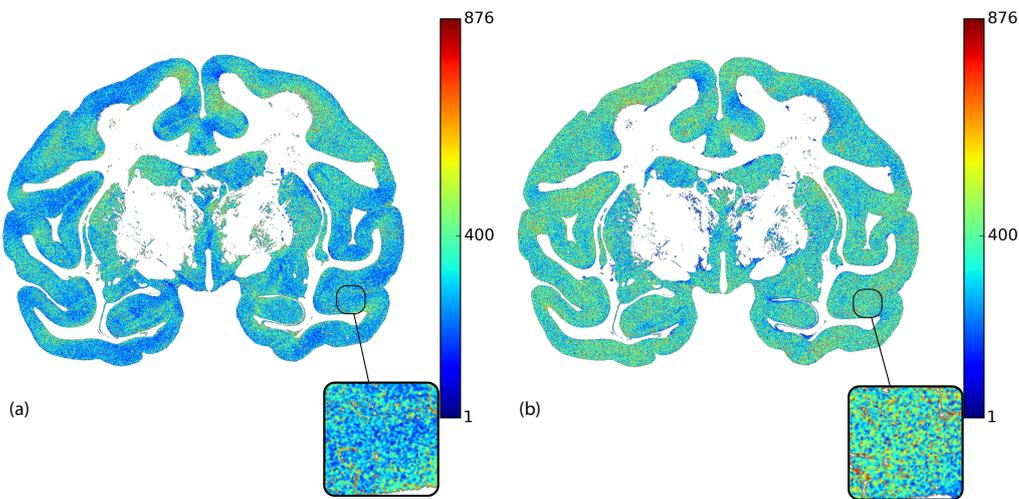


Figure 5.6.: $wrGOF$ values in arbitrary units per pixel generated after standard cICAP applied on the whole histological section (a) and the new acICA applied on GM and pcWM (b). Note that for comparison reasons only the processed pixels of GM and pcWM are illustrated.

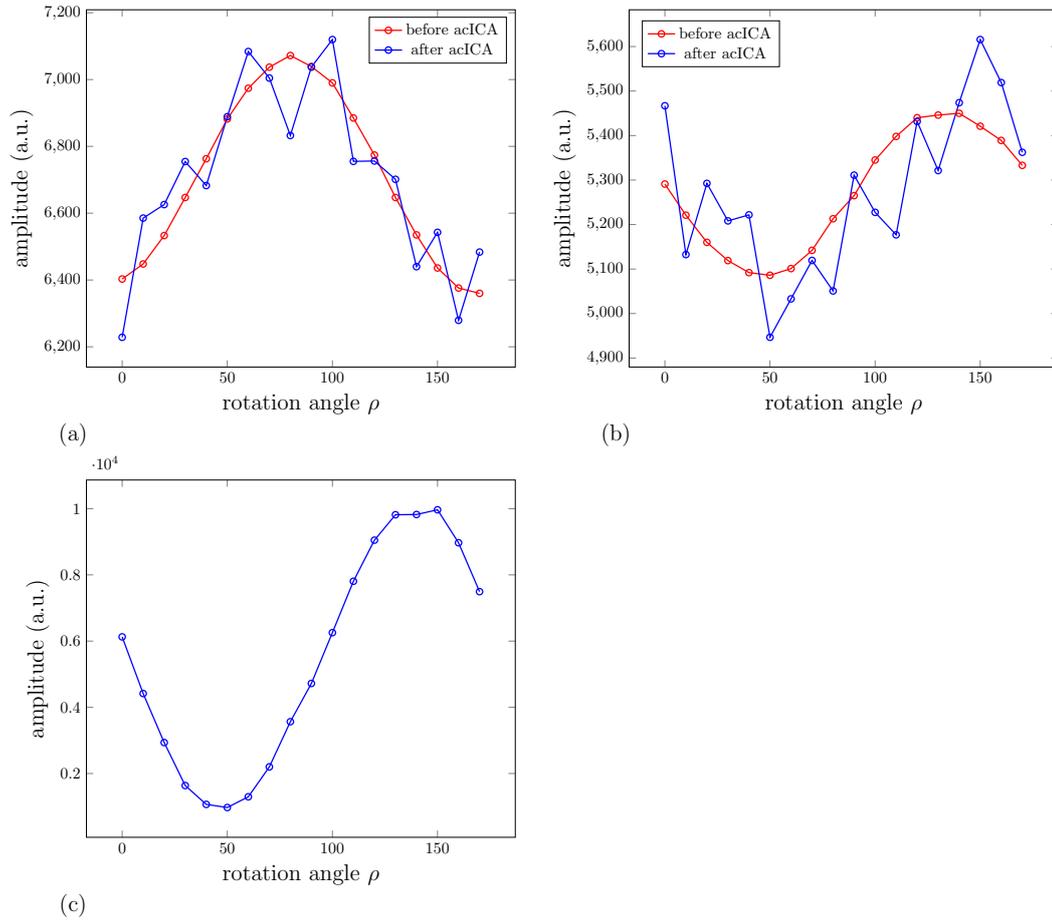


Figure 5.7.: Measured light intensities of one pixel at different rotation angles ρ before (blue) and after *acICA* (red). (a) and (b) illustrate two light intensity profiles of pixels located in GM and pcWM, respectively. The measured signals before *acICA* in (a) and (b) show how sensitive the GM and pcWM signals are to noise. In contrast, strong signals in WM are less affected to noise. One light intensity profile located in WM is shown in (c).

5.4. Discussion

The acICA approach newly introduced here is based on the cICAP, in which prior information of the sinusoidal profile in 3D-PLI is added for a faster extraction of signal sources and automatic identification of components of interest [21]. The new acICA adds further prior information to the existing cICAP, by incorporating GM and pcWM-specific cost functions for the LAP and PM. Hence, two types of source signals may be separated in the algorithm for each of the optical setups. For the remaining pixels of WM no acICA is applied as the level of magnitude of the sinusoidal WM signals is up to several orders larger than the noise level and thus they are less affected by deterioration (Fig. 5.7(c)). Overall, almost 94% of WM pixels provide a pronounced high SNR such that no improvement can be made by applying acICA, where only signal sources were found resulting in a $wrGOF=1$ (Fig. 5.3). Pixels in GM and between the boundaries of GM and WM, however, are afflicted with noise leading to a lower SNR (Fig. 5.7(a) and (b)).

In contrast to acICA, cICAP [21] and Infomax use the sigmoidal function as the nonlinear cost function, which presents a good compromise for a generalized source separation as it is known that many types of source activities follow a sigmoidal distribution [15]. However, an eligible signal separation can only be performed if the data driven cdfs of the applied cost function and the distribution of the signal sources are very similar to each other, yielding a faster cdf-match [103]. The advantage of an automatic identification of components of interest and the strategy to improve the “cdf-matching” has been employed in acICA.

In order to gain information about the relevant distributions of GM and of pcWM, 140 PLI image series of the LAP and 20 PLI image series of the PM were processed. For the LAP the distributions in both regions showed deviant characteristics compared to the sigmoidal function making the latter unsuitable for an optimal signal separation. On the other hand, the distributions in GM and pcWM for the PM present a higher resemblance to the sigmoidal function and differ from their distributions in the LAP. This may be explained by the different in-plane sampling resolution and sensitivities of both optical setups [95], considering that the PM provides more intensity information per LAP pixel. As a result of this work, GM and pcWM-specific cost functions for both optical setups have been generated separately. These cost functions derived from fitting the distributions to $g_a(\tilde{\mathbf{s}}(k))$ and integrated in acICA were tested on the 50 repro series and compared to the well

established Infomax and cICAP applications. The comparison showed that the here introduced acICA method performed 25% faster for the GM and 6% faster for the pcWM than cICAP, whereas the standard deviation decreased for both cases (Table 5.3). Additionally, due to the fact that the initial weight matrix \mathbf{W}_c is initialized with a random orthogonal matrix and the decorrelated data is permuted in each iteration step (see Section 3.5) for each acICA application and for all 50 repro series, local minima and computational biases are avoided during the estimation of an optimal weight matrix \mathbf{W}_{opt} [50]. Hence, the decreased standard deviation of acICA with the fast signal decomposition are indicators for a reliable convergence and reliable denoised PLI image series.

By taking the *wrGOF* of Infomax, cICAP and acICA into account, it can be seen that the algorithms provide approximately the same signal enhancement if they are applied independently on both regions (except for Infomax applied on pcWM), whereas a poor signal enhancement has been observed for the applications of Infomax and cICAP on whole histological sections (Table 5.4). The standard error, which measures the uncertainty of the *wrGOFs*, and the 95% confidence intervals show close similarities for acICA and cICAP indicating that the results of the new approach acICA agrees with the one of cICAP yielding reproducible *wrGOFs* for acICA.

As a result, the adapted cost function in acICA provides a 6 – 25% faster computation and tissue-specific signal enhancement with respect to GM and pcWM for each histological section.

6. Parameter Optimization of the Adapted Constrained Independent Component Analysis

Optimization or tuning of the intrinsic parameters η , t and ε (see Section 3.5) plays an important role in acICA, as these parameters affect the accuracy of signal separation as well as the convergence of the algorithm. Since the right choice of intrinsic parameters is dependent on the input images and may vary for each histological section, the question arises if the parameters are needed to be found for each histological section separately.

Breuer et al. [21] used a brute-force approach for finding the optimal parameters in cICAP, where multiple cICAP runs were performed on only one PLI image series. Subsequently, the optimal parameters were incorporated in cICAP for further applications on histological sections. However, it is still unknown how different histological sections across a full brain affect the optimal selection of the intrinsic parameters. Additionally, the brute-force approach is a naive and computational intensive optimization technique, where the entire parameter space must be searched for the right parameter candidates. Therefore, to speed up the search an alternative optimization technique is necessary, which needs less iterations than the brute-force approach and, at the same time, finds a local maximum/minimum throughout the parameter space.

This chapter goes a new way, focusing on finding the right parameter candidates for acICA at histological sections across a full brain by using the downhill simplex algorithm [87]. The ascertained parameters are evaluated as to their distribution across all histological sections.

For the sake of clarity and comprehension, the definition of variables and abbreviations for the following sections are summarized in Table 6.1.

Table 6.1.: Definition of variables and abbreviations in this chapter.

	definition	reference
$\widehat{\mathbf{A}}$	Orthogonal mixing matrix	see Chapter 3
$\widehat{a}^*(\rho)$	Basis vector with prior information	see Chapter 3
ϵ	threshold value	see Chapter 3
η	Confidence value	see Chapter 3
$f^{(k)}$	Function value of $x^{(k)}$	
$f(u_\rho)$	Theoretically expected function	see Chapter 2
κ_c	Contraction parameter	
κ_e	Expansion parameter	
κ_r	Reflection parameter	
κ_s	Shrinkage parameter	
t	tolerance value	
$wrGOF$	Weighted relative goodness-of-fit	see Chapter 2
$x^{(k)}$	Vertex of k^{th} iteration step	
acICA	Adapted constrained Independent Component Analysis	see Chapter 5
cICAP	Constrained Independent Component Analysis for 3D-PLI	see Chapter 3
DSA	Downhill Simplex Algorithm	
GM	Gray Matter	
LAP	Large Area Polarimeter	see Chapter 2
MSE	Mean Square Error	see Chapter 3
pcWM	Percentiles of White Matter	

6.1. Methods

The *Downhill Simplex Algorithm* (DSA), also referred to as the Nelder-Mead algorithm in the literature, is a widely used method for nonlinear unconstrained optimization [87, 108]. This method belongs to the class of direct search algorithms and requires only function values, without the need for gradient information. The DSA begins with a geometric figure, which is defined by its $n + 1$ vertices $x^{(k)}$, with k denoting the iteration steps. For each vertex the associated function value is calculated and sorted, such that

$$f_1^{(k)} \leq f_2^{(k)} \leq \dots \leq f_{n+1}^{(k)}, \quad (6.1)$$

where the function value $f_i^{(k)}$ describes $f(x_i^{(k)})$. The vertex with the highest function value $f_{n+1}^{(k)}$ is replaced with a new point of the form

$$x = (1 + \kappa)\bar{x} - \kappa x_{n+1}, \quad (6.2)$$

with $\bar{x} = \sum_{i=1}^n \frac{x_i}{n}$ being the centroid. The value κ can be interpreted as one of the parameters

$$\kappa_r > 0, \kappa_e > 1, 0 < \kappa_c < 1 \text{ and } 0 < \kappa_s < 1, \quad (6.3)$$

with $\kappa_e > \kappa_r$, depending on the function value $f_i^{(k)}$. The choices used for these parameters are typically $\kappa_r = 1$, $\kappa_e = 2$, $\kappa_c = 0.5$ and $\kappa_s = 0.5$. These parameters define the rules for reflection, expansion, contraction and shrinkage [74], respectively. The algorithm terminates if $f_{n+1}^{(k)} - f_1^{(k)}$ is sufficiently small or the total number of function evaluations has been exceeded. Lagarias et al. [74, 75] published a theoretical analysis of DSA, proving the convergence to a unique minimum for dimensions one and two.

With the requirement to maximize the *wrGOFs* for all pixel locations in GM and pcWM for the LAP, the intrinsic parameters η and t of acICA are estimated by DSA. As the parameter ε is determined by the *Mean Square Error* ($\text{MSE} = \frac{1}{N} \sum_{\rho} \left[\hat{a}_j^*(\rho) - f_j(u_{\rho}) \right]^2$, see Section 3.5) between components representing signals of interest and noise components, this parameter was optimized without DSA. Thus, the DSA used here constructs a triangle simplex in \mathbb{R}^2 with a starting vertex $\eta = 0.16$ and $t = 2.2 \cdot 10^{-7}$ [21] and initiation step size of 0.05. For iteratively computing a function value of a new vertex, Equation 6.2 is applied, resulting in a new pair of parameters η^* and t^* . These parameters are then used in acICA for processing GM and pcWM, respectively, until optimal η_{opt} and t_{opt} are found. As a control measure the *wrGOF* values are computed after each acICA application and a minimal *wrGOF* value was checked. The minimal *wrGOF* serves as a function value for DSA.

In order to evaluate the distributions of the intrinsic parameters of acICA, DSA was performed individually on GM and pcWM of the 140 PLI image series, resulting in optimal parameters η_{opt} and t_{opt} for both GM and pcWM of each PLI image series. Subsequently, the evolution of the MSE values for all iteration steps at each acICA run of the 140 PLI image series was used to determine the optimal stopping criteria

ε . For this, the means of both estimated parameters η_{opt} and t_{opt} were used and fixed for each acICA run.

6.2. Results

The identification of the optimal parameter candidates is important for the signal decomposition in acICA, where the weight update of $\hat{\mathbf{A}}_c$ is optimized by means of the incorporation of prior information, which models the expected signal in PLI by utilizing the Jones calculus (Eq. 2.8). Therefore, the optimal parameters η_{opt} and t_{opt} were determined using DSA on GM and pcWM of the 140 PLI image series.

As a starting point for each DSA run the optimal parameters reported by Breuer et al. [21] were used. Figure 6.1(a) and (b) provide a summary of these results. The confidence values for GM and pcWM settle at values close to 0.16, whereas the threshold values for GM can be clearly distinguished from the threshold values for pcWM. Table 6.2 shows the mean, median and standard deviation of the confidence and threshold values gained from the DSA. For simplicity, if only the means of η_{GM} and η_{pcWM} are considered, then a marginal difference of $1.2 \cdot 10^{-3}$ is observed. On the other hand, the mean of t_{GM} and t_{pcWM} have a significant visible difference of $8.503 \cdot 10^{-7}$ (Fig. 6.1(b)).

Table 6.2.: Mean, median and standard deviation of the optimized parameters η and t for GM and pcWM.

	mean	median	standard deviation
η_{GM}	0.164	0.1637	$8.2 \cdot 10^{-3}$
η_{pcWM}	0.1628	0.16	$7.5 \cdot 10^{-3}$
t_{GM}	$1.0739 \cdot 10^{-6}$	$1.0746 \cdot 10^{-6}$	$5.4 \cdot 10^{-8}$
t_{pcWM}	$2.2357 \cdot 10^{-7}$	$2.2 \cdot 10^{-7}$	$7.5 \cdot 10^{-9}$

By using the mean of η_{GM} and η_{pcWM} and respectively the mean of the threshold values, the stopping criterion ε , controlling the number of iterations, is determined for each acICA run. For this purpose, the largest and smallest error, which is expressed by the MSE, found in $\hat{a}_j^*(\rho)$ of $\hat{\mathbf{A}}_c$ is investigated. For all iterations in acICA the maximal MSE of all identified signal components and the minimal MSE of all noise components across the columns of $\hat{\mathbf{A}}_c$ were observed.

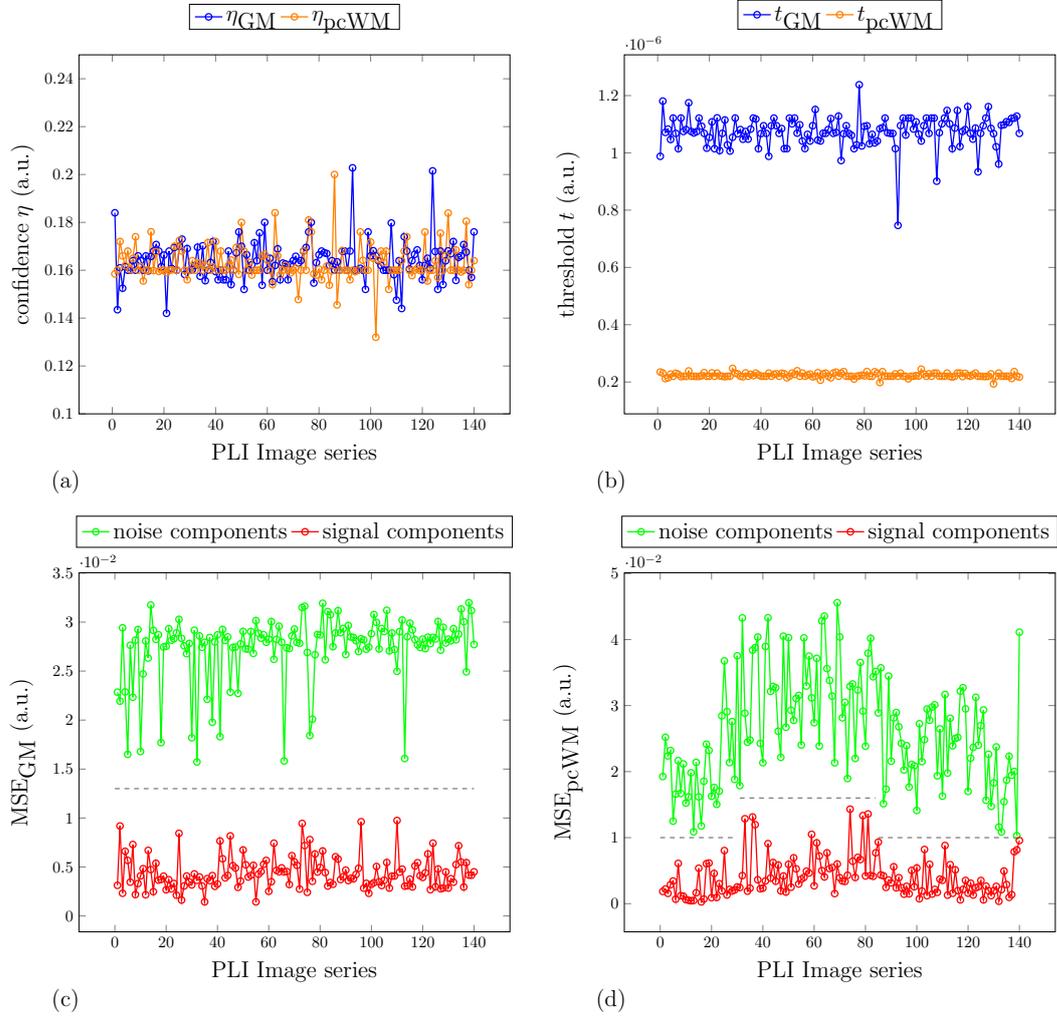


Figure 6.1.: Parameter benchmark of the confidence value η , the threshold value t and the stopping criterion ε . (a) and (b) show the resulting η_{opt} and t_{opt} of 140 PLI image series after computing DSA for GM (blue) and for pcWM (orange). The mean of the maximal and minimal MSE for GM and pcWM is shown in (c) and (d), respectively. Note that the minimal MSE is plotted for noise components (green), whereas the maximal MSE is plotted for signal components (red). Furthermore, the dashed gray line shows the threshold, where both components can be clearly identified.

Figure 6.1(c) and (d) depict the mean of the maximal and minimal MSE for GM and pcWM, respectively. Throughout all PLI image series the maximal MSE for GM was found to be well below 0.013 yielding an optimal parameter $\varepsilon_{\text{GM}} = 0.013$. In contrast, the distinction of the maximal and minimal MSE for pcWM can not be classified by one threshold value across the PLI image series. Thus, the optimization of the optimal parameter $\varepsilon_{\text{pcWM}}$ is performed on three partitions of the 140 PLI image series. In the first partition ranging from the first to 31st PLI image series, $\varepsilon_{\text{pcWM},1}$ was found at a value of 0.01. For the second partition ranging from the 32nd to 85th PLI image series $\varepsilon_{\text{pcWM},2}$ was set to 0.016. The last partition range from the 86th to the 140th PLI image series in which $\varepsilon_{\text{pcWM},3} = 0.01$.

6.3. Discussion

The aim of this chapter is to document and discuss the optimized parameters of acICA across the whole vervet brain. It is of importance to acquire information about the distribution of the determined parameters for an accurate signal decomposition throughout the 140 PLI image series. This may answer the question if unique parameters have to be found for each histological section.

For this purpose DSA was applied on GM and pcWM of all PLI image series yielding 140 values for each η and t . While the parameters η_{GM} and η_{pcWM} differ only marginally from each other with values of 0.164 and 0.1628 across all PLI image series and are thus less affected by the different tissue types, the parameter t showed noticeable differences between GM and pcWM. It is noted that the parameter t is the tolerance value where the entries of $\hat{a}_j^*(\rho)$ are fixed and if MSE is below t , $\hat{a}_j^*(\rho)$ and $f_j(u_\rho)$ become similar leading to the identification of the original sinusoidal profile of the 3D-PLI signal. Following this, t has a strong linkage to the underlying sources during signal separation and thus it may be more affected by the choice of the input data, i.e., GM and pcWM. Thus, when processing GM the parameters need to be set to $\eta_{\text{GM}} = 0.164$ and $t_{\text{GM}} = 1.0739 \cdot 10^{-6}$, whereas when processing pcWM the parameters need to be set to $\eta_{\text{pcWM}} = 0.1628$ and $t_{\text{pcWM}} = 2.2357 \cdot 10^{-7}$ (Table 6.2). Additionally, the means of t_{GM} show also a distinct difference to the published value of $t = 2.2 \cdot 10^{-7}$, whereas the means of η_{GM} and η_{pcWM} correspond roughly to the published value of $\eta = 0.16$ [21]. It is noted, however, that most DSA runs on pcWM exceeded the total number of function evaluations yielding optimal parameters to be equal to the initialized parameters $\eta = 0.16$ and $t = 2.2 \cdot 10^{-7}$. A similar noticeable

difference between GM and pcWM is also observed for the stopping criterion ε . The stopping criterion ε , which controls the number of iterations and has the condition of $\varepsilon \gg t$, is dependent on the distance between the maximal MSE of the sources of interest and the minimal MSE of noise components (Fig. 6.1). Therefore, ε is also dependent on the combination of η and t , which regulates the distance of the minimal and maximal MSE. By fixing the optimal parameter η and t for all PLI image series and evaluating the DSA results, it was observed that ε_{GM} can be set to 0.013 throughout the histological sections, whereas $\varepsilon_{\text{pcWM}}$ has to be determined for three partitions of the histological sections. While ε_{GM} and $\varepsilon_{\text{pcWM},2}$ differ slightly from the published value of 0.01, $\varepsilon_{\text{pcWM},1}$ and $\varepsilon_{\text{pcWM},3}$ are equal to $\varepsilon = 0.01$ [21]. Hence, the histological sections laying approximately in the frontal and occipital lobe need an $\varepsilon_{\text{pcWM}} = 0.01$. For the histological sections laying approximately in the temporal lobe $\varepsilon_{\text{pcWM}}$ needs to be set to 0.016. The three different partitions for $\varepsilon_{\text{pcWM}}$ from the frontal to the temporal lobe and from the temporal to the occipital lobe, respectively, may be explained by the increasing and then decreasing of number of pixels laying in the 4th percentile of WM.

7. A New Parallelization Concept for Adapted Constrained Independent Component Analysis

The analysis of PLI image series acquired with the PM poses two major difficulties: the involvement of high dimensional data and the usually low *Signal-to-Noise Ratio* (SNR) for birefringence signals with weak signal strengths. Huge amounts of memory and computing power are needed to analyze brain signals with ICA. Due to limited time and resources, these difficulties render a purely sequential procedure of ICA impossible. Various modalities in neuroscience, e.g., EEG, MEG and fMRI, share the same difficulties. However, in recent years, research in EEG, MEG and fMRI has overcome the constraint of sequential execution and memory limitations of ICA. One approach in EEG analysis is to use parallel computing clusters with the potential to exploit two different ICA implementations: FastICA relying on MPI, and Infomax using an *Open Multi-Processing* (OpenMP) implementation [69]. While the Infomax algorithm is the relevant one for our purposes, it was observed that the approach with OpenMP for Infomax of Keith et al. [69] shows lack of parallelism and scalability. A different way of introducing parallelism is the usage of SIMD architectures as proposed by Raimondo et al. [94], where one GPU was used for ICA calculation on EEG data. It was observed that the use of a common shared memory and one GPU could reduce the processing time by a factor of 25. While this may be only valid for smaller data sets that fit the GPU's onboard memory, very large data sets, which are common for the PM, would lead to a quite ineffective procedure when performed on one GPU only. Another strategy was proposed by Boubela et al. [19], where MR data of 300 subjects are distributed on multiple machines for processing ICA independently. Subsequently, a group-level analysis is applied, referred to as group ICA, which combines single subject ICA results [19]. By considering the current state of research of parallelized ICA conducted on different modalities in neuroscience, a new concept to parallelize acICA in 3D-PLI for

PM is introduced in this thesis. The new concept of parallelized acICA for PM, referred to as pICAP, needs to deal with huge amount of data and, at the same time, needs to provide a fast analysis of the PLI image series using the GPU-enhanced supercomputer JuDGE.

The development of a new parallelized ICA specifically designed for 3D-PLI is subject of this chapter. In Section 7.1 the concept and the implementation of pICAP is presented. Additionally, an approach is proposed which uses massive parallel processors in the GPU for linear algebra operations in pICAP. Finally, pICAP is compared to the sequential version acICA and a runtime analysis is performed. In Section 7.3 a summary of achievements is discussed.

For the sake of clarity and comprehension, the definition of variables and abbreviations for the following sections are summarized in Table 7.1.

Table 7.1.: Definition of variables and abbreviations in this chapter.

	definition	reference
$\widehat{\mathbf{A}}$	Orthogonal mixing matrix	see Chapter 3
$\widehat{\mathbf{a}}^*(\rho)$	Basis vector with prior information	see Chapter 3
α_{FP}	False positive rate	
α	Significance level	
B_I	Regularized incomplete beta function	
\mathbf{C}	Covariance matrix	
c	Iteration step	
clusterIdxs	Array of cluster indices	
\mathbf{D}	Diagonal matrix with eigenvalues	
\mathbf{D}_x	Whitening matrix	see Chapter 3
\mathbf{E}	Matrix containing the eigenvectors	
γ	Similarity	
$g_a(\widetilde{\mathbf{s}})$	Adapted cost-function	see Chapter 5
ν	Degrees of freedom	
p	Probability used to decide statistical significance	
\mathbf{p}	Array of p-values	
$\bar{\mathbf{p}}$	Mean of p-values for each cluster	
\mathbf{s}	Vector of source signals	see Chapter 3
τ	Learning rate	

\mathbf{W}^*	Estimated unmixing matrix	
\mathbf{W}_0	Bias weight	see Chapter 3
$wrGOF$	Weighted relative goodness-of-fit	see Chapter 2
\mathbf{x}	Vector of recorded sensor signals	see Chapter 3
acICA	Adapted constrained Independent Component Analysis	see Chapter 5
CPU	Central Processing Unit	
CUBLAS	CUDA Basic Linear Algebra Subroutines	
EEG	Electroencephalography	
fMRI	Functional Magnetic Resonance Imaging	
GB	Gigabyte	
GM	Gray Matter	
GPU	Graphics Processing Unit	see Chapter 4
HDF5	Hierarchical Data Format version 5	
ICA	Independent Component Analysis	see Chapter 3
JuDGE	Jülich Dedicated GPU Environment	see Chapter 4
MB	Megabyte	
MEG	Magnetoencephalography	
MPI	Message Passing Interface	see Chapter 4
OpenMP	Open Multi-Processing	
pcWM	Percentiles of White Matter	
pICAP	Parallelized Independent Component Analysis for 3D-PLI	
PM	Polarizing Microscope	see Chapter 2
SIMD	Single Instruction - Multiple Data	see Chapter 4
SNR	Signal-to-Noise Ratio	
3D-PLI	3D-Polarized Light Imaging	see Chapter 2

7.1. Concept Development, Implementation and Procedure

pICAP is a new method which incorporates all properties of acICA and performs a related group ICA on disjoint partitions of a PLI image series acquired with the PM. In general, group ICA is applied on recorded data from many subjects where sufficient similar components are found [23, 36]. The new approach is related to the method introduced by Himberg et al. [50], where the components are clustered by repetitively performing single ICA runs with different starting conditions. As

the components may be related to signals of interest or artifacts, a component selection and sorting is mandatory. Hyvärinen et al. [54, 58] proposed therefore an inter-subject consistency of components or mixing matrices, where ICA is performed separately for all subjects in either EEG, MEG or fMRI. The estimated components or mixing matrices are then clustered. Reliable source signals are found based on principles of statistical estimation theory, where similar sources are defined by the consistency of the columns of the mixing matrix or the spatial components across all subjects. In a last step, consistent sources are clustered together, where a cluster is restricted to contain only one component from each subject. In case of pICAP, the subjects can be represented by the disjoint partitions of PLI image series where the similarity of the estimated components is defined based on the columns $\hat{a}_j^*(\rho)$ of the mixing matrix $\hat{\mathbf{A}}_c^*$, resulting in clusters that represent source and artifact components, respectively (see Section 3.5). The combination of a related group ICA and an inter-subject consistency is incorporated in pICAP.

7.1.1. Development of a Parallelized ICA Concept

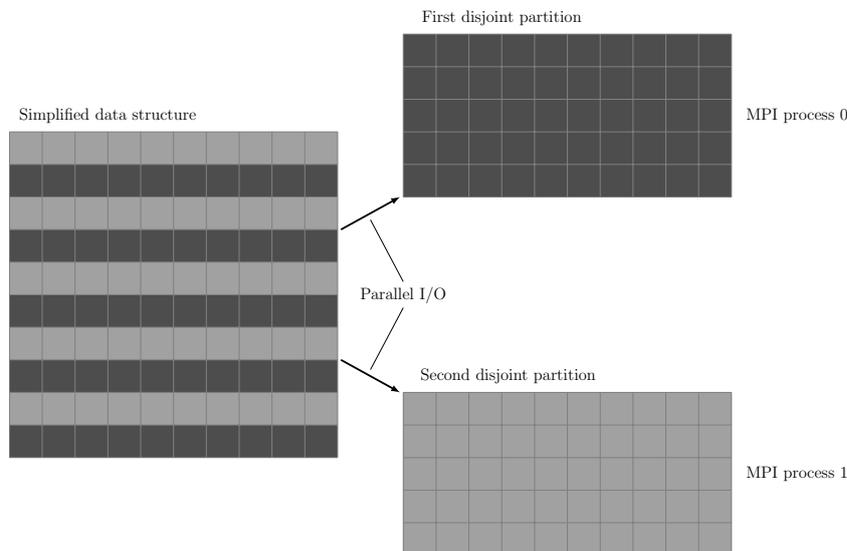


Figure 7.1.: Exemplary procedure to read data in a cyclic manner of one large data set resulting in a geometric decomposition (see Section 4.3). The 2 MPI processes hold a disjoint data partition of one large data set.

Figure 7.2 schematically explains the new concept of pICAP. The PLI image series and the corresponding mask, which are stored in a *Hierarchical Data Format ver-*

sion 5 (HDF5) [46, 106], are read simultaneously in a row cyclic manner resulting in disjoint data partitions in parallel (Fig. 7.1). Subsequently, the segmentation, centering and decorrelation of each partition is performed. On each decorrelated partition a single acICA is performed resulting in a parallel execution of acICA on the different partitions loaded. Each acICA generates at every iteration step one estimated unmixing matrix \mathbf{W}_c^* . In contrast to a standard group ICA, the estimated unmixing matrix \mathbf{W}_c^* is passed to the next acICA, which updates \mathbf{W}_c^* dependent on the present partition.

Looking at the update of the unmixing matrix \mathbf{W}_c^* , which is passed consecutively to the next acICA on each iteration step, a resemblance to the sequential procedure of acICA can be observed. The update is performed until convergence criteria are met. For simplicity reasons, a detailed description of the convergence criteria and the update of the mixing matrix is provided later in this chapter.

The subsequent method for inter-subject consistency [54, 58] uses the covariance matrix \mathbf{C} , which is defined as

$$\mathbf{C} = \frac{1}{nr} \sum_{i=1; k=1}^{i=n; k=r} \hat{a}_{ik}^* \cdot \hat{a}_{ik}^{*T}, \quad (7.1)$$

where \hat{a}_{ik}^* , $i = 1, \dots, n$, $k = 1, \dots, r$, are the columns of the mixing matrix $\hat{A}_k^* = [\hat{a}_{1k}^*, \hat{a}_{2k}^*, \dots, \hat{a}_{nk}^*]$ of r partitions and n independent components. From the eigenvalue decomposition of

$$\mathbf{C} = \mathbf{E} \cdot \mathbf{D} \cdot \mathbf{E}^T, \quad (7.2)$$

the diagonal matrix \mathbf{D} and the orthogonal matrix \mathbf{E} containing the eigenvectors are obtained. Here, the full rank of \mathbf{D} and \mathbf{E} was used. With the matrices \mathbf{E} and \mathbf{D} the similarities γ of \hat{a}_{ik}^* can be defined as

$$\gamma_{ij,kl} = \frac{|\hat{a}_{ik}^{*T} \cdot \mathbf{R} \cdot \hat{a}_{jl}^*|}{\sqrt{\hat{a}_{ik}^{*T} \cdot \mathbf{R} \cdot \hat{a}_{ik}^*} \sqrt{\hat{a}_{jl}^{*T} \cdot \mathbf{R} \cdot \hat{a}_{jl}^*}}, \quad \text{where} \quad (7.3)$$

$$\mathbf{R} = \mathbf{E} \cdot \mathbf{D}^{-1} \cdot \mathbf{E}^T, \quad (7.4)$$

for all $i, j = 1, \dots, n$ and $k, l = 1, \dots, r$, $k \neq l$. The goal here is to use a statistical test to determine if the different partitions have significantly similar basis vectors

7. A New Parallelization Concept for Adapted cICAP

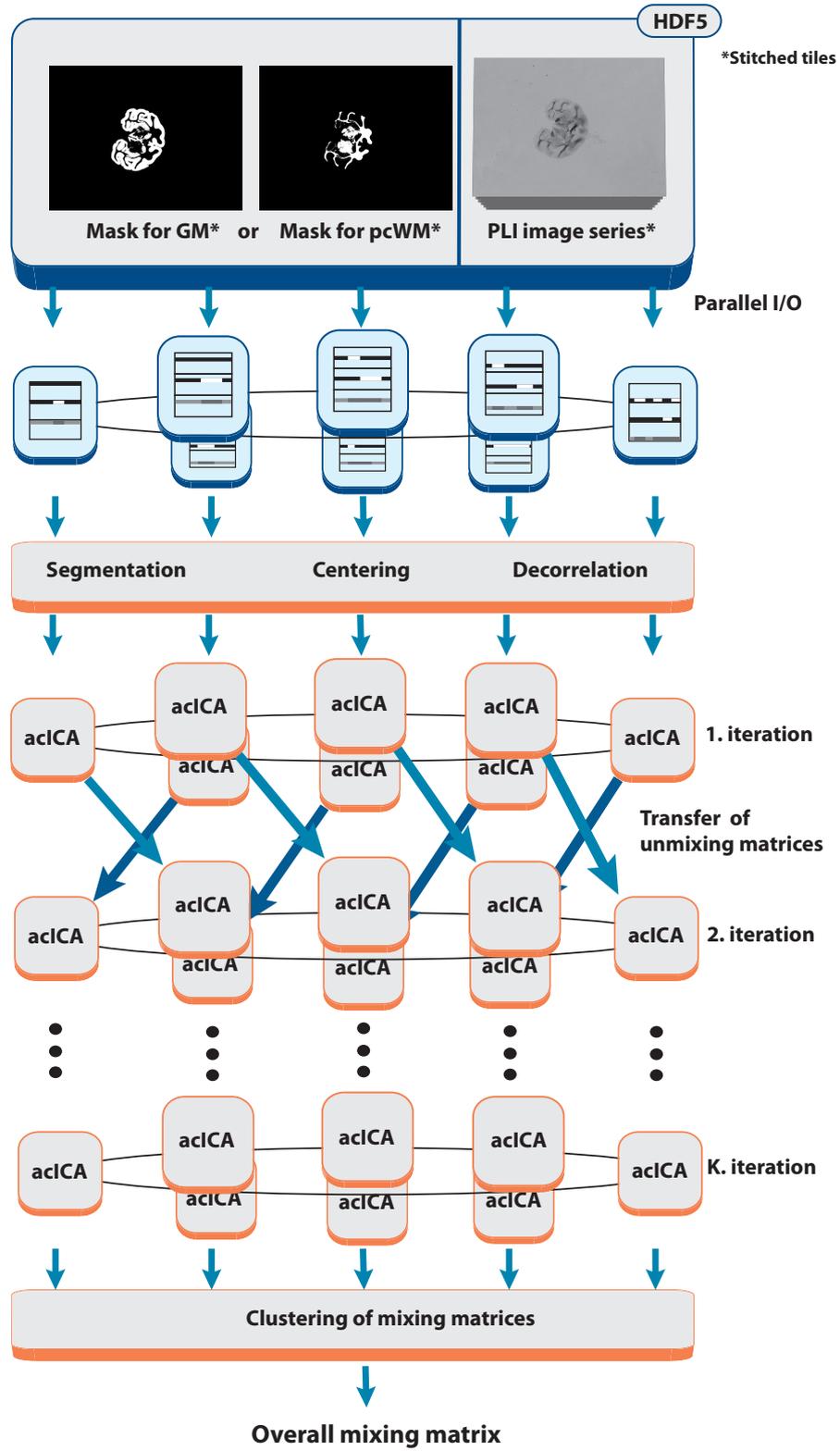


Figure 7.2.: Scheme of the estimation procedure for generating an overall mixing matrix $\hat{\mathbf{A}}_{tot}^*$ in pICAP.

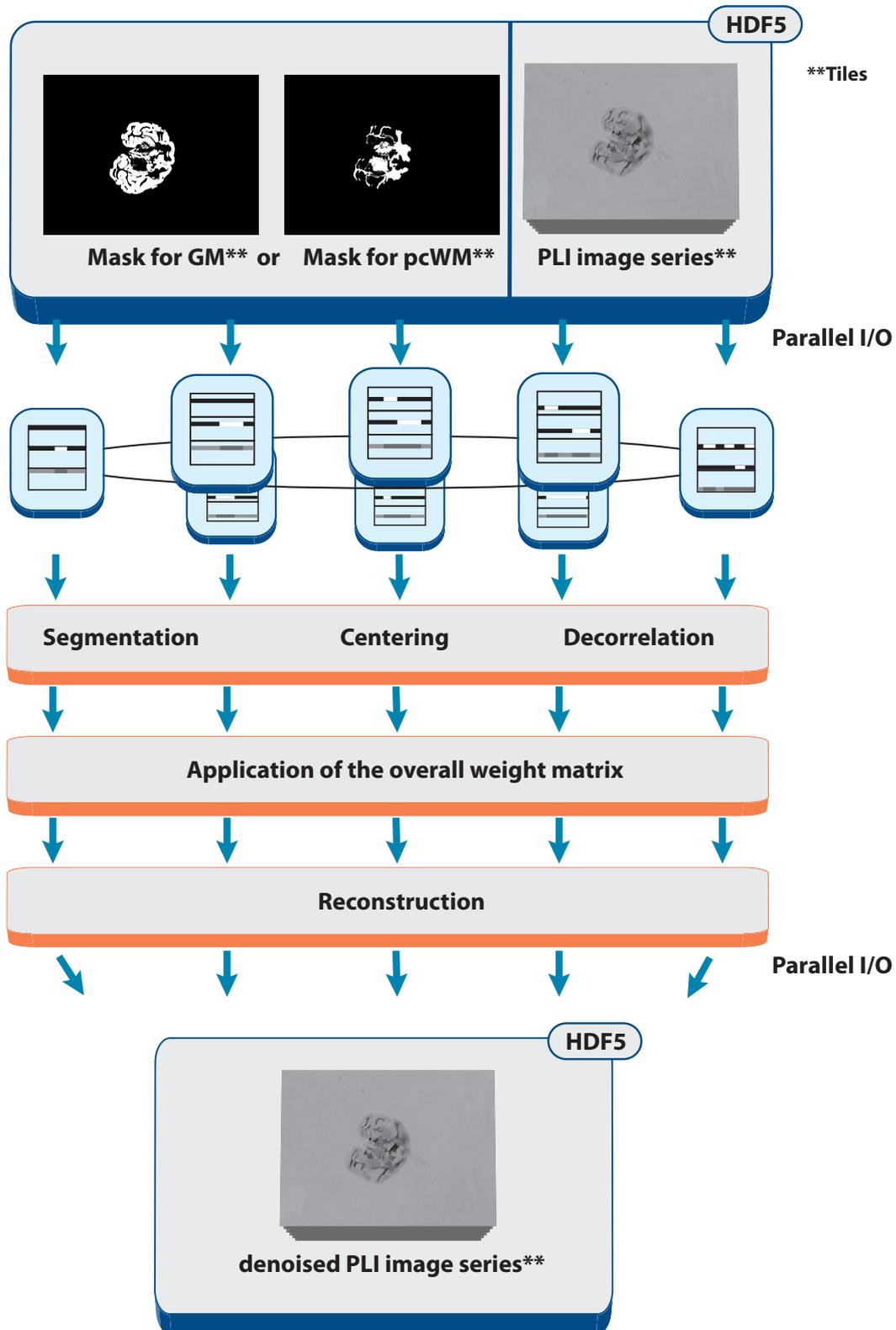


Figure 7.3.: Scheme of the reconstruction of the identified source components back into the PLI space. The reconstructed data is saved again as tiled PLI image series.

\widehat{a}_{ik}^* . The here assumed null hypothesis models the case where \widehat{a}_{ik}^* of the different partitions have no similarities. Hyvärinen et al. [54] showed that the distribution of the statistical test under the null hypothesis follows a beta distribution with the parameters $\frac{1}{2}$ and $\frac{n-1}{2}$. By knowing the distribution of the statistical test the similarities γ can be transformed to p-values. The p-values are calculated with

$$p_{ij,kl} = 1 - B_I \left(\gamma_{ij,kl}^2, \frac{1}{2}, \frac{n-1}{2} \right), \quad (7.5)$$

where $B_I \left(\gamma_{ij,kl}^2, \frac{1}{2}, \frac{n-1}{2} \right)$ describes the regularized incomplete beta function. Thus, the null hypothesis is rejected if the p-values $p_{ij,kl}$ are less than a significance level α and accepted otherwise. By means of the significant p-values clusters are defined [54].

This method clusters similar components of $\widehat{\mathbf{A}}_c^*$ from all partitions yielding the calculation of one unmixing matrix $\mathbf{W}_{tot}^* = \left(\widehat{\mathbf{A}}_{tot}^{*-1} \cdot \mathbf{D}_x \right)^{-1}$ (see Section 3.5). In the same manner as in acICA, pICAP is applied separately on GM and pcWM of a PLI image series acquired with the PM. Thus, two different unmixing matrices $\mathbf{W}_{tot,GM}^*$ and $\mathbf{W}_{tot,pcWM}^*$ are generated. For the estimation of the unmixing matrices the corresponding fitted cost functions $g_a(\widetilde{\mathbf{s}}(k))$ of GM and pcWM for PM were used (see Chapter 5). In order to manage an accurate signal separation and convergence the optimal parameter η_{opt} , t_{opt} and ε_{opt} for GM and pcWM of the LAP were used in the pICAP algorithm (see Chapter 6).

For avoiding that doubled or fourfold information are processed due to the 30% overlap area of the acquired tiles in the PM, the calibrated tiles were firstly stitched together resulting in stitched PLI image series. Analogously, the generated tiled masks are stitched creating a stitched mask. The stitched masks of GM or pcWM with the stitched PLI image series provide the input of pICAP.

Since each tile may be afflicted with noise and dust particles, which may affect the result of the stitching procedure, the estimated unmixing matrices $\mathbf{W}_{tot,GM}^*$ and $\mathbf{W}_{tot,pcWM}^*$ are used to denoise the tiles of the same histological section (Fig. 7.3). For this step the PLI image series of the tiles and the corresponding tiled masks are read simultaneously in a row cyclic manner. Analogously, the pre-processing steps, i.e., segmentation, centering and decorrelation, are performed. The weight matrices are applied individually on the decorrelated partitions. After keeping components of interest in $\widehat{\mathbf{A}}_{tot,GM}^*$ and $\widehat{\mathbf{A}}_{tot,pcWM}^*$, respectively, and rejecting noise and artifact components, the denoised partitions are reconstructed back into PLI space. Subse-

quently, the reconstructed partitions are written simultaneously as tiled PLI image series in HDF5. The denoised tiles are then stitched again and provide the basis for further analysis.

7.1.2. Implementation

In the following the pICAP algorithm is shown in pseudocode in order to give a detailed view and to emphasize relevant parts of the method.

Algorithm 3 Parallelized Independent Component Analysis for 3D-PLI (pICAP)

```

1: MPI initialization
2:  $comm \leftarrow \text{MPI\_COMM\_WORLD}$ 
3:  $rank \leftarrow \text{MPI\_Comm\_rank}(comm)$ 
4:  $nprocs \leftarrow \text{MPI\_Comm\_size}(comm)$ 

5: function LOADDATA( $rank$ )
6:    $\mathbf{x}_{rank} \leftarrow \text{HDF5ReadPLIImageSeries}(rank)$ 
7:    $\mathbf{m}_{rank} \leftarrow \text{HDF5ReadMask}(rank)$ 
8:   return  $\mathbf{x}_{rank}, \mathbf{m}_{rank}$ 
9: end function

10: function PREPROCESSING( $\mathbf{x}_{rank}, \mathbf{m}_{rank}$ )
11:    $\mathbf{idx}_{\text{dimX}}, \mathbf{idx}_{\text{dimY}} \leftarrow \text{GetIndicesGreaterZero}(\mathbf{m}_{rank})$ 
12:   Loadbalancing( $\mathbf{idx}_{\text{dimX}}, \mathbf{idx}_{\text{dimY}}$ )
13:    $\mathbf{x}_{rank, \text{masked}} \leftarrow \text{Segmentation}(\mathbf{x}_{rank}, \mathbf{idx}_{\text{dimX}}, \mathbf{idx}_{\text{dimY}})$ 
14:   Compute total sample mean  $\bar{\mathbf{x}}$  of  $\mathbf{x}_{rank, \text{masked}}$ 
15:    $\mathbf{x}_{rank, \text{centered}} \leftarrow \text{Centering}(\mathbf{x}_{rank, \text{masked}}, \bar{\mathbf{x}})$ 
16:   Compute total covariance matrix
17:    $\mathbf{x}_{rank, \text{whitened}} \leftarrow \text{Whitening}(\mathbf{x}_{rank, \text{centered}})$ 
18: end function

```

```

19:  $q_*, b_*, r_* \leftarrow \text{GetACICAPParameterGMOOrPCWM}()$ 
20:  $\eta_*, t_*, \varepsilon_* \leftarrow \text{GetOptParameterGMOOrPCWM}()$ 
21:  $right \leftarrow (rank + 1) \bmod nprocs$ 
22:  $left \leftarrow (rank - 1 + nprocs) \bmod nprocs$ 
23:  $\mathbf{W}_0 \leftarrow \mathbf{D}_W \cdot \mathbf{W}_\Pi$ 
24:  $\mathbf{W}_0^* \leftarrow \mathbf{I}$ 
25:  $\tau_0 \leftarrow 0.01$ 
26:  $c \leftarrow 0$ 
27: while  $\Delta \mathbf{W}_c > 10^{-10}$  AND  $c \leq 500$  do
28:    $\mathbf{x}_{rank, \text{whitened}} \leftarrow \text{Permute}(\text{SeedPoint}(c), \mathbf{x}_{rank, \text{whitened}})$ 
29:   if  $c > 0$  then
30:     Recieve unmixing matrix  $\mathbf{W}_{c, right}$ , bias weight matrix  $\mathbf{W}_{0, right}^*$ 
      and learning rate  $\tau_{right}$  from left neighbor
31:      $\mathbf{W}_{c, left}, \mathbf{W}_{0, left}^*, \tau_{left} \leftarrow \mathbf{W}_{c, right}, \mathbf{W}_{0, right}^*, \tau_{right}$ 
32:   else
33:      $\mathbf{W}_{c, left}, \mathbf{W}_{0, left}^*, \tau_{left} \leftarrow \mathbf{W}_0, \mathbf{W}_0^*, \tau_0$ 
34:   end if
35:    $\mathbf{W}_{c, right}, \mathbf{W}_{0, right}^*, \tau_{right} \leftarrow \text{Infomax}(\mathbf{x}_{rank, \text{whitened}}, \mathbf{W}_{c, left}, \mathbf{W}_{0, left}^*, \tau_{left}, q_*, b_*, r_*)$ 
36:   Incorporation of prior information by using  $\eta_*, t_*$  and  $\varepsilon_*$  (see Algorithm 2)
37:   Gather  $\hat{\mathbf{A}}_{c, right}^*$  of all MPI processes
38:    $\text{clusterIdxs}, \mathbf{p} \leftarrow \text{InterSubjectConsistency}([\hat{\mathbf{A}}_{c, 0}^* \quad \hat{\mathbf{A}}_{c, 1}^* \cdots \hat{\mathbf{A}}_{c, nprocs-1}^*])$ 
39:    $\bar{\mathbf{p}} \leftarrow \text{ComputeMean}(\mathbf{p})$ 
40:    $\text{idx}_{\bar{\mathbf{p}}} \leftarrow \text{GetIndices}(\bar{\mathbf{p}} < 0.05)$ 
41:   if  $\text{size}(\text{idx}_{\bar{\mathbf{p}}}) \geq \text{Indices of identified components}$  then
42:     break
43:   end if
44:   Send unmixing matrix  $\mathbf{W}_{c, right}^*$ , bias weight matrix  $\mathbf{W}_{0, right}^*$ 
      and learning rate  $\tau_{right}$  to the right neighbor
45: end while
46:  $\hat{\mathbf{A}}_{tot}^* \leftarrow \text{ComputeMedian}([\hat{\mathbf{A}}_{c, 0}^* \quad \hat{\mathbf{A}}_{c, 1}^* \cdots \hat{\mathbf{A}}_{c, nprocs-1}^*], \text{clusterIdxs})$ 
47:  $\mathbf{W}_{tot}^* \leftarrow (\hat{\mathbf{A}}_{tot}^{*-1} \cdot \mathbf{D}_x)^{-1}$ 

```

```
48: function LOADDATA(rank)
49:    $\mathbf{x}_{rank} \leftarrow \text{HDF5ReadPLIImageSeriesTiles}(\mathit{rank})$ 
50:    $\mathbf{m}_{rank} \leftarrow \text{HDF5ReadMaskTiles}(\mathit{rank})$ 
51:   return  $\mathbf{x}_{rank}, \mathbf{m}_{rank}$ 
52: end function

53: function PREPROCESSING( $\mathbf{x}_{rank}, \mathbf{m}_{rank}$ )
54:    $\text{idx}_{\text{dimX}}, \text{idx}_{\text{dimY}} \leftarrow \text{GetIndicesGreaterZero}(\mathbf{m}_{rank})$ 
55:    $\mathbf{x}_{rank, \text{masked}} \leftarrow \text{Segmentation}(\mathbf{x}_{rank}, \text{idx}_{\text{dimX}}, \text{idx}_{\text{dimY}})$ 
56:    $\mathbf{x}_{rank, \text{centered}} \leftarrow \text{Centering}(\mathbf{x}_{rank, \text{masked}}, \bar{\mathbf{x}})$ 
57:    $\mathbf{x}_{rank, \text{whitened}} \leftarrow \text{Whitening}(\mathbf{x}_{rank, \text{centered}})$ 
58: end function

59: function RECONSTRUCTION( $\hat{\mathbf{A}}_{tot}^*, \mathbf{W}_{tot}^*, \mathbf{x}_{rank, \text{whitened}}$ )
60:    $\text{idx}_{\text{SOI}} \leftarrow \text{IdentifySignalsOfInterest}(\hat{\mathbf{A}}_{tot}^*)$ 
61:    $\mathbf{s}'_{rank} \leftarrow \mathbf{W}_{tot}^* \cdot \mathbf{x}_{rank, \text{whitened}}$ 
62:    $\mathbf{x}_{rank, \text{denoised}} \leftarrow \text{ComputeReconstruction}(\mathbf{s}'_{rank}, \hat{\mathbf{A}}_{tot}^*, \text{idx}_{\text{SOI}}, \bar{\mathbf{x}})$ 
63: end function

64: function WRITEDATA(rank,  $\mathbf{x}_{rank, \text{denoised}}$ )
65:    $\text{HDF5WritePLIImageSeriesTiles}(\mathit{rank}, \mathbf{x}_{rank, \text{denoised}})$ 
66: end function
```

After each MPI process loads its partitions (step 6 and 7) of the stitched PLI image series and stitched mask, a load balancing of the masking indices is performed due to the unevenly distributed area of the histological section across the stitched image (step 12). The now evenly distributed pixels on all MPI processes are pre-processed individually (step 11-17). Thereafter, every MPI process initializes ($c = 0$) a random orthogonal weight matrix \mathbf{W}_0 and a bias weight matrix \mathbf{W}_0^* . By defining the right and left neighbors in terms of MPI processes a ring topology is constructed (step 21 and 22). For iteration step $c = 0$ all processes perform the Infomax algorithm with the adapted learning rule according to Equation 5.12 and 5.13 (step 35). In step 36 the optimal parameters are used to modify the basis vectors $\hat{a}_j(\rho)$ to select and to fix components of interest and to stop the algorithm, if convergence criteria are met (see Algorithm 2).

The following inter-subject consistency step needs the estimated mixing matrices of all MPI processes to compute reliable clusters. Each component that belongs to a specific cluster is subscripted with a corresponding cluster index. All indices are stored in an array referred to as **clusterIdxs** (step 38).

Furthermore, the consistency of a cluster of components is reflected by its p-value. A p-value is significant if it is smaller than a predefined threshold α_{FP}/ν , where α_{FP} describes the false positive rate and ν is the number of degrees of freedom. Here, the false positive rate α_{FP} was set to 0.05 according to [54, 58]. The p-values are also stored in an array referred to as **p**. The idea is to compute the mean of the p-values $\bar{\mathbf{p}}$ for each cluster, referred to as average-linkage strategy, where each entry in $\bar{\mathbf{p}}$ has to be significant (step 39). Here, the entries in $\bar{\mathbf{p}}$ are considered significant if they are below a defined threshold of 0.05, which is set purposely equal to α_{FP} (step 40).

The pICAP algorithm converges when the number of significant components are at least equal to the number of identified component of interest (step 41). This is the case due to stronger similarities among components of interest resulting in reliable clusters compared to the similarities among noise components. However, if one of the convergence criteria are not met, the updated unmixing matrix $\mathbf{W}_{c, \text{right}}^*$, the bias weight matrix $\mathbf{W}_{0, \text{right}}^*$ and the learning rate τ_{right} are send to the next MPI process (step 44). The simultaneous updating of the unmixing matrices at each iteration step, the clustering and the circular communication in the ring topology are performed until pICAP converges. The performance compromise between the parallelization effort in pICAP and the convergence criteria of each acICA process

(see Section 3.5) is explained in the following sections. At the final stage the estimated $\hat{\mathbf{A}}_{c, right}^*$ and **clusterIdxs** are gathered to form one total mixing matrix $\hat{\mathbf{A}}_{tot}^*$ and unmixing matrix \mathbf{W}_{tot}^* (step 46 and 47).

In order to denoise the tiles for further post-processing, the disjoint partitions of the tiled PLI image series \mathbf{x}_{rank} are first pre-processed and subsequently the components of interest of $\hat{\mathbf{A}}_{tot}^*$ are identified. The source components are then computed with $\mathbf{W}_{tot}^* \cdot \mathbf{x}_{rank, whitened}$ (step 61). By zeroing the columns of $\hat{\mathbf{A}}_{tot}^*$ (see Section 3.5), which represent noise and artifact components and apply the resulting matrix on \mathbf{s}'_{rank} , the denoised partitions $\mathbf{x}_{rank, denoised}$ can be computed.

7.1.3. Experimental Procedure

To determine the amount of time the pICAP algorithm needs to converge, a runtime analysis was performed. For this case the largest data set in terms of storage from the 20 PLI image series of the PM was used. This data set included both stitched and tiled PLI image series and stitched and tiled masks. The storage sizes of the generated data from different steps of the 3D-PLI workflow are shown in Table 7.2.

Table 7.2.: Size of data for different steps of the 3D-PLI workflow.

	size
stitched PLI Image series	127.46 GB
tiled PLI Image series	312.19 GB
stitched mask	1.8 GB
tiled mask	4.4 GB

The runtime analysis was performed by measuring the elapsed time of the **LoadData**, **PreProcessing**, **Reconstruction** and **WriteData** functions and the core algorithm of pICAP on JuDGE (see Algorithm 3). By increasing the number of MPI processes with the same size of workload the runtime scalability was measured. Therefore, each reserved compute node launched one MPI process with 90 GB allocated memory. The MPI process is mapped to one core of a JuDGE compute node. Thus, the total workload is distributed in form of data decomposition between the compute nodes. The allocation of 90 GB of memory per compute node is necessary due to the internal memory management of the Python interpreter (version 2.7.x), where the memory of a deallocated Python object is not released to the operating system but marked as a freed object in a pool's free list [93].

As a second step, the integration of GPUs was achieved for large vector-matrix and matrix-matrix multiplications. The integration of GPUs is restricted to the pre-processing and reconstruction steps due to two primary reasons. Firstly, the intrinsic dependencies in the learning rule of the here used Infomax and the inter-subject consistency are not able to be unraveled to parallelism. Secondly, the constant accumulating overhead of kernel launches on the GPU, due to the additional calls for memory transfer operations at each iteration step originated mainly by the random permutation of $\mathbf{x}_{rank, \text{whitened}}$ and by the receiving and sending of weight matrices, could cause a lack of improvement, thus making the GPU usage for the acICA routines unsuitable. For this reason, the computation of a covariance matrix $\text{Cov}(\mathbf{x}_{rank, \text{centered}}, \mathbf{x}_{rank, \text{centered}}^T)$ for large sizes of $\mathbf{x}_{rank, \text{centered}}$ in the pre-processing procedure is performed on GPUs. The same holds for the decorrelation of large $\mathbf{x}_{rank, \text{centered}}$ and the retrieval of the sources \mathbf{s}'_{rank} . Thus, the advantage of high arithmetic throughput of GPUs is exploited. Hence, PyCUDA with standard CUBLAS routines was used for the above mentioned parts of the pre-processing and reconstruction steps in pICAP and a subsequent comparison in terms of computing time with a standard CPU approach was performed. Note that, in the following, a single GPU is assigned to only one MPI process.

In order to verify the mixing matrices of pICAP produced and the *wrGOFs* generated hereafter, a comparison with the sequential acICA was carried out. Therefore, an approximately 9 times smaller data set taken from the 20 histological sections, which were acquired in the PM, was used. The smaller data set includes stitched PLI image series with a size of approximately 13.9 GB and a stitched mask with a size of approximately 197 MB. For verification purposes both algorithms used the same data set.

7.2. Results

7.2.1. Scalability

The main objective of pICAP is the accurate processing of PM images with a fast analysis of PLI image series. For the latter activity, a runtime analysis was carried out using the largest data set of the 20 PLI image series, with the optimal parameters η_{opt} , t_{opt} and ε_{opt} and the fitted cost function $g_a(\tilde{\mathbf{s}}(k))$ of GM and pcWM being incorporated in pICAP. By analyzing the GM part ($\approx 18 \times 576 \cdot 10^6$ pixels) of the 127.46 GB large PLI image series and increasing the used number of MPI processes

for each measurement, a reduction of the runtime up to 50 MPI processes is observed (Fig. 7.4). Note that the connected measurement points start with 10 MPI processes. A marginal growth of runtime was then observed for 60 MPI processes. pICAP performed with 20 MPI processes runs approximately 50% faster than pICAP with 10 MPI processes. Moreover, the optimal working point of 50 MPI processes found performed approximately 17.1% faster than with 40 MPI processes.

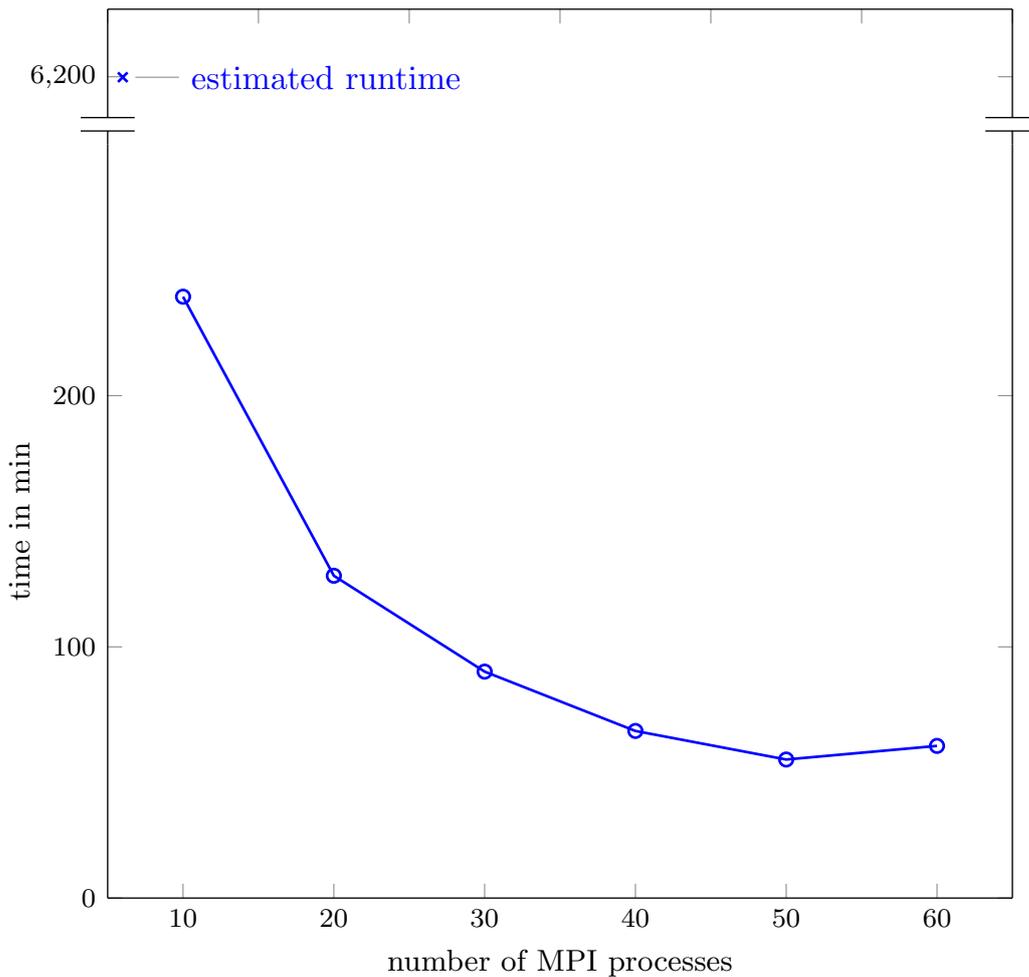


Figure 7.4.: Runtime measurements including the core method of pICAP with the mandatory pre-processing steps, which were applied on the GM of the stitched PLI image series with a size of 127.46 GB. The runtime for a single process was extrapolated on the basis of the sequential acICA by processing the GM of a PLI image series of size 13.9 GB.

Due to the limited memory of 96 GB per compute node the runtime measurement of

the GM part with one MPI process had to be extrapolated by using the sequential acICA, which was applied on the GM ($\approx 18 \times 100 \cdot 10^6$ pixels) of a PLI image series being 13.9 GB smaller. The estimated runtime for the sequential acICA amounts to approximately 6200 min. Here, only the runtime scalability of GM is described since the performance of processing the pcWM part would show the same characteristic. A detailed analysis of the entire pICAP application is shown in Figure 7.5, where `LoadData`, `PreProcessing`, `Reconstruction` and `WriteData` functions together with the core pICAP algorithm were profiled. pICAP was executed with the optimal number of 50 MPI processes on GM of the PLI image series of size 127.46 GB. It was observed that the core algorithm of pICAP consumed approximately 77% of the total computational time followed by parallel loading and writing taking 7.52% and 12.3% of the total runtime, respectively. Moreover, the pre-processing and reconstruction steps took each 0.6% and 2.4% of total time. The share of computing time for the initialization of the algorithm is negligible.

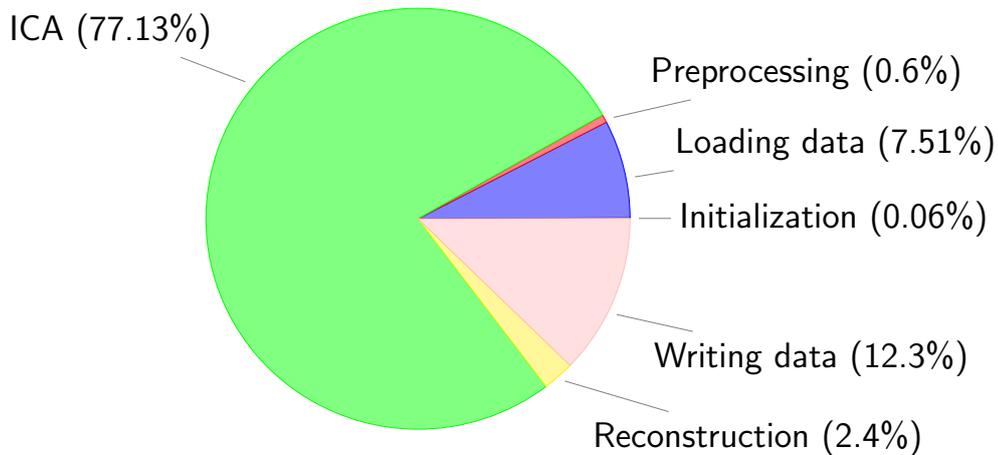


Figure 7.5.: Total runtime shares starting from the parallel reading to the parallel writing of tiled PLI image series of a complete pICAP application with the runtime optimal 50 MPI processes.

7.2.2. GPU Acceleration

In the next step the massive parallel processors in the GPU were used to reduce the computing time of linear algebra operations in the `PreProcessing` and

Reconstruction functions, which need to handle large matrices and vectors.

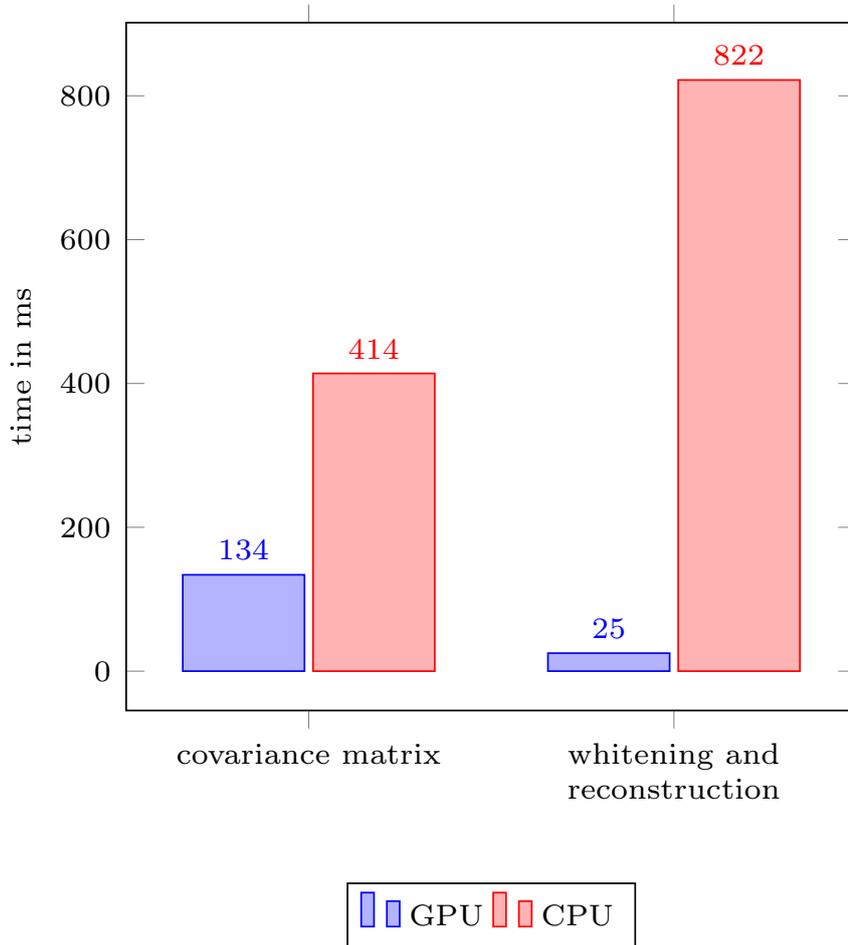


Figure 7.6.: Comparison of elapsed time between the kernel execution on the GPU and standard execution on the CPU for matrix-matrix and vector-matrix multiplication.

Note that the vectors described here hold, for each entry, a vectorized image of the PLI image series, resulting in 18 vectorized PLI image series. In order to use the GPUs, a smaller data set was used for fitting the disjoint partitions and the results in the memory of each GPU. It was found that with 50 MPI processes and a data set size of 13.9GB the disjoint partitions ($\approx 18 \times 4 \cdot 10^6$ pixels per MPI process) are able to be processed on a single GPU for each compute node. The matrix-matrix and vector-matrix calculations were performed on 50 compute nodes

using one GPU per compute node. Figure 7.6 shows a comparison of computing time between GPU kernel launches and CPU for the calculation of the covariance matrix, the whitening procedure and the reconstruction step. This Figure highlights the increased speed of the calculation of the covariance matrix by about a factor of 3, whereas an increased speed of the decorrelation and reconstruction step, which have the same linear algebra operation, by about a large factor of 32 is achieved.

7.2.3. Validation and Consistency

In order to verify the results of the pICAP algorithm with the sequential acICA, a comparison of consistency of the estimated mixing matrices from both methods was carried out. To this end, both algorithms were applied on the same data set by using the optimal parameters and the fitted $g_a(\tilde{\mathbf{s}}(k))$ of GM and pcWM for the PM, respectively. Figure 7.8 and 7.9 provide a summary of the resulting mixing matrices of the acICA and pICAP algorithms for GM. The red, green and orange highlighted basis vectors from pICAP and acICA were identified as components of interest, whereas the remaining basis vectors (blue), which differ from the expected sinusoidal profile, were rejected (see Section 3.5). By observing the course and the range of values of the identified signal components from both algorithms and by considering the sign ambiguity (see Section 3.3) a clear resemblance is observed. Similar basis vectors which were identified as signal components in both algorithms are colored in the same color (see Figure 7.8 and 7.9 the corresponding red, green and orange highlighted basis vectors). Furthermore, Figure 7.9 shows clearly separated clusters with their basis vectors gathered from all MPI processes. In particular, the identified signal components of each cluster in Figure 7.9 (a), (b) and (c) only show marginal deviations. The same applies for pcWM (see Appendix D).

After the estimation of the mixing matrices of both pICAP and acICA algorithms, components representing signal of interest are reconstructed back to PLI space. The subsequent wrGOF calculations are performed individually for both methods at each pixel location of the same data set. Figure 7.7 (b) shows the color coded wrGOF values for the acICA algorithm. In the entire data set, the mean value of the wrGOFs was found to be 396, which indicates a significant signal enhancement due to acICA. Moreover, in some pixel locations of the gray matter a better performance of acICA is observed (indicated by the bright colored pixels in Figure 7.7 (b)).

For the case of the pICAP algorithm Figure 7.7 (a) shows the achieved signal enhancements. Here, the overall mean of the wrGOFs was found to be 131 with an

increased local performance of pICAP in the outer boundaries of the gray matter. For both methods a signal improvement ($wrGOF > 1$) and a threefold signal enhancement are observed for the acICA algorithm on both GM and pcWM compared to the pICAP algorithm.

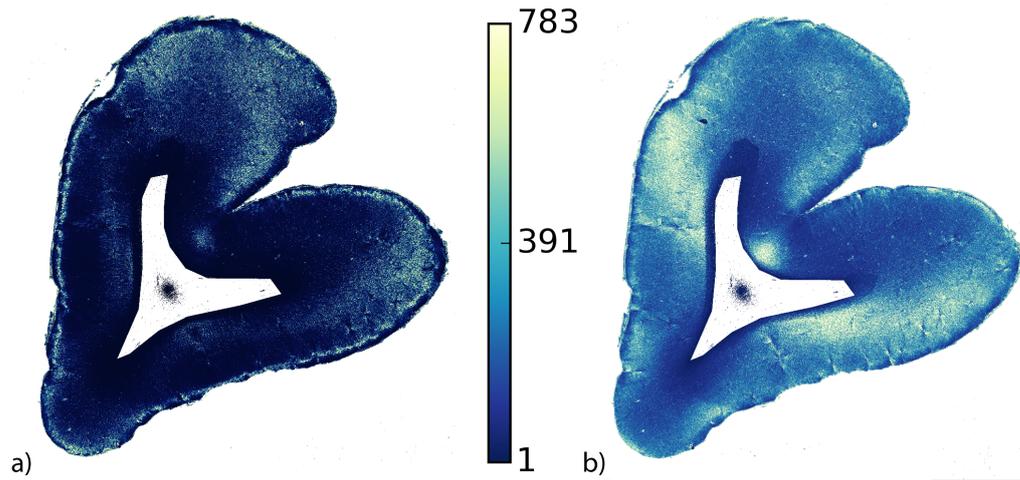


Figure 7.7.: $wrGOF$ values in arbitrary units per pixel generated after the new pICAP algorithm (a) and the sequential acICA (b) applied on GM and pcWM. Only the processed pixels of GM and pcWM of the left hemisphere of the histological section are illustrated.

7. A New Parallelization Concept for Adapted cICAP

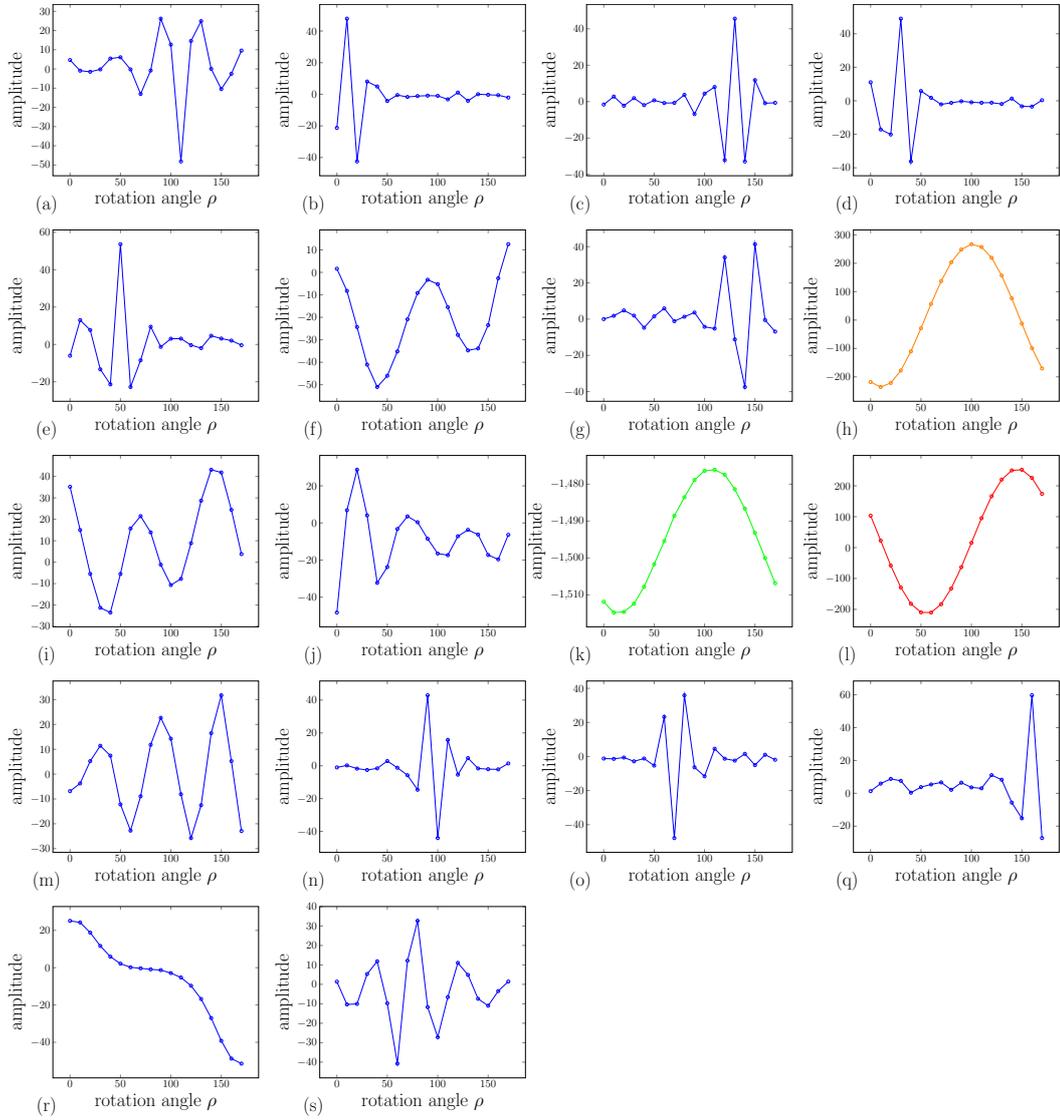


Figure 7.8.: Basis vectors $\hat{a}_j^*(\rho)$ of the mixing matrix $\hat{\mathbf{A}}_c^*$ after completion of the acICA algorithm applied on GM. The identified signal components are highlighted in red, green and orange.

7. A New Parallelization Concept for Adapted cICAP

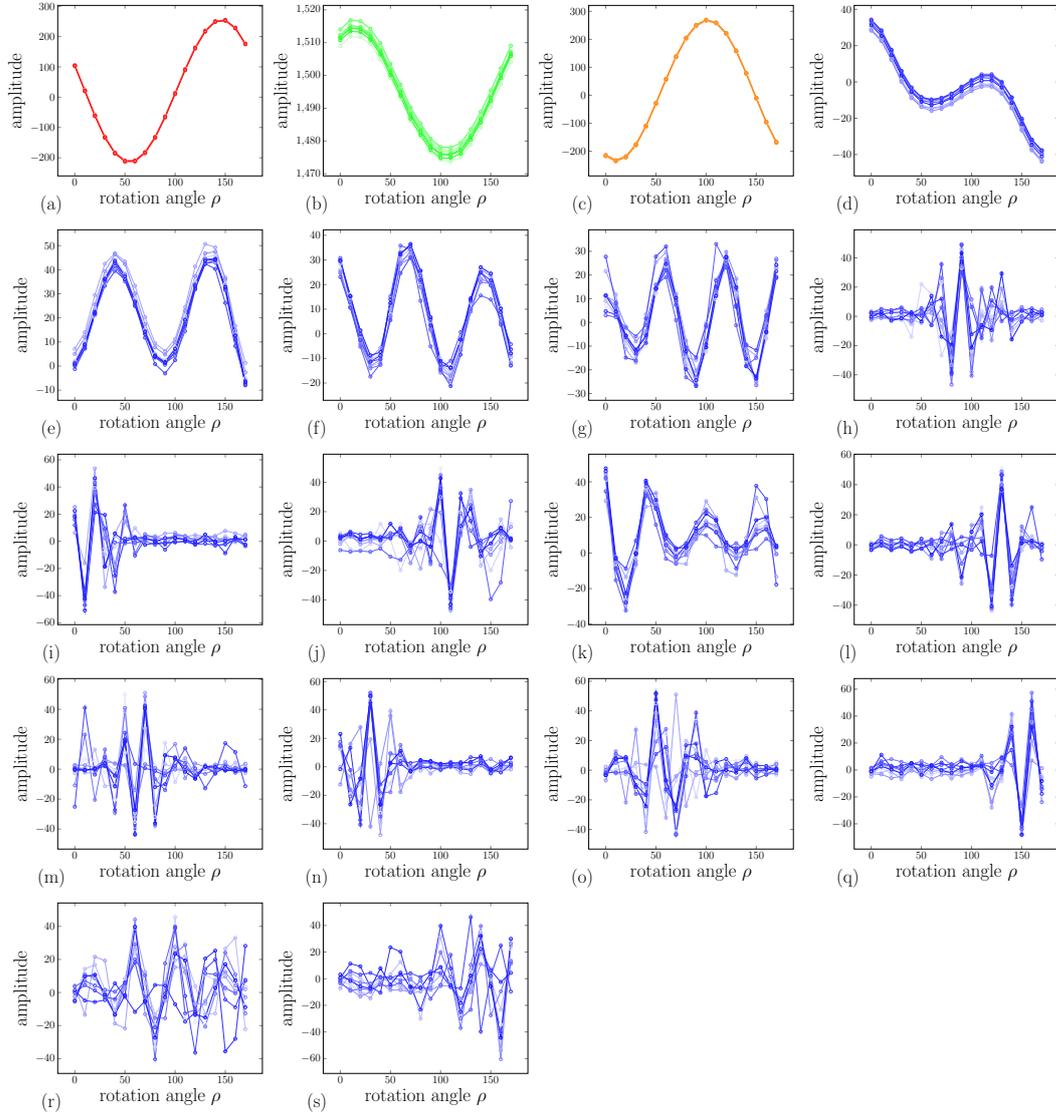


Figure 7.9.: Basis vectors $\hat{a}_{j,\text{rank}}^*(\rho)$ of the mixing matrices $\hat{\mathbf{A}}_{c,\text{rank}}^*$ from all MPI processes after completion of the pICAP algorithm and the clustering applied on GM. The identified clustered signal components are highlighted in red, green and orange. The basis vectors of each cluster are colored gradually in a transparent manner.

7.3. Summary of Achievements

In order to denoise large high-resolution data sets very fast and accurately, a novel ICA approach was developed. This new method, referred to as *pICAP*, encompasses the *acICA* approach. A demonstrator, *pICAP* was implemented on the supercomputer *JuDGE* at the Jülich Supercomputing Centre. *pICAP* is especially designed for image sizes of several gigabytes, acquired from the PM in 3D-PLI. Therefore, multiple, simultaneously running *acICA* processes with GM and *pcWM*-specific cost functions for the PM were used. The approach of simultaneously running *acICA* processes is based on the strategy of Himberg et al. [50] for MEG and fMRI data and on a group level ICA [19, 23, 36], where consistent spatial ICA components for an entire data set are found by combining the *acICA* results of single partitions. Additionally, the incorporation of prior knowledge of the underlying source signals enables a specific signal decomposition for both GM and *pcWM* on each *acICA*. Thus, the parallelized signal decomposition is performed fast and tissue-specific with respect to GM and *pcWM* for large high-resolution data sets.

The performance of this approach was analyzed on the GM of the largest data set from one of the 20 histological sections of a vervet monkey brain (Table 7.2). It was shown that the runtime optimal number of MPI processes is 50, since the runtime slightly increases for a number of MPI processes larger than 50. The increasing runtime can be explained by the increasing amount of communication among all MPI processes in the established ring topology, where the time of all executed communication processes may outperform the processing time of the *acICA*.

A first major achievement is that the entire runtime of 55.1 min for 50 MPI processes shows a 112 fold increased performance compared to the conservatively estimated sequential *acICA* runtime of at least 6199 min. This increased performance is induced primarily by the data decomposition applied, where each MPI process has a smaller portion to process.

Secondly, the newly included convergence criterion after the inter-subject consistency, which takes effect if the consistency of a cluster of ICA components is significant, will lead to an advanced termination of the *pICAP* algorithm.

A third substantial improvement could be achieved through the speed up of large vector-matrix and matrix-matrix multiplications by means of the NVIDIA GPUs of *JuDGE* avoiding processing bottlenecks of the *pICAP* algorithm. This results show that with almost no cost an increase of speed of 32 fold for the decorrelation

and reconstruction step and threefold for the computation of the covariance matrix can be achieved. By taking the profiling of the implementation of the algorithm (see Fig. 7.5) into account, it can be seen that the pre-processing and reconstruction steps consume 3% of the total computational time without GPUs. Due to the achieved speed-up of GPUs the 3% may be minimized to a neglectable processing time for both the pre-processing, i.e., the covariance matrix and the whitening procedure, and the reconstruction procedures. Furthermore, approximately 20% of the runtime is consumed by the parallel I/O, which may only be altered by the underlying hardware and network of the supercomputer.

When comparing the identified components, which represent signals of interest of the estimated mixing matrices for the pICAP and acICA algorithms a strong resemblance in the characteristics and data ranges is observed. Only the components with signals of interest are of importance since the remaining components are rejected during the reconstruction procedure. By reconstructing the components with signals of interest for both GM and pcWM for each algorithm, the *wrGOF* values are generated. The findings in terms of the *wrGOF* values for pICAP and acICA, respectively, showed a threefold different *wrGOF* range and slightly different local performances (see Fig. 7.7 (a) and (b)). The different *wrGOF* ranges are explained by the small differences in the almost similar mixing matrices of pICAP and acICA, which cause slightly different signal enhancements per pixel throughout the examined histological section. The small differences between the mixing matrices $\hat{\mathbf{A}}_{\text{acICA}}^*$ and $\hat{\mathbf{A}}_{\text{pICAP}}^*$ result in slight different sources $\mathbf{s}'_{\text{acICA}} = \left(\hat{\mathbf{A}}_{\text{acICA}}^{*-1} \cdot \mathbf{D}_{\mathbf{x}}\right)^{-1} \cdot \mathbf{x}_{\text{whitened}}$ and $\mathbf{s}'_{\text{pICAP}} = \left(\hat{\mathbf{A}}_{\text{pICAP}}^{*-1} \cdot \mathbf{D}_{\mathbf{x}}\right)^{-1} \cdot \mathbf{x}_{\text{whitened}}$ for both GM and pcWM. After keeping the signals of interest the denoised PLI image series $\mathbf{x}_{\text{denoised, acICA/pICAP}} = \hat{\mathbf{A}}_{\text{acICA/pICAP}}^* \cdot \mathbf{s}'_{\text{acICA/pICAP}}$ are transformed back in PLI space. The deviations of the estimated mixing matrices are now expressed in the slight differences of the denoised PLI images series $\mathbf{x}_{\text{denoised, acICA}}$ and $\mathbf{x}_{\text{denoised, pICAP}}$ which can explain the different signal improvements in Figure 7.7 (a) and (b). Moreover, the spots with less signal enhancements can be induced by a relatively higher local SNR of the unprocessed data set, whereas areas with lower SNR of the unprocessed data set may exhibit higher *wrGOF* values due to the ICA filtering. This may be caused by the inhomogenous preparation of the brain tissue on the glass slides. However, both algorithms showed on average an increased signal enhancement throughout the histological section.

In summary, the pICAP algorithm provides a fast and practical tool to effectively

7. A New Parallelization Concept for Adapted cICAP

denoise large high-resolution data sets with exploiting HPC techniques on super-computers.

8. Discussion

As an established method for signal decomposition, ICA found widespread use in neuroscience, both for noise and artifact removal [20, 31, 35, 66, 70, 80] and extraction of signals of interest [29, 83, 84, 89]. Over the last years ICA has also been demonstrated to be an effective tool to remove noise and artifacts in the acquired images of the LAP for 3D-PLI [21, 30, 32]. An ICA concept for the LAP was developed which incorporates the expected birefringence signal and component selection [21]. Generally, incorporating prior information and constraints in ICA improves the signal decomposition [10, 48, 52]. Additionally, Huang et al. [52] and Breuer et al. [20] demonstrated that by imposing the information of the underlying source signal, a faster and source signal sensitive signal decomposition occurs.

Motivated by this strategy a new data-driven approach in ICA for 3D-PLI was introduced in this thesis. The new acICA algorithm encompasses the prior information of the expected birefringence signal, the automatic component selection and the information of the underlying sources in the GM and parts of the WM. In order to accurately separate the underlying sources in the different regions of the histological section, two different cost functions were implemented: one for the GM and one for the pcWM. This allows for a fast and tissue-specific signal enhancement of each histological section acquired in the LAP.

In 3D-PLI the accurate determination of the orientation of nerve fibers in both optical setups, PM and LAP, is of importance. The reconstruction of nerve fibers strongly depends on the magnitude of the captured optical signal and its quality at the receiver side of the optical systems. Particularly, the different tissue types in a histological section, i.e., GM and WM, exhibit variations of the measured signal amplitude ([32], Fig. 5.7). Thus, strong signal variations indicating their birefringent properties are mostly located in the WM, whereas weaker signal variations are either located in the GM or at boundaries between GM and WM [32]. acICA considers the signal dynamics of both tissue types, since the impact of noise on weaker signals is higher compared to stronger signals located in the WM. For this work, each

histological section was therefore partitioned into three areas. The first area contains pixels located in the GM. The second area includes pixels located at the boundaries between GM and WM, referred to as pcWM, whereas the third area comprises pixels in the WM (see Chapter 5). Since the third area already provides a high SNR, an acICA filtering is considered as redundant. Using the newly introduced two separate cost functions the first two areas GM and pcWM could be denoised fast and optimally in terms of their unique birefringent properties. According to the findings described in Chapter 5, the acICA algorithm also showed a higher signal enhancement compared to the standard procedure of pICAP on an entire histological section [21].

There are three parameters involved in the acICA algorithm representing the constraints and prior information that affect the accuracy of signal separation and the convergence of the algorithm. Breuer et al. [21] demonstrated that a unique minimizer in \mathbb{R}^2 can be found via a naive brute-force method for the cICAP algorithm. The determined optimal parameters of one histological section were then applied to various cICAP applications of all histological sections across a post-mortem brain. This may be valid under the assumption that the percentage of white and gray matter on all histological sections and the amount of pixels to be processed is the same. For this study different percentages of white and gray matter ranging from frontal to the occipital lobe were observed due to the preliminary subdivision of GM and pcWM for the acICA algorithm. The DSA used here was applied on a set of representative histological sections of an entire post-mortem vervet brain. It can be assumed that for each histological section a unique minimizer is found due to the convex surface mapped by the function values of each parameter combination [21]. Chapter 6 showed that only one set of parameters throughout the post-mortem vervet brain for each GM and pcWM is needed, whereas the third parameter ε needs to be broken down into three brain regions: frontal, parietal/temporal and occipital lobe. In due consideration of these parameters subsequent applications of acICA on comparable post-mortem brains can be carried out accurately.

The PM in 3D-PLI established a pathway to investigate single fiber architectures at a high level of detail. The higher spatial resolution in the PM compared to the LAP comes at the cost of larger data volumes to be stored. For analyzing and reconstructing nerve fibers at the microscopical scale of the PM, the standard workflow in 3D-PLI (see Section 2.5) needs to handle the Big Data problems in an effective and fast fashion. For this purpose the usage of supercomputers and the exploitation of

HPC techniques provide a suitable strategy. However, an ICA filtering strategy for high-resolution data sets is still missing in the standard workflow. Since a sequential acICA application applied on a whole stitched PLI image series of the PM would require huge amounts of memory and take many days to process, a new parallelized concept of acICA was introduced in this thesis, which overcomes the constraint of sequential execution and memory limitations. In principle, the new pICAP algorithm estimates a global solution in terms of a mixing matrix by combining the results of multiple simultaneously running acICAs on single disjoint partitions of an entire data set (see Chapter 7). Different from the sequential acICA algorithm for the LAP, the cost functions for both GM and pcWM of the PM were used in pICAP (see Chapter 5). The resulting mixing matrices for both GM and pcWM are subsequently applied again on tiled PLI image series. This ensures an adequate stitching on denoised tiled data sets and post-processing, i.e., generation of FOMs and tractography.

The pICAP algorithm exhibits a 112-fold increased performance compared to the estimated sequential acICA runtime using 50 MPI processes. Furthermore, the advantage of SIMD architectures such as GPUs for the large scale linear algebra operations in the pre-processing and reconstruction steps of pICAP was taken into account. NVIDIA's CUDA with the available third party wrapper in Python [72] and CUBLAS were chosen for the GPU implementation, taking advantage from the straightforward implementation, extensive documentation and abstraction in Python. The usage of GPUs decreased the pre-processing and reconstruction steps of pICAP by a factor of 32 and 3, respectively. As shown in Chapter 7, the small aberrations of the mixing matrices compared to the mixing matrix of the sequential acICA algorithm caused unequal denoised PLI image series and therefore different signal enhancements per pixel throughout the histological section. Still a distinct tendency of high signal enhancement was maintained with pICAP, mainly due to the incorporation of prior information and the GM and pcWM-specific cost functions for the PM at each iteration step.

With the good portability and abstraction of Python for exploiting HPC on supercomputers, pICAP can be applicable ad-hoc on future supercomputer systems. Additionally, the pICAP approach will strongly benefit from the increased throughput and affordability of recent GPUs and performances of modern supercomputers.

9. Conclusion and Outlook

The aim of this work was to develop a tissue-specific signal separation method for both optical systems in 3D-PLI, the *Large Area Polarimeter* (LAP) and the *Polarizing Microscope* (PM). For the new acICA algorithm the learning rule of the natural-gradient version of Infomax was adapted in order to fit the newly introduced cost functions. For each of the optical systems two cost functions representing the cumulative distributions of GM and pcWM were generated. acICA was demonstrated to be faster and to exhibit a better quality of signal separation on GM and pcWM reflected by the increased *wrGOFs* compared to the cICAP algorithm. Thus, a higher SNR and signal restoration on areas prone to noise was achieved. Also whole-brain parameter sets for acICA were provided with the help of the downhill simplex method. With these parameter sets and the usage of the acICA algorithm an accurate signal restoration on various histological sections throughout an entire post-mortem brain is possible. The obstacle of filtering microscopical data sets due to the large size of data volumes was overcome by the development of a new parallelized ICA concept on a supercomputer. Processing ICA on large-size data volumes in a sequential fashion, e.g., on a single desktop computer, results in utterly long computation times. To tackle the time-intensive ICA procedure, pICAP discretizes tasks such as acICA processes on partitions in a concurrent manner, where computational bottlenecks in the pre-processing and reconstruction steps are outsourced to the GPUs. From these findings it can be concluded that the new pICAP algorithm provides a noise and artifact removal in a drastically shortened time due to the intrinsic tissue-specific cost functions. Since, accurate and reliable profiles are crucial for generating three-dimensional spatial orientations of the reconstructed nerve fibers in both LAP and PM datasets, acICA and pICAP are mandatory elements of 3D-PLI.

In the future a third cost function representing dust particles may be added to the specific applications of acICA and pICAP. A captured dust particle which is placed on one of the rotating polarizers in the optical systems may deteriorate the sinusoidal profile in the WM. For this purpose it is necessary to find a cost function related

to dust particles and identify components related to it in the WM. By transforming only the remaining components to the PLI space an efficient artifact rejection is achieved. With the introduced acICA and pICAP algorithms and the usage of a third cost function specifically designed to remove dust particles in the WM, the SNR in the WM could be enhanced.

At present pICAP and acICA are performed on single histological sections of a post-mortem brain. In order to filter multiple histological sections at a time a promising way in the future would be the usage of a multilinear ICA approach. This model uses multilinear (tensor) algebra to find sets of independent sources across the histological sections. The multilinear ICA approach would additionally benefit from the usage of a supercomputer and GPUs due to the increased data volume to process and the sophisticated multilinear algebra operations.

Finally, the replacement of hard drives with faster solid-state drives could improve in the future the data access rates and, therefore, significantly reduce the load and store time of pICAP.

A. Derivation of the 3D-PLI Signal

Jones matrices for an ideal linear horizontal polarizer (P_x) and an ideal linear vertical polarizer (P_y) are given by:

$$\begin{aligned} P_x &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \\ P_y &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned} \quad (\text{A.1})$$

A wave retarder is specified with a wave retardance γ for a specific wavelength. The wave retarder can be described by a Jones matrix, when the retarder introduces a phase shift, i.e., $\gamma = \frac{\delta}{2}$, with δ being the phase difference, along the fast axis (x-axis) and $-\gamma = -\frac{\delta}{2}$ along the slow axis (y-axis). Thus, the Jones matrix for a wave retarder is given by

$$M_{\text{ret}} = \begin{pmatrix} e^{i\gamma} & 0 \\ 0 & e^{-i\gamma} \end{pmatrix}. \quad (\text{A.2})$$

Furthermore, if an optical element is rotated counter-clockwise by an angle ψ , the rotated optical element will be expressed as:

$$J(\psi) = R(\psi) \cdot J \cdot R(-\psi), \quad (\text{A.3})$$

with J describing the Jones matrix and R denoting the rotation matrix. R is given by:

$$R(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix}. \quad (\text{A.4})$$

Hence, a quarter-wave retarder with $\gamma = \frac{\pi}{4}$ and $\psi = -\frac{\pi}{4}$ is described by:

$$\begin{aligned}
 M_{\lambda/4}(\psi) &= R(\psi) \cdot M_{\text{ret}} \cdot R(-\psi) \\
 &= \begin{pmatrix} \cos(-\frac{\pi}{4}) & -\sin(-\frac{\pi}{4}) \\ \sin(-\frac{\pi}{4}) & \cos(-\frac{\pi}{4}) \end{pmatrix} \cdot \begin{pmatrix} e^{i\frac{\pi}{4}} & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{pmatrix} \cdot \begin{pmatrix} \cos(-\frac{\pi}{4}) & \sin(-\frac{\pi}{4}) \\ -\sin(-\frac{\pi}{4}) & \cos(-\frac{\pi}{4}) \end{pmatrix} \\
 &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}. \tag{A.5}
 \end{aligned}$$

Each volume element of the brain tissue, which contributes to one image pixel, can be described as a single wave retarder where the x-axis is in direction of the optic axis of the nerve fibers. If all the optical elements are rotated simultaneously by angle ρ with respect to the nerve fiber direction φ , the rotation of the brain tissue will be expressed by $\psi = \varphi - \rho$. Thus, the brain tissue is described by:

$$\begin{aligned}
 M_{\text{tissue}}(\psi) &= R(\psi) \cdot M_{\text{ret}} \cdot R(-\psi) \\
 &= \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix} \cdot \begin{pmatrix} e^{i\gamma} & 0 \\ 0 & e^{-i\gamma} \end{pmatrix} \cdot \begin{pmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{pmatrix} \\
 &= \begin{pmatrix} \cos \gamma + i \sin \gamma \cos(2\psi) & i \sin \gamma \sin(2\psi) \\ i \sin \gamma \sin(2\psi) & \cos \gamma - i \sin \gamma \cos(2\psi) \end{pmatrix}. \tag{A.6}
 \end{aligned}$$

A.1. Jones Calculus for the LAP

As already shown in Equation 2.1 the 3D-PLI setup for the LAP can be described by the Jones calculus. For reason of simplicity P_y is denoted as the analyzer A and P_x is denoted as the polarizer P .

$$\begin{aligned}
 \vec{E}' &= A \cdot M_{\text{tissue}} \cdot M_{\lambda/4} \cdot P \cdot \vec{E} \\
 &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot M_{\text{tissue}} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} E_x \\ E_y \end{pmatrix} \\
 &= \frac{E_x}{\sqrt{2}} [i \sin \gamma \sin(2\psi) - i \cos \gamma - \sin \gamma \cos(2\psi)] \vec{e}_y \\
 &= \frac{E_x}{\sqrt{2}} [\sin \gamma \cos(2\psi) + i(\sin \gamma \sin(2\psi) - \cos \gamma)] \vec{e}_y \tag{A.7}
 \end{aligned}$$

For $I \sim |\vec{E}|^2$:

$$\begin{aligned}
 I &= \text{Re}^2(\vec{E}') + \text{Im}^2(\vec{E}') \\
 &= \frac{I_0}{2} \left[\sin^2 \gamma \cos^2(2\psi) + (\sin \gamma \sin(2\psi) - \cos \gamma)^2 \right] \\
 &= \frac{I_0}{2} [1 - \sin(2\gamma) \sin(2\psi)]
 \end{aligned} \tag{A.8}$$

For $\gamma = \frac{\delta}{2}$ and $\psi = \varphi - \rho$:

$$I = \frac{I_0}{2} \cdot [1 + \sin(2(\rho - \varphi)) \sin \delta]. \tag{A.9}$$

A.2. Jones Calculus for the PM

Since in the PM setup the quarter-wave retarder and the second linear polarizers are fixed, only the first linear polarizer is rotated (Fig. 2.1). The x-axis of the quarter-wave retarder is positioned by $\psi = -\frac{\pi}{4}$ with respect to the second linear polarizer. Hence, the quarter-wave retarder is described by:

$$\begin{aligned}
 M_{\lambda/4}(\psi) &= R(\psi) \cdot M_{\text{ret}} \cdot R(-\psi) \\
 &= \begin{pmatrix} \cos(-\frac{\pi}{4}) & -\sin(-\frac{\pi}{4}) \\ \sin(-\frac{\pi}{4}) & \cos(-\frac{\pi}{4}) \end{pmatrix} \cdot \begin{pmatrix} e^{i\frac{\pi}{4}} & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{pmatrix} \cdot \begin{pmatrix} \cos(-\frac{\pi}{4}) & \sin(-\frac{\pi}{4}) \\ -\sin(-\frac{\pi}{4}) & \cos(-\frac{\pi}{4}) \end{pmatrix} \\
 &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}.
 \end{aligned} \tag{A.10}$$

Since the emitted light of the light source in the PM is unpolarized, the Jones calculus is not applicable. To fulfill the Jones calculus's prerequisite the linearly polarized light after the first linear polarizer is considered. Thus, the incoming electric field vector of the PM setup is expressed as

$$\vec{E}(\rho) = E_0 \begin{pmatrix} \cos \rho \\ \sin \rho \end{pmatrix}, \tag{A.11}$$

with $E_x = E_y = E_0$, where ρ describes the rotation angle of the first linear polarizer. Therefore, the Jones calculus for the PM setup is expressed as

$$\begin{aligned}
 \vec{E}' &= P_y \cdot M_{\lambda/4} \cdot M_{\text{tissue}} \cdot \vec{E}(\rho) \\
 &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \cdot M_{\text{tissue}} \cdot E_0 \begin{pmatrix} \cos \rho \\ \sin \rho \end{pmatrix} \\
 &= \frac{E_0}{\sqrt{2}} \left[-i \cos \gamma \cos \rho + \sin \gamma \cos(2\varphi) \cos \rho + i \sin \gamma \sin(2\varphi) \cos \rho \right. \\
 &\quad \left. + \sin \gamma \sin(2\varphi) \sin \rho + \cos \gamma \sin \rho - i \sin \gamma \cos(2\varphi) \sin \rho \right] \vec{e}_y
 \end{aligned} \tag{A.12}$$

For $I \sim |\vec{E}'|^2$:

$$\begin{aligned}
 I &= \frac{E_0^2}{2} \left[\cos^2 \gamma + \sin^2 \gamma \sin^2(2\varphi) + \sin^2 \gamma \cos^2(2\varphi) - 2 \cos \gamma \sin \gamma \sin(2\varphi) \right. \\
 &\quad \left. \cdot (\cos^2 \gamma - \sin^2 \gamma) + 4 \cos \gamma \sin \gamma \cos(2\varphi) \cos \varphi \sin \rho \right] \\
 &= \frac{E_0^2}{2} [1 - \sin(2\gamma) \sin(2\varphi - 2\rho)] \\
 &= \frac{E_0^2}{2} [1 + \sin(2\gamma) \sin(2\rho - 2\varphi)] \\
 &= \frac{I_0}{2} [1 + \sin(2\gamma) \sin(2\rho - 2\varphi)].
 \end{aligned} \tag{A.13}$$

For $\gamma = \frac{\delta}{2}$:

$$I = \frac{I_0}{2} [1 + \sin(2(\rho - \varphi)) \sin \delta]. \tag{A.14}$$

A.3. Derivation of the Phase Retardation

When light passes through a birefringent medium, it is decomposed in two waves with perpendicular linear states of polarization. The waves are referred to as the ordinary and extraordinary wave. Since a birefringent medium has different refractive indices, a phase shift is induced between the ordinary and extraordinary wave. The phase shift δ depends on the extraordinary and ordinary refractive indices n_e and n_o , thickness d of the brain section, the wavelength λ and the out-of-plane angle α [18]:

$$\delta = \frac{2\pi d}{\lambda} (n_e(\alpha) - n_o). \quad (\text{A.15})$$

The two refractive indices describe an index ellipsoid [18, 111] with semi-axis of length n_o and n_E , where n_E corresponds to the maximal extraordinary refractive index. By choosing the coordinate system in which the light propagates parallel to the z-direction and the birefringent histological section is fixed in the xy-plane, the extraordinary refractive index $n_e(\alpha)$ which is depending on the out-of-plane angle α of the nerve fiber can be described as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} n_e \cos(\alpha) \\ n_e \sin(\alpha) \end{pmatrix}. \quad (\text{A.16})$$

A plane in an index ellipsoid is an ellipse with a mathematical expression of

$$\frac{x^2}{n_o^2} + \frac{y^2}{n_E^2} = 1. \quad (\text{A.17})$$

By using Equation A.16 in Equation A.17 then

$$\frac{n_e \cos(\alpha)}{n_o^2} + \frac{n_e \sin(\alpha)}{n_E^2} = 1, \text{ resulting to} \quad (\text{A.18})$$

$$n_e(\alpha) = \frac{1}{\sqrt{\frac{\cos^2(\alpha)}{n_o^2} + \frac{\sin^2(\alpha)}{n_E^2}}}. \quad (\text{A.19})$$

It is assumed that the birefringence of the histological section $\Delta n = n_E - n_o$ is small, thus, the expression $n_e(\alpha) - n_o$ can be approximated by $\Delta n \cos^2(\alpha)$ [76]. Equation A.15 can now be expressed as

$$\delta \approx \frac{2\pi d}{\lambda} \Delta n \cos^2(\alpha). \quad (\text{A.20})$$

B. Derivation of the Natural-gradient Version of Infomax

In order to determine the unmixing matrix \mathbf{W} , it is mandatory to know the description of the density of $\tilde{\mathbf{S}}(k) = g(\tilde{\mathbf{s}}(k))$ in terms of $\tilde{\mathbf{s}}(k)$.

For a simplified understanding, a univariate density of $\tilde{\mathbf{S}}(k)$ is first taken into account which will be denoted as the density of $\tilde{S}(k)$. To find $f_{\tilde{S}(k)}$ for a specific $\tilde{S}(k)$, $\tilde{s}(k)$ is dissolved in its roots, i.e., $\tilde{s}(k) = \tilde{s}_1(k)$ [91, 103],

$$\tilde{S}(k) = g(\tilde{s}_1(k)). \quad (\text{B.1})$$

If now a small interval $\Delta\tilde{s}(k)$ around a value $\tilde{s}_1(k)$ is determined, the probability that $\tilde{s}(k)$ would be in the set of $\tilde{s}_1(k) - \frac{\Delta\tilde{s}(k)}{2}$ and $\tilde{s}_1(k) + \frac{\Delta\tilde{s}(k)}{2}$ is

$$\text{P}\left(\tilde{s}_1(k) - \frac{\Delta\tilde{s}(k)}{2} < \tilde{s}(k) \leq \tilde{s}_1(k) + \frac{\Delta\tilde{s}(k)}{2}\right) = f_{\tilde{s}(k)}(\tilde{s}_1(k)) \Delta\tilde{s}(k). \quad (\text{B.2})$$

Analogously, the probability that $\tilde{S}(k)$ is in the set of $\tilde{S}_1(k) - \frac{\Delta\tilde{S}(k)}{2}$ and $\tilde{S}_1(k) + \frac{\Delta\tilde{S}(k)}{2}$ is

$$\text{P}\left(\tilde{S}_1(k) - \frac{\Delta\tilde{S}(k)}{2} < \tilde{S}(k) \leq \tilde{S}_1(k) + \frac{\Delta\tilde{S}(k)}{2}\right) = f_{\tilde{S}(k)}(\tilde{S}_1(k)) \Delta\tilde{S}(k). \quad (\text{B.3})$$

Using the increasing monotonic nonlinear cost-function g the value $\tilde{s}_1(k)$ can be mapped to $\tilde{S}_1(k)$. Furthermore, $\Delta\tilde{S}(k)$ can be determined by $\Delta\tilde{s}(k)$ with

$$\Delta\tilde{S}(k) = g\left(\tilde{s}_1(k) + \frac{\Delta\tilde{s}(k)}{2}\right) - g\left(\tilde{s}_1(k) - \frac{\Delta\tilde{s}(k)}{2}\right). \quad (\text{B.4})$$

Due to the mapping the areas $\tilde{s}_1(k) \pm \frac{\Delta\tilde{s}(k)}{2}$ and $\tilde{S}_1(k) \pm \frac{\Delta\tilde{S}(k)}{2}$ are equivalent (see

shaded areas in Figure 3.3),

$$f_{\tilde{\mathbf{s}}(k)}(\tilde{s}_1(k)) \Delta \tilde{s}(k) = f_{\tilde{\mathbf{S}}(k)}(\tilde{\mathbf{S}}_1(k)) \Delta \tilde{\mathbf{S}}(k). \quad (\text{B.5})$$

Thus, with $\lim_{\Delta \tilde{\mathbf{s}}(k) \rightarrow 0} \frac{\Delta \tilde{\mathbf{S}}(k)}{\Delta \tilde{\mathbf{s}}(k)}$ the generalized form is given by

$$f_{\tilde{\mathbf{S}}(k)}(\tilde{\mathbf{S}}(k)) = \frac{f_{\tilde{\mathbf{s}}(k)}(\tilde{\mathbf{s}}(k))}{\left| \frac{d\tilde{\mathbf{S}}(k)}{d\tilde{\mathbf{s}}(k)} \right|}, \quad (\text{B.6})$$

where the absolute values of $\frac{d\tilde{\mathbf{S}}(k)}{d\tilde{\mathbf{s}}(k)}$ are considered for avoiding negative values in the derivative when a monotonically decreasing cost-function g is used [91, 103].

In case of multivariate densities of $\tilde{\mathbf{S}}(k)$, the derivative of $\frac{d\tilde{\mathbf{S}}(k)}{d\tilde{\mathbf{s}}(k)}$ is a Jacobian matrix \mathbf{J} [103] and the vertical bars $|\cdot|$ describe the absolute value of the determinant \mathbf{J} ,

$$f_{\tilde{\mathbf{S}}(k)}(\tilde{\mathbf{S}}(k)) = \frac{f_{\tilde{\mathbf{s}}(k)}(\tilde{\mathbf{s}}(k))}{|\mathbf{J}|}, \text{ with} \quad (\text{B.7})$$

$$|\mathbf{J}| = \left| \frac{d\tilde{\mathbf{S}}(k)}{d\tilde{\mathbf{s}}(k)} \right| = g'(\tilde{\mathbf{s}}(k)). \quad (\text{B.8})$$

If now an unmixing matrix \mathbf{W} exists that maximizes the entropy of the mutual information $I(\tilde{\mathbf{S}}(k), \tilde{\mathbf{s}}(k)) = H(\tilde{\mathbf{S}}(k)) - H(\tilde{\mathbf{S}}(k) | \tilde{\mathbf{s}}(k))$ (Eq. 3.14) and is involved in the mapping from $\tilde{\mathbf{s}}(k)$ to $\tilde{\mathbf{S}}(k)$, $I(\tilde{\mathbf{S}}(k), \tilde{\mathbf{s}}(k))$ can be differentiated as follows:

$$\frac{d}{d\mathbf{W}} I(\tilde{\mathbf{S}}(k), \tilde{\mathbf{s}}(k)) = \frac{d}{d\mathbf{W}} H(\tilde{\mathbf{S}}(k)), \quad (\text{B.9})$$

where $H(\tilde{\mathbf{S}}(k) | \tilde{\mathbf{s}}(k))$ is a noise term and in the absence of noise the maximization of the mutual information is equal to the maximization of the entropy of $\tilde{\mathbf{S}}(k)$. If a change of variable is performed, using $f_{\tilde{\mathbf{s}}(k)} d\tilde{\mathbf{s}}(k) = f_{\tilde{\mathbf{S}}(k)} d\tilde{\mathbf{S}}(k)$ (Eq. B.5) with $d\tilde{\mathbf{S}}(k) = g'(\tilde{\mathbf{s}}(k)) d\tilde{\mathbf{s}}(k)$ (Eq. B.8), the entropy $H(\tilde{\mathbf{S}}(k))$ will be equal to the Kullback-Leibler distance of the pdf $f_{\tilde{\mathbf{s}}(k)}$ to the pdf $g'(\tilde{\mathbf{s}}(k))$ [86]:

$$\frac{d}{d\mathbf{W}} H(\tilde{\mathbf{S}}(k)) = \frac{d}{d\mathbf{W}} \left(- \int_{-\infty}^{\infty} f_{\tilde{\mathbf{s}}(k)} \log \left(\frac{f_{\tilde{\mathbf{s}}(k)}}{g'(\tilde{\mathbf{s}}(k))} \right) d\tilde{\mathbf{s}}(k) \right). \quad (\text{B.10})$$

Thus, maximizing the mutual information is equivalent to minimizing the Kullback-Leibler distance which in return infers the statistical independence of the signals

$\tilde{\mathbf{s}}(k)$.

Since the mixing matrix \mathbf{A} is nonsingular the above equation is replaced by the following natural gradient descent equation [3]:

$$\frac{d}{d\mathbf{W}}H(\tilde{\mathbf{S}}(k)) = \tau \frac{d}{d\mathbf{W}} \left(- \int_{-\infty}^{\infty} f_{\tilde{\mathbf{s}}(k)} \log \left(\frac{f_{\tilde{\mathbf{s}}(k)}}{g'(\tilde{\mathbf{s}}(k))} \right) d\tilde{\mathbf{s}}(k) \right) \mathbf{W}^T \mathbf{W}, \quad (\text{B.11})$$

where τ is the learning rate. If $E[\cdot]$ denotes the expected value equation B.11 gives

$$\frac{d}{d\mathbf{W}}H(\tilde{\mathbf{S}}(k)) = \frac{d}{d\mathbf{W}} (E[\tau \log g'(\tilde{\mathbf{s}}(k))] - E[\tau \log f_{\tilde{\mathbf{s}}(k)}]) \mathbf{W}^T \mathbf{W}. \quad (\text{B.12})$$

Because the term $E[\tau \log f_{\tilde{\mathbf{s}}(k)}]$ is considered unaffected by the changes of \mathbf{W} determining g , only the term $E[\tau \log g'(\tilde{\mathbf{s}}(k))]$ needs to be maximized [15]. Replacing the expectation values by their instantaneous values the derivation of $\tau \log g'(\tilde{\mathbf{s}}(k))$ is

$$\frac{d}{d\mathbf{W}} (\tau \log g'(\tilde{\mathbf{s}}(k))) = \tau (g'(\tilde{\mathbf{s}}(k)))^{-1} \frac{d}{d\mathbf{W}} (g'(\tilde{\mathbf{s}}(k))). \quad (\text{B.13})$$

If a sigmoidal logistic function $g(\tilde{\mathbf{s}}(k)) = \frac{1}{1+e^{-\tilde{\mathbf{s}}(k)}}$, $\tilde{\mathbf{s}}(k) = \mathbf{W} \cdot \mathbf{x}(k) + \mathbf{W}_0$, is used, the above terms will be evaluated as

$$g'(\tilde{\mathbf{s}}(k)) = \mathbf{W} \cdot g(\tilde{\mathbf{s}}(k)) (1 - g(\tilde{\mathbf{s}}(k))) \text{ and} \quad (\text{B.14})$$

$$\frac{d}{d\mathbf{W}} (g'(\tilde{\mathbf{s}}(k))) = g(\tilde{\mathbf{s}}(k)) (1 - g(\tilde{\mathbf{s}}(k))) [\mathbf{I} + \mathbf{W} \cdot \tilde{\mathbf{s}}(k) (1 - 2g(\tilde{\mathbf{s}}(k)))] \quad (\text{B.15})$$

Dividing Equation B.15 by Equation B.14 the following rule is calculated:

$$\Delta \mathbf{W} = \frac{d}{d\mathbf{W}}H(\tilde{\mathbf{S}}(k)) = \tau \left[(\mathbf{W}^T)^{-1} + (1 - 2g(\tilde{\mathbf{s}}(k))) \tilde{\mathbf{s}}(k) \right] \mathbf{W}^T \mathbf{W} \quad (\text{B.16})$$

$$= \tau \left[\mathbf{I} + (1 - 2g(\tilde{\mathbf{s}}(k))) \tilde{\mathbf{s}}(k)^T \right] \mathbf{W}. \quad (\text{B.17})$$

Analogously, the bias weight \mathbf{W}_0 is described without the learning rate τ by:

$$\Delta \mathbf{W}_0 = \mathbf{I} (1 - 2g(\tilde{\mathbf{s}}(k))). \quad (\text{B.18})$$

C. Nongaussianity for Source Signals in Independent Component Analysis

Nongaussian distributions for the source signals \mathbf{s} is a mandatory requirement in ICA. For understanding why gaussian variables are forbidden in ICA already whitened data is assumed.

Furthermore, it is assumed that the joint distribution of two source components s_1 and s_2 is gaussian. Thus, the joint pdf is described by

$$f_{s_1 s_2}(s_1, s_2) = \frac{1}{2\pi} e^{-\frac{s_1^2 + s_2^2}{2}} = \frac{1}{2\pi} e^{-\frac{\|\mathbf{s}\|^2}{2}}. \quad (\text{C.1})$$

Furthermore, lets assume that the mixing matrix \mathbf{A} is orthogonal as the data is already whitened. By using the change of pdf's in Equation B.7 and by using the rule of a orthogonal matrix $\mathbf{A}^{-1} = \mathbf{A}^T$, the above equation is formulated as the joint pdf of x_1 and x_2 :

$$f_{x_1 x_2}(x_1, x_2) = f_{s_1 s_2}(s_1, s_2) |\mathbf{A}^{-1}|, \quad (\text{C.2})$$

$$f_{x_1 x_2}(x_1, x_2) = \frac{1}{2\pi} e^{-\frac{\|\mathbf{A}^T \cdot \mathbf{x}\|^2}{2}} |\mathbf{A}^T|, \text{ with} \quad (\text{C.3})$$

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s} \iff \mathbf{s} = \mathbf{A}^{-1} \cdot \mathbf{x} \iff \mathbf{s} = \mathbf{A}^T \cdot \mathbf{x}, \quad (\text{C.4})$$

where $|\mathbf{A}^T|$ denotes the absolute value of the determinant of \mathbf{A}^T . Due to the fact that \mathbf{A} is orthogonal, the determinant $|\mathbf{A}^T| = 1$, i.e., $\|\mathbf{A}^T \cdot \mathbf{x}\|^2 = \|\mathbf{x}\|^2$. Hence, the joint pdf's of \mathbf{s} and \mathbf{x} are equal and the above equation reduces to

$$f_{x_1x_2}(x_1, x_2) = \frac{1}{2\pi} e^{\left(-\frac{\|\mathbf{x}\|^2}{2}\right)}, \quad (\text{C.5})$$

which proves that the orthogonal mixing matrix \mathbf{A} does not change the joint pdf as it is not included in the formula. Thus, the pdfs of the source signals and the acquired data are identical [56].

The joint pdf of x_1 and x_2 is illustrated in the following figure (Fig. C.1).

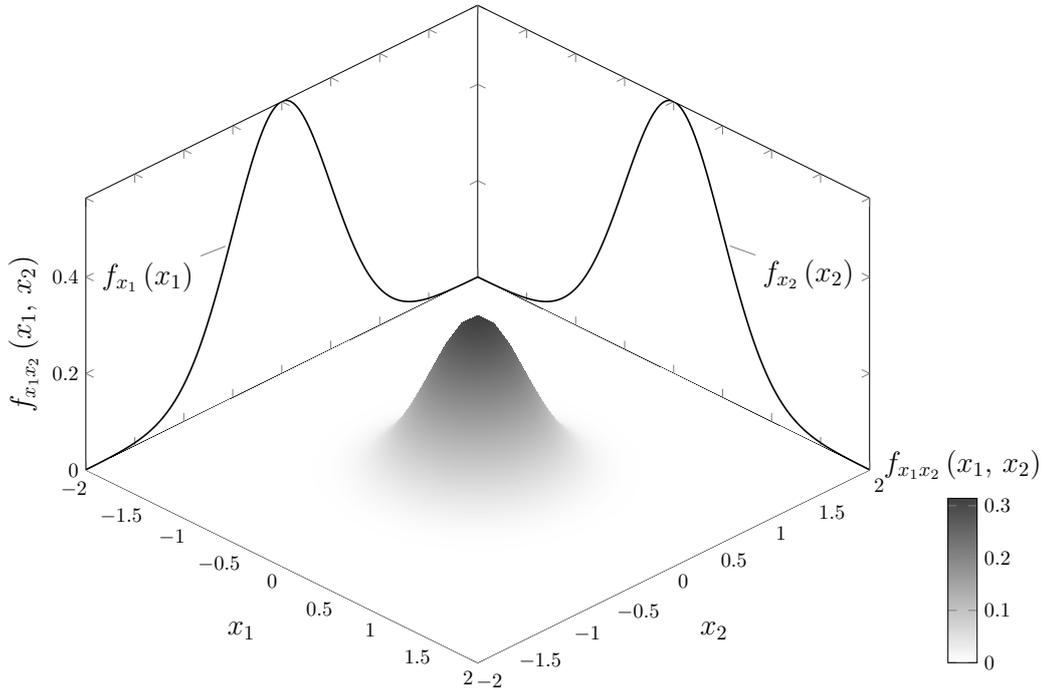


Figure C.1.: Joint pdf of two independent gaussian variables x_1 and x_2 . The joint pdf is rotationally symmetric, thus, no information on the directions of the columns of the mixing matrix \mathbf{A} can be inferred, i.e., \mathbf{A} is not determinable for gaussian variables. This results to an identical joint pdf of any orthogonalization of two gaussian distributed source signals s_1 and s_2 . Hence, x_1 and x_2 are statistically independent.

D. Basis vectors

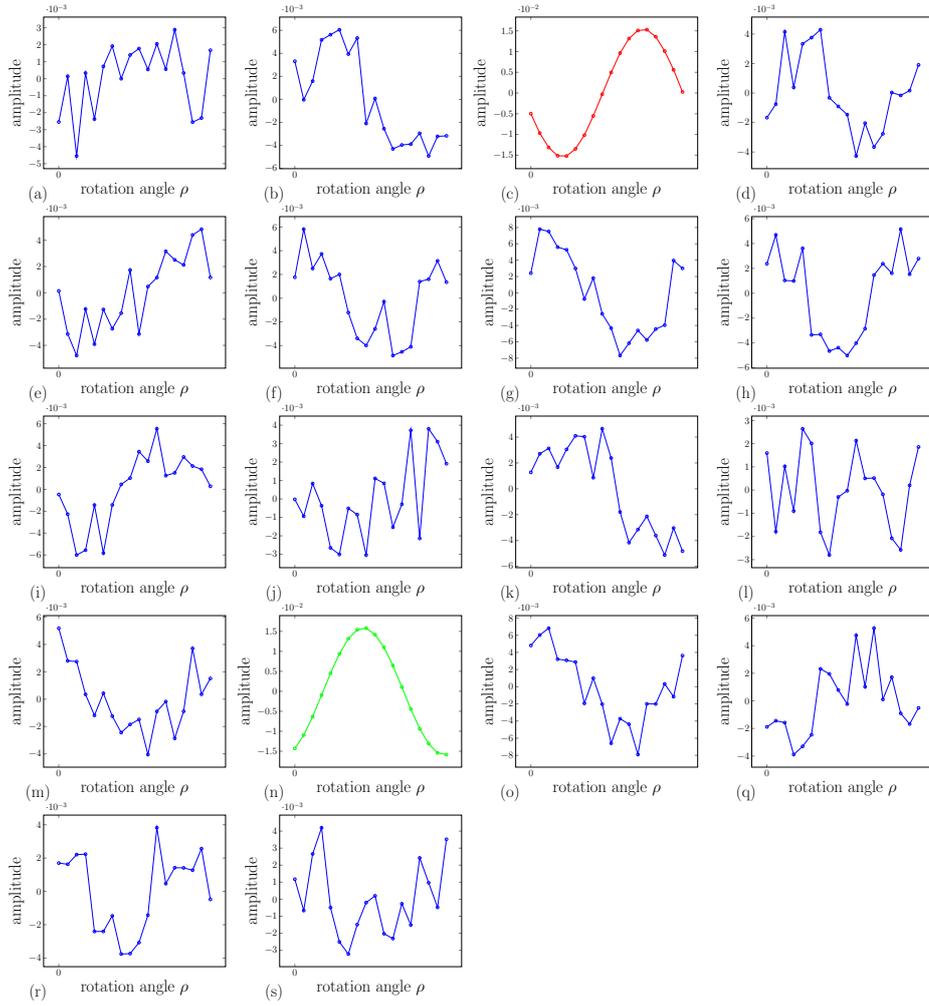


Figure D.1.: Basis vectors $\hat{a}_j^*(\rho)$ of the mixing matrix $\hat{\mathbf{A}}_c^*$ after completion of the acICA algorithm applied on pcWM. The identified signal components are highlighted in red and green.

D. Basis vectors

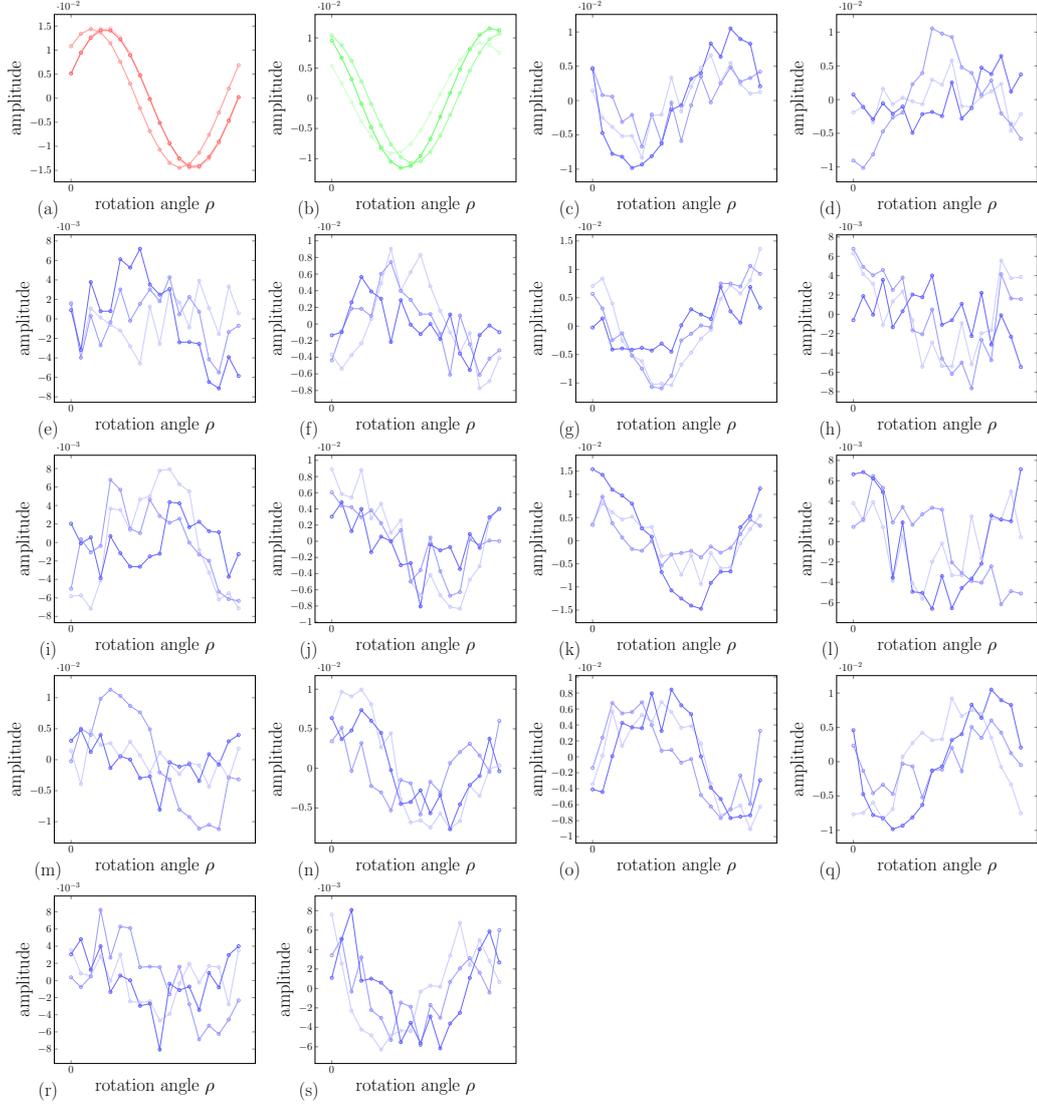


Figure D.2.: Basis vectors $\hat{a}_{j, \text{rank}}^*(\rho)$ of the mixing matrices $\hat{\mathbf{A}}_{c, \text{rank}}^*$ from all MPI processes after completion of the pICAP algorithm and the clustering applied on pcWM. The identified clustered signal components are highlighted in red and green. The basis vectors of each cluster are colored gradually in a transparent manner.

List of Figures

2.1.	Measurement setup for the LAP and the PM in 3D-PLI.	6
2.2.	Example of light intensities of two pixels at different rotation angles ρ	7
2.3.	Contrast enhanced flat images before and after calibration of the LAP.	11
2.4.	Measured light intensities of one pixel at different rotation angles ρ	14
2.5.	The stitching procedure combines all tiles together with the best correspondences.	16
2.6.	A color coded FOM in HSV color space of a transversal brain section acquired in the LAP.	18
3.1.	Schematic illustration of the mixing process in 3D-PLI and the decomposition in sICA (adapted from [21]).	20
3.2.	Block diagram showing the linearized BSS problem.	23
3.3.	Schematic diagram of the principle of Infomax where a uniform distribution is obtained if the transformation of the signal matches its own cdf (adapted from [15, 103]).	28
3.4.	Joint distribution of the first two centered images of one PLI image series \mathbf{x}_1 and \mathbf{x}_2 where only the first 5000 pixel values are shown.	32
3.5.	Joint distribution of the first two centered images of one PLI image series $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ after whitening.	32
3.6.	Whitened PLI image series.	34
3.7.	Two exemplary basis vectors with their theoretically expected functions $f(u_\rho)$	38
3.8.	ICA decomposed PLI image series.	40
4.1.	A schematic illustration of SISD, SIMD, MISD and MIMD.	45
4.2.	A master-slave setup with a task and communication workflow (adapted from [102]).	47
4.3.	A data decomposition strategy in which the core data structure is divided into chunks.	49

4.4.	Workflow of a pipeline strategy.	50
5.1.	A schematic of 1182 coronar sections from a post-mortem vervet brain, where 140 samples were taken and acquired in the LAP and PM.	56
5.2.	A histogram of one transmittance map of the 50 repro series with 130 bins (a) and its criterion function $J(T)$ (b).	61
5.3.	Percentiles of the white matter distribution of the transmittance map ($\text{percentile}_{\text{transmittance}}$) and retardation map ($\text{percentile}_{\text{retardation}}$) were processed by cICAP.	62
5.4.	Comparison of cdfs of GM (a) and pcWM (b) generated from the 140 PLI image series of the LAP.	62
5.5.	Comparison of cdfs of GM (a) and pcWM (b) generated from the 20 PLI image series of the PM.	63
5.6.	$wrGOF$ values after the standard cICAP and the new acICA algorithm.	65
5.7.	Measured light intensities of one pixel at different rotation angles ρ before and after acICA.	66
6.1.	Parameter benchmark of the confidence value η , the threshold value t and the stopping criterion ε	73
7.1.	Disjoint data partitions read in a cyclic manner.	80
7.2.	Scheme of the estimation procedure for generating an overall mixing matrix $\hat{\mathbf{A}}_{tot}^*$ in pICAP.	82
7.3.	Scheme of the reconstruction of the identified source components back into the PLI space.	83
7.4.	Runtime measurements of the pICAP algorithm	91
7.5.	Total runtime shares starting from the parallel reading to the parallel writing of tiled PLI image series of a complete pICAP application with the runtime optimal 50 MPI processes.	92
7.6.	Elapsed time on the GPU and CPU	93
7.7.	$wrGOF$ values per pixel generated after the new pICAP algorithm and the sequential acICA applied on GM and pcWM.	95
7.8.	Basis vectors of the acICA algorithm for the GM	96
7.9.	Basis vectors of the pICAP algorithm for the GM	97
C.1.	Joint pdf of two independent gaussian variables x_1 and x_2	118

D.1. Basis vectors of the acICA algorithm for the pcWM 119
D.2. Basis vectors of the pICAP algorithm for the pcWM 120

List of Tables

2.1. Definition of variables and abbreviations in this chapter.	3
3.1. Definition of variables and abbreviations in this chapter.	20
4.1. Definition of abbreviations in this chapter.	42
4.2. System configuration of JUDGE (adapted from [59])	43
4.3. Flynn's taxonomy classifying the four categories: Single Instruction - Single Data (SISD), Single Instruction - Multiple Data (SIMD), Multiple Instruction - Single Data (MISD) and Multiple Instruction - Multiple Data (MIMD) architecture.	44
5.1. Definition of variables and abbreviations in this chapter.	54
5.2. Parameters for the GM and pcWM of the LAP and PM.	63
5.3. A comparison of number of iterations from acICA, cICAP and Infomax.	64
5.4. A comparison of wrGOFs from acICA, cICAP and Infomax.	64
6.1. Definition of variables and abbreviations in this chapter.	70
6.2. Mean, median and standard deviation of the optimized parameters η and t for GM and pcWM.	72
7.1. Definition of variables and abbreviations in this chapter.	78
7.2. Size of data for different steps of the 3D-PLI workflow.	89

List of Algorithms

1. Constrained Independent Component Analysis for 3D-PLI 36
2. Incorporation of prior information 37
3. Parallelized Independent Component Analysis for 3D-PLI (pICAP) . 85

Bibliography

- [1] S. Amari. “Natural Gradient Works Efficiently in Learning.” In: *Neural Comput.* (1998). DOI: 10.1162/089976698300017746 (cit. on p. 27).
- [2] S. Amari and S. C. Douglas. “Why natural gradient?” In: *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. Vol. 2. 1998, 1213–1216 vol.2. DOI: 10.1109/ICASSP.1998.675489 (cit. on p. 28).
- [3] S. Amari, A. Cichocki, and H. H. Yang. “A New Learning Algorithm for Blind Signal Separation.” In: *Advances in Neural Information Processing Systems*. MIT Press, 1996, pp. 757–763 (cit. on pp. 27, 115).
- [4] J. Atick and A. Redlich. “Convergent Algorithm for Sensory Receptive Field Development.” In: *Neural Computation* 5.1 (1993), pp. 45–60. DOI: 10.1162/neco.1993.5.1.45 (cit. on p. 19).
- [5] H. Axer and D. Graf v. Keyserlingk. “Mapping of fiber orientation in human internal capsule by means of polarized light and confocal scanning laser microscopy.” In: *Journal of Neuroscience Methods* 94.2 (2000), pp. 165–175. DOI: [http://dx.doi.org/10.1016/S0165-0270\(99\)00132-6](http://dx.doi.org/10.1016/S0165-0270(99)00132-6) (cit. on p. 1).
- [6] H. Axer, S. Beck, M. Axer, F. Schuchardt, J. Heepe, A. Fluecken, M. Axer, A. Prescher, and O. W. Witte. “Microstructural analysis of human white matter architecture using Polarized Light Imaging (PLI): Views from neuroanatomy.” In: *Frontiers in Neuroinformatics* 5.28 (2011). DOI: 10.3389/fninf.2011.00028 (cit. on pp. 1, 3).
- [7] H. Axer, M. Axer, T. Krings, and D. Graf v. Keyserlingk. “Quantitative estimation of 3-D fiber course in gross histological sections of the human brain using polarized light.” In: *Journal of Neuroscience Methods* 105.2 (2001), pp. 121–131. DOI: [http://dx.doi.org/10.1016/S0165-0270\(00\)00349-6](http://dx.doi.org/10.1016/S0165-0270(00)00349-6) (cit. on p. 1).

- [8] M. Axer, K. Amunts, D. Grässel, C. Palm, J. Dammers, H. Axer, U. Pietrzyk, and K. Zilles. “A novel approach to the human connectome: Ultra-high resolution mapping of fiber tracts in the brain.” In: *NeuroImage* 54.2 (2011), pp. 1091–1101. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2010.08.075> (cit. on pp. 1, 3, 5, 9, 10, 13, 17).
- [9] M. Axer, D. Grässel, M. Kleiner, J. Dammers, T. Dickscheid, J. Reckfort, T. Hütz, B. Eiben, U. Pietrzyk, K. Zilles, and K. Amunts. “High-resolution fiber tract reconstruction in the human brain by means of three-dimensional polarized light imaging (3D-PLI).” In: *Frontiers in Neuroinformatics* 5.34 (2011). DOI: [10.3389/fninf.2011.00034](https://doi.org/10.3389/fninf.2011.00034) (cit. on pp. 1, 3, 5, 7, 58).
- [10] E. S. Barriga, M. Pattichis, D. Ts’o, M. Abramoff, R. Kardon, Y. Kwon, and P. Soliz. “Independent component analysis using prior information for signal detection in a functional imaging system of the retina.” In: *Medical Image Analysis* 15.1 (2011), pp. 35–44. DOI: <http://dx.doi.org/10.1016/j.media.2010.06.009> (cit. on p. 101).
- [11] M. S. Bartlett. “Information maximization in face processing.” In: *Neurocomputing* 70.13 (2007), pp. 2204–2217 (cit. on p. 1).
- [12] S. Becker and G. E. Hinton. “Self-organizing neural network that discovers surfaces in random-dot stereograms.” In: *Nature* 355.6356 (1992), pp. 161–163 (cit. on p. 19).
- [13] C. F. Beckmann and S. M. Smith. “Probabilistic independent component analysis for functional magnetic resonance imaging.” In: *IEEE Trans. Med. Imaging* 23.2 (2004), pp. 137–152 (cit. on p. 1).
- [14] C. F. Beckmann and S. M. Smith. “Tensorial extensions of independent component analysis for multisubject fMRI analysis.” In: *Neuroimage* 25.1 (2005), pp. 294–311 (cit. on p. 1).
- [15] A. J. Bell and T. J. Sejnowski. “An Information-maximization Approach to Blind Separation and Blind Deconvolution.” In: *Neural Comput.* 7.6 (1995), pp. 1129–1159. DOI: [10.1162/neco.1995.7.6.1129](https://doi.org/10.1162/neco.1995.7.6.1129) (cit. on pp. 2, 19, 23, 28, 29, 67, 115).
- [16] U. S. Bhalla and J. M. Bower. “Exploring parameter space in detailed single neuron models: simulations of the mitral and granule cells of the olfactory bulb.” In: *Journal of Neurophysiology* 69.6 (1993), pp. 1948–1965 (cit. on p. 46).

-
- [17] B. B. Biswal, P. A. Taylor, and J. L. Ulmer. “Use of Jackknife Resampling Techniques to Estimate the Confidence Intervals of fMRI Parameters.” In: *Journal of Computer Assisted Tomography* 25.1 (2001), pp. 113–120 (cit. on p. 59).
- [18] M. Born and E. Wolf. *Principles of Optics*. 7th ed. Cambridge University Press, 1999 (cit. on pp. 110, 111).
- [19] R. N. Boubela, W. Huf, K. Kalcher, R. Sladky, P. Filzmoser, L. Pezawas, S. Kasper, C. Windischberger, and E. Moser. “A highly parallelized framework for computationally intensive MR data analysis.” In: *Magnetic Resonance Materials in Physics, Biology and Medicine* 25.4 (2012), pp. 313–320. DOI: 10.1007/s10334-011-0290-7 (cit. on pp. 77, 98).
- [20] L. Breuer, J. Dammers, T. P. L. Roberts, and N. J. Shah. “A Constrained ICA Approach for Real-Time Cardiac Artifact Rejection in Magnetoencephalography.” In: *Biomedical Engineering, IEEE Transactions on* 61.2 (2014), pp. 405–414. DOI: 10.1109/TBME.2013.2280143 (cit. on pp. 53, 58, 101).
- [21] L. Breuer, M. Axer, and J. Dammers. “A new constrained {ICA} approach for optimal signal decomposition in polarized light imaging.” In: *Journal of Neuroscience Methods* 220.1 (2013), pp. 30–38. DOI: <http://dx.doi.org/10.1016/j.jneumeth.2013.08.022> (cit. on pp. 2, 19, 20, 35, 36, 38, 39, 53, 67, 69, 71, 72, 74, 75, 101, 102).
- [22] G. D. Brown, S. Yamada, and T. J. Sejnowski. “Independent component analysis at the neural cocktail party.” In: *Trends in Neuroscience* 24.1 (2001), pp. 54–63 (cit. on pp. 27, 28).
- [23] V. D. Calhoun, J. Liu, and T. Adali. “A review of group {ICA} for fMRI data and {ICA} for joint inference of imaging, genetic, and {ERP} data.” In: *NeuroImage* 45.1, Supplement 1 (2009). Mathematics in Brain Imaging, pp. 163–172. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2008.10.057> (cit. on pp. 79, 98).
- [24] C. E. Cherry. “Some Experiments on the Recognition of Speech, with One and with Two Ears.” In: *Journal of the Acoustical Society of America* 25.5 (1953), pp. 975–979. DOI: 10.1121/1.1907229 (cit. on p. 22).
- [25] D. Child. *Essentials of factor analysis*. 3rd ed. Previous ed.: London: Cassell, 1990. New York : Continuum, 2006 (cit. on p. 19).

- [26] A. Cichocki and S. Amari. *Adaptive blind signal and image processing: learning algorithms and applications*. Vol. 1. John Wiley & Sons, 2002 (cit. on pp. 23, 31, 35).
- [27] P. Comon. “Independent component analysis, a new concept?” In: *Signal processing* 36.3 (1994), pp. 287–314 (cit. on pp. 19, 22).
- [28] L. Dalcín, R. Paz, M. Storti, and J. D’Elía. “MPI for Python: Performance Improvements and MPI-2 Extensions.” In: *J. Parallel Distrib. Comput.* 68.5 (2008), pp. 655–662. DOI: 10.1016/j.jpdc.2007.09.005 (cit. on p. 52).
- [29] J. Dammers and M. Schiek. “Detection of Artifacts and Brain Responses Using Instantaneous Phase Statistics in Independent Components.” In: *Intech* (2011). DOI: 10.5772/27523 (cit. on pp. 53, 101).
- [30] J. Dammers, L. Breuer, M. Axer, M. Kleiner, B. Eiben, D. Graessel, T. Dickscheid, K. Zilles, K. Amunts, N. J. Shah, and U. Pietrzyk. “Automatic identification of gray and white matter components in polarized light imaging.” In: *NeuroImage* 59.2 (2012), pp. 1338–1347. DOI: 10.1016/j.neuroimage.2011.08.030 (cit. on pp. 1, 13, 19, 37, 53, 101).
- [31] J. Dammers, M. Schiek, F. Boers, C. Silex, M. Zvyagintsev, U. Pietrzyk, and K. Mathiak. “Integration of Amplitude and Phase Statistics for Complete Artifact Removal in Independent Components of Neuromagnetic Recordings.” In: *IEEE Trans. Biomed. Engineering* 55.10 (2008), pp. 2353–2362 (cit. on pp. 1, 53, 101).
- [32] J. Dammers, M. Axer, D. Graessel, C. Palm, K. Zilles, K. Amunts, and U. Pietrzyk. “Signal enhancement in polarized light imaging by means of independent component analysis.” In: *NeuroImage* 49.2 (2010), pp. 1241–1248. DOI: 10.1016/j.neuroimage.2009.08.059 (cit. on pp. 1, 10, 12, 19, 53, 101).
- [33] A. Delorme, T. Sejnowski, and S. Makeig. “Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis.” In: *NeuroImage* 34.4 (2007), pp. 1443–1449. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2006.11.004> (cit. on p. 1).
- [34] M. Dubois, M. Annavaram, and P. Stenström. *Parallel Computer Organization and Design*. New York, NY, USA: Cambridge University Press, 2012 (cit. on p. 46).

-
- [35] J. Escudero, R. Hornero, D. Abásolo, and A. Fernández. “Quantitative Evaluation of Artifact Removal in Real Magnetoencephalogram Signals with Blind Source Separation.” In: *Annals of Biomedical Engineering* 39.8 (2011), pp. 2274–2286. DOI: 10.1007/s10439-011-0312-7 (cit. on pp. 53, 101).
- [36] F. Esposito, T. Scarabino, A. Hyvarinen, J. Himberg, E. Formisano, S. Comani, G. Tedeschi, R. Goebel, E. Seifritz, and F. Di Salle. “Independent component analysis of fMRI group studies by self-organizing clustering.” In: *Neuroimage* 25.1 (2005), pp. 193–205 (cit. on pp. 1, 79, 98).
- [37] M. J. Flynn and K. W. Rudd. “Parallel Architectures.” In: *ACM Comput. Surv.* 28.1 (1996), pp. 67–70. DOI: 10.1145/234313.234345 (cit. on pp. 44, 46).
- [38] M. Flynn. “Computer Architecture.” In: *Wiley Encyclopedia of Computer Science and Engineering*. John Wiley & Sons, Inc., 2007. DOI: 10.1002/9780470050118.ecse071 (cit. on p. 45).
- [39] M. Flynn. “Some Computer Organizations and Their Effectiveness.” In: *Computers, IEEE Transactions on* C-21.9 (1972), pp. 948–960. DOI: 10.1109/TC.1972.5009071 (cit. on p. 43).
- [40] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 3.0*. Chapter author for Collective Communication, Process Topologies, and One Sided Communications. Sept. 2012 (cit. on pp. 51, 52).
- [41] I. Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995 (cit. on p. 50).
- [42] D. Freedman and P. Diaconis. “On the histogram as a density estimator: L2 theory.” In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 57.4 (1981), pp. 453–476. DOI: 10.1007/BF01025868 (cit. on p. 57).
- [43] J. H. Friedman and J. W. Tukey. “A Projection Pursuit Algorithm for Exploratory Data Analysis.” In: *Computers, IEEE Transactions on* C-23.9 (1974), pp. 881–890. DOI: 10.1109/T-C.1974.224051 (cit. on p. 19).
- [44] R. M. Gray. *Entropy and Information Theory*. SpringerLink : Bücher. Springer, 2011 (cit. on p. 25).
- [45] G. F. Göthlin. “Die doppelbrechenden Eigenschaften des Nervengewebes.” In: *Kungliga Svenska Vetenskapsakademiens* 51 (1913) (cit. on p. 1).

- [46] *HDF5 for Python*. <http://www.h5py.org/>. Accessed: 2015-05-01 (cit. on p. 81).
- [47] J. Héroult, C. Jutten, and B. Ans. “Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé.” In: *Proc. Xéme colloque GRETSI* (1985) (cit. on p. 22).
- [48] C. W. Hesse and C. J. James. “On Semi-Blind Source Separation Using Spatial Constraints With Applications in EEG Analysis.” In: *Biomedical Engineering, IEEE Transactions on* 53.12 (2006), pp. 2525–2534. DOI: 10.1109/TBME.2006.883796 (cit. on p. 101).
- [49] K. E. Hild and S. S. Nagarajan. “Source Localization of EEG/MEG Data by Correlating Columns of ICA and Lead Field Matrices.” In: *IEEE transactions on bio-medical engineering* (2009) (cit. on p. 53).
- [50] J. Himberg, A. Hyvärinen, and F. Esposito. “Validating the independent components of neuroimaging time series via clustering and visualization.” In: *NeuroImage* 22.3 (2004), pp. 1214–1222. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2004.03.027> (cit. on pp. 68, 79, 98).
- [51] H. Hotelling. “Analysis of a complex of statistical variables into principal components.” In: *J. Educ. Psych.* 24 (1933) (cit. on p. 19).
- [52] D. Huang and J. Mi. “A New Constrained Independent Component Analysis Method.” In: *Neural Networks, IEEE Transactions on* 18.5 (2007), pp. 1532–1535. DOI: 10.1109/TNN.2007.895910 (cit. on p. 101).
- [53] K. Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. 1st. McGraw-Hill Higher Education, 1992 (cit. on p. 46).
- [54] A. Hyvärinen. “Testing the ICA mixing matrix based on inter-subject or inter-session consistency.” In: *NeuroImage* 58.1 (2011), pp. 122–136. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2011.05.086> (cit. on pp. 80, 81, 84, 88).
- [55] A. Hyvärinen and E. Oja. “Independent component analysis: algorithms and applications.” In: *Neural Networks* 13 (2000), pp. 411–430 (cit. on pp. 19, 29, 31, 36).

-
- [56] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2004 (cit. on pp. 25, 29, 30, 118).
- [57] A. Hyvärinen, J. Hurri, and P. O. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. 1st. Springer Publishing Company, Incorporated, 2009 (cit. on pp. 29, 35).
- [58] A. Hyvärinen and P. Ramkumar. “Testing independent component patterns by inter-subject or inter-session consistency.” In: *Frontiers in Human Neuroscience* 7.94 (2013). DOI: 10.3389/fnhum.2013.00094 (cit. on pp. 80, 81, 88).
- [59] *IBM iDataPlex Cluster JUDGE*. http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUDGE/Configuration/Configuration_node.html. Accessed: 2015-05-01 (cit. on p. 43).
- [60] R. N. Ibbett and N. P. Topham. *Architecture of High Performance Computers Volume II: Array processors and multiprocessor systems*. Springer New York, 1989 (cit. on p. 44).
- [61] A. J. Izenman. “Review Papers: Recent Developments in Nonparametric Density Estimation.” In: *Journal of the American Statistical Association* 86.413 (1991), pp. 205–224. DOI: 10.1080/01621459.1991.10475021 (cit. on p. 57).
- [62] *JSC supercomputers*. http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/supercomputers_node.html. Accessed: 2015-05-01 (cit. on p. 41).
- [63] J. Jeffers and J. Reinders. *Intel Xeon Phi Coprocessor High Performance Programming*. 1st. Morgan Kaufmann Publishers Inc., 2013 (cit. on p. 44).
- [64] R. C. Jones. “A New Calculus for the Treatment of Optical Systems. IV.” In: *J. Opt. Soc. Am.* 32.8 (1942), pp. 486–493. DOI: 10.1364/JOSA.32.000486 (cit. on pp. 3, 8).
- [65] H. Hurwitz Jr and R. C. Jones. “A New Calculus for the Treatment of Optical Systems.” In: *J. Opt. Soc. Am.* 31.7 (1941), pp. 493–495. DOI: 10.1364/JOSA.31.000493 (cit. on pp. 3, 8).

- [66] T. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. J. Sejnowski. “Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects.” In: *Clinical Neurophysiology* 111.10 (2000), pp. 1745–1758. DOI: [http://dx.doi.org/10.1016/S1388-2457\(00\)00386-2](http://dx.doi.org/10.1016/S1388-2457(00)00386-2) (cit. on pp. 53, 101).
- [67] T. Jung, S. Makeig, C. Humphries, T. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski. “Removing electroencephalographic artifacts by blind source separation.” In: *Psychophysiology* 37 (02 2000), pp. 163–178. DOI: null (cit. on p. 1).
- [68] C. Jutten and J. Herault. “Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture.” In: *Signal Processing* 24.1 (1991), pp. 1–10. DOI: [http://dx.doi.org/10.1016/0165-1684\(91\)90079-X](http://dx.doi.org/10.1016/0165-1684(91)90079-X) (cit. on pp. 19, 22).
- [69] D. B. Keith, C. C. Hoge, R. M. Frank, and A. D. Malony. “Parallel ICA methods for EEG neuroimaging.” In: *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*. 2006, 10 pp.–. DOI: [10.1109/IPDPS.2006.1639299](https://doi.org/10.1109/IPDPS.2006.1639299) (cit. on p. 77).
- [70] J. W. Kelly, D. P. Siewiorek, A. Smailagic, J. L. Collinger, D. J. Weber, and W. Wang. “Fully Automated Reduction of Ocular Artifacts in High-Dimensional Neural Data.” In: *Biomedical Engineering, IEEE Transactions on* 58.3 (2011), pp. 598–606. DOI: [10.1109/TBME.2010.2093932](https://doi.org/10.1109/TBME.2010.2093932) (cit. on pp. 53, 101).
- [71] J. Kittler and J. Illingworth. “Minimum error thresholding.” In: *Pattern Recognition* 19.1 (1986), pp. 41–47. DOI: [http://dx.doi.org/10.1016/0031-3203\(86\)90030-0](http://dx.doi.org/10.1016/0031-3203(86)90030-0) (cit. on pp. 56, 57).
- [72] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih. “PyCUDA and PyOpenCL: A Scripting-Based Approach to GPU Run-Time Code Generation.” In: *Parallel Computing* 38.3 (2012), pp. 157–174. DOI: [10.1016/j.parco.2011.09.001](https://doi.org/10.1016/j.parco.2011.09.001) (cit. on pp. 51, 103).
- [73] S. Kullback and R. A. Leibler. “On Information and Sufficiency.” In: *Ann. Math. Statist.* 22.1 (1951), pp. 79–86. DOI: [10.1214/aoms/1177729694](https://doi.org/10.1214/aoms/1177729694) (cit. on p. 26).

-
- [74] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. “Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions.” In: *SIAM Journal of Optimization* 9 (1998), pp. 112–147 (cit. on p. 71).
- [75] J. C. Lagarias, B. Poonen, and M. H. Wright. “Convergence of the Restricted Nelder-Mead Algorithm in Two Dimensions.” In: *SIAM Journal on Optimization* 22.2 (2012), pp. 501–532 (cit. on p. 71).
- [76] L. Larsen, L. D. Griffin, D. Gräkel, O. W. Witte, and H. Axer. “Polarized light imaging of white matter architecture.” In: *Microscopy Research and Technique* 70.10 (2007), pp. 851–863. DOI: 10.1002/jemt.20488 (cit. on pp. 9, 111).
- [77] R. Linsker. “Local Synaptic Learning Rules Suffice to Maximize Mutual Information in a Linear Network.” In: *Neural Comput.* 4.5 (1992), pp. 691–702. DOI: 10.1162/neco.1992.4.5.691 (cit. on p. 19).
- [78] D. G. Lowe. “Distinctive image features from scale-invariant keypoints.” In: *International journal of computer vision* 60.2 (2004), pp. 91–110 (cit. on p. 15).
- [79] D. G. Lowe. “Object recognition from local scale-invariant features.” In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on.* Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410 (cit. on p. 15).
- [80] D. Mantini, R. Franciotti, G. L. Romani, and V. Pizzella. “Improving {MEG} source localizations: An automated method for complete artifact removal based on independent component analysis.” In: *NeuroImage* 40.1 (2008), pp. 160–173. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2007.11.022> (cit. on pp. 53, 101).
- [81] R. E. Martenson. *Myelin: Biology and Chemistry*. CRC Press, 1992 (cit. on p. 1).
- [82] T. Mattson, B. Sanders, and B. Massingill. *Patterns for Parallel Programming*. First. Addison-Wesley Professional, 2004 (cit. on pp. 48, 50).
- [83] M. J. McKeown, L. K. Hansen, and T. J. Sejnowsk. “Independent component analysis of functional MRI: what is signal and what is noise?” In: *Current Opinion in Neurobiology* 13.5 (2003), pp. 620–629. DOI: <http://dx.doi.org/10.1016/j.conb.2003.09.012> (cit. on pp. 53, 101).

- [84] J. Mehta, S. Jerger, J. Jerger, and J. Martin. “Electrophysiological correlates of word comprehension: Event-related potential (ERP) and independent component analysis (ICA).” In: *International Journal of Audiology* 48.1 (2009), pp. 1–11. DOI: 10.1080/14992020802527258 (cit. on pp. 53, 101).
- [85] R. G. Miller. “The jackknife—a review.” In: *Biometrika* 61.1 (1974), pp. 1–15. DOI: 10.1093/biomet/61.1.1 (cit. on p. 59).
- [86] J. Nadal and N. Parga. *Non Linear Neurons in the Low Noise Limit: A Factorial Code Maximizes Information Transfer*. 1994 (cit. on pp. 27, 114).
- [87] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization.” In: *The Computer Journal* 7.4 (1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308 (cit. on pp. 69, 70).
- [88] J. von Neumann. “First Draft of a Report on the EDVAC.” In: *IEEE Ann. Hist. Comput.* 15.4 (1993), pp. 27–75. DOI: 10.1109/85.238389 (cit. on p. 44).
- [89] V. V. Nikulin, G. Nolte, and G. Curio. “A novel method for reliable and fast extraction of neuronal EEG/MEG oscillations on the basis of spatio-spectral decomposition.” In: *NeuroImage* 55.4 (2011), pp. 1528–1535. DOI: <http://dx.doi.org/10.1016/j.neuroimage.2011.01.057> (cit. on pp. 53, 101).
- [90] N. Otsu. “A Threshold Selection Method from Gray-Level Histograms.” In: *IEEE Transactions on Systems, Man and Cybernetics* 9.1 (Jan. 1979), pp. 62–66. DOI: 10.1109/tsmc.1979.4310076 (cit. on p. 56).
- [91] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. 3rd. McGraw-Hill Companies, 1991 (cit. on pp. 24, 25, 113, 114).
- [92] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design*. 4th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008 (cit. on p. 44).
- [93] *Python*. <https://www.python.org/>. Accessed: 2015-09-17 (cit. on p. 89).
- [94] F. Raimondo, J. E. Kamienkowski, M. Sigman, and D. F. Slezak. “CUDA-ICA: GPU Optimization of infomax-ICA EEG Analysis.” In: *Intell. Neuroscience* 2012 (2012), 2:2–2:2. DOI: 10.1155/2012/206972 (cit. on p. 77).

-
- [95] J. Reckfort, H. Wiese, U. Pietrzyk, K. Zilles, K. Amunts, and M. Axer. “A multiscale approach for the reconstruction of the fiber architecture of the human brain based on 3D-PLI.” In: *Frontiers in Neuroanatomy* 9.118 (2015). DOI: 10.3389/fnana.2015.00118 (cit. on p. 67).
- [96] F. J. Richards. “A Flexible Growth Function for Empirical Use.” In: *Journal of Experimental Botany* 10.29 (1959), pp. 290–300 (cit. on p. 58).
- [97] T. H. Sander, G. Wübbeler, A. Lueschow, G. Curio, and L. Trahms. “Cardiac artifact subspace identification and elimination in cognitive MEG data using time-delayed decorrelation.” In: *IEEE Trans. Biomed. Engineering* 49.4 (2002), pp. 345–354 (cit. on p. 1).
- [98] J. Sanders and E. Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010 (cit. on pp. 44, 50, 51).
- [99] W. J. Schmidt. “Zur Doppelbrechung des Nervenmarks.” In: *Zeitschrift für wissenschaftliche Mikroskopie* 41 (1923) (cit. on p. 1).
- [100] F. O. Schmitt and R. S. Bear. “The optical properties of vertebrate nerve axons as related to fiber size.” In: *Journal of Cellular and Comparative Physiology* 9.2 (1937), pp. 261–273. DOI: 10.1002/jcp.1030090209 (cit. on p. 1).
- [101] J. S. Shao and D. Tu. In: *The Jackknife and Bootstrap*. Springer Series in Statistics. Springer New York, 1995. DOI: 10.1007/978-1-4612-0795-5 (cit. on p. 59).
- [102] L. Silva and R. Buyya. *High Performance Cluster Computing: Programming and Applications*. Ed. by Rajkumar Buyya. Vol. 2. Prentice Hall, 1999 (cit. on pp. 43, 47, 49).
- [103] J. V. Stone. *Independent Component Analysis: A Tutorial Introduction*. Cambridge, MA, USA: MIT Press, 2004 (cit. on pp. 19, 25, 27, 28, 35, 67, 113, 114).
- [104] Student. “The Probable Error of a Mean.” In: *Biometrika* 6.1 (1908), pp. 1–25. DOI: 10.1093/biomet/6.1.1 (cit. on p. 60).
- [105] G. Tabbì. “Analysis of Independent Signal Components Applied to Polarized Light Imaging.” Diplomarbeit. Technische Universität Dortmund, 2012 (cit. on p. 3).

- [106] The HDF Group. *Hierarchical data format version 5*. <http://www.hdfgroup.org/HDF5>. Accessed: 2015-05-01. 2000-2010 (cit. on p. 81).
- [107] A. Viebke and S. Pillana. “The Potential of the Intel Xeon Phi for Supervised Deep Learning.” In: *CoRR* (2015) (cit. on p. 44).
- [108] F. H. Walters, R. Lloyd Jr, S. L. Morgan, and S. N. Deming. *Sequential Simplex Optimization: A Technique for Improving Quality and Productivity in Research, Development, and Manufacturing*. Chemometrics series. CRC, 1991 (cit. on p. 70).
- [109] H. Wiese, D. Grässel, U. Pietrzyk, K. Amunts, and M. Axer. “Polarized light imaging of the human brain: A new approach to the data analysis of tilted sections.” In: SPIE, Baltimore(USA), 05/05/2014 - 05/10/2014. 2014 (cit. on p. 7).
- [110] N. Wilt. *The CUDA Handbook: A Comprehensive Guide to GPU Programming*. Addison-Wesley, 2013 (cit. on pp. 44, 50, 51).
- [111] F. Zernike and E. Midwinter. *Applied Nonlinear Optics*. John Wiley & Sons, 1973 (cit. on p. 111).

Acknowledgements

This challenging work would not have been possible without the support and advices of so many people who I want to thank.

First of all, I would like to thank Professor K. Amunts for providing me with the opportunity to work in her institute (INM-1) where I was able to gain valuable experiences. Furthermore, I would like to thank Professor T. Lippert for the support and advices during my PhD and his enthusiasm for my work.

I would like to express my gratitude to Professor U. Pietrzyk for his interest in my work, for always having an open door and for reviewing this thesis. The person, who I am also grateful, is Markus Axer. He introduced me to 3D-PLI, which sparked in me the interest on this research field. He gave me this interesting topic and had numerous suggestions and advices to provide.

My further gratitude goes to Professor A. Klümper and Professor A. Frommer for agreeing to be on the examining board.

The next persons I would like to thank are Jürgen Dammers and Lukas Breuer for their expert support and constructive discussions. Even when they were busy they found time for explanations and for answering questions.

The entire *Fiber Architecture* team helped me a lot with questions concerning data acquisition, polarimeter, etc. Therefore, I would like to thank all of them. A big thank you is reserved to the following colleagues and friends: Liliana Caldeira, Tim Hütz, Stefan Köhnen, Daniel Krötz, Julia Reckfort and Hendrik Wiese, for their moral support, for proof reading and for the awesome activities outside of work.

Words are not enough to express all my gratitude to Michaela Gaens, who encour-

Acknowledgements

aged me and gave me confidence during the struggle for this thesis. Unconditionally, during this troublesome period you gave me strength and support. Thank you for accentuating and enriching my life during the PhD time.

Finally, my special thanks go to my family. Even in tough times at work and in private life you gave me support and stood by me.