Energy Minimization Algorithms for Image-Guided Automatic Patch Clamp Systems

Ph.D. Thesis

Krisztián Koós

BIOMAG group

Biological Research Centre, Szeged

Supervisor:

Dr. Peter Horvath

Doctoral School of Computer Science

Faculty of Science and Informatics

University of Szeged

Szeged, 2019

Preface

Do your homework! This thought was all that *made it possible* for me to enroll in a PhD program. I did my homework, which never was easy. If I could roll back time I would only do the homework that interests me – because I always liked focusing on one thing at a time. This is what *made me enroll* in a PhD program: to excel in computer science.

I read somewhere that PhD was going to change me, which I didn't want. As time passed I realized it opened my mind in a way I could have never imagined or might have never experienced anywhere else. I am very grateful to my supervisor Dr. Peter Horvath for making it possible for me to come this way.

I would like to thank everyone who have helped me to achieve my goals so far. First of all, my wife Stella for her constant support and my parents for the environment they have created for me. I am thankful to József Molnár who was like a secondary supervisor. Many thanks to all my colleagues at Biomag for the invaluable discussions, especially to Tamás Balassa, Tivadar Danka, Csaba Molnár, Filippo Piccinini, Ábel Szkalisity, and Ervin Tasnádi. Special thanks to Gáspár Oláh, Márton Rózsa and Attila Ozsvár for their help in the wet lab work. I am also grateful for the teachers and participants of the university talent care program for the competitive atmosphere. Finally, I thank everyone who gave their opinion on PhD or helped me in choosing a topic.

My research was carried out at the Biological Research Centre of the Hungarian Academy of Sciences. We acknowledge the financial support from the National Brain Research Programme (MTA-SE-NAP B-BIOMAG); the TEKES FiDiPro Programme; the LENDULET-BIOMAG Grant (2018-342); the European Regional Development Funds (GINOP-2.3.2-15-2016-00006, GINOP-2.3.2-15-2016-00001, GINOP-2.3.2-15-2016-00026, GINOP-2.3.2-15-2016-00037).

Krisztián Koós, 2019

Contents

List of Figures

Chapter 1

Introduction

Image analysis algorithms and tools have pushed the frontiers of scientific computing in the last few decades. One of the fields where the modern computer has made a remarkable impact is the area of biological imaging. Advanced microscopes can now easily collect thousands of high-resolution images of cells daily. The first few large biological screens were manually analyzed by expert biologists. It was soon realized that automation of the process is beneficial to extract the full spectrum of information present in these images. Many software was developed for scientific image analysis, usually open source, such as the general purpose ImageJ/FiJi [1, 2, 3, 4] or CellProfiler [5, 6] which is specialized for high-throughput cell image analysis. Newer tools, e.g. ilastik [7] or FindMyCells [8] use classical machine learning or deep learning for the segmentation task. Analyzing the images simultaneously yields many informative measures of cells, including the intensity and localization of each fluorescently labeled cellular component (for example, DNA or protein) within each subcellular compartment, as well as the number, size, and shape of those subcellular compartments. The collected measurements can be further processed to reveal their connections by tools such as Advanced Cell Classifier [9] or CellProfiler Analyst [10]. Some tools combine the segmentation and analysis steps, e.g. for nucleoli morphology measurement [11]. A recent review paper [12] examines free phenotypic image analysis software and gives a perspective on future possibilities.

The simplest of all optical microscopy techniques is the brightfield microscopy that was invented by Anton van Leeuwenhoek [13] in the 17th century as an improved version of the modern microscope developed by Robert Hooke [14]. The visible light transmitted through or reflected from the sample through a system of lenses magnifies images of small objects [15]. This technique is the basis of advanced microscopy types. In 1994 the successful expression of green fluorescent protein in living organisms brought fluorescence to the forefront [16]. Proteins can be stained (labeled) by fluorescent substances which show their localization inside the cells. Most of the time, biological image analysis is carried out on fluorescence images. Moreover, the mentioned tools are primarily applied for such images. However, fluorescence is phototoxic and in most cases it is impossible to apply on living cells, which leads us to the main focus of the dissertation.

The scientific recognition that electrical phenomena are part of animal physiology is well known by now. The electrophysiology of brain cells is investigated by the patch clamp technique [17]. The process can be applied *in vivo*, on living animals, or on brain tissue slices in case of human samples to understand its complex structure and discover new cellular phenotypes. The technique is very laborious and requires expertise, but can be automated. An image guided automation requires novel algorithms for label-free microscopy. We have developed methods and a software system that fully automatically perform patch clamp recording on brain tissue slices. The dissertation details the designed algorithms and results.

The structure of the document is as follows:

- Part I focuses on processing images of a special label-free microscopy. We have developed a differential geometry based reconstruction algorithm that transforms the images to a format similar to fluorescence ones to make analysis easier. The quality of the reconstruction is state of the art, which is demonstrated in simulation and on two types of real image databases. In addition, we have checked the accuracy with which the phase and height of the objects can be calculated using this microscopy and reconstruction.

- In Part II three variants of the developed pipette tip detection algorithm is described. These are object fitting algorithms and are also based on variational frameworks. The first version is a 2-dimensional model which requires a minimum intensity projection image as input computed from an image stack of the pipette. The second model is the 3-dimensional extension of the previous which has more degrees of freedom and requires the image stack itself as input. The third variant is a 3 dimensional model for fluorescence imaging. In 2-photon microscopy, the patch clamp pipette can be filled with fluorescent material that the model tries to fit onto in the image stack. The models are evaluated on multiple images and compared with baseline algorithms and hand-picked solutions.

- Part III describes the developed image guided automatic patch clamp system. This is the first system that is based on label-free microscopy. Besides the technical challenge to control every hardware component, novel algorithms were required for the major steps of the patch clamping process. Part I was used to detect cells on microscopy images. The methods for pipette tip detection, as described in Part II, were built into the system. The next step was the detection of cells in image stacks which was solved by building an image database and training a deep learning model. Then, a cell tracking algorithm was developed which tracks the 3D position of the target cell based on the 2D imaging modality, throughout the patch clamping process. The whole automatic process is logged which standardizes laboratory notebook generation. The system was used for measuring hundreds of rodent and human neurons in brain tissue slices and proved to be capable of providing high quality data for further biological research.

Part I

DIC Image Processing

Chapter 2

Image Reconstruction Algorithms

2.1 Motivation

Brightfield microscopy is simple and requires minimal sample preparation which is a significant advantage. Its limitation is the low contrast of most biological samples which motivated the advancements in contrast-enhancing techniques. Phase-contrast microscopy was proposed in the early 1930s by Fritz Zernike who was awarded the 1953 Nobel Prize in Physics for his invention [18]. Phase-contrast microscopy is interference-based and allows the study of the internal structure of living cells. Two years later, in 1955, Georges Nomarski established the theoretical basis for differential interference contrast (DIC) microscopy [19] that gains information about the optical path length of the sample and shows features that are invisible in a brightfield microscope. Fig. 2.1 shows example DIC images of cell cultures and a tissue. However, label-free techniques are unable to selectively visualize subcellular components, processes, or localization of proteins which generated a demand for new methods in microscopy.

Biological image analysis is usually performed on fluorescence images which are quantitative, thus can easily be analyzed by suitable software. Despite their advantages, phase-contrast and DIC are qualitative methods. Although these images are easy to understand for the human eye, conventional algorithms are not applicable when it comes to image processing. However, label-free techniques have advantages over fluorescence microscopy: namely, cells can be observed without staining, so these techniques are less phototoxic and there is no need for chemical fixation of the sample in contrast to numerous fluorescent staining protocols. Thus, label-free techniques allow live cell analysis in a physiologically more relevant way.

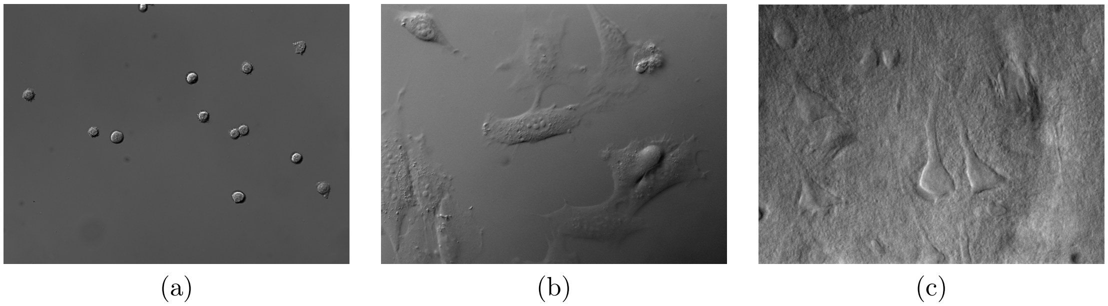


Figure 2.1: Example DIC images. (a) Chinese hamster ovary cell culture. (b) Mouse embryonic fibroblasts. (c) Brain tissue slice.

Nowadays, advanced label-free techniques are also used for the examination of biological samples. Quantitative phase-contrast microscopy [20, 21, 22] also gains information about the phase of cells. Diffraction phase microscopy (DPM) [23, 24] shifts the light all over the image and after a reconstruction step the phase differences are visible with high precision. Raman microscopy [25, 26] can reveal the location and amount of different constituents. Our focus here is on DIC microscopy due to its cost-effectiveness, widespread use in laboratories, and ability to perform depth-resolved imaging similar to confocal fluorescence microscopy. We have developed algorithms and tools to make DIC image analysis easier and more reliable.

We have proposed a novel DIC reconstruction algorithm and have given the most comprehensive overview of existing methods along with a wide range of comparisons. Our framework is based on energy minimization. The energy function uses a term that forces similarity to the original DIC image and a total variation-based regularization term. The first utilizes the PSF of the DIC microscope. The PSF is incorporated into our model by local integrals. We show that the derivation operation can be moved from the kernel to the image, which significantly accelerates the computations. Several other reconstruction methods are also described. The source code of the algorithms is made publicly available. We compare the reconstruction quality using three different test sets that are also made publicly available. The algorithms open up the possibility of using DIC as a quantitative imaging modality.

2.2 Image Formation Model

DIC microscopy images can be converted to show the optical path length distribution using the image formation model. The schematic setup of a DIC microscope is illustrated in Fig. 2.2. Image

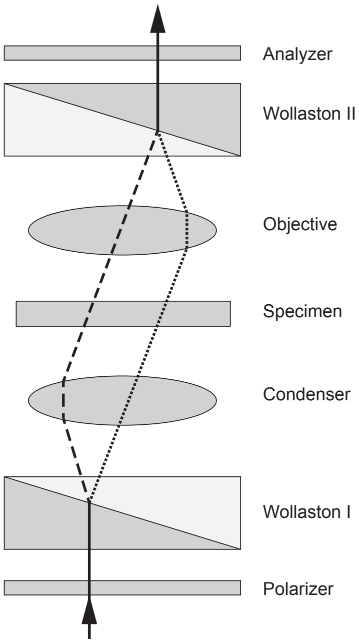


Figure 2.2: Setup of a DIC microscope. The arrows show the direction of the light. The dashed and dotted lines indicate the two mutually perpendicularly polarized light rays.

formation in a DIC microscope starts as a light ray enters into a polarizer that creates plane-polarized light. A Wollaston prism splits the light into two perpendicularly polarized light rays which are focused on the specimen by the condenser lens. These two light rays pass the sample by a minute shear. The distance between the rays, i.e. the shear distance, is smaller than the spatial resolution of the microscope's objective. Materials with different refractive indices and the specimen's thickness generate a phase shift between the light rays. Next a second lens, i.e. the objective, transmits the light rays to another Wollaston prism that recombines them. Finally, the second polarizer, called the analyzer creates plane-polarized light perpendicular to the light of the first polarizer. The two light rays interfere at this point and generate a contrast image [19]. Usually, the microscope is set to a positive bias by shifting the first DIC prism. In case of positive bias, the background in a DIC image is mid-gray and the optical path length differences are bright on one side and dark on the other side of the objects. However, the microscope can be set to extinction based on the position of the first prism. In this case the gradients of the regions in the sample that are thick or have large refractive indices appear as bright highlights on a black background in the images. Shifting the first prism to the opposite direction generates negative bias where intensities of the gradient features are reversed.

The prisms are often called DIC or Nomarski prisms. The Nomarski prism can be considered as the combination of the condenser or the objective lens and a Wollaston prism. The interference plane of a Nomarski prism is placed a few millimeters outside of its center, thus placing the specimen in the interference plane makes the use of lenses unnecessary.

Many mathematical models have been developed for the DIC image formation, some of which are listed below. The light in a DIC system is partially coherent and so it is not well defined [27, 28]. The level of coherence may vary inside the system which should also be taken into account. Moreover, the samples are usually not purely phase objects, hence the image is formed by both the phase difference and amplitude change. Theoretical developments were also carried out to reconstruct the amplitude along with the phase using the Transport of Intensity theorem [29, 30, 31]. DIC was converted into a quantitative technique earlier by modifying it to Phase-shifting DIC (PS-DIC) [32, 33]. PS-DIC utilizes an additional device that is able to introduce a constant phase shift to the light rays. Virtually it has the effect of rotating the sample – in other words, it changes the shear direction of the microscope, thus allowing edges otherwise invisible to be observed by the technique. Reconstructing the scene from multiple images using a linear model provides a result with a 15% error [34, 35]. However, physically rotating the sample is not possible after we have inserted the slide and using more images for a single scene reconstruction increases the computational complexity and acquisition time. Attempts were made to reconstruct the true phase from single images using different reconstruction algorithms [36], but the results have not been validated.

We have used a linear approximation which considers the image formation as the convolution of the original object information (optical path length distribution) I and the point spread function (PSF) of the DIC microscope. In linear models the PSF is often chosen to be the first derivative of a Gaussian function or the difference of two Dirac delta functions [37, 38]. Images are considered as real-valued intensity functions parameterized by image coordinates (x, y) on the domain Ω. The convolution is written as a local integral which has a window $W = [-d, d] \times [-d, d]$, where $2d + 1$ is the window size and $(\xi, \eta) \in W$. Furthermore, let $d\Omega = dx\, dy$ and $dW = d\xi\, d\eta$ represent the infinitesimal area elements of the integrals in the image and the local domains, respectively. The shear direction is denoted by a unit vector $\mathbf{u} = [u\ v]^T$, $u^2 + v^2 = 1$. The model is generalized with an appropriately chosen kernel function (which will be examined later) and becomes:

$$G(x,y) = \mathbf{u} \cdot \nabla \left(\iint\limits_{W} K\left(\xi, \eta\right) I\left(x + \xi, y + \eta\right) dW \right) = \mathbf{u} \cdot \nabla \hat{I}\left(x, y\right). \tag{2.1}$$

Note that the integral in (2.1) is parametric w.r.t. the image coordinates x and y.

The presented linear image formation limits the accuracy by ignoring diffraction and partial coherence effects. However, we have compared simulated DIC images generated by both the linear model and *microlith* [39], an optically accurate simulation tool. Our comparison shows that the differences are visually negligible and the correlations between images of these two kinds are very high, typically between 0.96-0.99.

2.3 Methods

This section describes our variational framework for DIC image reconstruction along with 6 other algorithms which were used for comparison. These algorithms include three Fourier transform-based ones, namely the Hilbert transform [38, 40], Wiener filtering [38, 41] and a general inverse filtering technique [42] (referred to as Yin), two linear programming approaches, which are the sparseness-enhanced multiplicative update [37] (SEMU) and a second-order cone program [37] (SOCP), and finally an earlier variational framework [43] (referred to as Feineigle).

2.3.1 Proposed variational framework

The goal of DIC reconstruction task is to reconstruct I given G and K. In a variational framework [44, 45] an energy function has to be defined. The energy function usually has the form of:

$$E = E_{data} + \lambda E_{reg}. \tag{2.2}$$

We propose a model, where E_{data} term has a low value when I convolved with K is similar to G. E_{reg} ensures spatial regularization to have locally smooth regions where G is smooth (i.e. there are no edges in the original image). To find the minimizer, the variable I is usually updated in the gradient of the function.

We propose the following variational framework [46, 47]. The energy function is formulated as

$$E \doteq \frac{1}{2} \iint\limits_{\Omega} \left[\underbrace{\mathbf{u} \cdot \nabla \hat{I} - (G - c)}_{S} \right]^2 d\Omega + \lambda \iint\limits_{\Omega} |\nabla I| \; d\Omega. \tag{2.3}$$

where c is an arbitrary constant that does not affect the solution, partial derivatives are denoted by

∂_x and ∂_y, and

$$S = \iint\limits_{\xi,\eta} K\left(u\partial_x I + v\partial_y I\right)\, d\xi\, d\eta - (G - c)\,. \tag{2.4}$$

The first term in functional (2.3) has a special form, its Lagrangian is given pointwise at image coordinates (x, y) by the known function G and the local integrals of the partial derivatives $\partial_z I\,(x - \xi, y - \eta)$, $z \in \{x, y\}$ of its minimizer I.

The data term ensures that the energy is low when the reconstructed image convolved by the PSF is similar to the input DIC image. Our choice of the smoothness term is the Total Variation (TV) term, where λ is a weight function, providing trade-off between data-driven reconstruction and denoising. TV denoising model [48] preserves discontinuities, i.e. its associated gradient descent equation does not smooth across edges. The energy function is minimized by the gradient descent method. The gradient descent iteratively takes steps proportional to the negative gradient. Faster methods like Gauss-Seidel or Successive Over Relaxation may also be used. However, at this point, their convergence criteria are not yet investigated. To define the steps we calculate the Euler-Lagrangian (EL) equations.

The Euler-Lagrange Equations

For the data term the first variation of the EL equation is:

$$\begin{gathered} \left.\frac{dE_{data}\left(\partial_x I + \varepsilon\partial_x h, \partial_y I + \varepsilon\partial_y h\right)}{d\varepsilon}\right|_{\varepsilon=0} = \\ \iint\limits_{x,y} S\, \left.\frac{dS\left(\partial_x I + \varepsilon\partial_x h, \partial_y I + \varepsilon\partial_y h\right)}{d\varepsilon}\right|_{\varepsilon=0} dx\, dy \end{gathered} \tag{2.5}$$

includes the derivatives of the perturbation $\partial_z h\,(x - \xi, y - \eta)$, $z \in \{x, y\}$ under local integrals such that:

$$\begin{gathered} S\left(\partial_x I + \varepsilon\partial_x h, \partial_y I + \varepsilon\partial_y h\right) = \\ \iint\limits_{\xi,\eta} K\left(u\partial_x\left(I + \varepsilon h\right) + v\partial_y\left(I + \varepsilon h\right)\right) d\xi\, d\eta - (G - c) \end{gathered} \tag{2.6}$$

To overcome this difficulty we require that the reconstructed primary image I, and hence its perturbation h as well, are piecewise analytic functions, i.e. their Taylor series are convergent around each image point at least in the domain of the local integral. If this requirement is satisfied, we can replace the perturbation function with its Taylor series around x, y and bring its derivatives from behind the local integral. Let ∂_x^n and ∂_y^n denote the nth order partial derivatives $(n > 1)$.

Performing the necessary amount of partial integration steps, the first variation (2.5) becomes

$$\iint\limits_{x,y} h \left[\sum_{m,n=0}^{\infty} w\left(m,n\right) \partial_x^m \partial_y^n \left(u \partial_x S + v \partial_y S\right) \right] dx\, dy \tag{2.7}$$

with weights

$$w\left(m,n\right) = \frac{\left(-1\right)^{m+n+1}}{m!n!} \iint\limits_{\xi,\eta} K \xi^m \eta^n \, d\xi\, d\eta \tag{2.8}$$

and the partial derivatives of S:

$$\partial_x^k \partial_y^l S = \partial_x^k \partial_y^l G + \iint\limits_{\xi,\eta} K \left(u \partial_x^{k+1} \partial_y^l I + v \partial_x^k \partial_y^{l+1} I\right) \, d\xi\, d\eta. \tag{2.9}$$

The EL equation is given by setting the expression in the brackets in (2.7) to zero. Assuming that lower-order terms contribute significantly more than higher-order terms, for sufficiently small local integrals the "bilinear" approximation of the EL equation can be formed as:

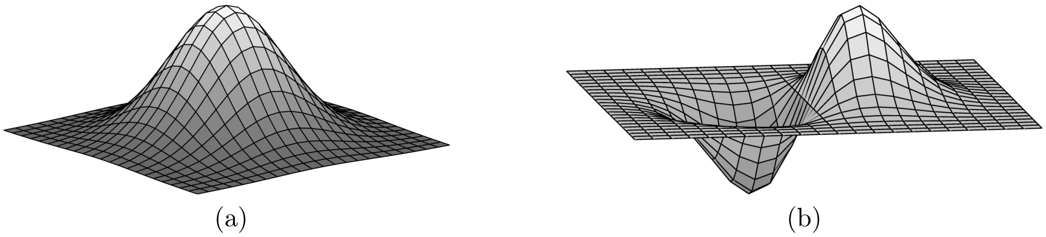
$$\begin{aligned} 0 = &- \left(u \partial_x S + v \partial_y S\right) \iint\limits_{\xi,\eta} K \, d\xi\, d\eta \\ &+ \left(u \partial_x^2 S + v \partial_x \partial_y S\right) \iint\limits_{\xi,\eta} K \xi \, d\xi\, d\eta \\ &+ \left(u \partial_x \partial_y S + v \partial_y^2 S\right) \iint\limits_{\xi,\eta} K \eta \, d\xi\, d\eta \\ &- \left(u \partial_x^2 \partial_y S + v \partial_x \partial_y^2 S\right) \iint\limits_{\xi,\eta} K \xi \eta \, d\xi\, d\eta \end{aligned} \tag{2.10}$$

The EL equation of the TV smoothness term is defined pointwise as follows:

$$\left|\nabla I\right| = \sqrt{\left(\partial_x I\right)^2 + \left(\partial_y I\right)^2}. \tag{2.11}$$

2.3.2 Kernel selection

The finite kernel function can be chosen arbitrarily. At least two simplifications can be made using special kernels. An obvious, special choice for K is a rotationally symmetric function K_R (Fig. 2.3a). In this case, the one-dimensional slices of the function along the parameter lines x or y become even functions, hence all terms having odd m or n in (2.7) are eliminated and the bilinear approximation



(a)

(b)

Figure 2.3: Possible kernel functions. (a) A rotationally symmetric kernel function K_R. (b) Kernel function K which is a correct approximation of the DIC PSF.

(2.10) reduces to "piecewise constant" approximation:

$$\begin{aligned} 0 &= u\partial_x G + v\partial_y G - \iint\limits_{\xi,\eta} K_R \left(u^2\partial_x^2 I + 2uv\partial_x\partial_y I + v^2\partial_x^2 I\right)\, d\xi\, d\eta \\ &= \mathbf{u}\cdot\nabla G - \iint\limits_{\xi,\eta} K_R \mathbf{u}\cdot\nabla\nabla I\cdot\mathbf{u}\, d\xi\, d\eta. \end{aligned} \tag{2.12}$$

Equation (2.12) is given after the multiplication by the weight factor $w\left(0,0\right) = \iint_{\xi,\eta} K\, d\xi\, d\eta$. ∇G is the gradient of the DIC image, $\nabla\nabla I$ is the Hessian of the primary image. Note that the rotational symmetry is not a necessary condition, axis symmetry with axis aligned to the shear angle and going through the origin of the local integration window is enough.

However, the DIC PSF and its approximations (Fig. 2.3b) do not satisfy the symmetry requirements. After applying parameter transformations $\xi \rightarrow \xi - x$, $\eta \rightarrow \eta - y$ to the local integral in (2.1), its equivalent is

$$\hat{I}\left(x,y\right) = \iint\limits_{\substack{\{x-d,x+d\}\\ \{y-d,y+d\}}} K\left(\xi - x, \eta - y\right) I\left(\xi,\eta\right)\, dW. \tag{2.13}$$

Moving the calculations of the derivations from the image to the kernel function allows the use of the correct PSF. The partial derivatives w.r.t variables x and y become more complex because of their occurrence in the boundary of the integration. For example, the first derivative $\partial_x \hat{I}(x,y)$ is

$$\begin{aligned}
& - \iint\limits_{\substack{\{x-d,\, x+d\} \\ \{y-d,\, y+d\}}} \partial_x K(\xi - x, \eta - y)\, I(\xi, \eta)\; d\xi\, d\eta \\
& + \int\limits_{\{y-d,\, y+d\}} K(d + x, \eta - y)\, I(d, \eta)\; d\eta \\
& - \int\limits_{\{y-d,\, y+d\}} K(d - x, \eta - y)\, I(d, \eta)\; d\eta.
\end{aligned} \tag{2.14}$$

Terms other than the first can be canceled in many ways. The simplest choice for such kernel function K_0 has zero value derivatives all over the boundary. Note that infinite, bounded kernels automatically satisfy this property. By using K_0, (2.14) simplifies to

$$\partial_x^k \partial_y^l \hat{I}(x,y) = (-1)^{k+l} \iint\limits_W \partial_x^k \partial_y^l K_0(\xi - x, \eta - y)\, I(\xi, \eta)\; dW. \tag{2.15}$$

Symmetric Gaussian kernel K_G with the size of at least $6\sigma - 1$, where σ is the standard deviation parameter of the Gaussian normal distribution, closely approximates both properties. The rationale behind using (2.12) (i.e. leaving derivations on the image and using the integral of the PSF as the kernel) is that it requires less calculation. Although the partial kernel derivatives can be precomputed, using them for local integration is computationally more expensive than calculating image derivatives. Hereafter this algorithm is referred to as the proposed one.

2.3.3 Hilbert transform

The Hilbert transform [38, 40] is considered to be the first algorithm used to reconstruct DIC images. The 1D Hilbert transform H of a function f is the convolution of the function by $1/\pi x$:

$$f_H(x) = H\{f(x)\} = f(x) * \frac{1}{\pi x} = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(x')}{(x - x')}\, dx'. \tag{2.16}$$

The required convolution in Fourier space is a multiplication. In 2D the spatial frequencies k and l that define the signum function have to be perpendicular to the DIC shear direction. Let H now denote the 2D Hilbert transform and $\mathbf{F}$ denote the Fourier transform. The real part of the inverse

Fourier transform gives the resulting Hilbert transformed image:

$$H\left\{G\right\} = \mathbf{F}^{-1}\left\{-i\,\mathrm{sgn}\left(vk - ul\right)\mathbf{F}\left\{G\right\}\right\}. \tag{2.17}$$

2.3.4 Wiener filtering

If we consider the image formation model written in equation (2.1) as a convolution with the DIC PSF, the determination of I would be a simple division in Fourier space of the DIC image with the PSF. However, dividing by zero would produce large errors. Wiener filtering [38, 41] regularizes the equation by a small term based on the signal S and noise N ratio:

$$\mathbf{F}\left(I\right) = \frac{\overline{\mathbf{F}\left(\mathbf{K}\right)}}{\left|\mathbf{F}\left(\mathbf{K}\right)\right|^{2} + N/S}\mathbf{F}\left(\mathbf{G}\right), \tag{2.18}$$

where the overline is the conjugate transpose.

2.3.5 Inverse filtering

Wiener filtering is one technique to avoid division by zero in Fourier space. If we add regularization terms to the objective function, they can also prevent the occurrence of values close to zero in the divisor. In such an approach [49] a smoothness and a sparsity term are present in the objective function and the equation is solved in frequency space. In the original paper [42] the algorithm is designed to reconstruct an image from multiple images of the same scenario but with different shear angles. Changing the shear direction in a microscope is not possible. The microscope stage can be moved though, however, if cells are not fixed and get shifted, then matching the corresponding images becomes a challenging task and generates a new problem. For a single DIC image the equation to get the reconstructed image is the following:

$$\mathbf{F}\left(I\right) = -\left(\mathbf{F}\left(\mathbf{u}\cdot\nabla K\right)\cdot\mathbf{F}\left(G\right)\right)./\left(\omega_s A + \omega_r - \mathbf{F}\left(\mathbf{u}\cdot\nabla K\right)^{2}\right), \tag{2.19}$$

where A is the Fourier transform of the 2D discrete Laplace operator and ./ is the point-wise division. This algorithm is referred to as ‘Yin’.

2.3.6 Sparseness-enhanced multiplicative update (SEMU)

The reconstruction task can be formulated as a linear equation system [37]. Let $\mathbf{H}$ denote the transfer matrix of the image formation model, $\mathbf{g}$ denote G and $\mathbf{f}$ denote I by converting them to column vectors. The transfer matrix $\mathbf{H}$ has to be matched with the column- or row-order conversion of the variables G and I. The general objective function can be written as:

$$\mathrm{O}(f) = \|\mathbf{g} - \mathbf{Hf}\|_2^2 + \gamma\,\mathrm{Smoothness}(f) + \beta\,\mathrm{Sparsity}(f)\,, \text{ subject to } \mathbf{f} \geq 0. \tag{2.20}$$

The smoothness term in the objective function forces the reconstruction have smooth areas, while the sparsity term makes the pixel intensities in the reconstruction to have small values, close to zero. The described algorithm differs from the original such that the flat-field correction [50] step is ignored. Bias-correction is a different field and skipping it makes the comparison of the algorithms fair.

One special form of the objective function uses L2 norm and weighted L1 sparsity besides the data term. The matrix R stands for the smoothness term which represents the Laplace operator. The matrix W which stands for the sparsity term is a diagonal matrix with positive weights on the diagonals and zeros elsewhere. The objective function is written as:

$$\mathrm{O}_1(f) = \|\mathbf{g} - \mathbf{Hf}\|_2^2 + \gamma\|\mathbf{Rf}\|_2^2 + \beta\|\mathbf{Wf}\|_1, \text{ s.t. } \mathbf{f} \geq 0. \tag{2.21}$$

Equation (2.21) can be expressed as a nonnegative-constrained quadratic program. The algorithm ensures that the non-negativity of the result is preserved during the iterations. The weights in W can optionally be updated during the iterations by a log-sum penalty function, which can speed up the convergence. This algorithm is referred to as Sparseness-Enhanced Multiplicative Update (SEMU).

2.3.7 Second order cone program (SOCP)

The general objective function written in equation (2.20) can be specialized such that it incorporates the TV smoothness term instead of L2 regularization which is used in equation (2.21). Furthermore, the weighting function from the sparsity term can be removed. The objective function becomes:

$$\mathrm{O}_2(f) = \|\mathbf{g} - \mathbf{Hf}\|_2^2 + \gamma\|\mathbf{f}\|_{\mathrm{TV}} + \beta\|\mathbf{f}\|_1, \text{ s.t. } \mathbf{f} \geq 0. \tag{2.22}$$

Equation (2.22) can be solved as a second-order cone program [37] using a general-purpose convex optimization tool [51].

2.3.8 An earlier variational framework

Previously an energy function for DIC image reconstruction was formulated as [43]:

$$E = \iint_{\Omega} \left|\mathbf{u} \cdot \nabla I - G\right|^2 + \lambda_1 \left|\mathbf{n} \cdot \nabla I\right|^2 + \lambda_2 b \left|I - I_{\text{db}}\right| \, dx \, dy, \tag{2.23}$$

where $\mathbf{n}$ is the normal vector of the shear direction, b is a binary function that indicates whether a pixel belongs to the background or not, λ_1 and λ_2 are weight constants and I_{db} is the desired background pixel value. The directional derivative perpendicular to the shear removes the bright stripes that appear in the shear direction as artifacts due to noise and the imperfection of the image formation model. There is no straightforward way to determin variable b since objects can appear anywhere in the image. One way is to use the smoothed version of the variance filtered DIC image or its thresholded version for b, but it does not guarantee a correct solution. The Euler equation of the energy function is shown in equation (16).

$$\lambda_2 b \left(I - I_{\text{db}}\right) - \left(u^2 - \lambda_1 v^2\right) \delta_x^2 I - \left(v^2 - \lambda_1 u^2\right) \delta_y^2 I - 2\left(1 - \lambda_1\right) uv \delta_x \delta_y I = 0. \tag{2.24}$$

Let ‘Feineigle’ refer to this algorithm, which solves the problem by iteratively updating the reconstruction based on this function.

2.4 Results

After reconstructing the DIC images by each algorithm, we computed the similarity of the results to the corresponding ground truth images. After removing the meaningless negative values, we normalized both the result and the ground truth images between 0 and 1 to compare reconstruction quality. A general approach is to calculate the Mean Squared Error (MSE) between the result and the ground truth images. A lower MSE value means better reconstruction. Higher pixel intensity differences increase the MSE value more than smaller differences. We remark that other conventional metrics such as correlation, norm and mean average error gave the same ranking of the algorithms.

In real cases when no ground truth is available, comparing the convolved result image to the input DIC image might be the only way to measure quality. However, it can be a misleading metric because of the imperfect PSF, the noise and the optional smoothness term. In the real tests we show two problem-specific ways of obtaining ground truth images for real DIC images.

2.4.1 Synthetic dataset

For the quality test of the algorithms on a synthetic dataset, we created 20 ground truth images of simple shapes, e.g. ellipse, rectangle or pentagon. The images were grayscale, but 15 of them contained only the minimum and maximum intensities, i.e. they can be considered as binary. The other 5 contained a range of different intensities. The simulated DIC images were created by convolving the ground truth images with an estimated PSF which is included in the dataset. The PSF we used is the difference of two Dirac delta functions which is a close approximation of the first derivative of the Gaussian function with $\sigma = 0.5$. The generated DIC images and the PSF were the inputs of the

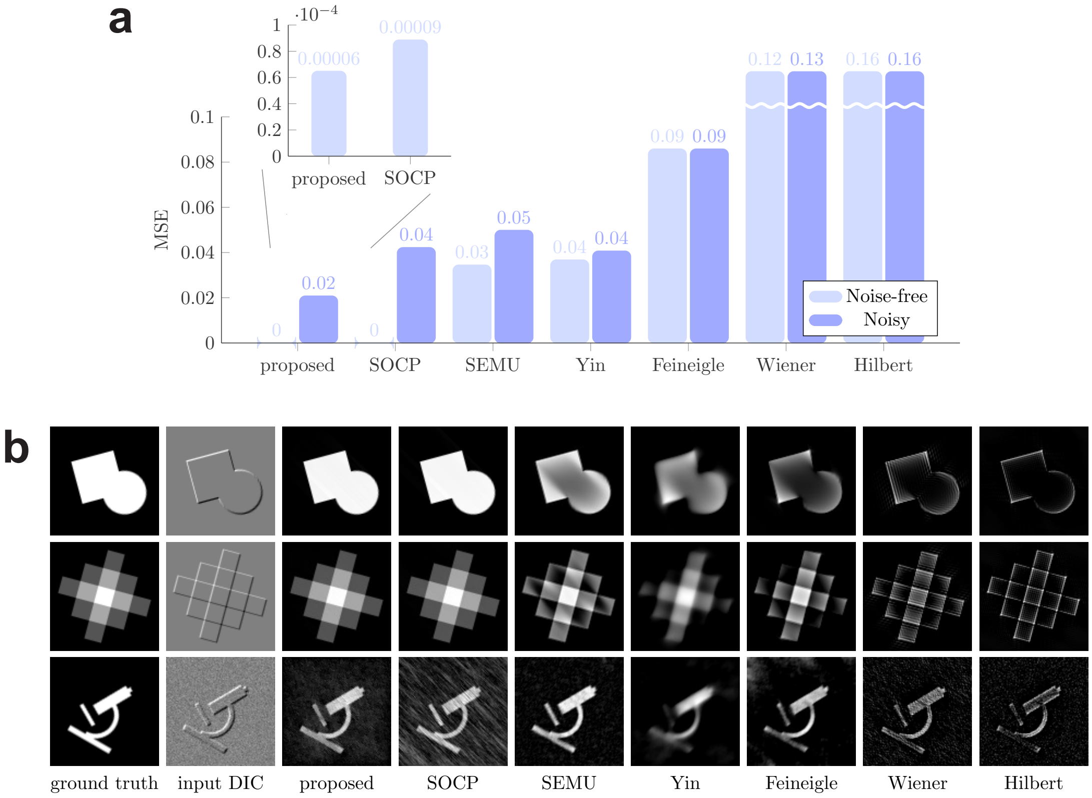


Figure 2.4: Comparison of algorithms on synthetic data. (a) The bars show the average mean squared error values. The algorithms are ordered based on their performance on the noise-free data. The enlarged part for the first two algorithms has a different scale. (b) Example synthetic ground truth images, simulated DIC images, and reconstructions.

algorithms. The mentioned σ value assumes ideal conditions which are not present in a microscope. However, since some of the algorithms do not support any other PSF, $\sigma = 0.5$ allows us to perform a fair comparison. Furthermore, we have a noisy test set by adding 20 dB noise to the images.

Fig. 2.4a shows the average MSE values of the algorithms in both noise-free and noisy cases. Different images were weighted equally regardless of the number of rotations. Fig. 2.4b shows example images. In the noise-free images both the proposed and the SOCP algorithms seem to provide a desired result. Other algorithms are usually unable to reconstruct bright regions in the middle of larger objects, thus may only reconstruct the edges. The noisy tests change the quality ranking based on the MSE values, and the Yin algorithm performs remarkably well compared to SOCP. Overall, the best few algorithms, namely the proposed, the SOCP, the SEMU, and the Yin algorithms perform well in general, but the quality of the result depends on the object and the noise level.

2.4.2 Fluorescent nanoprinted structures

For the quality test of the algorithms on microstructures we created polymerized fluorescent surface-attached objects. Our dataset of polymerized objects consists of 6 different shapes: squares, square-shaped blocks of increasing height (referred to as step-like structures), triangles, rings, circles, and hangar-like forms. The surface-attached test structures were polymerized by two-photon polymerization (TPP) [52] on a custom-developed optical system constructed around a Zeiss Axiovert 200 inverted microscope [53]. In this process the beam of an ultrafast laser (C-Fiber A 780, Menlo Systems GmbH, Germany, $\Delta\tau$ =100 fs, λ = 785 nm, 100 MHz repetition rate) was focused into an 18 µm thick layer of SU-8 2007 photoresist (Microchem, Newton, MA, USA) mixed with rhodamine 6G by a 100x oil immersion objective (NA=1.25, Zeiss Apochromat). The sample was mounted on and translated by a piezo translation system (model P-731.8L in X-Y and P-721.10 in Z, Physik Instrumente, Germany) in 3 dimensions during the laser illumination. The photoresist layer was produced by spin coating 30 µl of SU-8 on a 170 µm thick coverslip and soft-baking at 95 °C for 10 minutes to remove its solvent. For the step-like structures only one laser beam was used, but for the other structures the original single laser beam was split into four beams with a spatial light modulator (SLM, Pluto NIR, Holoeye GmbH, Germany) in order to produce four identical structures in a rectangular arrangement [54]. The power of each beam was set to 6mW, and the scanning speed was set to 80 µm/s for all structures. After illumination, the sample was baked again for 10 minutes at 95 °C to complete the polymerization, then developed in its matching developer (mr-Dev

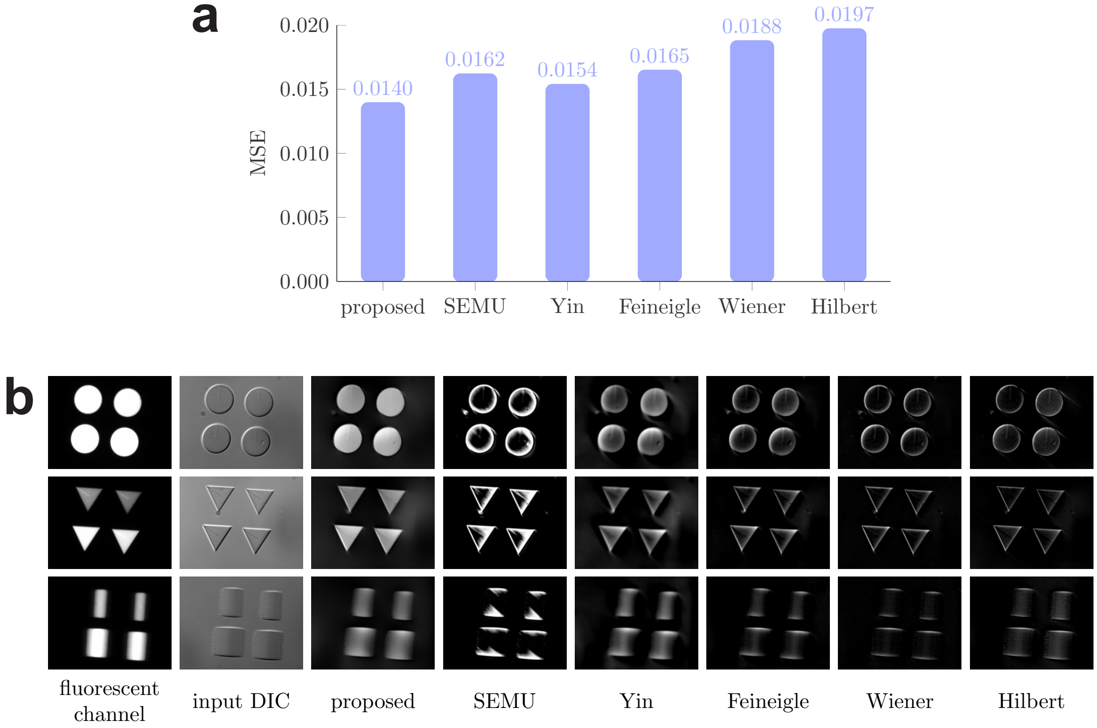


Figure 2.5: Reconstruction results of the algorithms on polymerized structures. (a) Average mean squared error values between the fluorescence and the reconstructed images of nanoprinted structures. (b) Example images of microstructures in fluorescence channel and DIC mode and their corresponding reconstructions.

600, Microchem, Newton, MA, USA) to remove the non-polymerized surplus and finally rinsed with ethanol and dried in a stream of nitrogen. The sample was immersed in oil with a refractive index of 1.515 (type DF, Cargille, Cedar Grove, NJ, USA), because the phase shift at the edges would be bigger than what DIC could correctly visualize. We designed squares and step-like structures with a side length of 20 µm, rings with an outer and inner diameter of 20 µm and 13 µm, respectively, circles with a diameter of 20 µm, triangles with a side length of 20 µm and hangar-like structures with a tilt angle of about 9.5 degrees and a width of 20 µm. We took both fluorescence and DIC images of the objects by an Olympus Cell-R microscope which can work in both modes without having to move the stage, thus the objects remain in place for the two modalities. Fluorescence microscopy can be considered quantitative and for thin samples the recorded intensity is linearly proportional to the thickness of a homogenous sample. We compared the reconstructions of the qualitative DIC images to the fluorescence ones. We calculated the MSE, but any other metrics described in section 2.4

may be used. The total number of image pairs taken of the nanoprinted samples was 60. Fig. 2.5a shows the average MSE values for the different reconstruction algorithms run on the nanoprinted structures dataset. The SOCP algorithm is not included due to its excessive runtime. Quality ranking of the algorithms is very similar to that described in the synthetic dataset. Fig. 2.5b shows some DIC and the corresponding fluorescence images with the results of the different algorithms. Again, it is obvious that some of the algorithms reconstruct only the edges of the objects. It is an issue particularly for the hangar-like objects (bottom row), where the shape is designed to test whether small changes are visible in the image.

2.4.3 Cell images

The third quality test of the algorithms consisted of 60 DIC images of CHO cells. The images were taken at the time when the cells had just started their attachment to the bottom of the dish. We made hand-segmentation of all the images which we used as ground truth for quality measurement of the reconstructions. Since the reconstructions are not binary images, comparing them to the binary hand-segmentations is not a reliable quality measurement method. Segmentation is a different and wide area of image processing and is out of the focus of the current work. Basic segmentation algorithms find a global threshold value derived from the examined image's histogram. To overcome the problem of thresholding Receiver Operating Characteristics (ROC) Area Under Curve (AUC) was used. To calculate the AUC, the result image was thresholded by every possible value, and both the True Positive Rate (TPR) and the False Positive Rate (FPR) were determined for every threshold value. The ROC space is then defined by FPR and TPR as X and Y axes, respectively. The AUC is the area under the FPR/TPR plot. Higher AUC value indicates better reconstruction. An AUC value of 1 indicates that the result is equivalent to the ground truth and the thresholded image is always the expected regardless of the threshold value used. Fig. 2.6a shows a result image of the proposed algorithm and the corresponding DIC image. All of the algorithms tested are able to reconstruct DIC images that can be subject to further processing. Some images are blurred or artifacts, bright lines may appear at the edges of the cells in the shear direction. Fig. 2.6b shows the average ROC curves for every algorithm. The SOCP algorithm is not included due to its excessive runtime. The proposed variational framework has the highest AUC value, followed by SEMU and their differences to the rest of the algorithms are remarkable.

It is important to note that the proposed algorithm was tested to reconstruct DIC images of tissue samples. A DIC image of a brain tissue and the corresponding reconstruction is shown in

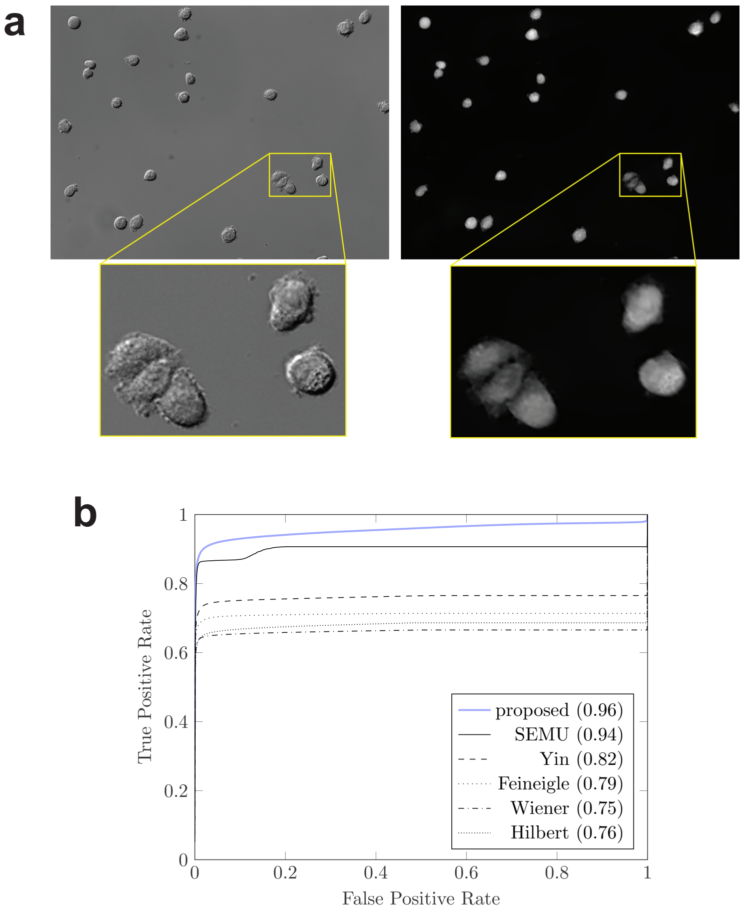


Figure 2.6: Comparison of algorithms on cell images. (a) Example DIC image of CHO cells and its reconstruction with the proposed algorithm. Yellow boxes show enlarged parts of the images. (b) Average ROC Curves of cell images. The numbers in the legend show the AUC values.

Fig. 2.7. Although extracellular matrix, axons, and dendrites can be seen as well in DIC images, the cell bodies are clearly visible in the reconstructions as bright intensity regions. Extending the algorithm from slices to volumes is future work.

2.5 Discussion

We described 7 different DIC reconstruction algorithms of 3 types: inverse filtering algorithms, linear equation solvers and energy minimization frameworks. The algorithms are tested on 3 new image

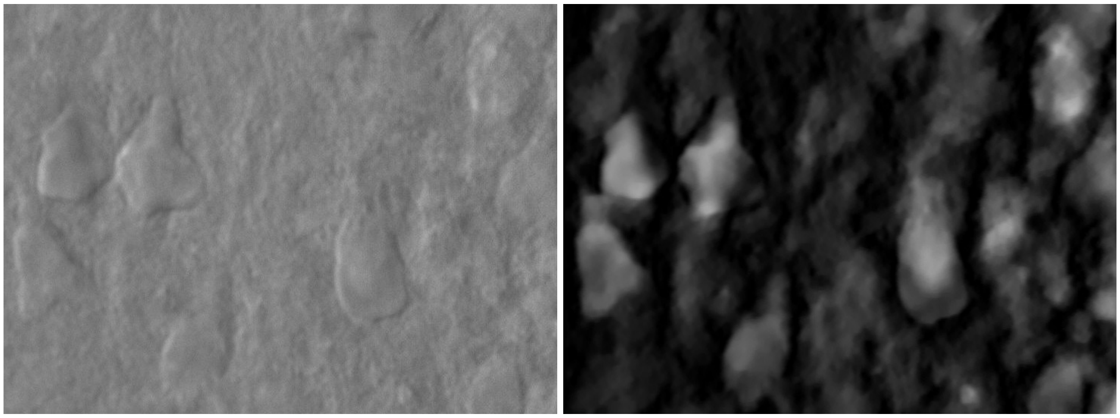


Figure 2.7: Example of brain tissue image and its reconstruction with the proposed algorithm.

sets, namely on a synthetic dataset with a generated PSF, on real DIC images of fluorescent polymerized microstructures and on images of cells in biological samples with hand-segmentation. The image sets are made publicly available in BBBC. The reconstructions are compared to the ground truth images of the corresponding sets. Every algorithm is able to reconstruct DIC images so that the resulting images are visually acceptable. Some techniques generate artifacts like bright stripes in the shear direction. When it comes to processing thousands of images, small errors can accumulate resulting in false segmentation. This work along with the publicly available source code provides a basis for automated high-throughput label-free microscopy. The provided implementations can be used to convert the qualitative DIC images into quantitative ones after calibrating the system. The converted, quantitative DIC images are similar to what other quantitative phase imaging techniques provide with the advantage that DIC is widely available and it is a cost-effective solution. The reconstructed images are appropriate for morphological analysis or tracking of living cells. Furthermore, a fluorescence channel is spared which can be used to stain specific compartments or speed up the acquisition.

Chapter 3

Phase and height measurement

Here, we describe the developed algorithm for phase and height measurement using DIC microscopy [55]. The validation of the algorithm is presented on microbeads of different sizes. Then an example application is shown on images of cells, where the height of cells has also been calculated.

3.1 Algorithm for phase and height reconstruction

In this section, we describe an algorithm used to measure the phase and height of unknown objects. The algorithm starts with a calibration process that aims to determine the relation between the values in the reconstructed images and the phase of the examined objects. The assumption of a linear image formation model has to be made because it is already required by the reconstruction algorithms. The linear image formation model (in terms of phase) means that the sample is a phase object and does not absorb light, i.e. the amplitude of light does not change. Cells only absorb about 10% of light and in practice they are considered as phase objects.

As a first step, the microscope settings have to be fixed: the illumination and exposure time have to be set to a constant value, and a bandpass filter should be used if available. Using filters makes it easier to calculate a weighted average of wavelengths or a narrower range can be used later to calculate the phase. Furthermore, any other microscope specific parameters should be fixed that affect the image acquisition, i.e. image histogram operations should be disabled. Most microscope software can automatically determine these parameters, but only for the sample being observed. The final settings can be based on the automatic values but they should be set manually so that they are suitable for both the calibration objects and further samples.

The next step is the phase calibration. Images of some calibration samples should be taken. Calibration objects usually have simple shapes and are homogeneous. Moreover, the refractive index of both the sample and the surrounding medium has to be known in advance. After reconstructing the DIC images of the calibration samples, the reconstructed object values are fit to the theoretical phase profile. Now the function that maps the reconstructed values to phase values can be calculated. This function can be used later (without altering the microscope settings) to determine the phase of unknown objects, but with known refractive indices. If the objects are not homogeneous, an averaged refractive index value can be used to get an approximation as will be demonstrated later in Section 3.3.

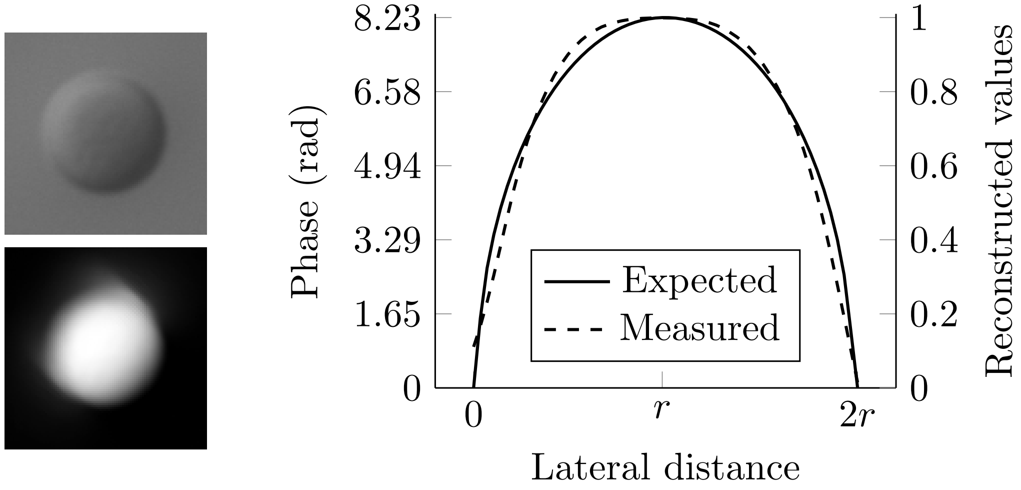
To perform the phase calibration, we used polystyrene microbeads of diameter $d = 9\,\mu\text{m}$ (Polybead, Polysciences, Warrington, PA, USA). The refractive index of the beads was $n_{bead} = 1.595$ and $n_{oil} = 1.515$ for the oil (type DF, Cargille, Cedar Grove, NJ, USA) in which they were immersed. We used an Olympus IF550 bandpass filter for our tests which had an average weighted wavelength λ of about 550 nm. This λ value was then used to calculate the theoretical phase profile of the beads. The following equation gives the phase difference between the surrounding medium and the center of the bead:

$$\varphi = \frac{2\pi}{\lambda}\Delta n d, \tag{3.1}$$

where $\Delta n = n_{bead} - n_{oil}$ in this case. Fig. 3.1 shows an example DIC and reconstructed image of a microbead and the averaged data fit to the theoretical phase profile. The data was normalized to the theoretical profile, which is a reliable method in the case of spheres, because the phase distribution is not uniform. The two line plots highly correlate. The flat-top behavior of the measured data is attributed to the slight smoothing effect of the TV term, which preserves high jumps and flattens small differences. Based on the scales shown for the two Y axes, a unit change in pixel intensity can be expressed as a phase change. This value can be used to express the phase of unknown objects. However, this approach does not consider diffraction effects due to the linearity of the model. If the phase difference is too high between the surrounding medium and the calibration object, diffraction causes artifacts in the DIC image which negatively affects the calibration.

The physical height of homogeneous samples can easily be calculated from the phase φ, if we know the wavelength λ and the refractive index difference Δn of the sample and the medium, using the following formula, which gives the height in nanometers:

$$h = \frac{\varphi\lambda}{2\pi\Delta n} \tag{3.2}$$



(a) (b)

Figure 3.1: Phase calibration on microbeads. (a) Example DIC image of a microbead and its reconstruction. (b) Line profile of the expected phase distribution of a microbead and that of the reconstructions, where r is the radius of a microbead. The plot of the reconstruction was averaged over 16 beads after normalization, in 4 directions (horizontal, vertical and two diagonals) through the center of the beads.

Algorithm 1 summarizes the process used for calibration and height estimation:

Algorithm 1 Measuring phase using DIC microscopy

```
1: Select a calibration sample with known dimensions and refractive index
2: Perform a phase calibration
3: Create images of the sample
4: Reconstruct the DIC images
5: Transform the reconstructed images to phase images
6: Optionally convert phase to height
```

If more than one type of sample needs to be imaged in one session, the calibration only has to be performed once, and from the second type the algorithm can be started from Step 3.

3.2 Phase reconstruction on microbeads

To verify that the calibration is accurate, we created images of microbeads of different but known sizes under the same conditions, in the same medium. Then the images were reconstructed and converted

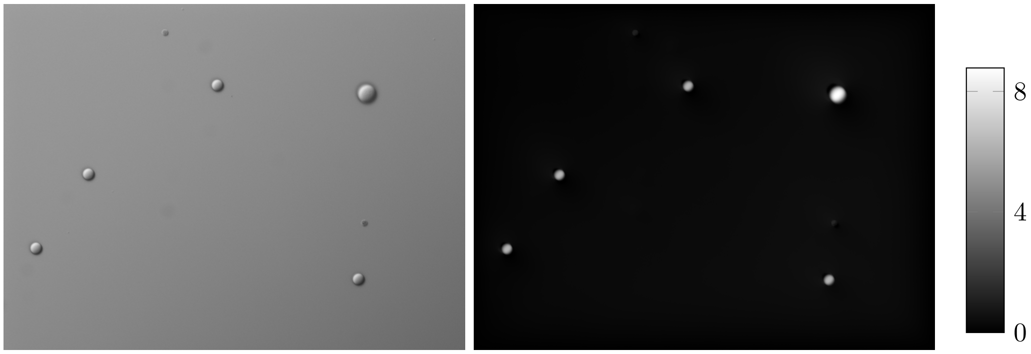


Figure 3.2: Example DIC image (left) of multiple sized microbeads and its reconstructed phase image (right) with a colormap. The phase was calculated after the calibration process (described in Section 3.1) had been performed.

to phase images using the calibration function that was determined in the calibration process in Section 3.1. Finally, the observed peak values of the beads were compared to the theoretical peaks.

The diameters of the microbeads used for this test were 6 and 9 µm. The microbeads, originally supplied in distilled water, were mixed and put on a coverslip. Then the coverslip was dried under heat to ensure that the microbeads were not drawn to each other when the water evaporated, thus reconstructed images should be more accurate. 7 DIC images were made of the coverslip and the peak values of 29 smaller and 7 bigger microbeads were determined. The average reconstructed phase for the 6 µm beads was 5.81 rad, which is 0.33 rad more than the value that (3.1) gives. However, the variance in 6 and 9 µm beads were only 0.12 and 0.14 rad, respectively. We think that the phase error between the different-sized beads originates from the nonlinearity of the image formation and it is not a problem of the reconstruction process, because the variance is much less among the same-sized beads. Furthermore, 0.33 rad converted to distance is about 180 nm, which is fairly below the 10% size variance of the beads as given by the manufacturer. Fig. 3.2 shows an example DIC image of microbeads and its reconstructed phase image.

3.3 Cell height measurement

To demonstrate the practical utility of phase reconstruction from DIC images, pictures of mouse embryonic fibroblast (MEF) cells were taken and reconstructed. Fig. 3.3 contains example DIC images and their computed phase images. Rounded-up cells are brighter, while elongated (deformed) ones that have become attached to the bottom of the dish are mid-gray. Rounded-up cells are thought to have a similar volume to others, but they are taller. Since OPL is formed by the product

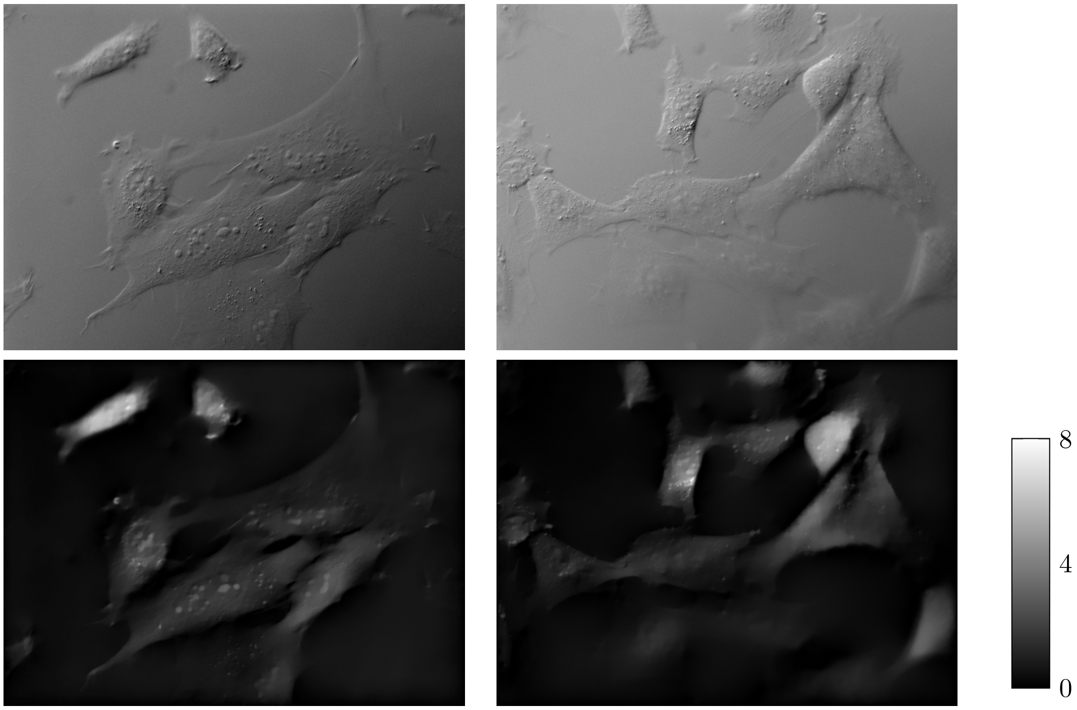


Figure 3.3: Example DIC images of MEF cells and their reconstructed phase images. The color bar is in radians. The phases were calculated after the calibration process. Rounded-up cells are taller, hence appear brighter in the reconstructed images. The nucleoli and other compartments of the cells are clearly visible.

of refractive index and thickness, taller cells appear brighter in the reconstructions. The nucleoli and other subcellular compartments appear as regions which demonstrate the resolution capability of the technique. The OPL can be visualized as a heightmap as shown in Fig. 3.4. Since (3.2) is linear, in the case of cells this transformation usually does not reflect the true height value. Although the nucleoli in Fig. 3.3 have higher OPL, this is due to their higher refractive index and the cell height being smooth. If height information of cells is required, some statistical phase value (e.g. mean or median) should be used in the calculation. Furthermore, note that in spite of the microbeads having similar radian values in Fig. 3.3 to Fig. 3.2, it does not imply that the cells are of similar size to the microbeads. The refractive index difference Δn has a different value in the two cases, which results in significantly different height values when (3.2) is applied. The calculated height is about 10 µm and 20 µm for attached and rounded-up cells, respectively, which is in the range of the physical cell size.

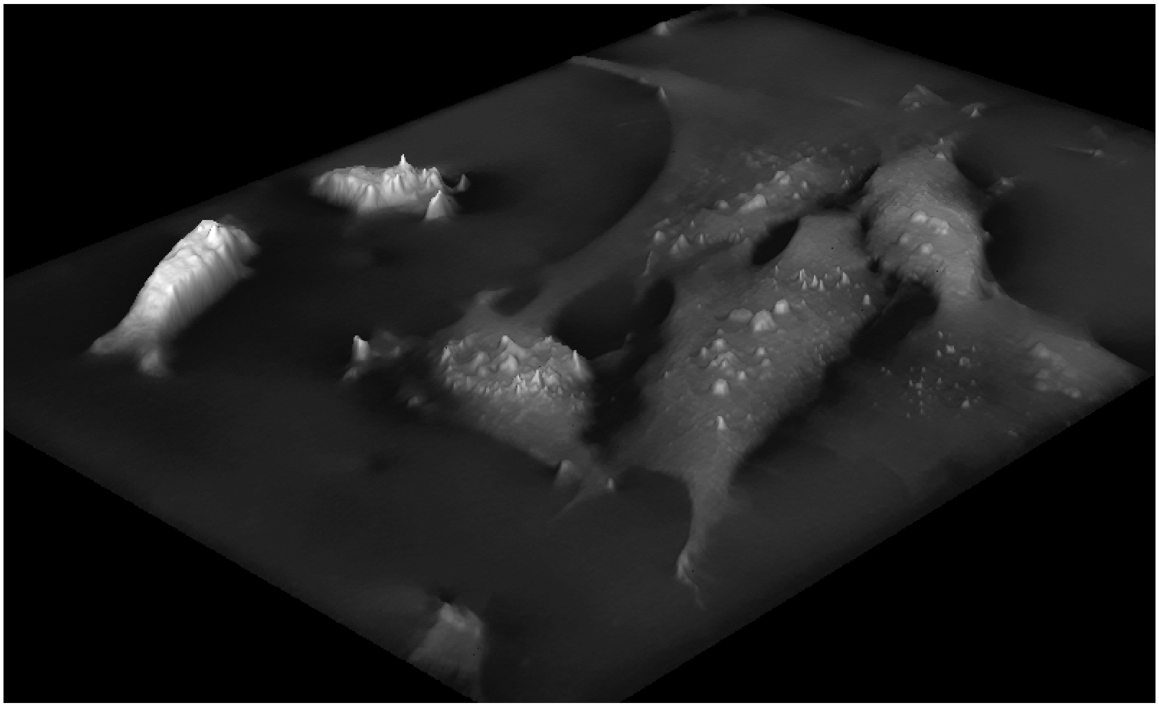


Figure 3.4: Reconstruction interpreted as heightmap.

3.4 Discussion

A computational method was described to measure the phase of near-transparent objects using DIC microscopy. The method is based on a calibration process using an object with known dimensions and refractive index. The DIC images are reconstructed using a variational framework. The reconstructed images are then converted to phase or height images. The process is described in a step-by-step fashion in Algorithm 1. The calculations are verified on microbeads of different sizes and applied to cell images. The technique has a precision of about one-third wavelength. Compared to other microscopy techniques that were mentioned earlier (e.g. PS-DIC and DPM), our technique provides a simple and cost-effective solution, it does not require additional microscope parts to be installed, and it reliably restores the phase information using just a single image.

Part II

Patch Clamp Pipette Detection Models

Chapter 4

Pipette Hunter

4.1 Introduction

Segmentation of objects with well-defined geometries is a fundamental problem in image analysis. Several methods were proposed to detect lines, ellipses or rectangles to identify roads [56], trees [57], or houses [58], respectively using marked point processes. Another way is to compromise strict geometries and use variational methods. For example higher-order active contours (HOAC) that can describe various objects with defined shape allowing slight variations of the boundaries. HOACs were successfully used to model circular objects [59] or complex road structures [60]. Recently a family of hybrid variational models was proposed [61, 62] that is capable of capturing circular and elliptical objects by minimizing only a few parameters. We have presented a variational method that extends the latter model to detect elongated straight object pairs that have a common reference point. We use this model to segment pipette tips under a microscope and automatically navigate these tips with micrometer precision for patch clamping and measure properties of nerve cells.

Patch clamping is a technique to study ion channels in cells. The technique was invented by Erwin Neher and Bert Sakmann in the early 1980s who received the Nobel Prize in Physiology or Medicine in 1991 for their work. Although the technique can be applied to a wide variety of cells, it is especially useful for measuring the electrophysiological properties of neurons.

The schematic process of patch clamping a single cell is the following. A glass pipette is pulled onto an electrode. The tip of the glass pipette is open, thus the measured signal originates only from the pipette tip because the glass does not transfer electricity. The pipette is then pushed next to a cell. When a tight connection, called "gigaseal" is formed between the cell and the pipette,

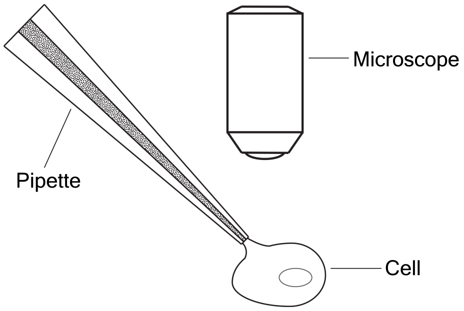


Figure 4.1: Schematic whole-cell patch clamping.

the cell membrane is broken by vacuum or relatively high voltage pulses. This way the whole-cell patch clamping configuration is established as illustrated in Fig. 4.1, the electric signal is passed to an amplifier and it is ready to be recorded.

The patch clamping process has to be repeated manually for every target cell. Experienced biologists can usually do only 10-30 successful patch clamping a day. The process is repetitive and monotonous, thus error-prone as the researchers get fatigued. Recently, efforts have been made to automate the technique. In [63] authors used their automatic patch-clamp setup for *in vivo* applications. In [64] the authors extended the technique to a multi-electrode system using up to 12 pipettes. A MATLAB implementation of an automatic patch-clamp software is publicly available [65]. A detailed description of building an automatic patch-clamp setup can be found in [66]. In [67] automatic patch clamping has been successfully used for cardiomyocytes. An issue of automatic patch clamping is that glass pipettes have to be changed after every patch clamping, limiting the throughput. In [68] the authors describe a way to clean the pipettes which allows them to be used about 10 times. Unlike *in vivo* applications, patch clamping is often used in tissue slices when there is an imaging modality that visualizes the target cells and the pipette. However, changing the pipettes introduces another problem. The pipettes are not perfectly identical and the tip can be slightly translated after the change. In [69] a method is proposed for automatic pipette tip detection using fluorescence channels. However, fluorescent materials can damage cell functions and are not always applicable. Recently, a method for pipette detection in label-free microscopy was proposed in combination with fluorescent cell detection in tissues [70]. The method is used in low (4x) magnification which provides a sharp image of the pipette due to the relatively thick focus plane, even if it is tilted. The detection is based on finding intersecting lines using Hough transform, which calculates the lateral position of the tip. The z position is refined using a focus detection algorithm.

In this part we describe three variants of the proposed novel Pipette Hunter tip detection algorithm which is based on a variational framework. The method works on differential interference contrast, oblique (Hoffman modulation contrast), and fluorescence microscopy image stacks that contain optically sliced images of a pipette. The label-free microscopy techniques provide dark image regions at the edge of the glass pipette. The method tries to fit two rectangles in 2D and two cylinders in 3D, with a common reference point on different projections of the image stack, or the image stack itself, respectively. In fluorescent modality the 3D object to be fit is a single cone. The idea of fitting a primitive shape to the image is inspired by the Snakuscule [61] algorithm that segments circular objects. Besides the exact location of the tip's endpoint in 3D, our detection algorithm determines the orientation and tilt angle as well. The results of the algorithms are compared to hand picked ground truth tip positions and they are implemented in a patch clamp software detailed in the next part.

4.2 2D Detection Model

The first variant of the tip detection method relies on the image formation of DIC (and similarly of oblique) microscopy. DIC is able to show very small optical path length differences in the sample. Diffraction, refraction, reflection, and very high optical path length differences can cause effects that are shown incorrectly by DIC. The light rays hit the sides of the pipette at a very flat angle, which results in strong effects of the mentioned principles. Due to these side-effects, the regions of pipette edges in the image will be dark, which provides information about the position of the pipette. Our method relies on this observation. Usually, only a small portion of the pipette is in focus in a single image, which is illustrated in Fig. 4.2a-b. We have developed a 2D model that works on the minimum

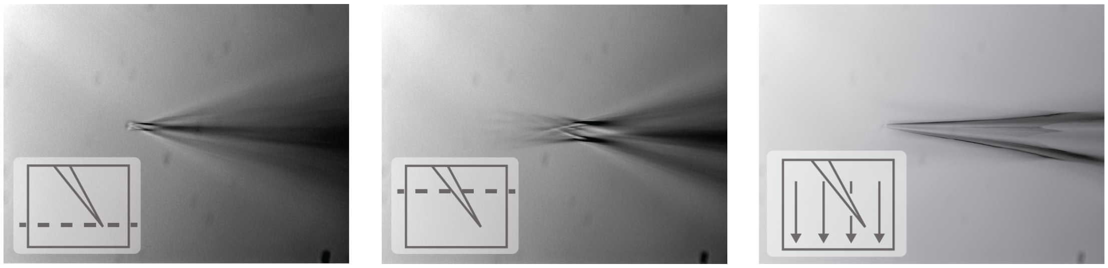

(a) (b) (c)

Figure 4.2: Example images of a pipette. (a) The pipette tip is nearly in focus. (b) The z level is moved 20 micrometers upwards compared to (a). (c) Minimum intensity projection along the third dimension of the image stack.

intensity projection (MIP) of an image stack. The MIP image contains dark stripes in the position of the pipette edges (Fig. 4.2c). We apply the proposed algorithm for all three possible projection directions to determine the exact position of the pipette tip in space.

4.2.1 The Pipette Hunter Model

The pipette capturing configuration is illustrated in Fig. 4.3. Let $\mathbf{i}$ and $\mathbf{j}$ define our standard basis at the image origin. The coordinates in the standard basis are denoted by x^1, x^2 and $I(x^1, x^2)$ is the image data. The main idea is to cover dark image regions (the edges of the pipette) with two wide rectangles given some constraints. The rectangles are aligned with two line segments that have a common end point, also called a pivot point. The pivot point is the reference point of the two rectangles. The line segments are called legs. The pivot point is given by its position vector $\mathbf{r} = x^1\mathbf{i} + x^2\mathbf{j}$. The rotation of the legs around the pivot point φ^1, φ^2 are measured from the $\mathbf{i}$ axis in the positive direction. The unit direction vectors of the legs are denoted by $\mathbf{e}_1, \mathbf{e}_2$ and their unit normals by $\mathbf{n}_1, \mathbf{n}_2$ respectively. ξ_{i1} and ξ_{i2} are distances from the pivot point in the direction of $\mathbf{e}_i$ that define the placement along the leg and the length of a rectangle ($i \in \{1, 2\}$, $\xi_{i2} > \xi_{i1} \geq 0$). η_{i1} and η_{i2} are distances in the normal directions that define the perpendicular placement and the thickness of a rectangle ($i \in \{1, 2\}$, $\eta_{i2} > \eta_{i1}$). Note, that η values can be given such that the rectangles are not symmetric to the legs, which allows fine-tuning of the pivot point in the algorithm. The model has 4 degrees-of-freedom (DOF), 2 for the coordinates of $\mathbf{r}$ and 1-1 for φ^1 and φ^2.

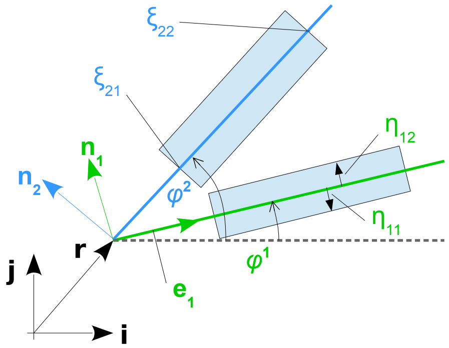


Figure 4.3: The 2D Pipette Hunter model.

4.2.2 The associated extreme-value problem

The points of the rectangles $\mathbf{p}_i$, $i \in \{1, 2\}$ can be decomposed either in the directions of the standard basis vectors or in the directions defined by their respective legs such that

$$\begin{aligned}\mathbf{p}_i &= \mathbf{r} + \xi_i \mathbf{e}_i + \eta_i \mathbf{n}_i \\ &= \left[x^1 + \xi_i \left(\cos\varphi^i\right) - \eta_i \left(\sin\varphi^i\right)\right] \mathbf{i} + \left[x^2 + \xi_i \left(\sin\varphi^i\right) + \eta_i \left(\cos\varphi^i\right)\right] \mathbf{j} \\ &= \left[x^1 \left(\cos\varphi^i\right) + x^2 \left(\sin\varphi^i\right) + \xi_i\right] \mathbf{e}_i + \left[-x^1 \left(\sin\varphi^i\right) + x^2 \left(\cos\varphi^i\right) + \eta_i\right] \mathbf{n}_i,\end{aligned} \tag{4.1}$$

where ξ_i and η_i are the local coordinates w.r.t. the pivot point. The area of the two rectangles can be simply given as:

$$A = \sum_{i=1}^{2} \left(\xi_{i2} - \xi_{i1}\right)\left(\eta_{i2} - \eta_{i1}\right). \tag{4.2}$$

We define the energy of the described system as the sum of the energies of the individual legs $E = \sum_{i=1}^{2} E_i$:

$$E\left(x^1, x^2, \varphi^1, \varphi^2\right) \doteq \sum_{i=1}^{2} \frac{1}{A} \int\limits_{\eta_i=\eta_{i1}}^{\eta_{i2}} \int\limits_{\xi_i=\xi_{i1}}^{\xi_{i2}} f(\xi_i, \eta_i) I\left(\mathbf{p}_i\right) \, d\xi_i \, d\eta_i, \tag{4.3}$$

where f is an appropriately chosen function representing any filter that rotates with the legs. Note that upper indices indicate variables on which the energy depends and are not powers. The easiest way to understand the energy function is to consider f to be identical to 1. In this case, the energy is low when the mean image intensity under the rectangles defined by the Pipette Hunter is low. The components of the energy gradient w.r.t. the coordinates of the pivot point are:

$$\begin{aligned}\frac{\partial E}{\partial x^1} &= \frac{\partial E}{\partial \mathbf{r}} \cdot \mathbf{i} \\ \frac{\partial E}{\partial x^2} &= \frac{\partial E}{\partial \mathbf{r}} \cdot \mathbf{j},\end{aligned} \tag{4.4}$$

where $\frac{\partial E}{\partial \mathbf{r}} \cdot \mathbf{b}$ is the scalar (dot) product of the gradient vector $\frac{\partial E}{\partial \mathbf{r}} \equiv E\nabla$ with one of the standard basis vectors $\mathbf{b} \in \{\mathbf{i}, \mathbf{j}\}$. The gradient vector itself is a sum of two vectors (i.e. the coordinates of the pivot point dependent on the energies of both legs), each of which can be decomposed in the directions of its own leg, such that the integration boundaries become constants:

$$\frac{\partial E}{\partial \mathbf{r}} = \sum_{i=1}^{2} \left[\left(\frac{\partial E_i}{\partial \mathbf{r}} \cdot \mathbf{e}_i\right) \mathbf{e}_i + \left(\frac{\partial E_i}{\partial \mathbf{r}} \cdot \mathbf{n}_i\right) \mathbf{n}_i\right]. \tag{4.5}$$

The energy gradient w.r.t. the rotations φ^1 and φ^2 are:

$$\frac{\partial E}{\partial \varphi^i} = \sum_{j=1}^{2} \frac{1}{A} \int\limits_{\eta_j=\eta_{j1}}^{\eta_{j2}} \int\limits_{\xi_j=\xi_{j1}}^{\xi_{j2}} f\left(\xi_j, \eta_j\right) I \nabla \left(\mathbf{p}_j\right) \cdot \frac{\partial \mathbf{p}_j}{\partial \varphi^i}\, d\xi_j\, d\eta_j. \tag{4.6}$$

From (4.1) the derivatives of the position vector $\mathbf{p}_j$ are:

$$\begin{aligned} \frac{\partial \mathbf{p}_j}{\partial \varphi^i} &= \delta_{ij} \left\{ \xi_j \left[-\left(\sin \varphi^i\right) \mathbf{i} + \left(\cos \varphi^i\right) \mathbf{j} \right] + \eta_j \left[-\left(\cos \varphi^i\right) \mathbf{i} - \left(\sin \varphi^i\right) \mathbf{j} \right] \right\} \\ &= \delta_{ij} \left(\xi_j \mathbf{n}_i - \eta_j \mathbf{e}_i\right). \end{aligned} \tag{4.7}$$

where δ_{ij} is the Kronecker delta function, indicating that the rotations of the legs contribute to the system energy independently, unlike the pivot coordinates.

Using equations (4.4)-(4.7) together with the following identities $\mathbf{i} \cdot \mathbf{e}_i = \cos \varphi^i$, $\mathbf{i} \cdot \mathbf{n}_i = -\sin \varphi^i$, $\mathbf{j} \cdot \mathbf{e}_i = \sin \varphi^i$, $\mathbf{j} \cdot \mathbf{n}_i = \cos \varphi^i$, and the definitions of the directional derivatives $I_{\xi_i} = I \nabla \cdot \mathbf{e}_i$, $I_{\eta_i} = I \nabla \cdot \mathbf{n}_i$, the complete system is written as:

$$\begin{aligned} \frac{\partial E}{\partial x^1} &= \frac{1}{A} \sum_{i=1}^{2} \int\limits_{\eta_{i1}}^{\eta_{i2}} \int\limits_{\xi_{i1}}^{\xi_{i2}} f\left(\xi_i, \eta_i\right) \left[\cos \varphi^i I_{\xi_i}\left(\mathbf{p}_i\right) - \sin \varphi^i I_{\eta_i}\left(\mathbf{p}_i\right)\right] d\xi_i\, d\eta_i \\ \frac{\partial E}{\partial x^2} &= \frac{1}{A} \sum_{i=1}^{2} \int\limits_{\eta_{i1}}^{\eta_{i2}} \int\limits_{\xi_{i1}}^{\xi_{i2}} f\left(\xi_i, \eta_i\right) \left[\sin \varphi^i I_{\xi_i}\left(\mathbf{p}_i\right) + \cos \varphi^i I_{\eta_i}\left(\mathbf{p}_i\right)\right] d\xi_i\, d\eta_i \\ \frac{\partial E}{\partial \varphi^i} &= \frac{1}{A} \int\limits_{\eta_{i1}}^{\eta_{i2}} \int\limits_{\xi_{i1}}^{\xi_{i2}} f\left(\xi_i, \eta_i\right) \left[\xi_i I_{\eta_i}\left(\mathbf{p}_i\right) - \eta_i I_{\xi_i}\left(\mathbf{p}_i\right)\right] d\xi_i\, d\eta_i, \quad i = 1, 2. \end{aligned} \tag{4.8}$$

Note that in the first two lines $\cos \varphi^i$ and $\sin \varphi^i$ could be brought before the integrals.

4.2.3 Simplification

Consider the simple case when no filter function is used: $f\left(\xi_i, \eta_i\right) \equiv 1$. Then the calculations will be limited to the boundaries of the rectangles. By using filters, the calculations can be expanded to the internal regions of the rectangles. Our simplified model uses no filter function. Let the primed ξ', η' variables note the variables measured from the origin of the standard basis in the directions of the respective local systems $\mathbf{e}_i$, $\mathbf{n}_i$, i.e. $(\xi_i', \eta_i') = (\mathbf{e}_i \cdot \mathbf{r} + \xi_i,\ \mathbf{n}_i \cdot \mathbf{r} + \eta_i)$. Note that the primed variables ξ', η' differ from their unprimed counterparts only by a displacement, hence $d\xi = d\xi'$, $d\eta = d\eta'$.

The gradient components of the energy w.r.t. the pivot point (i.e. the first two lines of the extreme value equations (4.8)) become single integrals:

$$
\begin{aligned}
\frac{\partial E}{\partial x^1} &= \frac{1}{A}\sum_{i=1}^{2}\left(\cos\varphi^i \int\limits_{\eta_i=\eta_{i1}}^{\eta_{i2}} I\left(\xi'_{i2},\eta'_i\right) - I\left(\xi'_{i1},\eta'_i\right)\, d\eta_i \right.\\
&\qquad \left. - \sin\varphi^i \int\limits_{\xi_i=\xi_{i1}}^{\xi_{i2}} I\left(\xi'_i,\eta'_{i2}\right) - I\left(\xi'_i,\eta'_{i1}\right)\, d\xi_i \right)\\
\frac{\partial E}{\partial x^2} &= \frac{1}{A}\sum_{i=1}^{2}\left(\sin\varphi^i \int\limits_{\eta_i=\eta_{i1}}^{\eta_{i2}} I\left(\xi'_{i2},\eta'_i\right) - I\left(\xi'_{i1},\eta'_i\right)\, d\eta_i \right.\\
&\qquad \left. + \cos\varphi^i \int\limits_{\xi_i=\xi_{i1}}^{\xi_{i2}} I\left(\xi'_i,\eta'_{i2}\right) - I\left(\xi'_i,\eta'_{i1}\right)\, d\xi_i \right).
\end{aligned}
\tag{4.9}
$$

Note that the integrands are the differences of the image intensity values on the opposite boundaries of the regions (that is, the opposite edges of the rectangles).

Similarly, the gradient components of the energy w.r.t. the angles φ^1, φ^2 (the third line of the extreme value equations (4.8)) become:

$$
\begin{aligned}
\frac{\partial E}{\partial \varphi^1} &= \frac{1}{A}\left(\int\limits_{\xi_1=\xi_{11}}^{\xi_{12}} \xi_1\left[I\left(\xi'_1,\eta'_{12}\right) - I\left(\xi'_1,\eta'_{11}\right)\right] d\xi_1 \right.\\
&\qquad \left. - \int\limits_{\eta_1=\eta_{11}}^{\eta_{12}} \eta_1\left[I\left(\xi'_{12},\eta'_1\right) - I\left(\xi'_{11},\eta'_1\right)\right] d\eta_1 \right)\\
\frac{\partial E}{\partial \varphi^2} &= \frac{1}{A}\left(\int\limits_{\xi_2=\xi_{21}}^{\xi_{22}} \xi_2\left[I\left(\xi'_2,\eta'_{22}\right) - I\left(\xi'_2,\eta'_{21}\right)\right] d\xi_2 \right.\\
&\qquad \left. - \int\limits_{\eta_2=\eta_{21}}^{\eta_{22}} \eta_2\left[I\left(\xi'_{22},\eta'_2\right) - I\left(\xi'_{21},\eta'_2\right)\right] d\eta_2 \right).
\end{aligned}
\tag{4.10}
$$

Note that unlike in the case of the pivot equations, the integrands (of the single integrals) are the weighted differences of the image intensity values on the opposite boundaries of the regions.

4.2.4 Solving the equations

One way to minimize the energy $E\left(q^i\right)$ of a system that depends on general variables q^i, $i = 1, 2, ...n$ is to find the stationary solution for the gradient descent evolution equation: $\frac{\partial q^i}{\partial \tau} = -\frac{\partial E}{\partial q^i}$, where τ is the "artificial" time, and at the stationary point $\frac{\partial q^i}{\partial \tau} = 0$ (hence $\frac{\partial E}{\partial q^i} = 0$).

In our case, the dimensions of the pivot point equations (4.9) and the rotation angle equations (4.10) are different. The first two are expressed in length units, while the second two are in radians, thus we perform a normalization. The complete system, using the local coordinate system, consists of four coupled differential equations:

$$\begin{aligned} \frac{\partial x^1}{\partial \tau} &= -\frac{\partial E}{\partial x^1} \\ \frac{\partial x^2}{\partial \tau} &= -\frac{\partial E}{\partial x^2} \\ \frac{\partial \varphi^1}{\partial \tau} &= -\frac{2}{\left(\xi_{11} + \xi_{12}\right)} \frac{\partial E}{\partial \varphi^1} \\ \frac{\partial \varphi^2}{\partial \tau} &= -\frac{2}{\left(\xi_{21} + \xi_{22}\right)} \frac{\partial E}{\partial \varphi^2}. \end{aligned} \tag{4.11}$$

The quantities on the right-hand side are defined in (4.9) and (4.10).

4.2.5 Properties and notes

The Pipette Hunter is neutral (does nothing) in a homogeneous environment. Weak "external forces", i.e. the constants in (4.11), may be added to avoid freezing in these regions.

In the simplified case, the integrals are calculated only on the edges of the rectangle-shaped region, hence only the image intensity distribution at the boundary is taken into account. The intensity function can freely vary inside. This is not always acceptable. To impose some regularity requirements inside the rectangles, an appropriate filter function can be applied, hence the general equations (4.8) need to be used. Minimizing the most significant Fourier coefficients of the intensity function (i.e. using a combination of the Fourier basis functions as a filter) allows minor variance inside the rectangles.

The regions of interest, and thus the boundaries of the local integrals can be asymmetric. For almost parallel stripes, the boundaries should also be asymmetrically defined in normal directions to prevent the pivot point falling far from the rectangles.

4.2.6 Implementation Details

We have implemented (4.11) in Matlab. If some part of the legs is outside the image boundaries, we use the median value of the image for calculations, which is usually very close to the background intensity. For ξ_{i1} we use 0, and for the length of the legs $(\xi_{i2} - \xi_{i1})$ we use half of the longer side of the image. We find this value to work well when the pipette covers about half of the image (it can be the shorter side as well) or more.

The η values should be chosen such that if a rectangle is fit on a pipette edge, its sides which are aligned to $\mathbf{e}_i$ lie on the opposite sides of the dark region. For our tests, we have empirically set 15 pixels for both η values which satisfied the above requirement, and thus the sides are symmetric to the corresponding leg. Note that the distance between the sides $(\eta_{i1} + \eta_{i2})$ becomes 30 pixels. If this distance is too short, the algorithm is not able to fit the model in dark regions. Similarly, if the distance is too high, the result can be inaccurate.

We start multiple instances of the Pipette Hunter mechanism in swarm to cover the whole image, which is not possible by using only 2 legs. A simple swarm setup is to place a few mechanisms with different rotations on grid points over the image. This number can be minimized to keep the runtime low. As we require the pipette to be around or over the center of the image and set the length of

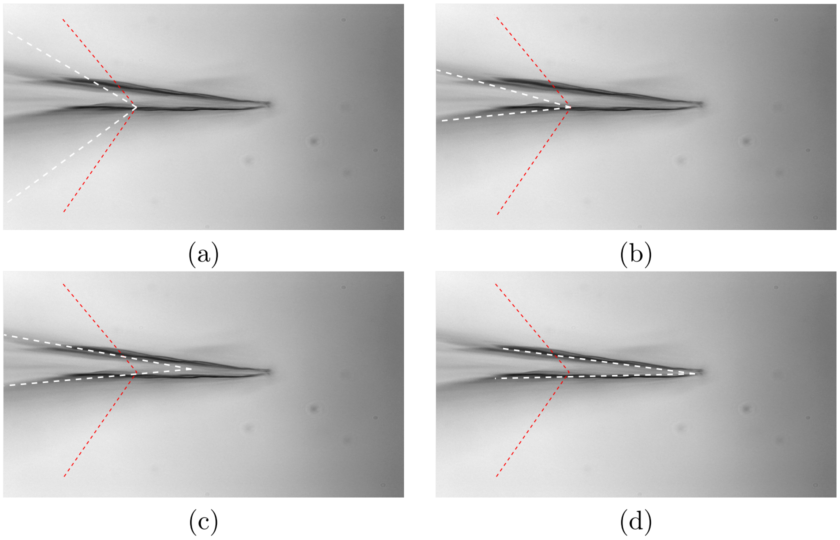

(a)
(b)
(c)
(d)

Figure 4.4: Images captured during the iterations of the algorithm. Red dashed lines are the starting position of the legs, which is kept in all the images for comparison. White dashed lines are the state of the legs in a specific iteration. (a) During the first phase both the introduced force and the image intensities pull the legs towards the dark regions. (b) End of the first phase. (c) In the second phase, the pivot point is also updated. The legs are moved and rotated to cover darker image regions. (d) The result of the detection.

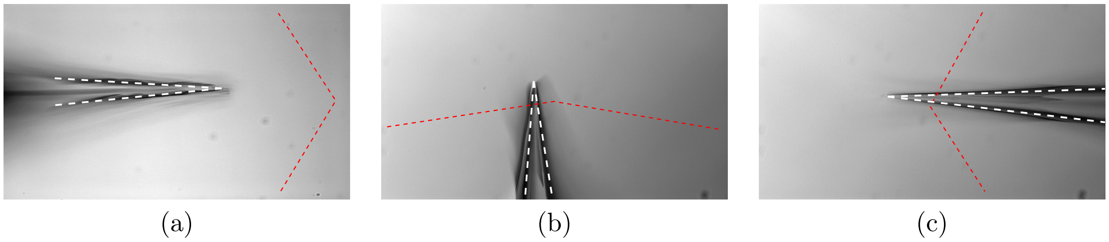


Figure 4.5: Example configurations and results.

the legs to be half of the longer side of the image, putting mechanisms to the sides, the corners, and the center of the image will be enough to find the pipette region with at least one instance. More specifically, we put one instance in each corner, one in the middle of each side, and four in the center, each with 90 degrees between the legs, which sums up to 12 instances in total. Note that this is a specific case and the search can fail if the requirements are not satisfied. In the general case where no assumption is made on the pipette's position, the runtime can be much higher as it depends also on the number of instances in the swarm. Furthermore, we use a 2-phase run of the algorithm. In the first phase, we only update the angles and apply a force that pulls the two legs towards each other. The phase ends if the changes of the angles are small or the legs get closer to each other than 0.1 rad. This allows the initialization of an instance with high angle difference (even $\pi/2$ rad or more). In the second phase we turn off the pulling force, update the pivot point as well and keep the restriction that does not let the legs get too close to each other. A few iterations of the algorithm on an example image using one mechanism is illustrated in Fig. 4.4.

4.2.7 Results

Comparison

To compare the detection of the Pipette Hunter to a reliable solution, we have manually determined the pipette tip positions in 31 stack images. Fig. 4.5 contains images where the pipette orientation and the starting points of the algorithm differs. The average absolute difference between the algorithm's result and the hand-picked focus points is 3.53 ± 2.47 µm, which is 32.97 ± 23.10 pixels and the image size was 1388 by 1040 pixels. This error is acceptable in automatic patch clamping since a cell's diameter is 10-20 µm and if the pipette tip is aimed at the cell's centroid, given the above error it will still reliably hit the cell. Furthermore, our results are better than the value reported in [69] for final tip-target distance in *in vitro* experiments, which was 12.06 ± 4.30 µm.

We have developed a simple baseline algorithm which can be compared to our method. The

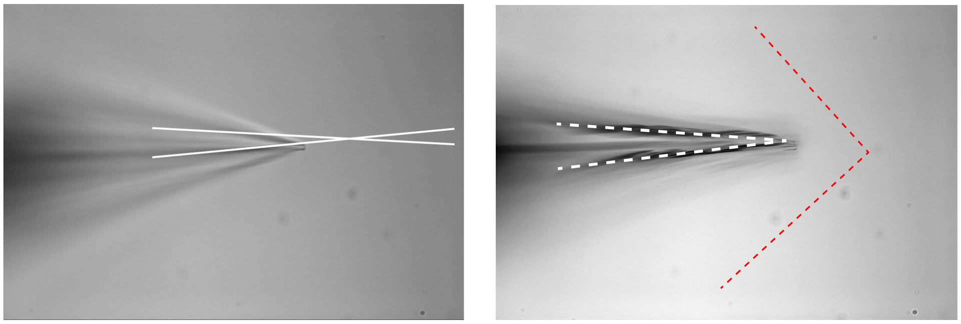


Figure 4.6: Comparison to the baseline algorithm (left) on the same image stack.

baseline model also searches for dark image regions. The model works if the pipette orientation is 0 rad. First, the algorithm searches for minimum point-pairs along the Y-axis in every slice, then fits two lines on them. The intersection of the two lines will be considered as the pipette tip. The algorithm has a linear runtime, but poor quality. The baseline model often over-detects the pipette tip, which is illustrated and compared to the proposed model in Fig. 4.6. On 15 appropriate image stacks (where the pipette orientation is 0 rad.) the mean absolute difference to the hand-picked solutions is 17.92 ± 9.49 µm (167.52 ± 88.68 pixels).

Application to fluorescence images

Patch clamping is sometimes performed in two-photon (fluorescence) imaging mode. The proposed pipette detection method can be applied to fluorescence images after an edge detection step (Sobel filter in our case) and inverting the image. The edges of the pipette will be dark regions. Fig. 4.7 shows the algorithm applied to a fluorescence image after the discussed preprocessing steps. Because the dark regions are narrow, we used smaller η values and longer legs.

4.3 3D Pipette Detection

The initial Pipette Hunter model described in Chapter 4.2 is a shape detection algorithm that takes 2D images as input. However, applying it for pipette tip detection requires two separate runs to get a 3D point as a result. A theoretically more elegant way to detect the tip position in space would be the 3D extension of this model which takes an image stack of the pipette as input and results in a 3D point in one pass. Since 3D algorithms usually have many DOFs it is reasonable to apply some initialization algorithm to determine the starting point instead of a swarm start.

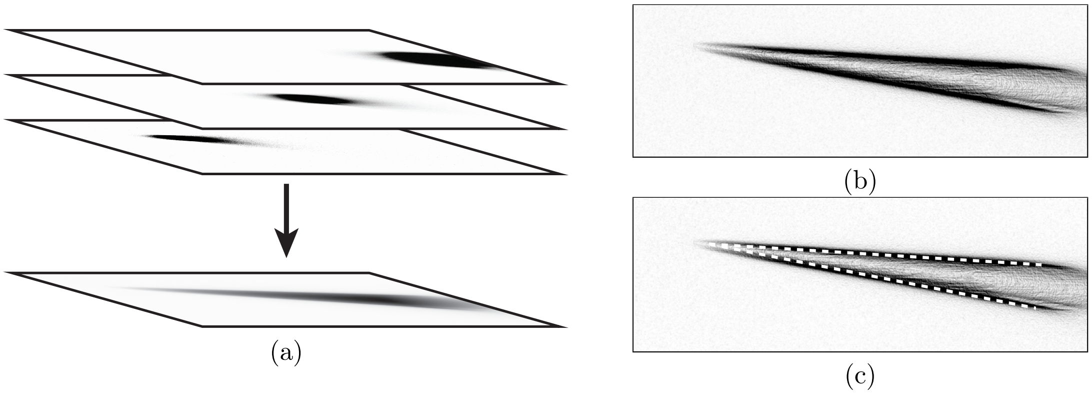


Figure 4.7: Example fluorescence images and the detection result. (a) Images of a fluorescent pipette in different z-levels and the projected fluorescence image. (b) The projected image filtered with the Sobel operator. (c) Result of the pipette detection algorithm.

The 3D pipette detection pipeline is introduced here. It consists of three steps. First, an image stack is acquired of the pipette. Ideally, the pipette should cover most of the image and should be nearly in focus. Moreover, the tip should be lowered near the sample to have similar light conditions during detection to what is present when the pipette is moved into the sample. Fig. 4.8 shows example images of a stack at characteristic focus levels. Then a fast initialization algorithm is run on the acquired stack. The initialization is based on a moving line profile in the image and is detailed in Section 4.3.1. Finally, the result of the initialization is refined using the main tip detection algorithm referred to as Pipette Hunter 3D (PH3D). The algorithm is the 3-dimensional extension of the base Pipette Hunter model [71]. This section describes this extension and the results are compared to the initialization algorithm and ground truth data.

4.3.1 Line profile estimation

An initialization is advantageous to the main detection algorithm as variational frameworks are known to be robust but require many iterations and the result is sensitive to the starting point. The developed initialization algorithm operates on minimum intensity projected (MIP) images of the stack containing the pipette. Two independent runs are required on different projection images (usually Z and X projections) to get a 3D point estimation.

The initialization algorithm estimates the tip position by analyzing a line profile in the MIP image. First, it is assumed that the orientation of the pipette is known either by setting this value manually or using the value set by the pipette calibration step. Then a line is placed in front of the

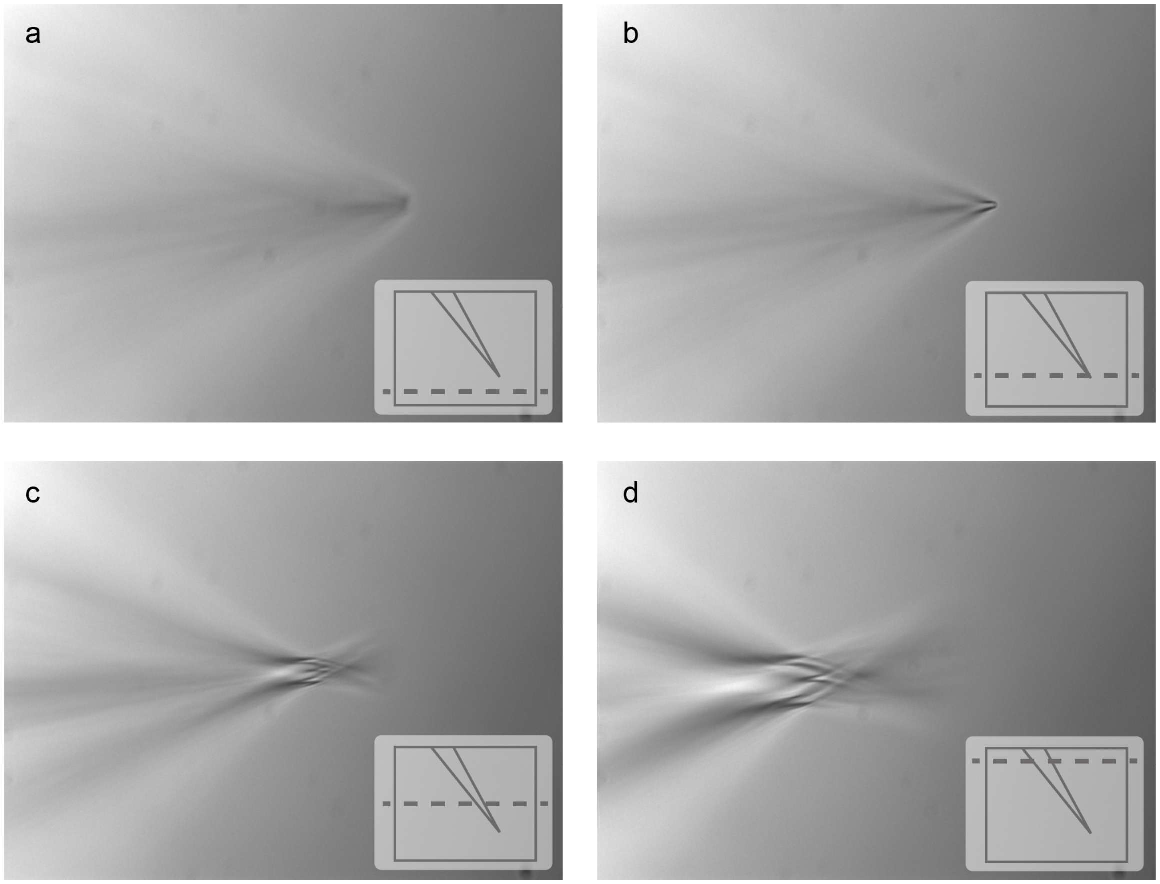
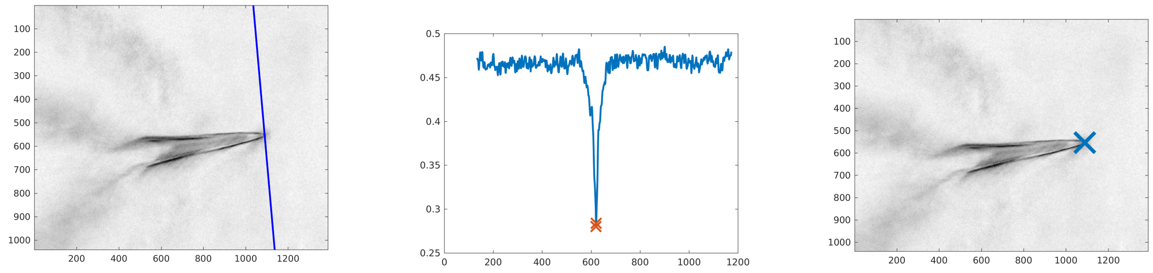


Figure 4.8: Images of a pipette from an image stack. Dashed lines show the position of the image slice in the stack. (a) The focus level is below the pipette tip. (b) The pipette tip is exactly in the focus plane. (c) The focus level is above the tip. (d) The last image of the pipette in the stack. Note that the part of the pipette in focus is not close to the side of the image which is generally the case.

pipette on the edge of the image and rotated such that it is perpendicular to the orientation of the pipette. The line profile is analyzed in those positions where it covers the image. Since the image is often noisy we compute the 90th percentile of the pixel values from the line profile which will be a reference value. If the profile includes at least one value which is significantly smaller than the reference value it is considered to be caused by the edge of the pipette. The significantly smaller value is defined as a 40% intensity drop but it can be set as a parameter to the algorithm. If more than one relevant value is present the smallest is chosen. Afterward, the position in the image of the chosen relevant small value is computed from the parameters of the line, the algorithm terminates and outputs this value as the position of the pipette tip. If no such value is present in the profile than the line is pushed towards the pipette tip by one pixel distance and the algorithm repeats from



(a) (b) (c)

Figure 4.9: Initialization using line profile estimation. (a) A line aligned perpendicularly to the orientation of the pipette is pushed against the pipette from the edge of the image. (b) The line profile of the image where the first significant intensity drop was found. (c) The position of the intensity drop in the line profile is converted back to a position in the image.

analyzing the line's profile. The different steps of the algorithm are presented in Fig. 4.9. The quality measurement of this initialization step is described in Section 4.3.3.

4.3.2 The 3D Model for Label-Free Images

Here we define the energy and derive its gradient for the two-wand mechanism designed to detect the pipette (see Fig. 4.10). The assumptions are the following:

- original configuration: the moving local frame $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$ with coordinates ξ^1, ξ^2, ξ^3 is aligned with the standard basis $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ with coordinates denoted by x^1, x^2, x^3
- the axes of the wands are situated on the plane spanned by $\mathbf{e}_2, \mathbf{n}$, satisfying the equation $\xi^2 = \pm\eta^1 \pm \xi^3 \tan\alpha^1$
- the symmetrical arrangement of the wands is retained during the iterative approach to the energy minimum
- ζ is the distance along the wand axis measured from the plane spanned by $\mathbf{e}_1$ and $\mathbf{e}_2$, ρ is the distance to the wand axis and ϕ is the angle around it, measured from $\mathbf{e}_1$ in the plane $\mathbf{e}_1, \mathbf{e}_2$.

The wand positions are completely determined (besides the symmetry condition above) by

- the position vector $\mathbf{r}\left(x^1, x^2, x^3\right) = x^1\mathbf{i} + x^2\mathbf{j} + x^3\mathbf{k}$ that designates the origin (or pivot point) of the local frame $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$
- the rotation of the local frame around its pivot point (we parameterize the space of rotations with proper Euler angles $\varphi^1, \varphi^2, \varphi^3$, using extrinsic $z - y - z$ scheme)

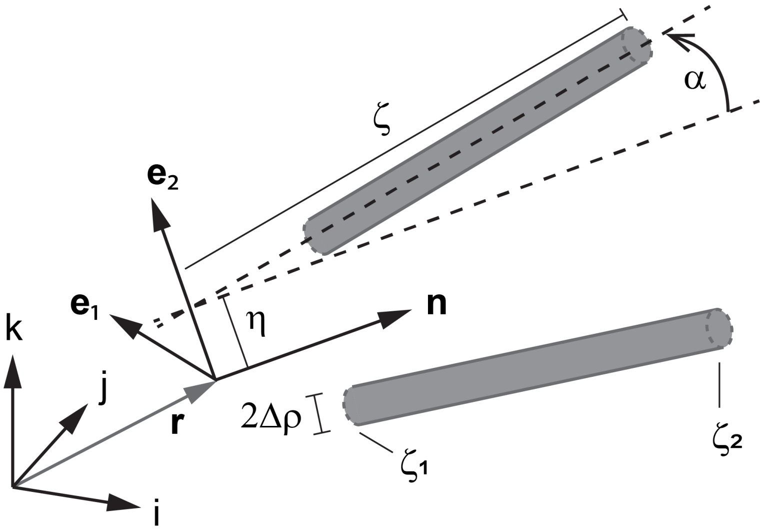


Figure 4.10: The 8 DOF pipette detection mechanism for 3D label-free images.

- the coordinates η^1, α^1 which are given relative to the moving local frame
- the (fixed parameter) values ζ_1, ζ_2 representing the start and the endpoints of the wands along their axes.

The system has therefore 8 DOFs given by the variables having upper indices in the above list: $x^1, x^2, x^3, \varphi^1, \varphi^2, \varphi^3, \eta^1, \alpha^1$.

The local coordinates of a point w.r.t. the local frame $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$ inside the cylinder-shaped wands, using cylindrical coordinates ρ, ϕ, ζ are given by:

$$\begin{aligned} \xi^1(\rho,\phi,\zeta) &= \rho\cos\phi \\ \xi^2(\rho,\phi,\zeta) &= \pm\left(\eta^1+\zeta\sin\alpha^1\right)+(\rho\sin\phi)\cos\alpha^1 \\ \xi^3(\rho,\phi,\zeta) &= \zeta\cos\alpha^1\mp(\rho\sin\phi)\sin\alpha^1, \end{aligned} \tag{4.12}$$

$\rho\in[0,\triangle\rho]$, $\zeta\in[\zeta_1,\zeta_2]$, $\phi\in[0,2\pi)$, then the internal point of wands $\mathbf{P}\left(x^i,\varphi^k,\eta^1,\alpha^1\right)=\mathbf{r}+\xi^1\mathbf{e}_1+\xi^2\mathbf{e}_2+\xi^3\mathbf{n}$, where i, $k=1,2,3$ is identified by the formula:

$$\begin{aligned} \mathbf{P}_\pm = \mathbf{r}\left(x^i\right) \pm \eta^1\mathbf{e}_2\left(\varphi^k\right) + \zeta\left(\cos\alpha^1\mathbf{n}\left(\varphi^k\right) \pm \sin\alpha^1\mathbf{e}_2\left(\varphi^k\right)\right) \\ + \rho\left[\sin\phi\left(\cos\alpha^1\mathbf{e}_2\left(\varphi^k\right)\mp\sin\alpha^1\mathbf{n}\left(\varphi^k\right)\right)+\cos\phi\mathbf{e}_1\left(\varphi^k\right)\right]\ . \end{aligned} \tag{4.13}$$

Note that in (4.13) the second line identifies the points of the wand axes. Now we are ready to define the energy of the system as the integral of the voxel intensities over the volume occupied by the wands:

$$E\left(x^i,\varphi^k,\eta^1,\alpha^1\right) \doteq \int_V \left[I\left(\mathbf{P}_+\right)+I\left(\mathbf{P}_-\right)\right]dV = \int_{\zeta_1}^{\zeta_2}\int_0^{2\pi}\int_0^{\triangle\rho}\left[I\left(\mathbf{P}_+\right)+I\left(\mathbf{P}_-\right)\right]\rho\,d\rho\,d\phi\,d\zeta. \quad (4.14)$$

The local coordinates as the functions of the variables of the integration ρ, ϕ, ζ are given by (4.12). Energy (4.14) has its minimal value wherever its derivatives w.r.t. the variables defining the system DOFs are all zero. The list of the required derivatives is:

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{r}} &= \int_V \left[I\nabla\left(\mathbf{P}_+\right)+I\nabla\left(\mathbf{P}_-\right)\right]dV \\
\frac{\partial E}{\partial \varphi^k} &= \int_V \left[I\nabla\left(\mathbf{P}_+\right)\cdot\frac{\partial \mathbf{P}_+}{\partial \varphi^k}+I\nabla\left(\mathbf{P}_-\right)\cdot\frac{\partial \mathbf{P}_-}{\partial \varphi^k}\right]dV \\
\frac{\partial E}{\partial \eta^1} &= \int_V \left[I\nabla\left(\mathbf{P}_+\right)\cdot\frac{\partial \mathbf{P}_+}{\partial \eta^1}+I\nabla\left(\mathbf{P}_-\right)\cdot\frac{\partial \mathbf{P}_-}{\partial \eta^1}\right]dV \\
\frac{\partial E}{\partial \alpha^1} &= \int_V \left[I\nabla\left(\mathbf{P}_+\right)\cdot\frac{\partial \mathbf{P}_+}{\partial \alpha^1}+I\nabla\left(\mathbf{P}_-\right)\cdot\frac{\partial \mathbf{P}_-}{\partial \alpha^1}\right]dV\,.
\end{aligned} \quad (4.15)$$

In the case of the pivot point (1st line) the coordinate system is aligned to the wands:

$$\begin{aligned}
\mathbf{e}_{2+} &= \cos\alpha^1\mathbf{e}_2 - \sin\alpha^1\mathbf{n} \\
\mathbf{n}_+ &= \sin\alpha^1\mathbf{e}_2 + \cos\alpha^1\mathbf{n} \\
\mathbf{e}_{2-} &= \cos\alpha^1\mathbf{e}_2 + \sin\alpha^1\mathbf{n} \\
\mathbf{n}_- &= -\sin\alpha^1\mathbf{e}_2 + \cos\alpha^1\mathbf{n}
\end{aligned} \quad (4.16)$$

Then the equation simply leads to the dot product of the image gradient and the unit vectors of the local coordinate system:

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{R}} = \mathbf{e}_1 &\int\limits_{\zeta_1}^{\zeta_2}\int\limits_0^{2\pi}\int\limits_0^{\Delta\rho} I_\xi\left(\mathbf{P}_+\right)\rho\,d\rho\,d\phi\,d\zeta + \mathbf{e}_{2+}\int\limits_{\zeta_1}^{\zeta_2}\int\limits_0^{2\pi}\int\limits_0^{\Delta\rho} I_\eta\left(\mathbf{P}_+\right)\rho\,d\rho\,d\phi\,d\zeta \\
+\,\mathbf{e}_1 &\int\limits_{\zeta_1}^{\zeta_2}\int\limits_0^{2\pi}\int\limits_0^{\Delta\rho} I_\xi\left(\mathbf{P}_-\right)\rho\,d\rho\,d\phi\,d\zeta + \mathbf{e}_{2-}\int\limits_{\zeta_1}^{\zeta_2}\int\limits_0^{2\pi}\int\limits_0^{\Delta\rho} I_\eta\left(\mathbf{P}_-\right)\rho\,d\rho\,d\phi\,d\zeta
\end{aligned}$$

$$+\,\mathbf{n}_{+}\int_{0}^{2\pi}\int_{0}^{\Delta\rho}\left[I\left(\zeta_2\right)-I\left(\zeta_1\right)\right]\rho\,d\rho\,d\phi\,d\zeta+\mathbf{n}_{-}\int_{0}^{2\pi}\int_{0}^{\Delta\rho}\left[I\left(\zeta_2\right)-I\left(\zeta_1\right)\right]\rho\,d\rho\,d\phi\,d\zeta$$

$$\frac{\partial E}{\partial x^1}=\mathbf{i}\cdot\frac{\partial E}{\partial\mathbf{R}},\quad\frac{\partial E}{\partial x^2}=\mathbf{j}\cdot\frac{\partial E}{\partial\mathbf{R}},\quad\frac{\partial E}{\partial x^3}=\mathbf{k}\cdot\frac{\partial E}{\partial\mathbf{R}}\tag{4.17}$$

For the derivation w.r.t. the rotation variables we use the following idea. The local coordinates in (4.12) are known for all values $\rho\in[0,\triangle\rho]$, $\zeta\in[\zeta_1,\zeta_2]$, $\phi\in[0,2\pi)$. We wish to express the quantities that occur in (4.15) with these local coordinates. One natural way to describe the orientation of a unit direction vector is the usage of spherical angles, where φ^1 is the angle measured from the standard basis vector $\mathbf{i}$ and φ^2 is the angle measured from the standard basis vector $\mathbf{k}$. This can be described with two rotations, first about the Y-axis, then about the Z-axis (using extrinsic description). The resulting rotation sequence applicable to (column) vectors from the left is: $\mathbf{R}_{\varphi^1}\mathbf{R}_{\varphi^2}$. Allowing full degrees of rotational freedom, the natural extension is to enable the object spinning around itself by a third angle φ^3, hence using the complete Euler-angles configuration Z-Y-Z or $\mathbf{R}_{\varphi^1}\mathbf{R}_{\varphi^2}\mathbf{R}_{\varphi^3}$:

$$\mathbf{R}_{\varphi^1}\mathbf{R}_{\varphi^2}\mathbf{R}_{\varphi^3}=\begin{bmatrix}\cos\varphi^1 & -\sin\varphi^1 & 0\\ \sin\varphi^1 & \cos\varphi^1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\varphi^2 & 0 & -\sin\varphi^2\\ 0 & 1 & 0\\ \sin\varphi^2 & 0 & \cos\varphi^2\end{bmatrix}\begin{bmatrix}\cos\varphi^3 & -\sin\varphi^3 & 0\\ \sin\varphi^3 & \cos\varphi^3 & 0\\ 0 & 0 & 1\end{bmatrix}=$$
$$\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{n}\end{bmatrix}.\tag{4.18}$$

The basis vectors of the local system $(\mathbf{e}_1,\mathbf{e}_2,\mathbf{n})$ are expressed in the standard basis (the first two angles φ^1 and φ^2 designate the azimuthal and polar coordinates of $\mathbf{n}$ respectively while φ^3 represents the spin of the plane spanned by $\mathbf{e}_1,\mathbf{e}_2$). The derivatives are determined by direct calculation, the results can be expressed by the angles and the local basis vectors as:

$$\begin{array}{lll}\dfrac{\delta\mathbf{e}_1}{\delta\varphi^1}=\cos\varphi^2\mathbf{e}_2-\sin\varphi^2\sin\varphi^3\mathbf{n}, & \dfrac{\delta\mathbf{e}_1}{\delta\varphi^2}=-\cos\varphi^3\mathbf{n}, & \dfrac{\delta\mathbf{e}_1}{\delta\varphi^3}=\mathbf{e}_2\\ \dfrac{\delta\mathbf{e}_2}{\delta\varphi^1}=-\cos\varphi^2\mathbf{e}_1-\sin\varphi^2\sin\varphi^3\mathbf{n}, & \dfrac{\delta\mathbf{e}_2}{\delta\varphi^2}=\sin\varphi^3\mathbf{n}, & \dfrac{\delta\mathbf{e}_2}{\delta\varphi^3}=-\mathbf{e}_1\\ \dfrac{\delta\mathbf{n}}{\delta\varphi^1}=\sin\varphi^2\left(\sin\varphi^3\mathbf{e}_1+\cos\varphi^3\mathbf{e}_2\right) & \dfrac{\delta\mathbf{n}}{\delta\varphi^2}=\cos\varphi^3\mathbf{e}_1-\sin\varphi^3\mathbf{e}_2, & \dfrac{\delta\mathbf{n}}{\delta\varphi^3}=0\end{array}\tag{4.19}$$

This leads to the following partial derivatives of $\mathbf{P}$ which can be used in the second line of (4.15):

$$\begin{aligned}\frac{\partial\mathbf{P}_{\pm}}{\partial\varphi^1}=&\rho\cos\phi\left(\cos\phi^2\mathbf{e}_2-\sin\varphi^2\sin\varphi^3\mathbf{n}\right)\\&-\left(\pm\eta^1\pm\zeta\sin\alpha^1+\rho\cos\alpha^1\sin\phi\right)\left(\cos\varphi^2\mathbf{e}_1+\sin\varphi^2\cos\varphi^3\mathbf{n}\right)\end{aligned}$$

$$
\begin{aligned}
&+ \left(\zeta \cos\alpha^1 \mp \rho \sin\alpha^1 \sin\phi\right) \cos\varphi^2 \left(\sin\varphi^3 \mathbf{e}_1 + \cos\varphi^3 \mathbf{e}_2\right) \qquad (4.20)\\
\frac{\partial \mathbf{P}_\pm}{\partial \varphi^2} &= + \left[\left(\pm\eta^1 \pm \zeta \sin\alpha^1 + \rho \sin\phi \cos\alpha^1\right) \sin\varphi^3 - \rho \cos\phi \cos\varphi^3\right] \mathbf{n}\\
&+ \left(\zeta \cos\alpha^1 \mp \rho \sin\alpha^1 \sin\phi\right) \left(\cos\varphi^3 \mathbf{e}_1 - \sin\varphi^3 \mathbf{e}_2\right)\\
\frac{\partial \mathbf{P}_\pm}{\partial \varphi^3} &= \rho \cos\phi \mathbf{e}_2 - \left(\pm\eta^1 \pm \zeta \sin\alpha^1 + \rho \cos\alpha^1 \sin\phi\right) \mathbf{e}_1
\end{aligned}
$$

For η and α the partial derivatives of $\mathbf{P}$ become:

$$
\begin{aligned}
\frac{\partial \mathbf{P}_\pm}{\partial \eta^1} &= \pm\mathbf{e}_2 \qquad (4.21)\\
\frac{\partial \mathbf{P}_\pm}{\partial \alpha^1} &= \left(\pm\zeta \cos\alpha^1 - \rho \sin\alpha^1 \sin\phi\right) \mathbf{e}_2 + \left(-\zeta \sin\alpha^1 \mp \rho \cos\alpha^1 \sin\phi\right) \mathbf{n}
\end{aligned}
$$

4.3.3 Results

The performance of the algorithms was evaluated on 5 acquired image stacks of different pipettes. The pipettes were lowered around the working distance to have similar light conditions to the case of experiments with samples. The pipette tip was moved over at least half of the image and it was assured that the stack always contains it. The voxel position of the tip was manually selected as ground truth in the stacks. Then the initialization algorithm was run and its results were the starting point of PH3D. The results of both the initialization algorithm and PH3D were saved and compared to the manually picked positions. Fig. 4.11 shows three different projections of a result and an enlarged image part with the different points for comparison. The mean difference between the line profile estimation and the ground truth data was 14.64 ± 4.23 voxels, while for PH3D the error was 8.63 ± 4.80 voxels. The pixel size was 115 nm and the distance between the slices in the stack was 1 µm, thus the error expressed in micrometers is 0.99 ± 0.55 µm. This error value is fairly low to reliably hit a cell if the pipette is oriented to its center since the diameter of cells is usually in the range of 5-20 µm. Furthermore, this error value is smaller than the reported 3.53 ± 2.47 µm or 32.97 ± 23.10 pixels in case of the previous, 2D model.

4.4 3D Model for Fluorescence Images

A simpler but still 8 DOF model configuration which can be used to detect pipettes filled with fluorescent material is illustrated in Fig. 4.12. The mechanism is intended to align its nappe with a rotational symmetric truncated cone image. Since it has rotation symmetry, two independent angles

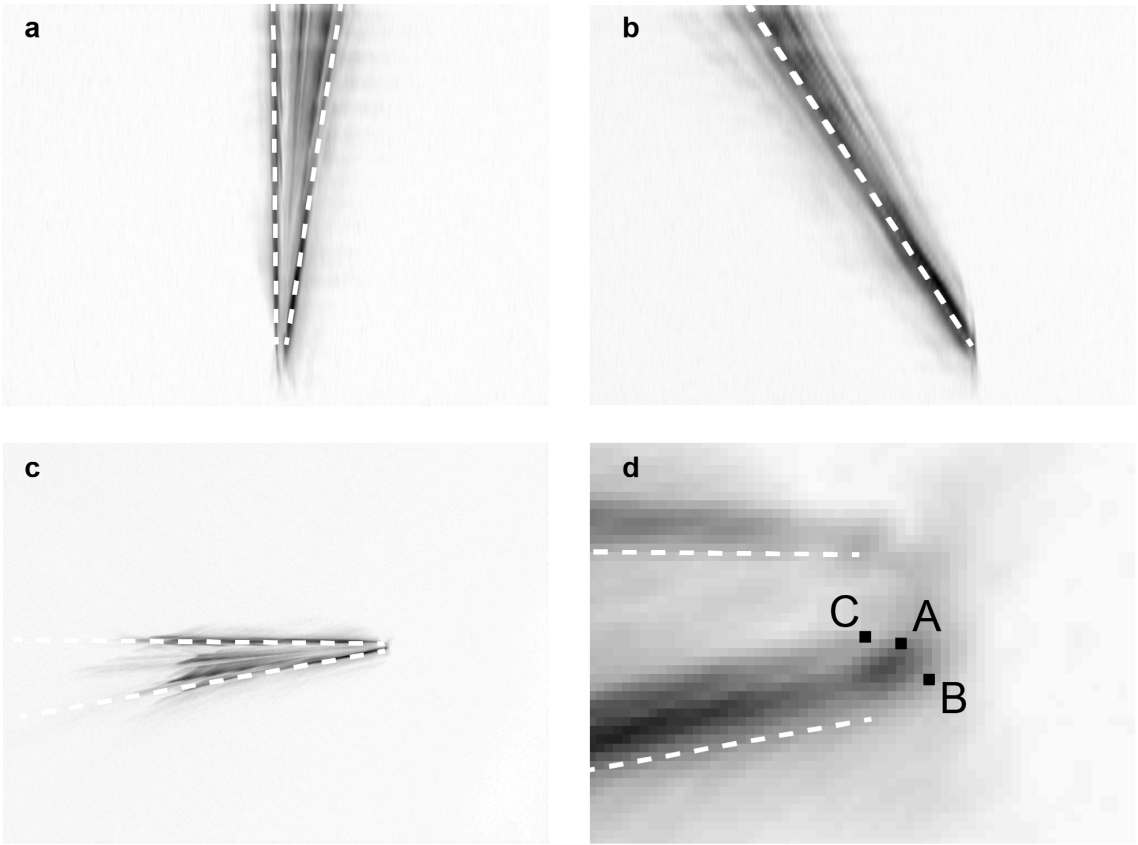


Figure 4.11: Result images of PH3D. (a) X projection. (b) Y projection. (c) Z projection. (d) Enlarged Z projection at the tip. The ground truth (A) and the result points of the initialization algorithm (B) and the PH3D (C) are highlighted.

(e.g. the spherical angles) determine its orientation. However, this symmetry is not utilized *a priori* in the energy but taken into account later during the analysis of the associated extreme-valued equation. The notation mostly matches with what is used in Section 4.3.2 and the differences are pointed out. Here $\mathbf{R}$ denotes the reference point the model is described around. Furthermore, the radii of the truncated cone are given by an initial radius variable r along the aligned axis and a slope variable m. The length of the cone l is also a variable, i.e. not fixed. The method could benefit from an initialization algorithm similar to the one described in Section 4.3.1, however, it is not included in the tests.

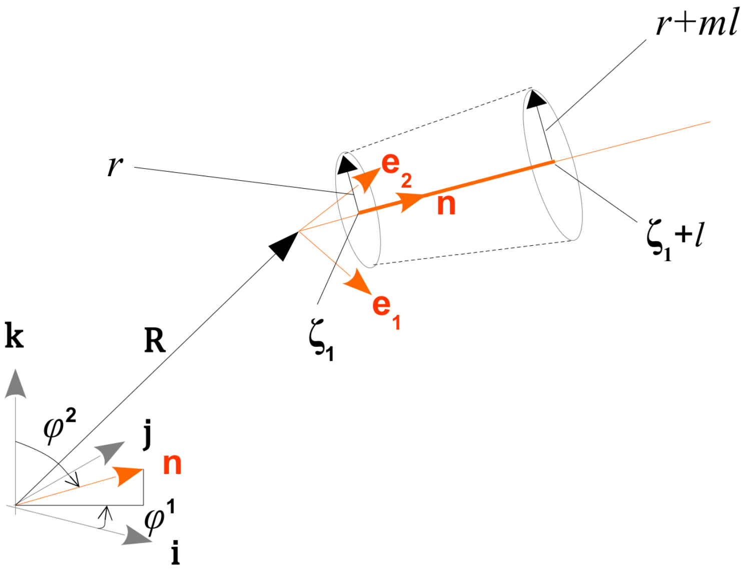


Figure 4.12: The 8 DOF 3D fluorescent pipette detection mechanism.

4.4.1 The Fluorescence 3D Model

A point inside the truncated cone using the spatial coordinates of the pivot point and the local coordinates in the $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{n}$ basis as derived in (4.18) is formulated as:

$$\begin{aligned}
\mathbf{P} &= \mathbf{R}\left(x^1, x^2, x^3\right) + \zeta\,\mathbf{n}\left(\varphi^1, \varphi^2, \varphi^3\right) + \rho\left[\cos\phi\,\mathbf{e}_1\left(\varphi^1, \varphi^2, \varphi^3\right) + \sin\phi\,\mathbf{e}_2\left(\varphi^1, \varphi^2, \varphi^3\right)\right] \\
\mathbf{P} &= \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix} + \zeta \begin{bmatrix} \cos\varphi^1 \sin\varphi^2 \\ \sin\varphi^1 \sin\varphi^2 \\ \cos\varphi^2 \end{bmatrix} \\
&+ \rho \left(\cos\phi \begin{bmatrix} \cos\varphi^1 \cos\varphi^2 \cos\varphi^3 - \sin\varphi^1 \sin\varphi^3 \\ \sin\varphi^1 \cos\varphi^2 \cos\varphi^3 + \cos\varphi^1 \sin\varphi^3 \\ -\sin\varphi^2 \cos\varphi^3 \end{bmatrix} + \sin\phi \begin{bmatrix} -\cos\varphi^1 \cos\varphi^2 \sin\varphi^3 - \sin\varphi^1 \cos\varphi^3 \\ -\sin\varphi^1 \cos\varphi^2 \sin\varphi^3 + \cos\varphi^1 \cos\varphi^3 \\ \sin\varphi^2 \sin\varphi^3 \end{bmatrix} \right)
\end{aligned} \tag{4.22}$$

Then the normalized energy of the capturing mechanism can be defined as follows:

$$\begin{aligned}
E\left(x^1, x^2, x^3, \varphi^1, \varphi^2, \varphi^3, r, m, l\right) &= \frac{1}{V} \int\limits_V I\left(\mathbf{P}\right)\, dV \\
&= \frac{1}{V} \int\limits_{\phi=0}^{2\pi} \int\limits_{\zeta=\zeta_1}^{\zeta_1+l} \int\limits_{\rho=0}^{r+m(\zeta-\zeta_1)} I\left(\mathbf{P}\right) \rho\, d\rho\, d\zeta\, d\phi,
\end{aligned} \tag{4.23}$$

where V is the volume of the mechanism. The expression includes the radii of the truncated cones as the function of the variables r, m and the length l:

$$r\left(\zeta; r, m\right) = r + m\left(\zeta - \zeta_1\right) = \left(r - m\zeta_1\right) + m\zeta \quad \Rightarrow \quad V = \pi l \left(r^2 + mlr + \frac{1}{3} l^2 m^2\right) \tag{4.24}$$

The following equations describe the energy gradients without detailed derivation, which is sufficient for implementation. First, the equations for r, m and l are the most straightforward, because additional quantities will be introduced for the others. For component r it is:

$$\begin{aligned}\frac{\partial E}{\partial r} = -\frac{\pi l}{V^2}\left(2r + ml\right) \int_{0}^{2\pi} \int_{\zeta=\zeta_1}^{\zeta_1+l} \int_{\rho=0}^{r+m(\zeta-\zeta_1)} I\left(\mathbf{P}\right) \rho \, d\rho \, d\zeta \, d\phi \\ + \frac{1}{V} \int_{0}^{2\pi} \int_{\zeta=\zeta_1}^{\zeta_1+l} \left[r + m\left(\zeta - \zeta_1\right)\right] I|_{\rho=r+m(\zeta-\zeta_1)} \, d\zeta \, d\phi\end{aligned} \tag{4.25}$$

In the second term, the weighted intensities are integrated over the nappe points. The equation for component m is:

$$\begin{aligned}\frac{\partial E}{\partial m} = -\frac{\pi l^2}{V^2}\left(r + \frac{2}{3} lm\right) \int_{0}^{2\pi} \int_{\zeta=\zeta_1}^{\zeta_1+l} \int_{\rho=0}^{r+m(\zeta-\zeta_1)} I\left(\mathbf{P}\right) \rho \, d\rho \, d\zeta \, d\phi \\ + \frac{1}{V} \int_{0}^{2\pi} \int_{\zeta=\zeta_1}^{\zeta_1+l} \left[r\left(\zeta - \zeta_1\right) + m\left(\zeta - \zeta_1\right)^2\right] I|_{\rho=r+m(\zeta-\zeta_1)} \, d\zeta \, d\phi\end{aligned} \tag{4.26}$$

Similarly to r the integral in the second term incorporates the nappe points. Finally, the equation for l is:

$$\frac{\partial E}{\partial l} = -\frac{\pi}{V^2}\left(r^2 + 2lmr + l^2 m^2\right) \int_{0}^{2\pi} \int_{\zeta=\zeta_1}^{\zeta_1+l} \int_{\rho=0}^{r+m(\zeta-\zeta_1)} I\left(\mathbf{P}\right) \rho \, d\rho \, d\zeta \, d\phi + \frac{1}{V} \int_{\phi=0}^{2\pi} \int_{\rho=0}^{r+m(\zeta-\zeta_1)} \rho I|_{\zeta_1+l} \, d\rho \, d\phi \tag{4.27}$$

For the remaining elements of the gradient the following variables are introduced: $\xi = \rho\cos\phi$, $\eta = \rho\cos\phi$, $I_\xi = I\nabla\cdot\mathbf{e}_1$, $I_\eta = I\nabla\cdot\mathbf{e}_2$ and $I_\zeta = I\nabla\cdot\mathbf{n}$. Then the Euler-angle equations can be summarized as follows:

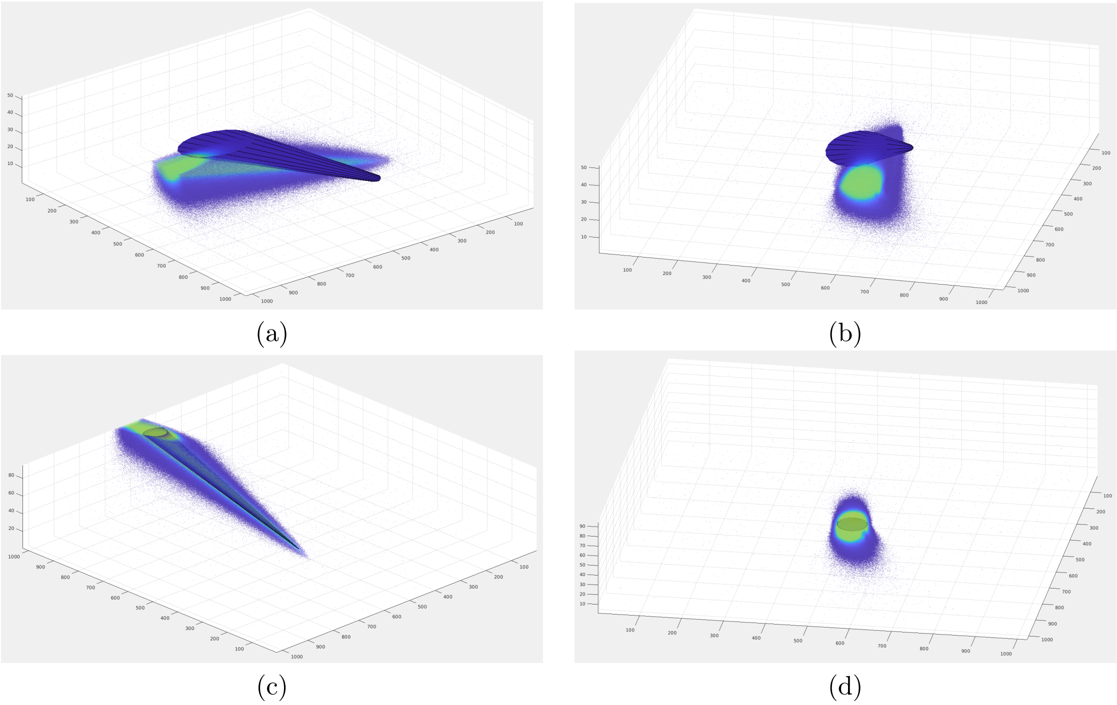
$$\begin{aligned}
\frac{\partial E}{\partial \varphi^1} &= -\frac{\sin \varphi^2}{V} \iiint\limits_V (\eta I_\zeta - \zeta I_\eta)\ dV \\
\frac{\partial E}{\partial \varphi^2} &= \frac{1}{V} \iiint\limits_V (\zeta I_\xi - \xi I_\zeta)\ dV \\
\left(\frac{\partial E}{\partial \varphi^3} \right. &= \left. 0 \right)
\end{aligned} \tag{4.28}$$

Finally, the energy-gradient of the pivot point is expressed in the local system:

$$\begin{aligned}
&\frac{1}{V} \iiint\limits_V I \nabla (\mathbf{P})\ dV = \frac{1}{V} \iiint\limits_V (I_\xi \mathbf{e}_1 + I_\eta \mathbf{e}_2 + I_\zeta \mathbf{n})\ dV \\
&\quad \Rightarrow \\
&\frac{\partial E}{\partial x^1} = \frac{\cos \varphi^1 \cos \varphi^2}{V} \iiint\limits_V I_\xi\, dV - \frac{\sin \varphi^1}{V} \iiint\limits_V I_\eta\, dV + \frac{\cos \varphi^1 \sin \varphi^2}{V} \iiint\limits_V I_\zeta\, dV \\
&\frac{\partial E}{\partial x^2} = \frac{\sin \varphi^1 \cos \varphi^2}{V} \iiint\limits_V I_\xi\, dV + \frac{\cos \varphi^1}{V} \iiint\limits_V I_\eta\, dV + \frac{\sin \varphi^1 \sin \varphi^2}{V} \iiint\limits_V I_\zeta\, dV \\
&\frac{\partial E}{\partial x^3} = -\frac{\sin \varphi^2}{V} \iiint\limits_V I_\xi\, dV + \frac{\cos \varphi^2}{V} \iiint\limits_V I_\zeta\, dV
\end{aligned} \tag{4.29}$$

4.4.2 Results

The input images to this algorithm were taken of pipettes filled with fluorescent material as shown in Fig. 4.7a. Similarly to the previous tests, the result of the algorithm was compared to manually picked ground truth tip positions. The test set contained 8 image stacks. The results were expressed only in voxels and the Z dimension was scaled by a factor of 10 to have nearly cubic voxels. The average error was 75.66 ± 30.97 voxels. This error is fairly high which could be due to the following observations. As the example images show in Fig. 4.13, the model fits the data points well but the tip looks shifted along its middle axis. This is due to the low concentration of fluorescent dyes in the tip of the pipette which appear as less bright intensity regions in the image. On the other hand, where the radius of the pipette is greater and more material condenses the image is brighter and attracts the model. Another negative effect is that when the exact distances are not known in the lateral plane and along the Z-axis then the pipette is not interpreted as a (truncated) cone. However, the model does not allow such deformations and fits the data anyway (see Fig. 4.13d), leading to errors compared to the manually selected tip positions. Finally, the 8 DOFs in this case almost make


(a)

(b)

(c)

(d)

Figure 4.13: Example images during the iterations of the 3D fluorescence Pipette Hunter algorithm. The point cloud is the slice view of the image data without the zero elements. The blue truncated cone is the current state of the model. (a) Initial state of the model. (b) Initial state of the model from another angle. (c) Result of the algorithm. (d) Result of the algorithm from an angle that shows that the circular base of the cone differs from the elliptical nature of the data.

the algorithm unstable and it is challenging to find the best parameters for the different cases.

4.5 Discussion

The Pipette Hunter variational framework has been proposed for patch clamp pipette detection. Pipettes have to be often replaced and by such algorithms slight changes in the length and shape can be detected in a robust way which helps in automating the patch clamp process. The base 2D method works on minimum intensity projection of DIC images. The main idea is to fit rectangles on dark image regions that are the sides of the pipette. Running the algorithm on two different projections of the same image stack results in a 3D point which is the tip position in space. The model has two different 3D extensions for different microscopy techniques that accept an image stack as input. The label-free extension models the edge of the pipette as two cylinders which also have a common reference point. The fluorescence 3D extension simplifies to a single truncated cone that tries to cover the inner part of the pipette. However, the radii of their bases are included in the variables thus the number of DOFs remain the same.

Due to the nature of gradient descent which requires many iterations for a reasonable result and the amount of data, 3D cases require a deliberate implementation to keep the runtime low. The equations in all three models are highly parallelizable and can be derived to matrix operations which makes Matlab a good language choice. In an object fitting task the underlying image does not change, thus only the voxel positions have to be determined which are covered by the model and have to be indexed. The image gradient can be precomputed. The code is vectorized the following way. (Vectorization is a Matlab terminology and technique for optimizing loop-based, scalar-oriented code.) First, the possible parameter combinations are computed without the variables (that change). Then in every iteration, the variables are applied to the equations and the resulting voxel positions are stored using linear indexing. This results in a "flat" (vectorized) code where the only loop variable is the iteration number. A stopping criteria is still beneficial to the algorithm. In every 100th iteration it is checked whether the variables changed more than a threshold value. If they did not change or the sign of the change alternates then the algorithm terminates and the result is the last point.

The algorithms are evaluated on real image stacks of pipettes. The detected tip positions are compared to manually determined ground truth positions. The algorithms are also compared to each other or to a baseline algorithm, where relevant. The results include the orientation and tilt angles of the pipettes as well, however, these are not included in the tests or used later in the automated system. The quality of the results is fairly high to reliably hit target cells. We believe our method can be used in other applications, e.g. road intersection detection, neural circuit segmentation.

Part III

Automated Microscopy and Applications

Chapter 5

Algorithms for Automated Patch Clamp Systems

5.1 Introduction

Research of the past decade uncovered the unprecedented cellular heterogeneity of the mammalian brain. It is well accepted now, that the complexity of the rodent and human cortex can be best resolved by classifying individual neurons into subsets by their cellular phenotypes [72, 73, 74]. By characterizing molecular, morphological, connectional, physiological and functional properties several neuronal subtypes have been defined. Revealing cell-type heterogeneity is still incomplete and challenging since classification based on quantitative features requires large amount, often thousands or more, of individual cell samples encompassing a highly heterogeneous cell population. Recording morphological, electrophysiological, and transcriptional properties of neurons requires different techniques combined on the same sample such as patch clamp electrophysiology, post hoc morphological reconstruction or single-cell transcriptomics. The most laborious part of individual neuron characterization is patch clamp recording of biophysical properties of the cell during intracellular current injection.

Recently, patch clamp technique has been automated to a more advanced level [75, 76]. "Blind patch clamping" moves the pipette forward *in vivo* [63, 66, 77] and the brain cells are automatically detected by a resistance increase. The technique soon emerged to image guided systems using multiphoton microscopy on genetically modified rodents [69, 78, 79]. Further improvements include the

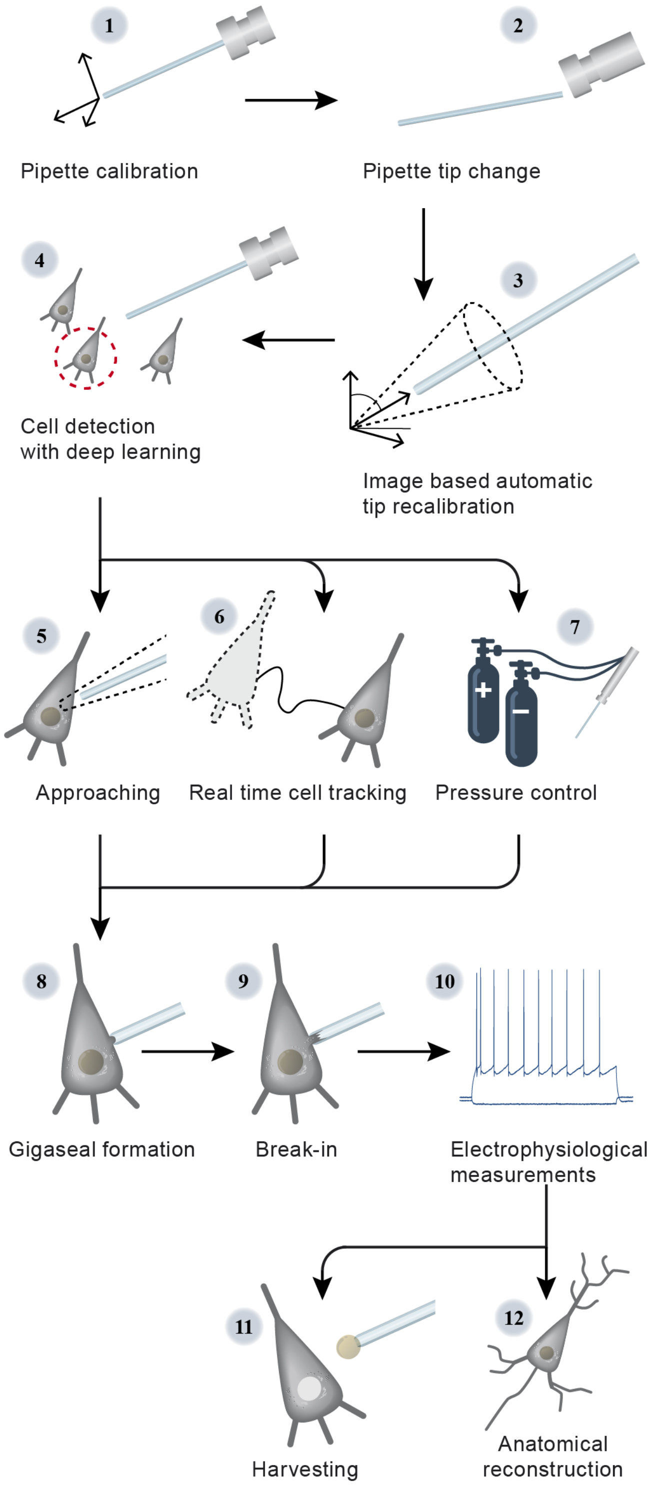


Figure 5.1: Steps of visual autopatching.

integration of behavioral equipments [65], design of an obstacle avoidance algorithm for approaching the target cell [80] or the development of a pipette cleaning method which allows the reuse of the pipettes up to ten times [68, 81]. Automated multi-pipette systems were developed to study synaptic connections [64, 82]. It is also shown that cell morphology can be examined using automated systems [83]. One crucial step for the image guided automation is the pipette localization. Different pipette detection algorithms were compared in [71]. The other crucial step is the automatic detection of the cells which was only performed in two-photon images so far. It is currently not possible to efficiently fluorescently stain human brain tissues. Alternatively, detection of cells in label-free images would open up new application possibilities *in vitro* [70], e.g. experiments on surgically removed human tissues. Most recently, deep learning has been emerging to a level that in the case of well-defined tasks outperforms humans, and often reaches human performance on ill-defined problems like detecting astrocyte cells [8].

To overcome time consuming and expertise-intense neuron cell characterization and collection, we describe a fully automated label-free image guided patch clamping system combining (DIC-like) 3D infrared video microscopy, cell detection using deep convolutional neural networks and glass microelectrode guiding system to approach, attach, break-in and record biophysical properties of target cells.

The steps of the developed visual patch clamp recording process is illustrated in Fig. 5.1. In order to use the system for the first time, a pipette calibration step has to be performed to allow for the pipette to be moved relatively to the camera's plane (1). Thereafter, a position update is made after every pipette change (2) using one of the built-in pipette detection algorithms to overcome the problem caused by length differences of pipettes (3). At this point, the system is ready to select and approach a cell to perform recording on it. We have acquired and annotated a single cell database on brain tissue images, to our knowledge the largest 3D set of this kind. A convolutional deep neural network was trained for cell detection. The system can automatically select a cell for recording (4). When a cell is selected, multiple subsystems are started parallelly that perform the patch clamp process:

i) A subsystem adjusts the electrode and slowly pushes it forward next to the cell. If any obstacle is found on the way, an avoidance algorithm tries to dodge it (5).

ii) A cell tracking system follows the possible shift of the cell (6) and adjusts the pipette's trajectory if necessary.

iii) During the whole process, a pressure regulator system assures that the requested pressure is available (7).

Once the pipette touches the cell (cell-attached configuration) the system performs gigaseal formation (8), breaks in the cell membrane (9) and then automatically starts the electrophysiological measurements (10).

At the end of the measurements, the implemented pipette cleaning method can be performed or the next patch clamp recording can be started from the pipette tip position update step. An event logging system collects information during the patch clamp process, including the target locations and the outcome success, and report files can be generated at the end. The report files are compatible with the Allen Cell Types Database [84].

Our system was tested on rodent and human samples *in vitro*. The electrophysical recording quality strongly correlates to that of an expert biologist. We used the system for harvesting cytoplasm and nuclei from the recorded cells (11) and performed anatomical reconstruction on the samples later (12). Our system is the first that can operate on unstained tissues using deep learning that reaches and even outperforms the cell detection accuracy of human experts, and the number of recordings can be multiplied while preserving high quality measurements.

5.2 Methods

5.2.1 Hardware control and development

The schematic hardware setup of the proposed system is shown in Fig. 5.2. The software system we developed controls each hardware using their drivers on application programming interface (API) level, which makes the system modular and new devices can easily be attached. An Olympus BX61 (Tokyo, Japan) microscope was used for 3D imaging with a 40x objective. For moving the pipette and the microscope stage we used Luigs and Neumann Mini manipulators with SM-5 controllers (Ratingen, Germany). The electrophysiological signals were measured by a HEKA EPC-10 amplifier (HEKA Elektronik, Lambrecht, Germany). The signals were digitized at 100 kHz and Bessel filtered at 10 kHz. To apply different air pressures on the pipette in the different stages of the patch clamping procedure we built a custom pressure controller system. The stages of the procedure were determined by the impedance value of the pipette tip. The electrophysiological signal was transferred to the control software via a USB digitizer board (National Instruments, USB-6009). If the pipette resistance indicated a new stage of the protocol then first the desired air pressure was mixed in the pressure regulator tank and then applied to the pipette. Two analog pressure sensors were used

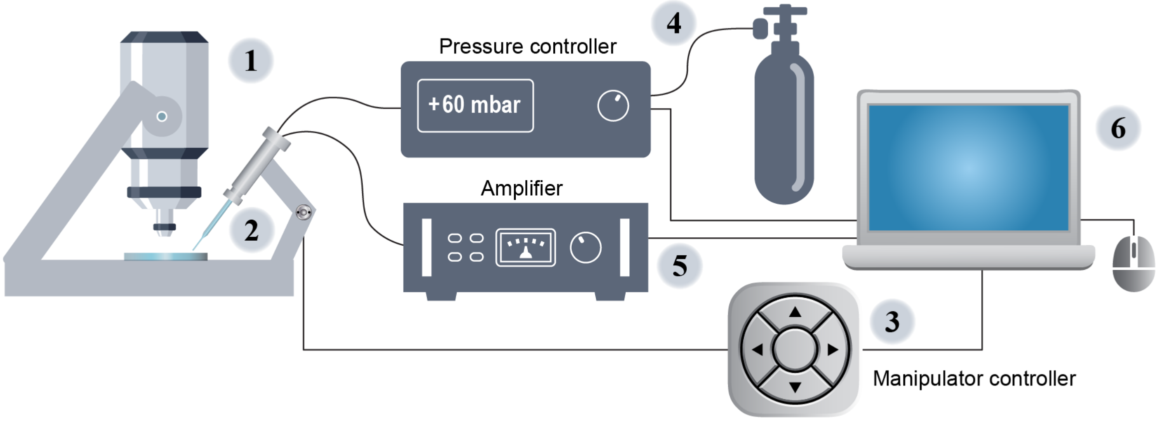


Figure 5.2: Schematic hardware setup. 1: Fixed stage microscope. 2: Micromanipulator. 3: Stage and micromanipulator controller. 4: Pneumatic pressure regulator. 5: Electrophysiological amplifier. 6: Host computer for the software.

for monitoring the actual air pressure: one attached to the tank and another to the pipette. The analog voltage signals of the sensors were measured by the USB digitizer's analog input channels. The solenoid valves of the regulator were controlled with TTL signals of the digital output channels of the USB digitizer board.

5.2.2 Pipette Calibration and Automatic Detection

The pipette calibration is a one-time process which determines the coordinate system transformation between the pipette and the stage axes. The calibration consists of moving the pipette along its axes with known distances, finding it with the camera and detecting the exact tip position in the image. Calibration allows the pipette to be moved to any position in the coordinate system of the microscope's stage. Note that no assumptions are made on the pipette's orientation or tilt angles.

The glass pipettes usually differ in length, thus the tip position should be updated changing them. To automate this step we have developed algorithms for pipette detection in DIC images. First, we use a fast initialization heuristic and then refine the detection. The refinement step is the extension of our previous differential geometry-based method to 3 dimensions [46]. Detailed description of the algorithms with test images and equation derivations can be found in Section 4.3 of the dissertation. The pipette is modeled as two cylinders that have a common reference point and an orientation. The model is updated by the gradient descent method such that it covers dark regions introduced by the pipette in the image. The algorithm has an accuracy of 0.99 ± 0.55 µm compared to manually selected tip positions. This allows reliably hitting the cells, which have a diameter of 10 µm in average, with pipettes oriented towards their centroids.

5.2.3 Pipette Cleaner

We implemented a pipette cleaning method described in [68] to our system. The cleaning procedure requires two cleaning agents: Alconox, a commercially available cleaning detergent, and artificial cerebrospinal fluid (aCSF). We 3D printed a holder for two PCR tubes containing the liquids. The holder can be attached to the microscope objective and should be reachable by the pipette tip. The cleaning is performed by pneumatically soaking up and then blowing out the agents into and from the pipette. The vacuum strength used for the intake of the liquids was -300 mBar and the pressure used for the removal was +1,000 mBar. The method consists of three steps. First, the pipette is moved to the cleaning agent bath and vacuum is applied for 4 seconds. Then, to physically agitate glass-adhered tissue, pressure and vacuum are alternated, each for 1 second and repeated 5 times in total. Finally, pressure is applied for 10 seconds for safety to remove the detergent. In the second step, the pipette is moved to the aCSF bath and any remaining detergent is expelled by applying pressure for 10 seconds. In the third step the pipette is moved back to the experimental preparation where the cleaning was initiated. In the original paper, it is shown that these pressure values and the duration of the different steps are more than enough to cycle the volume of agents necessary to clean the pipette tip. We provide a graphical window in our software to calibrate the pipette positions of the liquid tubes and to start the cleaning process.

5.3 Developed algorithms

5.3.1 Cell Detection System

Comparison of Machine Learning Software Tools

Phenotypic image analysis is the task of recognizing variations in cell properties using microscopic image data. These variations, produced through a complex web of interactions between genes and the environment, may hold the key to uncover important biological phenomena or to understand the response to a drug candidate. Modern high-throughput microscopes can easily produce terabytes of data daily which necessitates computational solutions.

Fully automated high-throughput screening (HTS) assays allow researchers to quickly conduct

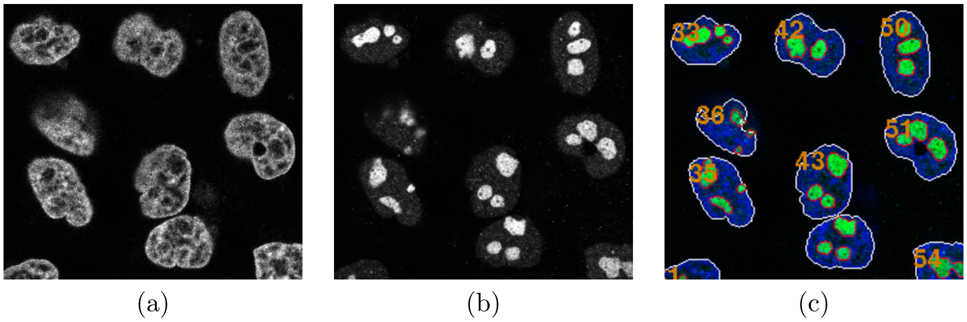


Figure 5.3: Example images of the developed software for nucleoli morphology analysis. Input images of nuclei (a) and nucleoli (b) channels. (c) Segmentation results with contours.

millions of chemical, genetic, and pharmacological tests to rapidly identify active compounds, antibodies, or genes that affect a particular biological process. High-content analysis applies image analysis methods to automate cellular measurements, including the quantification of cellular products such as proteins, or detection of changes in morphology [85]. High-content screening (HCS) is the combination of high-content analysis and HTS [86], offering richer data, higher throughput, and increased flexibility [87]. Today, phenotypic analysis is rarely performed completely by hand.

We have developed a tool to analyze nucleoli morphology and intensity-based properties in HTS assays [11]. A graphical user interface was developed allowing interactive semimanual analysis and data browsing. Multiple fluorescence images can be loaded, each consisting of up to 3 spectral channels. The first channel, containing the DNA staining, was used to identify individual nuclei, while the second channel was used to detect nucleoli based on immunofluorescence. Images were segmented using the Otsu thresholding algorithm [88, 89]. A nucleolus was considered to belong to a nucleus if its segmented area was inside the nuclear area. Objects that were segmented improperly were excluded manually by the user. Example input images and a result segmentation is shown in Fig. 5.3. After the selection process, a comma-separated values (CSV) statistics file containing the intensity and morphological parameters of all the selected nucleoli was generated and used for statistical and phenotypic analysis.

The most common follow-up step of single-cell image segmentaiton in HCS is phenotypic classification. To this end, we have compared 15 publicly available machine learning tools in [12] and a few of them are discussed below. Some of these are appropriate for the segmentation task too, however, this is not a common feature of them. One of the most well known tool is CellProfiler Analyst (CPA) [10, 90]. CPA features a graphical user interface allowing the user to define a set of phenotypes and annotate individual cells accordingly. Annotations created by the user can be used to train a supervised machine-learning algorithm. Following the release of CPA, several other software packages have appeared which also support machine-learning analysis applied to microscopic images of cell-based assays. Phaedra [91] and cellXpress [92] are open-source platforms for visualization and analysis of HCS data. They excel in data visualization, including plate heatmaps, charts, tables, image viewers, and dose-response curves. However, they do not provide a graphical option to directly annotate images for training. HCS-Analyzer [93] works with features extracted by another image processing software such as CellProfiler or Columbus. Phenotype classification is performed either in a supervised (SVM, neural network, k-nearest neighbors, or random forest) or unsupervised manner (k-means, expectation-maximization, or hierarchical clustering). Unsupervised phenotype classification requires the user to provide the number of classes and features that should be considered, and the algorithm groups cells without any labeling. WND-CHARM is a software tool that takes a holistic approach to phenotype analysis. It is designed to remove the need for parameter tuning [94]. It not only relies on single-cell segmentation for analysis but considers the image as a whole. WND-CHARM computes a number of features on the entire image, including polynomial decompositions, high-contrast features, pixel statistics, and texture features. It selects the most informative features according to Fisher's discriminant score [95], which finds features that maximize distance between classes and minimize distance within classes. It then performs classification using weighted-neighbor distances on the selected features. For training, WND-CHARM only requires image-level annotations and therefore does not provide a GUI for annotating.

Some challenges arise in phenotypic image analysis that represent the limitations of current technologies. It is often unknown which phenotypes will be present in an experiment or how they will appear. If a phenotype is rare or the dataset is large, phenotypes are likely to be overlooked. Some methods for data mining and discovery exist that can help ensure completeness of training data. Advanced Cell Classifier [9] provides a tool to help uncover rare phenotypic classes. Many interesting biological processes are continuous by nature. For example, cell differentiation [96], cell development [97], cell adhesion, and cell death. Many other biological processes have important continuous aspects, such as endocytosis or drug uptake. Most existing phenotypic analysis tools rely on machine-learning algorithms designed for classification, even though discrete categorization does not reflect the biological reality. To make use of these tools, researchers are faced with the difficult task of defining artificial cutoffs to discretize inherently continuous processes. CellCognition was one of the first attempts to apply machine learning to recognize patterns in a continuous biological process [98].

Deep networks have been successfully applied to different visual recognition tasks since their ground-breaking performance in the 2012 ImageNet challenge [99] including pose estimation, object tracking, object retrieval, activity recognition, super-resolution, etc. The state-of-the-art for several important tasks used in phenotypic image analysis is now dominated by deep learning approaches, including semantic segmentation, feature extraction, image enhancement, and object recognition. Deep learning is an end-to-end approach to learning. It takes raw data (images in the case of phenotypic profiling) as input and produces the desired output by learning from training examples. This eliminates the need for hand-crafted features, which are cumbersome to design and give suboptimal discriminatory power. In deep learning, a network of artificial neurons organized into layers embeds all the necessary processing. Each layer takes the output of the previous layer as input and applies simple operations that transform the previous representation into a higher, slightly more abstract representation. Although the individual computations are simple, unprecedented discriminatory power is achieved through the compositional nature of the architecture. Modern networks may contain hundreds of layers [100]. In contrast to the traditional approach (image processing, feature extraction, and classification), the end-to-end approach of deep learning is attractive because it offers a holistic solution and often yields better performance. Deep networks have a number of advantages over the traditional paradigm. One advantage is that deep networks can learn to do multiple tasks using the same network. Due to its ability to model highly complex functions, deep learning can tackle problems that are infeasible with traditional approaches. Another important aspect of deep learning is that the features learned are highly general and thus transferable [101], even to unseen tasks and images.

Due to the expansion of deep learning algorithms, we chose this technology to develop a cell detection system and integrate it into our automated microscopy system.

Training a Deep Neural Network

We applied deep learning to establish a system that detects cells in DIC images and makes suggestions from them for automatic patch clamp recording. Object-detection in label-free tissue images is a great challenge [8]. Numerous methods, as well as software [5, 7], are used to segment neurons in cell cultures, but they do not provide satisfactory results when applied to tissues. To obtain a reliable object detector we built a pipeline consisting of three steps: data annotation, model training, and inference (see Fig. 5.4).

Our software provides a labeling tool that offers a platform to generate an applicable annotated

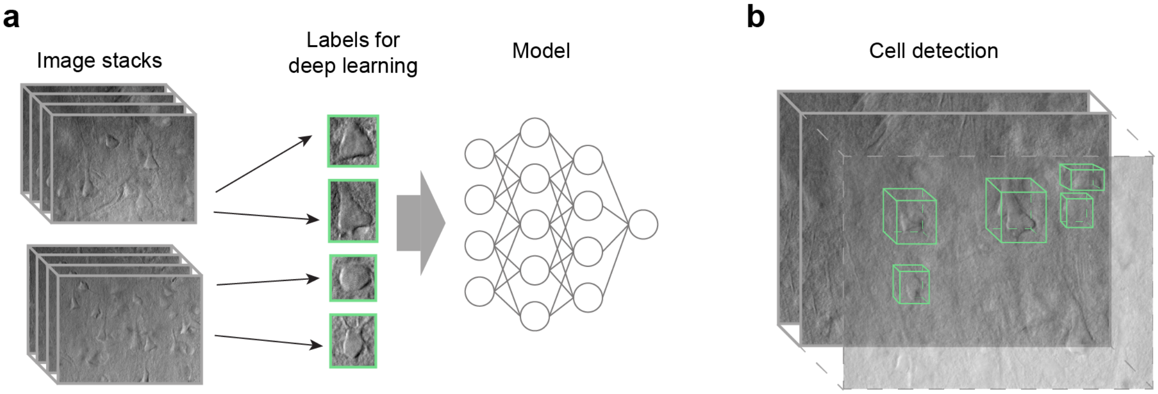


Figure 5.4: Cell detection in label-free tissues using deep learning. (a) Training dataset generation: 265 z stack (60-100 images per stack with 1 µm frame distance) captured from human and rodent neocortical slices with DIC video microscopy. 31,720 objects as healthy cells (green boxes) labeled on every slice of the z stack by 4 experts. (b) After the training session the system detects cells in unstained living neocortical tissues.

dataset. Field experts labeled 6344 cells on 265 image stacks which took about 132 hours. The annotation procedure consisted of putting bounding boxes on the recognized cells over multiple slices in the stack. The acquired labeled training set was converted into the format required by the deep learning framework we used.

The number of deep learning frameworks has increased in the past few years. We chose the *caffe* [102] deep learning framework because of its running time and support of both python and Matlab programming languages. The framework supports adding, removing and modifying the elements of deep learning architectures which was a requirement.

As an extension of *caffe*, Nvidia's Deep Learning GPU Training System (DIGITS [103]) allows non-advanced deep learning users to evaluate the finest trainings at ease. DIGITS also includes a predefined neural network architecture called DetectNet [104]. DetectNet can be split into training and validation sections and these can further be divided into three and two subsections, respectively. The training part contains the data augmentation, the fully connected network (GoogLeNet [105]) and the loss functions, while the bounding box clustering and the mean average precision (mAP) statistics are the elements of the validation part.

DIGITS was used to train DetectNet. The solver used for the training process was Adaptive Moment Estimation (ADAM) [106]. The learning rate was 1e-5 with fixed learning rate policy. The pretrained weights of the ImageNet dataset were used for initializing the GoogLeNet to speed up the training process. The training was performed on a PC with Intel Core i7-4770K 3.5 GHz CPU, 16 GB memory and an Nvidia Titan Xp graphics card. The number of epochs was 2500 which took 6 days and 15 hours.

The detection was performed on image slices from the stack (i.e. 2D images) and the results were extended to 3D where the refined position of the cells was determined in an increased quality. The extension was done by merging overlapping detection bounding boxes over adjacent slices by using their intersection regions. The same applies during the operation of the software. The required overlapping ratio can be set as a parameter. The merging can still happen even if a cell is not detected on every consecutive slice. If one slice does not contain a detection in the position of a cell but the previous and the subsequent does then the slice in question is assumed to contain the object and it will also be included. The merging algorithm compensates for the noise sensitivity of the detection.

Evaluation

To evaluate the performance of the proposed framework we measured precision, recall, and F1-score provided by two different annotators. The object matching between the ground truth data and the detection results was done manually. This resulted in better matching when multiple ground truth bounding boxes were close to each other or even overlapped. A detection was considered to be correct (true positive, TP) when it significantly overlapped with a ground truth bounding box at least on one image slice in the stack. Otherwise, it was treated as a false positive (FP) detection. Ground truth objects not paired with a detection were treated as false negatives (FN). Based on these aspects the detection accuracy was calculated as precision P=TP/(TP+FP), recall R=TP/(TP+FN), and F1-score=2*P*R/(P+R). For the final result we evaluated four different comparisons. First, we determined intra- and inter-accuracies of human experts. The experts were asked to annotate an image stack they already had at an earlier time. The intra-accuracy proved that cell detection is a complex task as the two annotators reached 0.771 (P = 83.12%, R = 71.91%) and 0.778 F1-scores (P = 87.5%, R = 70%), respectively. To compare the experts they were asked to separately annotate one randomly selected image stack. The inter-accuracy measurement resulted in 0.752 F1-score (P = 81.54%, R = 69.74%). For computing all the accuracies, we used the first annotation set of the first annotator as a ground truth. As the final test, we compared the ground truth to the bounding boxes predicted by our method. This outperformed both annotators with 0.835 F1-score (P = 81.72%, R = 85.39%) as shown in Fig. 5.5.

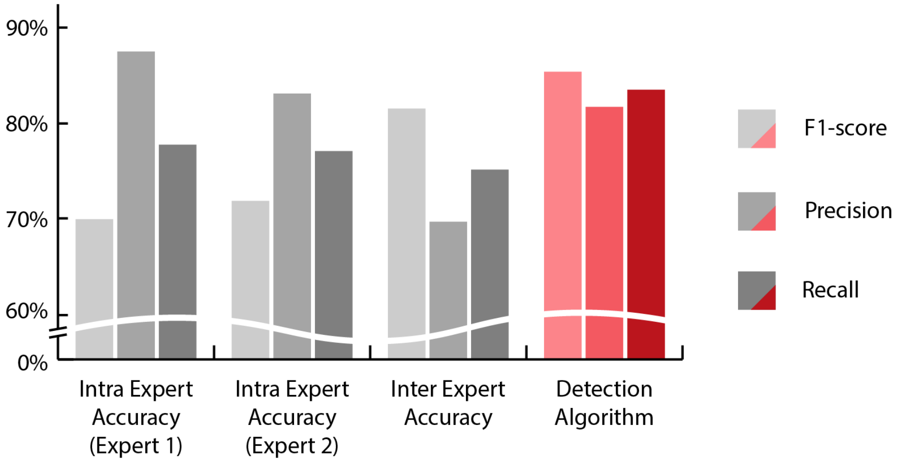


Figure 5.5: Accuracy of the automated cell detection pipeline. Both intra- and inter-expert measurements were performed. The quality of the automated system outperforms the annotators.

Prediction on a server

We developed a tool which we refer to as RemoteWorker to perform cell detection on a server computer. The tool can easily be configured to run any Matlab script on a different computer. We used it for remote cell detection and the code is included in the software and can be configured from a file. The microscope computer operates as a client which sends the image to the server and downloads the detection result, thus GPUs can be shared between microscopes. Below an example shows how to set up both the client and server code to perform the simple task of adding 5 to a function parameter. The server-side script should register the supported commands as follows:

```
s = RemoteWorker.Server();
s.registerCommand('addFive', @(header,data) data+5);
s.run();
```

The client sends a job, polls if the result is ready and then downloads the return value:

```
c = RemoteWorker.Worker();
c.host = '127.0.0.1';
c.sendJob('addFive', 1);
while ~c.isJobReady()
   pause(0.01);
end
result = c.downloadResult(); % returns 6
```

5.3.2 Cell Tracking System

Introduction

The target cell can be shifted during the patch clamping process while the pipette is pushed towards it. The shift can occur in any direction in the tissue and based on our experiments it is usually between 3 and 10 µm (6.98 ± 3.91 µm, n=19). Tracking the cell under the microscope is a challenging task because a 3-dimensional tracking is required with a 2-dimensional label-free modality. The developed online tracking system has two parts which run in separate phases. One part performs lateral tracking in the XY plane while the other tracks the cell in the Z dimension. Both parts require a template image of the target cell which is acquired before starting the patch clamp process and when the cell is in focus in the image. The lateral tracking is always performed in the image corresponding to the latest focus level. The Z tracking algorithm operates on a small image stack. Acquiring an image stack is time-consuming as the objective has to be moved physically to different focus levels. Therefore, when Z tracking is being performed the pipette is not moved to ensure that the cell is not pushed away meanwhile.

Lateral Cell Tracking

The Kanade-Lucas-Tomasi (KLT) feature tracking algorithm was used as a lateral cell tracker. Feature points are detected by the minimum eigenvalue algorithm in a window in the template image [1,2]. The algorithm is computationally inexpensive and it is performed often (usually per second) except when Z tracking is in progress. Our experimental verification shows that in 95%

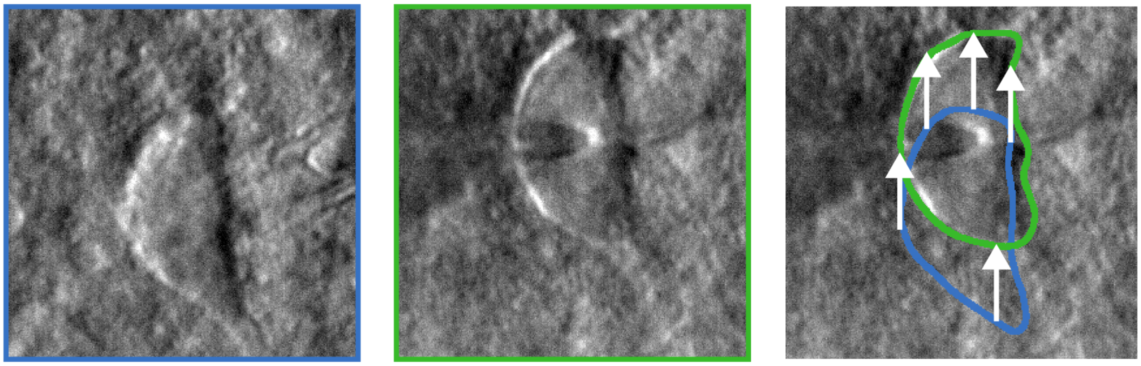


Figure 5.6: Lateral tracking of the cell movement. DIC images of the targeted (blue box) and patch clamped cell (green box). The cell drifted from its initial location (arrows in the right panel) during the pipette maneuver.

of the cases the algorithm was able to track single cells for more than 100 frames, even in case of extreme motion (greater than 3 µm/frame). The default window size for the tracker was 241x241 pixels which is enough to cover almost every cell (except some rare large human cells). The feature points are detected using a 5x5 filter. If the number of valid feature points significantly drops, for example, due to a sudden movement of the cell in the Z direction, the points are reinitialized. A drop of feature points is considered significant when the number of valid points is less than 10% of their original number or less than or equal to 10. Fig. 5.6 shows an image of a cell before and after patch clamping, and the result of the lateral tracking algorithm.

Tracking in the Z Dimension

Tracking in the Z dimension is based on focus detection algorithms. As a major principle, we assume that an object which is in focus should have sharp edges, and this results in large pixel intensity differences in the image. In the first step of the algorithm, a small image stack is acquired around the last known position of the target cell. The template image is compared to every slice of the stack by calculating the absolute standard deviation difference in a window of the images. The position of the window is determined by the lateral tracker. Then the maximum of the values corresponding to the images below the middle element is selected. A value for the images above the middle element

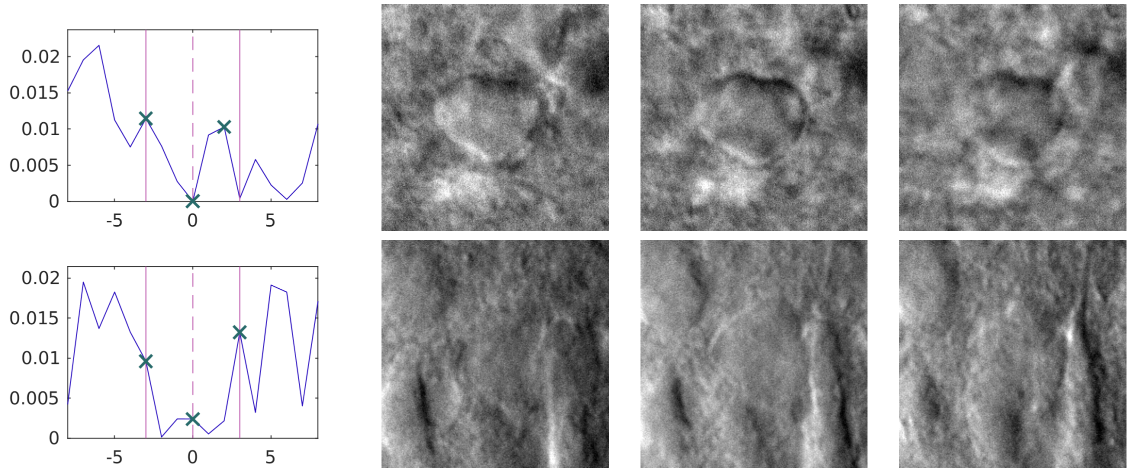


Figure 5.7: Examples of the Z tracking algorithm. The plots in the first column show the calculated values used for determining the direction of the shift. The X-axis is the index difference from the middle slice. The x markers on the plot show the locations of the images in the next three columns. The final decision, whether the cell has shifted, is based on the values in these locations too. The values outside the red lines are not used in the calculations and only show the robustness of the algorithm.

is selected similarly. The middle slice corresponds to the last known focus position. Finally, the minimum of these three values, the two maximums and the one of the middle slice, determines the direction of the shift of the cell. Fig. 5.7 shows examples of the computed values and the related images which were used to make a decision. If the lowest value corresponds to the middle slice, the algorithm predicts that the cell has not moved. If the cell has moved downwards relative to its previous position then the minimum value will correspond to one of the lower elements. Similarly, we can determine if the cell has moved upwards. Although the approach does not determine the value of displacement but the direction of it the tracking is reliable because the cells do not move rapidly. The stack size used for the tracking is 7 slices by default, which is set based on empirical tests, and thus the middle element is the 4th. The data shown in Fig. 5.7 are calculated from bigger image stacks, consisting of 60-100 slices; this allowed for showing more than just 7 values in the figure and demonstrate the reliability of the algorithm.

5.3.3 Automated Microscopy System

Automated Patch Clamping Steps

After pipette calibration and cell detection the patch clamping procedure can be started. First, the software calculates the trajectory of the pipette movement on which the manipulator moves the pipette tip close to the cell and applies medium air pressure to the pipette. The initial trajectory is a straight line on the manipulators X-axis thus the distortion made in the tissue is minimal. Note that the X-axis of the manipulator is tilted against the plane of the stage. The pipette is first moved a few micrometers above the cell and then pushed on it, which makes the approaching process more reliable. The pipette is moved closer to the target in small steps per second. The impedance of the tip is continuously monitored during the movement. An early increase in resistance denotes the presence of an obstacle in front of the pipette, e.g. a blood vessel or another cell. If an obstacle is hit, an avoidance algorithm takes over control: the pipette is pulled back, slightly moved laterally and when the obstacle is passed the pipette is oriented back to the initial trajectory towards the target [80]. If multiple obstacles are found, the pipette is moved laterally in a spiral pattern after each hit. Meanwhile, the described 3D tracking algorithm compensates for the possible displacement of the target cell in the trajectory. When the pipette tip reaches the target position above the cell the pressure is decreased to a low positive value. Then the pipette is lowered in Z direction and the resistance of the tip is monitored by injecting -10 pA current for 5 ms periodically. If the impedance increases more than a predefined value (0.7-1.2 MΩ) the cell membrane penetration

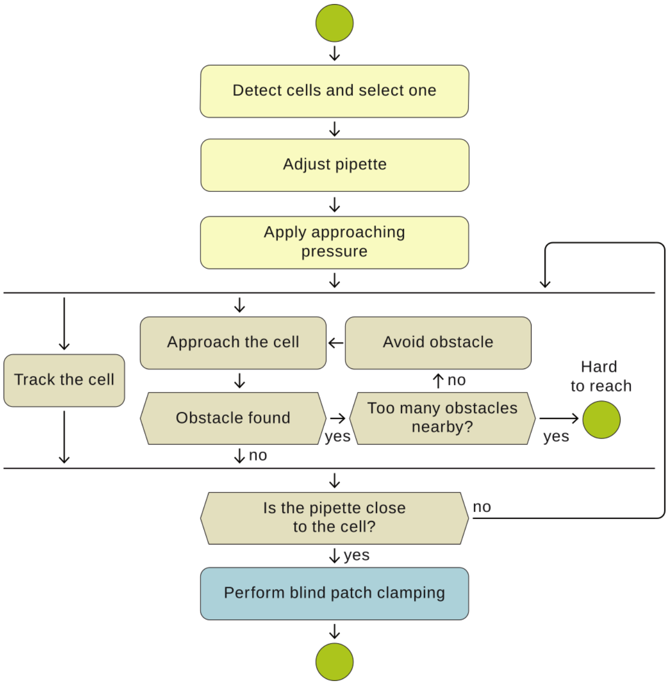

Detect cells and select one
Adjust pipette
Apply approaching pressure
Track the cell
Approach the cell
Avoid obstacle
no
Hard to reach
Obstacle found
yes
Too many obstacles nearby?
yes
no
Is the pipette close to the cell?
no
yes
Perform blind patch clamping

Figure 5.8: Flowchart of the system performing visual patch clamping.

protocol is initiated. The cell-attached configuration is set up by the immediate cease of pressure. Small negative pressure is applied to acquire tight sealing of the cell membrane into the glass. If seal resistance reaches 1 GΩ ("gigaseal") on -60 mV holding potential then vacuum pulses of increasing length are applied to break-in the membrane and reach the whole-cell configuration. The flowchart of the process is illustrated in Fig. 5.8. A representative patch clamp process is demonstrated in Fig. 5.9 including the detected cells in an image stack, the trajectory of the pipette, and the measured pressure and resistance values. Information about the process is always up to date in the GUI windows. Every parameter that has effect on the process can be set by the user.

Software

The control software is written in MATLAB and the source code is publicly available. The visual patch clamping process can be started from a user-friendly GUI which allows every parameter to be set and the process can be monitored in real-time by the operator. A screenshot of the two main windows are shown in Fig. 5.10. Throughout the session the Patch Clamp Diary module collects information about recording attempts, including their location and outcome status. Additionally, the user can include positions of the sample side and the sample holders. The information collected by

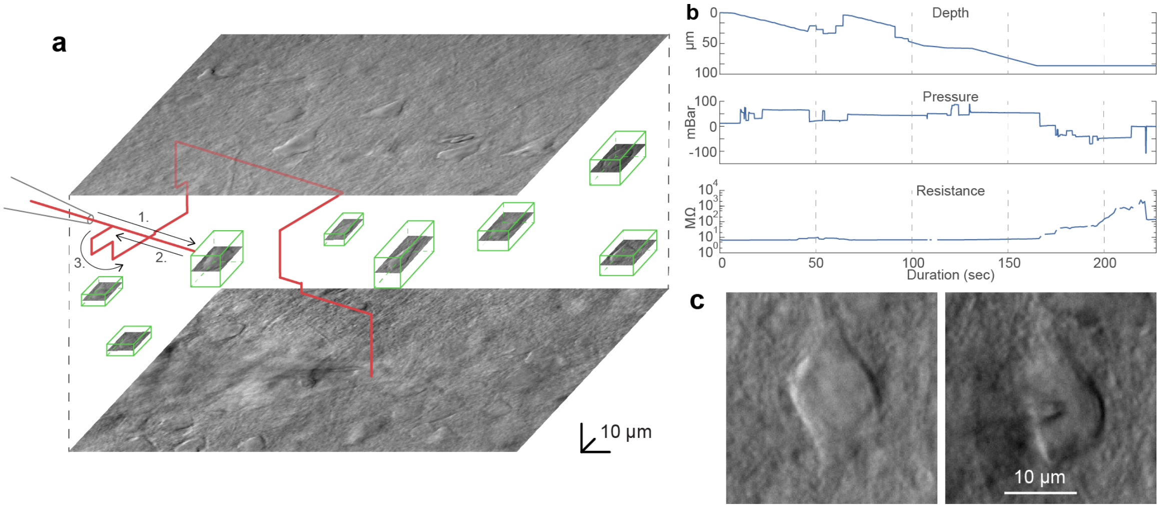
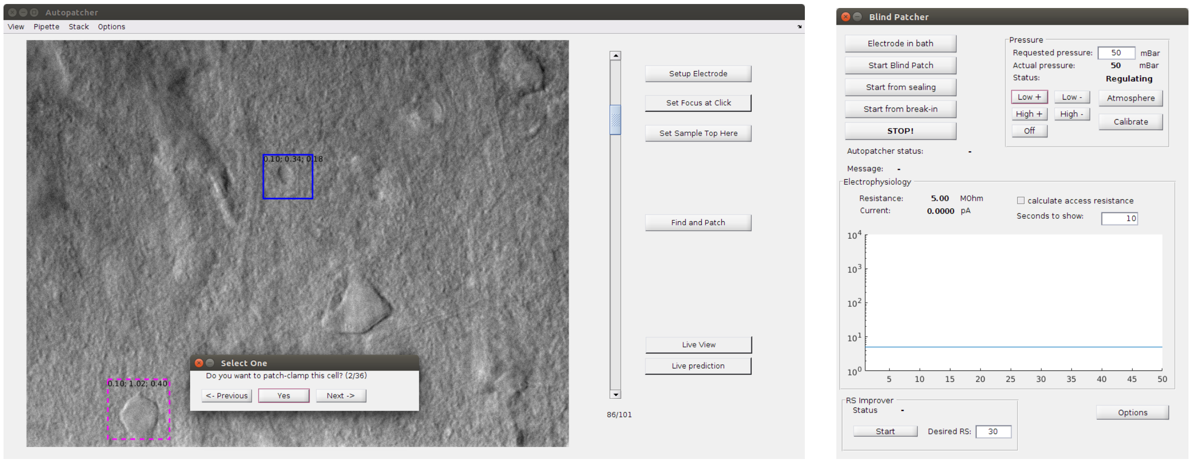


Figure 5.9: Representative example of a whole-cell recording. (a) Trajectory of the pipette tip in the tissue and the distribution of the detected cells. (b) Plots of the depth of the pipette tip in the tissue, the applied air pressure, and the measured series resistance during the approach. (c) Image of a cell before and after performing patch clamp recording on it.

the diary module can be visualized and saved into a comma-separated values (CSV) file. Many utility features are present to help everyday experimenting. Single images or image stacks can be acquired, saved or loaded later from the menu bar. The acquired images can be processed by performing background illumination correction or DIC image reconstruction which can help in identifying cells



Do you want to patch-clamp this cell? (2/36)

Requested pressure: 50 mBar
Actual pressure: 50 mBar
Status: Regulating

Autopatcher status: -

Message: -

Resistance: 5.00 MOhm
Current: 0.0000 pA

□ calculate access resistance
Seconds to show: 10

Status -
Desired RS: 30

(a)

(b)

Figure 5.10: Screenshots of the two main windows of the software. (a) Main window with live camera image. (b) Electrophysiology and pressure window.

and their features. The reconstruction algorithm is the GPU extension of our previous algorithm [47] which works about 1000x faster now. The built-in labeling tool allows image database generation which can be further used to train deep networks. We have implemented the recent automation techniques for *in vivo* patch clamping, including pipette cleaning or hit reproducibility check. The XML configuration file makes the adaptation easy between different setups and the software can also operate as a general microscope controller. The multithreading limitations in MATLAB are overcome by timers, events, polling, and client-server design. A logging system is used for maintainability purposes.

5.4 Biological Evaluation

5.4.1 In vitro preparation of human and rat brain slices

All procedures were performed according to the Declaration of Helsinki with the approval of the University of Szeged Ethics Committee. Human slices were derived from material that had to be removed to gain access for the surgical treatment of deep-brain tumors from the left and right frontal, temporal, and parietal regions with written informed consent of female (n = 10, aged 49.7 ± 25.5 years) and male (n = 7, aged 49.7 ± 11.7 years) patients prior to surgery. Anesthesia was induced with intravenous midazolam and fentanyl (0.03 mg/kg, 1-2 µg/kg, respectively). A bolus dose of propofol (1-2 mg/kg) was administered intravenously. To facilitate endotracheal intubation, the patient received 0.5 mg/kg rocuronium. After 120 s the trachea was intubated and the patient was ventilated with a mixture of O2 and N2O at a ratio of 1:2. Anaesthesia was maintained with sevoflurane at monitored anaesthesia care (MAC) volume of 1.2-1.5. After surgical removal the tissue blocks were immediately immersed in ice-cold solution containing (in mM) 130 NaCl, 3.5 KCl, 1 NaH2PO4, 24 NaHCO3, 1 CaCl2, 3 MgSO4, 10 d(+)-glucose, saturated with 95% O2 and 5% CO2. Slices were cut perpendicular to cortical layers at a thickness of 350 µm with a vibrating blade microtome (Microm HM 650 V) and were incubated at room temperature for 1 hour in the same solution. The aCSF used during recordings was similar to the slicing solution but it contained 3 mM CaCl2 and 1.5 mM MgSO4.

Coronal slices (also 350 µm thick) were prepared from the somatosensory cortex of male Wistar rats (P18-25, n = 23, RRID: RGD_2312511) [107]. Recordings were performed at 36°C. Micropipettes (3.5-5 MΩ) were filled with low [Cl]i intracellular solution for whole-cell patch clamp recording (in mM): 126 K-gluconate, 4 KCl, 4 ATP-Mg, 0.3 GTP-Na2, 10 HEPES, 10 phosphocreatine, and 8 biocytin (pH 7.20; 300 mOsm).

5.4.2 Anatomical processing and reconstruction of recorded cells

Following electrophysiological recordings, slices were transferred into a fixative solution containing 4% paraformaldehyde, 15% (v/v) saturated picric acid, and 1.25% glutaraldehyde in 0.1 M phosphate buffer (PB; pH = 7.4) at 4°C for at least 12 h. After several washes with 0.1 M PB, slices were frozen in liquid nitrogen, thawed in 0.1 M PB, embedded in 10% gelatin, and further sectioned into 60 µm slices. Sections were incubated in a solution of conjugated avidin-biotin horseradish peroxidase (ABC; 1:100; Vector Labs) in Tris-buffered saline (TBS, pH = 7.4) at 4°C overnight. The enzyme reaction was revealed by 3' 3-diaminobenzidine tetra-hydrochloride (0.05%) as chromogen and 0.01% H2O2 as oxidant. Sections were postfixed with 1% OsO4 in 0.1 M PB. After several washes in distilled water, sections were stained in 1% uranyl acetate and dehydrated in an ascending series of ethanol. Sections were infiltrated with epoxy resin (Durcupan) overnight and embedded on glass slides. Three-dimensional light-microscopic reconstructions were carried out using a Neurolucida system (MicroBrightField) with a 100x objective.

5.4.3 Application of visual guided autopatcher in brain slices

To test the performance and effectiveness of our system we obtained a series of recordings on slice preparation of rat somatosensory and visual cortices (n=23 animals) and human temporal and association cortices (n=17 patients). Successful automatic whole-cell patch clamp trials without experimenter assistance were achieved in a total number of n=100 and n=74 (rodent and human

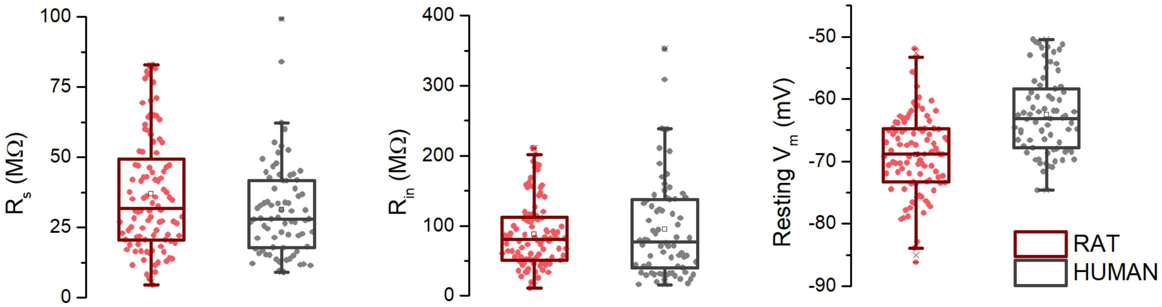


Figure 5.11: Main electrophysiological features of the patched cells. The series resistance (R_s) of all successful measurement (right), the input resistance of the cells (middle) and the resting membrane potential of the cells patched with lower than 30 MΩ R_s value (right).

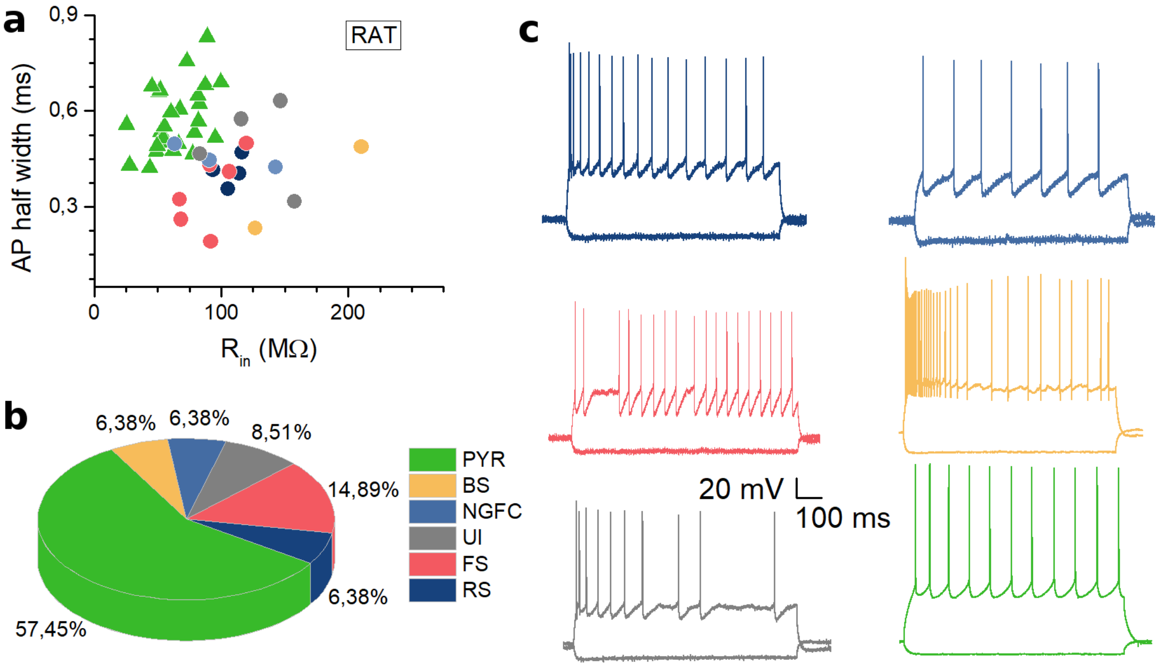


Figure 5.12: Identified cell types of the autopatched cells in rodent samples. (a) Plot of two features, action potential half-width, and input resistance. (b) The proportion of recorded cell types in the dataset. (c) Representative firing patterns of the cell types.

cortex, respectively) out of n=157 and n=198 attempts. Recording quality was supervised on the recorded current traces by measuring series resistance (Rs) with step current injection (Fig. 5.11). We found a wide range of Rs values within successful attempts in both species: 34.52 ± 18.99 MΩ in rat and 31.39 ± 16.67 MΩ in human recordings. Trials with Rs value exceeding 100 MΩ were discarded. Access resistance in 48.28% of our recordings was under 30 MΩ which we denoted as high quality and used for further analysis. We applied a standard stimulation protocol and recorded membrane voltage responses to injected currents. Based on the extracted common physiological features and firing patterns we grouped neurons based on the criteria established by the Petilla convention [108]. We identified six types of neurons in automatically patch clamped (autopatched) rat samples, which is shown in Fig. 5.12: pyramidal cell (pyr, green triangles), fast-spiking basket cell (fs, red dots), neurogliaform cell (ngfc, light blue dots), burst firing interneuron (bs, orange dots), regular spiking interneuron (rs, dark blue dots) and those that could not be categorized (ui, grey dots). Cell types were identified according to electrophysiological features. In recordings done in human tissues we identified four neuron types in which 93% of all the autopatched neurons were grouped: pyramidal cell (pyr), fast-spiking basket cell (fs), neurogliaform cell (ngfc) and the stuttering rosehip cell [109],

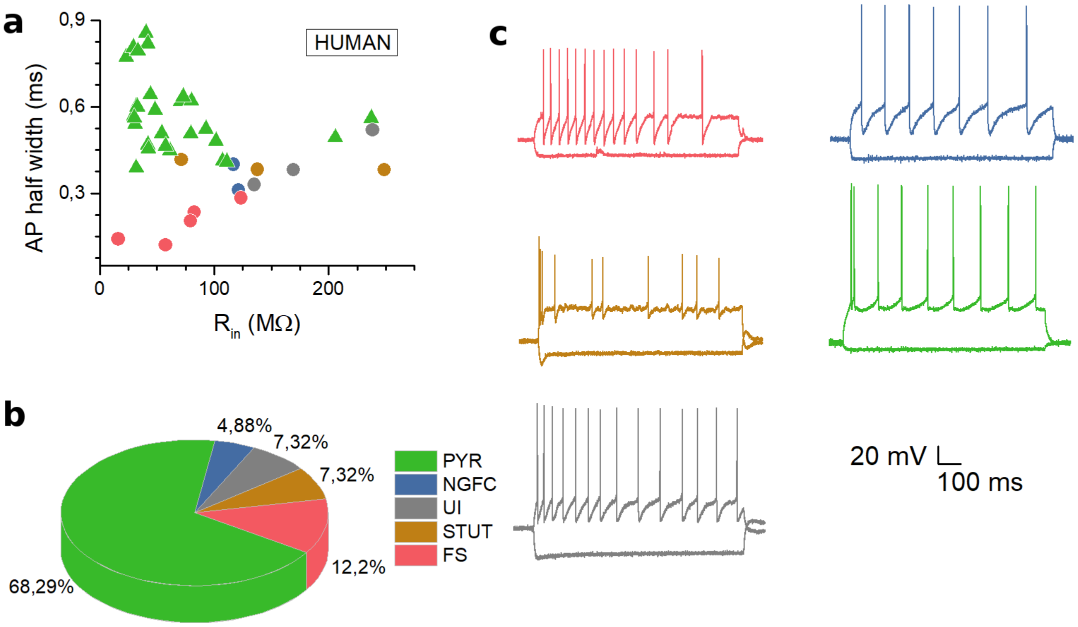


Figure 5.13: Identified cell types of the autopatched cells in human samples. (a) Distribution of phenotypes based on action potential half width and input resistance. (b) The proportion of recorded cell types in the dataset. (c) Representative firing patterns of the cell types.

as illustrated in Fig. 5.13 similarly to rodent data .

Each electrophysiological recording was performed using biocytin based intracellular solution for further 3,3'-Diaminobenzidine (DAB) staining and anatomical investigation. Among autopatched neurons with <30 MΩ access resistance 80% and 60.71% full recovery was achieved from rat and human samples, respectively. Two selected neurons were reconstructed in 3D with Neurolucida system and are shown in Fig. 5.14.

We next tested if single-cell RNA sequencing is achievable from the collected nuclei of autopatched neurons. After performing automated whole-cell clamp and injecting test pulses for 15-20 minutes the nuclei of recorded cells have been harvested into the recording pipettes. Immediately after this, the pipettes were removed and the nuclei were extracted from the pipettes into tubes for transcriptomic profiling. From six extracted nuclei from the human temporal cortex (four interneurons, two pyramidal) we detected 6317-12956 genes and used h-score method to evaluate mapping quality (0.38-1).

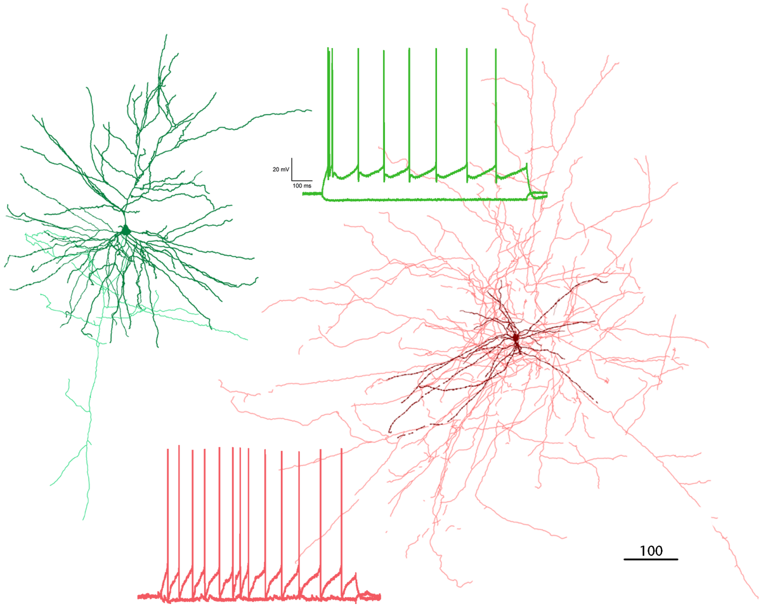


Figure 5.14: Two anatomically reconstructed autopatched neurons. The darker colors represent the somata and dendrites of the pyramidal (green) and the interneuron (red) cells. The brighter color shows the axonal arborization. The firing patterns of the cells are color-coded as the reconstructions.

5.5 Discussion and Future Plans

The developed system is able to fully automatically perform patch clamp recording on single neurons in rodent and human brain cortex. This is a step forward towards characterizing and understanding the phenotypic heterogeneity and cellular diversity of the brain. A central part of the method was the detection of single neurons in label-free 3D images using deep convolutional neural networks reaching super-human precision. The system we developed is fully controlled by a single software, including all hardware components, data handling, and visualization. For the patch clamping procedure it uses single-cell approaching strategies, such as micrometer precision calibration, registration and navigation of the pipette tip, elastic real-time 3D tracking of the target cell, bypassing obstacles e.g. cells or other tissue components while the tip approaches its target, and an advanced pressure controller system that substitutes the classical manual pressure protocols throughout the process. The control software has its internal very comprehensive logging system that allows the tracking of approaching phase and harvesting conditions to a level of details that might only be necessary in the future. In addition, it can connect to and save database entries compatible with the Allen Brain Atlas single neuron database.

In this work, we demonstrated the power of this system that was capable of measuring a large set of rodent and human neurons in the brain cortex. The results show strong correlation to the earlier manually measured literature results in terms of quality and phenotypic composition of the cell type heterogeneity. Records of measured cells were inserted to the Allen Brain Atlas database and a subset of them was isolated from its tissue environment and molecularly characterized and then single-cell RNA sequencing confirmed unique transcriptomal properties of cell types. Furthermore, we successfully demonstrated that autopatched neurons can be anatomically reconstructed.

The main advantage of the proposed system is its easy portability to any existing setups. Although we do not believe that it will fully substitute human experts, it is a great choice for specific complex tasks, allows parallelization and speeds up discovery. It is important to emphasize the need for a standardized and fully documented patch clamping procedure, which is guaranteed by using the developed system. The choice of advanced image analysis and deep learning techniques made it possible to work with the least harmful imaging modalities at a human expert level of single-cell detection that was impossible so far. Further possibilities are more widespread and potentially enable or accelerate discoveries. Combining cell detection with intelligent single-cell selection strategies, the proposed system can be the ultimate tool to reveal and describe cellular heterogeneity. Combining multiple pipettes could be used to describe the connectome at cellular level. Here, we presented the system's application for brain research, but other fields, such as cardiovascular or organoid research will also benefit from it. Based on its nearly complete automation, it can help in education.

The system currently selects cells based on detection probability, but lacks an advanced sampling strategy that would choose cells based on more relevant aspects, such as spatial distribution, phenotypic heterogeneity or morphology. Currently, it is computationally infeasible to migrate the proposed semi-3D deep learning single-cell detection to 3D in real-time. In the future, we aim to extend our software to advance user experience, such as better visual experience, more advanced whole tissue imaging and navigation and easier documentation steps. Our ultimate aim is to sample the scanned space as efficiently as possible and resolve morphological phenotypic distribution. The system will then allow connecting spatial, morphological and molecular information, and – by using multiple pipettes – the electrophysiological connections between pairs, triplets or higher number of cells at a time.

Chapter 6

Conclusion

This thesis presents our research in microscopic image analysis applied for brain research. The topics are connected by the special contrast-based microscopy (DIC) used for the experiments which allows the examination of living cells in cultures and even in thin tissue slices. The main objective of this work was to build an automated microscope system controlled by a software that performs electrophysiological measurements on neurons in brain tissues.

The first part of the dissertation is an analysis of the image formation model of DIC microscopy. We have described a variational framework capable of converting the images to a quantitative type. Three different datasets were designed and prepared to compare the algorithm to numerous others. The results show that the proposed one gives the best reconstructions based on ground truth data. The reconstructed images can be considered as phase- or height-maps. We presented a calibration algorithm for such conversions to interpret the reconstructed DIC images quantitatively.

The second part of the document describes a novel object fitting algorithm used for precisely determining the position of patch clamp pipettes. These glass pipettes are used for approaching the target cell with an electrode for measurement. The fitting method is based on a mathematical shape model of the pipette. The model is iteratively updated using an energy minimization framework to cover the region of the pipette in the image. The algorithm is defined for both 2D and 3D images, as well as for DIC and fluorescence imaging modalities. The variations of the method are compared to each other and to baseline algorithms, and tested on ground truth data. We are not aware of other algorithms for this purpose that determine the spatial position of the pipette tip.

The third part of the dissertation presents our automated patch clamp microscopy system. All the hardware components are controlled by a single software using image-guided and deep learning algorithms which resulted in a fully automated electrophysiological measurement system. We have generated an image database and trained a deep convolutional neural network to detect the neurons in label-free microscopy images of brain tissues. A cell tracking system determines the spatial displacement of the target cell while approaching it with the patch clamp pipette. The software forms gigaseal connection, breaks the cell membrane, and then performs electrophysiological measurements. Afterward, the pipette can be cleaned or replaced and the tip position can be refined to reliably hit the next target. The whole process is recorded by the built-in laboratory notebook module. Our system is the first one that works on label-free imaging modality and allows the analysis of human samples. The software was used for electrophysiological measurements of hundreds of rodent and human neurons. We have shown that further biological experiments can be performed on the cells measured by the automated system.

Although label-free microscopy is the first one which was used for biological experiments, its full potential is still unexplored. This dissertation presents algorithms for DIC microscopy images and covers multiple biological applications. Our future plans include the extension of the reconstruction algorithm to phase-contrast microscopy or other modalities. Phase contrast is also widely used and the interpretation of the images is slightly more straightforward. Modern fluorescence microscopes are often provided with digital phase contrast modality which justifies the need for such algorithms.

The pipette detection algorithms, despite the vectorized code, could be further sped up with GPU implementation. The shape fitting idea that variables of a model are optimized in a variational framework can be extended to other applications as well, e.g. road intersection detection.

The main application of the developed algorithms, the image-guided automated patch clamp system can be extended to a multi-patcher system. Such a system would allow the examination of the connectome at cellular level and help in understanding how the brain works. We plan to build a database of the autopatched cells that collects visual, electrophysiological and molecular information. The collected data could be used to develop a model that predicts the phenotype of cells based only on the visual information.

Appendix A

Summary in English

Image analysis algorithms and tools have pushed the frontiers of scientific computing in the last few decades. One of the fields where the modern computer has made a remarkable impact is the area of biological imaging. Advanced microscopes can now easily collect thousands of high-resolution images of cells daily. It was soon realized that automation of the process is beneficial for extracting the full spectrum of information present in these images.

Brightfield microscopy is a simple optical microscopy technique where visible light transmitted through the sample and a system of lenses magnifies images of small objects. Contrast enhancing techniques, for example, differential interference contrast (DIC) microscopy, are able to show features that are invisible otherwise. Such techniques allow the study of the internal structure of living cells.

The scientific recognition that electrical phenomena are part of animal physiology is well known by now. The electrophysiology of brain cells is investigated by the patch clamp technique. This helps in understanding the complex structure of the brain and discover new cellular phenotypes. The technique is very laborious and requires expertise, but can be automated. The dissertation details the algorithms we have developed for a software system that fully automatically performs patch clamp recording on brain tissue slices. The results are grouped into three major thesis points which are summarized below. The relation between these thesis points and their supporting publications is shown in Table A.1.

Part	[46]	[47]	[55]	[71]	[110]	[11]	[12]
I.	●	●	●				
II.				●	●		
III.					●	●	●

Table A.1: Thesis contributions and supporting publications

DIC Image Processing

DIC images – and label-free images in general – are easy to understand for the human eye but conventional algorithms are not applicable when it comes to image processing. We have developed algorithms and tools to make DIC image analysis easier and more reliable.

We have developed a differential geometry based reconstruction algorithm which transforms the images to a format similar to fluorescence ones for easier analysis. The method incorporates the point spread function of the microscope which leads to better reconstructions. Mathematically this is described by a double integral in the energy function and it is challenging to calculate its Euler-Lagrangian functions. The state of the art quality of the algorithm is demonstrated on three different datasets which we generated. One dataset consists of simulated images. The other two are real images of miniaturized objects with fluorescence ground truth and cell images with hand-segmentations.

The transformed images directly contain morphological information of the sample but their extension in the Z dimension is unclear. We have proposed an algorithm for phase and height measurement using DIC microscopy. This is a calibration step performed by taking images of microbeads with a known refractive index and calculating a function that maps the pixel intensities to physical properties. The technique provides a simple and cost-effective solution to phase measurement without the need of additional microscope parts, and it performs reliably using just a single image.

Patch Clamp Pipette Detection Models

Segmentation of objects with well-defined geometries is a fundamental problem in image analysis. We presented a method that extends a model to detect elongated straight object pairs that have a common reference point. We use it to segment pipette tips under a microscope to automatically navigate these tips with micrometer precision for patch clamping and measuring properties of nerve cells.

Three variants of the proposed Pipette Hunter tip detection is presented which are all based on a variational framework. The methods work on DIC, oblique, and fluorescence microscopy image stacks that contain optically sliced images of a pipette. The inputs are either different projections of the image stack or the stack itself, depending on the number of dimensions supported by the algorithm. The 2D method fits two rectangles, while the 3D method in case of label-free imaging fits two cylinders. In both cases, the objects being fit have a common reference point. However, for fluorescence images the 3D algorithm fits a truncated cone.

The quality of the algorithms is measured by comparing the results to hand-picked ground truth tip positions. The accuracies of the detections are high enough to reliably hit target cells. The algorithms are implemented in our patch clamp system.

Automated Microscopy and Applications

Recently, the automation of patch clamp technique has emerged to image guided systems. We have developed the first fully automated label-free patch clamp system enabling the study of human samples. The system is controlled by a single software, including all hardware components, data handling, and visualization. For the patch clamping procedure it uses single cell approaching strategies, such as micrometer precision calibration, registration, and navigation of the pipette tip, elastic real-time 3D tracking of the target cell, bypassing obstacles e.g. cells or other tissue components while the tip approaches its target at a few micrometer precision, and an advanced pressure controller system that substitutes the classical manual pressure protocols throughout the process. A central part of the method was the detection of single neurons in label-free 3D images by generating an image database and training a deep convolutional neural network.

We demonstrated the power of this system and its capability of measuring a large set of rodent and human neurons in the brain cortex. The measured records are compatible with and were inserted into the database of the Allen Institute for Brain Science. The nuclei or the cytoplasm of some measured cells were extracted after the recording and characterized molecularly. Single-cell RNA sequencing confirmed unique transcriptomal properties of cell types. The system is easily portable to any existing microscope setups. Although it does not completely substitute human experts, it speeds up the recording process, shortens the learning curve of electrophysiologists and one operator can control multiple devices at a time. Our future plans include the extension to multiple pipettes to study the connection between pairs, triplets or a higher number of neurons at a time.

Appendix B

Magyar nyelvű összefoglaló

A képfeldolgozó algoritmusok és eszközök az utóbbi évtizedekben a tudományos számítás határait feszegették. A biológiai képalkotás egy azon területek közül, melyekre a modern számítógép jelentős hatással volt. A fejlett mikroszkópok ma már napi szinten könnyen képek ezreit állítják elő magas felbontásban. Hamar világossá vált, hogy az automatizálás hasznos lehet a képekből származó információ teljes spektrumának kinyeréséhez.

A fénymikroszkópia egy egyszerű optikai mikroszkópos technika, mely esetben a látható fény áthalad a vizsgált mintán és egy lencserendszer felnagyítja a kis objektumról alkotott képeket. A kontraszt növelő technikák, például a differenciális interferencia kontraszt (DIC) mikroszkópia, képesek a minta olyan jellemzőit is megmutatni, amik másképp rejtve maradnak. Ezek a technikák lehetővé teszik az élő sejtek belső struktúrájának vizsgálatát is.

A tudományos felfedezés, mely szerint az elektromosság része az élőlények fiziológiájának, mára már jól ismert tény. Az agysejtek elektrofiziológiájának vizsgálatára a patch clamp (folt-feszültségzár) technika használható. Ez segítséget nyújt az agy komplex struktúrájának megértéséhez és új sejttípusok felfedezéséhez. A módszer alkalmazása fárasztó folyamat és nagy szakértelmet igényel, de lehetséges az automatizálása. A disszertációm az általam fejlesztett algoritmusokat és egy szoftverrendszert tárgyal, mely képes a patch clamp folyamatot teljesen automatikusan elvégezni agyszeleteken. Az eredményeket három fő tézispontba csoportosítottam, melyek összefoglalója alább található. A tézispontjaim és publikációim kapcsolatait B.1 táblázat mutatja.

Rész	[46]	[47]	[55]	[71]	[110]	[11]	[12]
I.	•	•	•				
II.				•	•		
III.					•	•	•

B.1. táblázat. Tézispontok és publikációk kapcsolata.

DIC Képfeldolgozás

A DIC képek – és általánosságban a fluoreszcens jelölés mentes képek – az emberi szem számára könnyen értelmezhetőek, de a klasszikus képfeldolgozó algoritmusok nem alkalmazhatóak rájuk. Az általunk fejlesztett algoritmusok és eszközök megkönnyítik és megbízhatóbbá teszik a DIC képfeldolgozást.

Kifejlesztettünk egy differenciális geometrián alapuló rekonstrukciós algoritmust mely a bemeneti képeket fluoreszcens jellegűre transzformálja az elemzésük megkönnyítése céljából. A módszer felhasználja a mikroszkóp pontválasz függvényét, mely így jobb rekonstrukcióhoz vezet. Matematikailag a probléma egy dupla integrált tartalmazó energiafüggvényként van felírva. A függvény Euler-Lagrange egyenletének kiszámítása kihívást jelentő feladat. Az algoritmus csúcsminőségét három különböző, általunk generált adatsoron demonstráltuk. Az első adatsor szimulált képeket tartalmaz. A másik kettő valós képekből áll, egyik miniatűr objektumokról készült, melyeknek fluoreszcencia képét a keresendő megoldásnak tekintjük, a másik pedig sejtekről készült képeket és manuálisan szegmentált változataikat tartalmazza.

A transzformált képek már tartalmazzák a minta morfológiai információját, de a Z tengely mentén való kiterjedésük nem egyértelmű. Ennek megoldására javasoltam egy algoritmust, mely fázis és magasság becslésére használható DIC mikroszkópiával. A módszer egy kalibrációs lépéssorozat, mely során képeket készítünk ismert fénytörési mutatóval rendelkező mikrogyöngyökről, majd kiszámoljuk azt a függvényt, ami a pixel intenzitásokat fizikai tulajdonságokra képezi. A technika egyszerű és költséghatékony módszert jelent fázismérésre, nem igényel további mikroszkóp alaktrészeket és már egyetlen kép alapján is megbízhatóan működik.

Patch Clamp Pipetta Detekciós Modellek

A jól definiált geometriával rendelkező objektumok szegmentálása a képfeldolgozásnak egy fontos része. A dolgozatban egy modell kiterjesztéseként bemutattam egy variációs módszert, amely közös referenciaponttal rendelkező, hosszúkás, egyenes objektum-párok detektálására használható.

A Pipette Hunter pipettahegy detektáló algoritmus három változatát mutattam be, melyek mindegyike variációs keretrendszeren alapul. A módszerek egyaránt működnek DIC, oblique és fluoreszcens mikroszkópia stack képein (kvázi-3D képeken), melyek mikropipettáról készült optikai képszeleteket tartalmaznak. A módszerek bemenetét vagy a stack képről készült különböző irányú projekciók, vagy maga a stack kép jelenti, az algoritmus által támogatott dimenziók számától függően. A 2 dimenziós módszer két négyzetet, míg a 3 dimenziós módszer jelölés mentes képeken két hengert próbál meg illeszteni a képre. A két illesztendő objektumnak mindkét esetben közös referenciapontja van. A fluoreszcens 3 dimenziós modell esetén ellenben az algoritmus egyetlen csonkakúpot illeszt a képre.

Az algoritmusok minőségének mérésére az eredmény pozíciókat összehasonlítottuk manuálisan választott megoldásokkal. A detekciók pontossága lehetővé teszi a sejtek mikropipettával történő megbízható eltalálását. A módszerek beépítésre kerültek az automatizált patch clamp rendszerünkbe.

Automatizált Mikroszkópia és Alkalmazások

Az utóbbi időben a patch clamp technika automatizálásában megjelentek a képpel vezérelt rendszerek. A disszertációm harmadik részében bemutattam az általam fejlesztett, első teljesen automatikus jelölés mentes patch clamp rendszert, mely lehetővé teszi humán minták vizsgálatát. A rendszert egyetlen szoftver vezérli, beleértve a az összes hardver elemet, az adatfeldolgozást és vizualizációt. A patch clamp folyamat elvégzéséhez egysejt alapú stratégiát használunk: mikrométer pontosságú kalibrációt, a pipetta hegyének regisztrációját és mozgatását, a célsejt valós idejű térbeli elasztikus követését, akadályok (pl. egyéb sejteknek vagy a szövet más alkotóelemeinek) kikerülését a célsejt néhány mikrométer pontosságú megközelítése során, és egy fejlett nyomásszbályzó rendszert amely a klasszikus manuális nyomás protokollokat helyettesíti a teljes folyamat során. A módszer központi eleme az egyes neuronok 3D képeken való detekciója volt, melyhez egy képi adatbázist készítettünk egy konvolúciós neurális hálózat tanításához.

A rendszer adottságait és képességeit patkány és humán agyszövetben lévő neuronok nagyszámú vizsgálatával mutattuk be. A mérések formátuma kompatibilis az Allen Institute for Brain Science adatbázisával és beillesztésre kerültek abba. Néhány sejt sejtmagja vagy citoplazmája kiemelésre került a mérés után, melyeket molekulárisan jellemeztünk. Az egysejt RNS szekvenálás eredménye megerősítette a különböző sejttípusok transzkriptomikai jellemzőit. Rendszerünk könnyen átültethető bármilyen létező mikroszkóp eszközre. Habár az emberi szakértőt nem váltja ki teljese mértékben, lényegesen felnyorsítja a mérési folyamatot, lerövidíti az elektrofiziológusok betanulási idejét és egyetlen operátor egyidőben akár több eszközt is üzemeltethet. További terveink között szerepel a többpipettás rendszerre való kiterjesztés, mely lehetővé tenné sejt-párok, -hármasok vagy magasabb számú kapcsolatok egyidejű vizsgálatát.

Bibliography

[1] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram, "Image processing with imagej," *Biophotonics international*, vol. 11, no. 7, pp. 36–42, 2004.

[2] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "Nih image to imagej: 25 years of image analysis," *Nature methods*, vol. 9, no. 7, p. 671, 2012.

[3] T. J. Collins, "Imagej for microscopy," *Biotechniques*, vol. 43, no. S1, pp. S25–S30, 2007.

[4] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, *et al.*, "Fiji: an open-source platform for biological-image analysis," *Nature methods*, vol. 9, no. 7, p. 676, 2012.

[5] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, *et al.*, "Cellprofiler: image analysis software for identifying and quantifying cell phenotypes," *Genome biology*, vol. 7, no. 10, p. R100, 2006.

[6] M. R. Lamprecht, D. M. Sabatini, and A. E. Carpenter, "Cellprofiler: free, versatile software for automated biological image analysis," *Biotechniques*, vol. 42, no. 1, pp. 71–75, 2007.

[7] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht, "Ilastik: Interactive learning and segmentation toolkit," in *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pp. 230–233, IEEE, 2011.

[8] I. Suleymanova, T. Balassa, S. Tripathi, C. Molnar, M. Saarma, Y. Sidorova, and P. Horvath, "A deep convolutional neural network approach for astrocyte detection," *Scientific reports*, vol. 8, 2018.

[9] F. Piccinini, T. Balassa, A. Szkalisity, C. Molnar, L. Paavolainen, K. Kujala, K. Buzas,

M. Sarazova, V. Pietiainen, U. Kutay, *et al.*, "Advanced cell classifier: user-friendly machine-learning-based software for discovering phenotypes in high-content imaging data," *Cell systems*, vol. 4, no. 6, pp. 651–655, 2017.

[10] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter, "Cellprofiler analyst: data exploration and analysis software for complex image-based screens," *BMC bioinformatics*, vol. 9, no. 1, p. 482, 2008.

[11] F. Wandrey, C. Montellese, K. Koos, L. Badertscher, L. Bammert, A. G. Cook, I. Zemp, P. Horvath, and U. Kutay, "The NF45/NF90 heterodimer contributes to the biogenesis of 60S ribosomal subunits and influences nucleolar morphology," *Molecular and cellular biology*, vol. 35, no. 20, pp. 3491–3503, 2015.

[12] K. Smith, F. Piccinini, T. Balassa, K. Koos, T. Danka, H. Azizpour, and P. Horvath, "Phenotypic image analysis software tools for exploring and understanding big image data from cell-based assays," *Cell systems*, vol. 6, no. 6, pp. 636–653, 2018.

[13] A. Schierbeek, *Measuring the invisible world: the life and works of Antoni van Leeuwenhoek.* No. 37, Abelard-Schuman, 1959.

[14] R. Hooke, *Micrographia: or some physiological descriptions of minute bodies made by magnifying glasses, with observations and inquiries thereupon.* Courier Corporation, 2003.

[15] S. Bell and K. Morris, *An introduction to microscopy.* CRC Press, 2009.

[16] C. Vonesch, F. Aguet, J.-L. Vonesch, and M. Unser, "The colored revolution of bioimaging," *IEEE signal processing magazine*, vol. 23, no. 3, pp. 20–31, 2006.

[17] A. Molleman, *Patch clamping: an introductory guide to patch clamp electrophysiology.* John Wiley & Sons, 2003.

[18] D. Murphy and M. Davidson, *Phase contrast microscopy and darkfield microscopy*, pp. 115–133. John Wiley & Sons, Inc., 2012.

[19] D. Murphy and M. Davidson, *Differential Interference Contrast Microscopy and Modulation Contrast Microscopy*, pp. 173–197. John Wiley & Sons, Inc., 2012.

[20] E. Cuche, P. Marquet, and C. Depeursinge, "Simultaneous amplitude-contrast and quantitative phase-contrast microscopy by numerical reconstruction of fresnel off-axis holograms," *Applied optics*, vol. 38, no. 34, pp. 6994–7001, 1999.

[21] P. Marquet, B. Rappaz, P. J. Magistretti, E. Cuche, Y. Emery, T. Colomb, and C. Depeursinge, "Digital holographic microscopy: a noninvasive contrast imaging technique allowing quantitative visualization of living cells with subwavelength axial accuracy," *Optics letters*, vol. 30, no. 5, pp. 468–470, 2005.

[22] C. J. Mann, L. Yu, C.-M. Lo, and M. K. Kim, "High-resolution quantitative phase-contrast microscopy by digital holography," *Optics Express*, vol. 13, no. 22, pp. 8693–8698, 2005.

[23] T. Kim, R. Zhou, L. L. Goddard, and G. Popescu, "Solving inverse scattering problems in biological samples by quantitative phase imaging," *Laser & Photonics Reviews*, vol. 10, no. 1, pp. 13–39, 2016.

[24] B. Bhaduri, C. Edwards, H. Pham, R. Zhou, T. H. Nguyen, L. L. Goddard, and G. Popescu, "Diffraction phase microscopy: principles and applications in materials and life sciences," *Advances in Optics and Photonics*, vol. 6, no. 1, pp. 57–119, 2014.

[25] C. W. Freudiger, W. Min, B. G. Saar, S. Lu, G. R. Holtom, C. He, J. C. Tsai, J. X. Kang, and X. S. Xie, "Label-free biomedical imaging with high sensitivity by stimulated raman scattering microscopy," *Science*, vol. 322, no. 5909, pp. 1857–1861, 2008.

[26] D. Zhang, P. Wang, M. N. Slipchenko, D. Ben-Amotz, A. M. Weiner, and J.-X. Cheng, "Quantitative vibrational imaging by hyperspectral stimulated raman scattering microscopy and multivariate curve resolution analysis," *Analytical chemistry*, vol. 85, no. 1, pp. 98–106, 2012.

[27] S. B. Mehta and C. J. Sheppard, "Equivalent of the point spread function for partially coherent imaging," *Optica*, vol. 2, no. 8, pp. 736–739, 2015.

[28] S. B. Mehta and C. J. Sheppard, "Partially coherent image formation in differential interference contrast (DIC) microscope," *Optics express*, vol. 16, no. 24, pp. 19462–19479, 2008.

[29] S. Kou, L. Waller, G. Barbastathis, and C. Sheppard, "Transport-of-intensity approach to differential interference contrast (TI-DIC) microscopy for quantitative phase imaging," *Optics letters*, vol. 35, no. 3, pp. 447–449, 2010.

[30] L. Waller, L. Tian, and G. Barbastathis, "Transport of intensity phase-amplitude imaging with higher order intensity derivatives," *Optics express*, vol. 18, no. 12, pp. 12552–12561, 2010.

[31] L. Waller, M. Tsang, S. Ponda, S. Yang, and G. Barbastathis, "Phase and amplitude imaging from noisy images by Kalman filtering," *Optics express*, vol. 19, no. 3, pp. 2805–2815, 2011.

[32] C. Cogswell, N. Smith, K. Larkin, and P. Hariharan, "Quantitative DIC microscopy using a geometric phase shifter," in *BiOS'97, Part of Photonics West*, pp. 72–81, International Society for Optics and Photonics, 1997.

[33] S. King, A. Libertun, C. Preza, and C. Cogswell, "Calibration of a phase-shifting DIC microscope for quantitative phase imaging," in *Proc. SPIE*, vol. 6443, p. 64430M, 2007.

[34] M. Arnison, K. Larkin, C. Sheppard, N. Smith, and C. Cogswell, "Linear phase imaging using differential interference contrast microscopy," *Journal of microscopy*, vol. 214, no. 1, pp. 7–12, 2004.

[35] C. Preza, S. King, and C. Cogswell, "Algorithms for extracting true phase from rotationally-diverse and phase-shifted DIC images," in *Biomedical Optics 2006*, International Society for Optics and Photonics, 2006.

[36] E. Van Munster, L. Van Vliet, and J. Aten, "Reconstruction of optical pathlength distributions from images obtained by a wide-field differential interference contrast microscope," *Journal of Microscopy*, vol. 188, no. 2, pp. 149–157, 1997.

[37] K. Li and T. Kanade, "Nonnegative mixed-norm preconditioning for microscopy image segmentation," in *International Conference on Information Processing in Medical Imaging*, pp. 362–373, Springer, 2009.

[38] B. Heise, A. Sonnleitner, and E. Klement, "DIC image reconstruction on large cell scans," *Microscopy research and technique*, vol. 66, no. 6, pp. 312–320, 2005.

[39] S. B. Mehta and R. Oldenbourg, "Image simulation for biological microscopy: microlith," *Biomedical optics express*, vol. 5, no. 6, pp. 1822–1838, 2014.

[40] M. Arnison, C. Cogswell, N. Smith, P. Fekete, and K. Larkin, "Using the hilbert transform for 3d visualization of differential interference contrast microscope images," *Journal of microscopy*, vol. 199, no. 1, pp. 79–84, 2000.

[41] E. Van Munster, L. Van Vliet, and J. Aten, "Reconstruction of optical pathlength distributions from images obtained by a wide-field differential interference contrast microscope," *Journal of Microscopy*, vol. 188, no. 2, pp. 149–157, 1997.

[42] Z. Yin, K. DFE, and T. Kanade, "Restoring DIC microscopy images from multiple shear directions," in *IPMI*, pp. 384–397, Springer, 2011.

[43] P. Feineigle, A. Witkin, and V. Stonick, "Processing of 3D DIC microscopy images for data visualization," in *ICASSP*, vol. 4, pp. 2160–2163, IEEE, 1996.

[44] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[45] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[46] K. Koos, J. Molnár, and P. Horvath, "DIC microscopy image reconstruction using a novel variational framework," in *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–7, IEEE, 2015.

[47] K. Koos, J. Molnár, L. Kelemen, G. Tamás, and P. Horvath, "DIC image reconstruction using an energy minimization framework to visualize optical path length distribution," *Scientific reports*, vol. 6, p. 30420, 2016.

[48] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[49] Z. Yin and T. Kanade, "Restoring artifact-free microscopy image sequences," in *Biomedical Imaging: From Nano to Macro*, pp. 909–913, IEEE, 2011.

[50] K. Smith, Y. Li, F. Piccinini, G. Csucs, C. Balazs, A. Bevilacqua, and P. Horvath, "Cidre: an illumination-correction method for optical microscopy," *Nature methods*, vol. 12, no. 5, p. 404, 2015.

[51] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[52] S. Kawata, H.-B. Sun, T. Tanaka, and K. Takada, "Finer features for functional microdevices," *Nature*, vol. 412, no. 6848, p. 697, 2001.

[53] A. Búzás, L. Kelemen, A. Mathesz, L. Oroszi, G. Vizsnyiczai, T. Vicsek, and P. Ormos, "Light sailboats: Laser driven autonomous microrobots," *Applied Physics Letters*, vol. 101, no. 4, p. 041111, 2012.

[54] G. Vizsnyiczai, L. Kelemen, and P. Ormos, "Holographic multi-focus 3d two-photon polymerization with real-time calculated holograms," *Optics express*, vol. 22, no. 20, pp. 24217–24223, 2014.

[55] K. Koos, B. Peksel, and L. Kelemen, "Phase measurement using DIC microscopy," *Acta Cybernetica*, vol. 23, no. 2, pp. 629–643, 2017.

[56] C. Lacoste, X. Descombes, and J. Zerubia, "Point processes for unsupervised line network extraction in remote sensing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1568–1579, 2005.

[57] G. Perrin, X. Descombes, and J. Zerubia, "Tree crown extraction using marked point processes," in *2004 12th European Signal Processing Conference*, pp. 2127–2130, IEEE, 2004.

[58] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny, "Automatic building extraction from dems using an object approach and application to the 3d-city modeling," *ISPRS Journal of photogrammetry and remote sensing*, vol. 63, no. 3, pp. 365–381, 2008.

[59] C. Molnar, I. H. Jermyn, Z. Kato, V. Rahkama, P. Östling, P. Mikkonen, V. Pietiäinen, and P. Horvath, "Accurate morphology preserving segmentation of overlapping cells based on active contours," *Scientific reports*, vol. 6, p. 32412, 2016.

[60] M. Rochery, I. H. Jermyn, and J. Zerubia, "Higher order active contours," *International Journal of Computer Vision*, vol. 69, no. 1, pp. 27–42, 2006.

[61] P. Thevenaz and M. Unser, "Snakuscules," *IEEE Transactions on image processing*, vol. 17, no. 4, pp. 585–593, 2008.

[62] P. Thevenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382–393, 2011.

[63] S. B. Kodandaramaiah, G. T. Franzesi, B. Y. Chow, E. S. Boyden, and C. R. Forest, "Automated whole-cell patch-clamp electrophysiology of neurons in vivo," *Nature methods*, vol. 9, no. 6, p. 585, 2012.

[64] R. Perin and H. Markram, "A computer-assisted multi-electrode patch-clamp system," *Journal of visualized experiments: JoVE*, no. 80, 2013.

[65] N. S. Desai, J. J. Siegel, W. Taylor, R. A. Chitwood, and D. Johnston, "Matlab-based automated patch clamp system for awake behaving mice," *American Journal of Physiology-Heart and Circulatory Physiology*, 2015.

[66] S. B. Kodandaramaiah, G. L. Holst, I. R. Wickersham, A. C. Singer, G. T. Franzesi, M. L. McKinnon, C. R. Forest, and E. S. Boyden, "Assembly and operation of the autopatcher for automated intracellular neural recording in vivo," *Nature protocols*, vol. 11, no. 4, p. 634, 2016.

[67] B. Amuzescu, O. Scheel, and T. Knott, "Novel automated patch-clamp assays on stem cell-derived cardiomyocytes: Will they standardize in vitro pharmacology and arrhythmia research?," *Journal of Physical Chemistry & Biophysics*, vol. 4, no. 4, p. 1, 2014.

[68] I. Kolb, W. Stoy, E. Rousseau, O. Moody, A. Jenkins, and C. Forest, "Cleaning patch-clamp pipettes for immediate reuse," *Scientific reports*, vol. 6, p. 35001, 2016.

[69] B. Long, L. Li, U. Knoblich, H. Zeng, and H. Peng, "3d image-guided automatic pipette positioning for single cell experiments in vivo," *Scientific reports*, vol. 5, p. 18426, 2015.

[70] Q. Wu, I. Kolb, B. M. Callahan, Z. Su, W. Stoy, S. B. Kodandaramaiah, R. L. Neve, H. Zeng, E. S. Boyden, C. R. Forest, *et al.*, "Integration of autopatching with automated pipette and cell detection in vitro," *American Journal of Physiology-Heart and Circulatory Physiology*, 2016.

[71] K. Koos, J. Molnár, and P. Horvath, "Pipette Hunter: Patch-clamp pipette detection," in *Scandinavian Conference on Image Analysis*, pp. 172–183, Springer, 2017.

[72] B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, *et al.*, "Adult mouse cortical cell taxonomy revealed by single cell transcriptomics," *Nature neuroscience*, vol. 19, no. 2, p. 335, 2016.

[73] B. Tasic, Z. Yao, L. T. Graybuck, K. A. Smith, T. N. Nguyen, D. Bertagnolli, J. Goldy, E. Garren, M. N. Economo, S. Viswanathan, *et al.*, "Shared and distinct transcriptomic cell types across neocortical areas," *Nature*, vol. 563, no. 7729, p. 72, 2018.

[74] H. Zeng, E. H. Shen, J. G. Hohmann, S. W. Oh, A. Bernard, J. J. Royall, K. J. Glattfelder, S. M. Sunkin, J. A. Morris, A. L. Guillozet-Bongaarts, *et al.*, "Large-scale cellular-resolution gene profiling in human neocortex reveals species-specific molecular signatures," *Cell*, vol. 149, no. 2, pp. 483–496, 2012.

[75] H.-J. Suk, E. S. Boyden, and I. van Welie, "Advances in the automation of whole-cell patch clamp technology," *Journal of neuroscience methods*, p. 108357, 2019.

[76] Y. Peng, F. X. Mittermaier, H. Planert, U. C. Schneider, H. Alle, and J. R. Geiger, "High-throughput microcircuit analysis of individual human brains through next-generation multi-neuron patch-clamp," *BioRxiv*, p. 639328, 2019.

[77] S. B. Kodandaramaiah, *Robotics for in vivo whole cell patch clamping.* PhD thesis, Georgia Institute of Technology, 2012.

[78] H.-J. Suk, I. van Welie, S. B. Kodandaramaiah, B. Allen, C. R. Forest, and E. S. Boyden, "Closed-loop real-time imaging enables fully automated cell-targeted patch-clamp neural recording in vivo," *Neuron*, vol. 95, no. 5, pp. 1037–1047, 2017.

[79] L. A. Annecchino, A. R. Morris, C. S. Copeland, O. E. Agabi, P. Chadderton, and S. R. Schultz, "Robotic automation of in vivo two-photon targeted whole-cell patch-clamp electrophysiology," *Neuron*, vol. 95, no. 5, pp. 1048–1055, 2017.

[80] W. A. Stoy, I. Kolb, G. Holst, Y. J. Liew, A. Pala, B. Yang, E. S. Boyden, G. B. Stanley, and C. R. Forest, "Robotic navigation to sub-cortical neural tissue for intracellular electrophysiology in vivo," *American Journal of Physiology-Heart and Circulatory Physiology*, 2017.

[81] I. Kolb, C. R. Landry, M. C. Yip, C. F. Lewallen, W. A. Stoy, J. Lee, A. Felouzis, B. Yang, E. S. Boyden, C. J. Rozell, *et al.*, "Patcherbot: a single-cell electrophysiology robot for adherent cells and brain slices," *Journal of neural engineering*, vol. 16, no. 4, p. 046003, 2019.

[82] S. B. Kodandaramaiah, F. J. Flores, G. L. Holst, A. C. Singer, X. Han, E. N. Brown, E. S. Boyden, and C. R. Forest, "Multi-neuron intracellular recording in vivo via interacting autopatching robots," *Elife*, vol. 7, p. e24656, 2018.

[83] L. Li, B. Ouellette, W. A. Stoy, E. J. Garren, T. L. Daigle, C. R. Forest, C. Koch, and H. Zeng, "A robot for high yield electrophysiology and morphology of single neurons in vivo," *Nature communications*, vol. 8, p. 15604, 2017.

[84] "Allen brain atlas. *Allen Cell Types Database.*" `http://help.brain-map.org/display/celltypes`. Accessed: 2019-03-26.

[85] F. Zanella, J. B. Lorens, and W. Link, "High content screening: seeing is believing," *Trends in biotechnology*, vol. 28, no. 5, pp. 237–245, 2010.

[86] M. Bickle, "The beautiful cell: high-content screening in drug discovery," *Analytical and bioanalytical chemistry*, vol. 398, no. 1, pp. 219–226, 2010.

[87] M. M. Usaj, E. B. Styles, A. J. Verster, H. Friesen, C. Boone, and B. J. Andrews, "High-content screening for quantitative cell biology," *Trends in cell biology*, vol. 26, no. 8, pp. 598–611, 2016.

[88] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[89] X. Xu, S. Xu, L. Jin, and E. Song, "Characteristic analysis of otsu threshold and its applications," *Pattern recognition letters*, vol. 32, no. 7, pp. 956–961, 2011.

[90] T. R. Jones, A. E. Carpenter, M. R. Lamprecht, J. Moffat, S. J. Silver, J. K. Grenier, A. B. Castoreno, U. S. Eggert, D. E. Root, P. Golland, *et al.*, "Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning," *Proceedings of the National Academy of Sciences*, vol. 106, no. 6, pp. 1826–1831, 2009.

[91] F. Cornelissen, M. Cik, and E. Gustin, "Phaedra, a protocol-driven system for analysis and validation of high-content imaging and flow cytometry," *Journal of biomolecular screening*, vol. 17, no. 4, pp. 496–506, 2012.

[92] D. Laksameethanasan, R. Z. Tan, G. W.-L. Toh, and L.-H. Loo, "cellxpress: a fast and user-friendly software platform for profiling cellular phenotypes," *BMC bioinformatics*, vol. 14, no. 16, p. S4, 2013.

[93] A. Ogier and T. Dorval, "Hcs-analyzer: open source software for high-content screening data correction and analysis," *Bioinformatics*, vol. 28, no. 14, pp. 1945–1946, 2012.

[94] N. Orlov, L. Shamir, T. Macura, J. Johnston, D. M. Eckley, and I. G. Goldberg, "Wnd-charm: Multi-purpose image classification using compound image transforms," *Pattern recognition letters*, vol. 29, no. 11, pp. 1684–1693, 2008.

[95] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pp. 41–48, Ieee, 1999.

[96] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature biotechnology*, vol. 32, no. 4, p. 381, 2014.

[97] S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, E. F. Simonds, T. J. Chen, D. K. Shenfeld, G. P. Nolan, and D. Pe'er, "Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development," *Cell*, vol. 157, no. 3, pp. 714–725, 2014.

[98] M. Held, M. H. Schmitz, B. Fischer, T. Walter, B. Neumann, M. H. Olma, M. Peter, J. Ellenberg, and D. W. Gerlich, "Cellcognition: time-resolved phenotype annotation in high-throughput live cell imaging," *Nature methods*, vol. 7, no. 9, p. 747, 2010.

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[101] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1790–1802, 2015.

[102] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, ACM, 2014.

[103] L. Yeager, J. Bernauer, A. Gray, and M. Houston, "Digits: the deep learning gpu training system," in *ICML 2015 AutoML Workshop*, 2015.

[104] A. Tao, J. Barker, and S. Sarathy, "Detectnet: Deep neural network for object detection in digits," *Parallel Forall*, vol. 4, 2016.

[105] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[106] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[107] G. Molnár, N. Faragó, Á. K. Kocsis, M. Rózsa, S. Lovas, E. Boldog, R. Báldi, É. Csajbók, J. Gardi, L. G. Puskás, *et al.*, "Gabaergic neurogliaform cells represent local sources of insulin in the cerebral cortex," *Journal of Neuroscience*, vol. 34, no. 4, pp. 1133–1137, 2014.

[108] G. A. Ascoli, L. Alonso-Nanclares, S. A. Anderson, G. Barrionuevo, R. Benavides-Piccione, A. Burkhalter, G. Buzsáki, B. Cauli, J. DeFelipe, A. Fairén, *et al.*, “Petilla terminology: nomenclature of features of gabaergic interneurons of the cerebral cortex,” *Nature Reviews Neuroscience*, vol. 9, no. 7, p. 557, 2008.

[109] E. Boldog, T. E. Bakken, R. D. Hodge, M. Novotny, B. D. Aevermann, J. Baka, S. Bordé, J. L. Close, F. Diez-Fuertes, S.-L. Ding, *et al.*, “Transcriptomic and morphophysiological evidence for a specialized human cortical gabaergic cell type,” *Nature neuroscience*, vol. 21, no. 9, p. 1185, 2018.

[110] K. Koos, G. Oláh, T. Balassa, N. Mihut, M. Rózsa, A. Ozsvár, E. Tasnádi, J. Molnár, P. Barzó, G. Molnár, G. Tamás, and P. Horvath, “Deep learning driven label-free image guided automatic patch clamp system for human and rodent in vitro slice physiology,” *Manuscript.*