Dissertation

zur Erlangung des Doktorgrades der Technischen Fakultät der Albert-Ludwigs-Universität Freiburg im Breisgau

Segmentation of Cells and Sub-cellular Structures from Microscopic Recordings

Margret Keuper

2012

Albert-Ludwigs-Universität
Freiburg im Breisgau

Dekan:	Prof. Dr. Bernd Becker

Prüfungskommission:

Prof. Dr. Matthias Teschner	(Vorsitz)
Junior-Prof. Dr. Olaf Ronneberger	(Beisitz)
Prof. Dr.-Ing. Thomas Brox	(1. Gutachter)
Prof. Dr. Fred Hamprecht	(2. Gutachter)

Datum der Disputation: 30.11.2012

TO JANIS AND PHILINE

Abstract

In life-sciences, technical innovations in microscopy and the ever growing availability of modern fluorescence microscopes allow for the investigation of more and more complex biological processes. To facilitate the analysis of the recorded microscopic data, the development of automated methods is indispensable, because these methods allow for the evaluation of a sufficient amount of data without bias.
The analysis of fluorescence microscopic data comes with some specific challenges. The recorded fluorescence signal is noisy, the signal intensity of the identical structure potentially varies from one recording point to the other and the recording resolution is strongly anisotropic due to technical limitations. Additionally, the optical system introduces blur, which is commonly described by a convolution with the so called *point spread function*.

The presented thesis primarily focuses on the segmentation of objects, mostly cells, cell nuclei, or other cell organelles, from microscopic recordings. Another key aspect is the deconvolution of the recordings in order to alleviate the blur introduced by the optical system. Especially when processing data from widefield microscopy, deconvolution is often a crucial preprocessing step. The presented methods for blind deconvolution are based on the regularization of the reconstructed point spread function. On our test datasets, they lead to improved, sharpened images with a better signal to noise ratio. A comparison to state-of-the-art methods confirms the aptitude of our approach.

The proposed methods for the segmentation of objects from volumetric data are based on parametric active surfaces. An important contribution is the generation of appropriate data terms for the representation of the object boundaries in noisy data with weak fluorescence signal. Thus, we present methods for the generation of data terms that allow for the segmentation of fluorescently labeled cell nuclei and nucleoli from widefield microscopic recordings. Furthermore, we propose a method to automatically generate new, very specific data terms for the segmentation of objects from volumetric data based on very simple user input. Experiments show that our methods allow an accurate segmentation of the objects under investigation, which, with conventional active surface and level set methods, could not be achieved. Thereby, our dataterms can be easily adapted to changing recording conditions.

Finally, we present a method that allows to learn topological knowledge that can be used for the simultaneous classification and segmentation of objects. This approach is based on hierarchical Markov Random Fields. By means of this method, the segmentation accuracy in a dataset of electron microscopic recordings could be greatly improved compared to conventional Markov Random Fields.

Zusammenfassung

Aufgrund technischer Innovationen im Bereich der Mikroskopie und der stetig wachsenden Verbreitung moderner Fluoreszenzmikroskope geraten immer komplexere biologische Abläufe und Modelle in den Fokus der Bio-Wissenschaften. Um die Analyse der aufgenommenen mikroskopischen Daten zu ermöglichen, ist die Entwicklung automatischer Methoden unabdingbar, denn nur so kann eine zuverlässige Auswertung hinreichend großer Datenmengen ohne systematische Fehler garantiert werden.
Dabei ist die Auswertung fluoreszenzmikroskopischer Daten mit einigen Herausforderungen verbunden. Das aufgenommene Fluoreszenzsignal ist rauschbehaftet, die Signalstärke ein und derselben Struktur kann in Abhängigkeit der Position des Datenpunktes innerhalb einer Aufnahme stark variieren und die Auflösung der aufgenommenen Daten ist aufgrund technischer Beschränkungen meist stark anisotrop. Aufgrund physikalischer Limitierungen wird durch die Optik Unschärfe in die Aufnahme eingeführt, die in aller Regel als Faltung des Datensatzes mit einer sogenannten *Punktbildfunktion* modelliert wird.

Die vorliegende Arbeit befasst sich vorrangig mit der Segmentierung von Objekten, meist Zellen, Zellkernen oder anderen subzellulären Strukturen, in mikroskopischen Aufnahmen. Ein weiterer zentraler Aspekt ist die Entfaltung der Aufnahmen zur Verringerung der durch die Optik eingeführten Unschärfe.
Gerade bei der Verarbeitung weitfeld-mikroskopischer Aufnahmen ist die Entfaltung der Daten oft ein zentraler Vorverarbeitungsschritt. Die von uns vorgestellten Methoden zur blinden Entfaltung basieren auf der Regularisierung der rekonstruierten Punktbildfunktion. Mit ihrer Hilfe konnte die Bildschärfe und damit das Signal zu Rausch Verhältnis in unseren Testdaten stark verbessert werden. Auch der Vergleich mit State-of-the-art Methoden bestätigt die Qualitäten unseres Ansatzes.

Die vorgestellten Methoden zur Segmentierung von Objekten aus volumetrischen Aufnahmen basieren auf parametrischen aktiven Oberflächen. Das Erstellen geeigneter Datenterme, welche die Objektgrenzen auch in rauschbehafteten Daten mit schwachem Fluoreszenzsignal gut repräsentieren, ist dabei ein wesentlicher Beitrag. Zum einen stellen wir verschiedene Datenterme vor, mit deren Hilfe fluoreszent markierte Zellkerne und Nukleoli aus weitfeld-mikroskopischen Aufnahmen segmentiert werden können. Zum anderen präsentieren wir eine Methode zur automatischen Erstellung neuer, sehr spezifischer Datenterme für die Segmentierung von Objekten aus volumetrischen Daten, die auf einer sehr einfachen Benutzereingabe basiert.
Experimente zeigen, dass die vorgestellten Methoden, im Gegensatz zu konventionellen aktiven Oberflächen und Level Set Methoden, eine genaue Segmentierung der untersuchten Objekte ermöglichen. Dabei können die Datenterme leicht an neue Aufnah-

mebedingungen angepasst werden.

Schließlich stellen wir eine Methode vor, die es ermöglicht, topologisches Wissen zu lernen und zur gleichzeitigen Klassifikation und Segmentierung von Objekten zu nutzen. Dieser Ansatz beruht auf hierarchischen Markov Random Fields. Im Vergleich zu konventionellen Markov Random Fields konnte eine deutliche Verbesserung in der Genauigkeit der Segmentierung eines Datensatzes elektronenmikroskopischer Aufnahmen gezeigt werden.

Acknowledgments

Despite my devotion to the pursuit of the research goals this thesis is committed to, its successful completion is not only my personal merit. It is therefore a great pleasure to thank everyone who helped and supported me finish this project successfully.

First I want to thank my supervisor Prof. Dr. Thomas Brox for his support and guidance. I also want to thank his predecessor Prof. Dr.-Ing. Hans Burkhardt who supported me a lot in the beginning of my thesis and trusted in my research decisions. Together with Junior-Prof. Dr. Olaf Ronneberger, these two made the image processing lab a very pleasant place to work. Thank you all for your support and for being open to my problems and concerns. Olaf, thank you especially for the countless inspiring discussions!

I would also like to thank Prof. Dr. Fred Hamprecht for his interest in my work and for agreeing to be co-examiner of this thesis. I would also like to thank Prof. Dr. Matthias Teschner for agreeing being member of the examination committee.

I am obliged to many of my colleagues who supported me in different ways: Janina Schulz and Alexandra Teynor encouraged me to pursue my own ideas in the very beginning of my time as a PhD student, where this advice was especially helpful, Thorsten Schmidt, Maja Temerinac-Ott, Peter Ochs and Robert Bensch were part of many fruitful discussions. Thorsten, I owe you thanks for many more reasons: especially for reassuring me in difficult situations and for coping with my temper in these exact situations.

I want to thank Cynthia Findlay for her help with organizational questions and Stefan Teister for being there whenever I encountered technical problems.

Last but not least, I want to thank my family. My parents taught me to believe in my abilities. They always believe in me and give me all their love and support. Janis, without your help it would even not have been possible to finish this thesis. My greatest thanks belong to you and Philine - for your love, your appreciation of my work although it cost much of our precious time, and for making me believe in happy endings.

Contents

1. Introduction

In life sciences, more and more complex biological models are to be investigated. The emerging field of systems biology aims for building models integrating the regulatory processes of organisms from the genome over cell organelles up to the behavior of the whole organism [Kitano, 2002]. The role of proteins as "dynamic actors" [Bu and Callaway, 2011] is thereby of major interest. Besides technical instruments for the automated analysis of e.g. certain properties of cell populations by flow cytometry or Western Blots, image-based analysis is a very important basis for new findings. Images can, in contrast to biophysical or biochemical approaches, provide not only quantitative but also contextual information. Recent advances in fluorescence microscopy lead to the possibility to rapidly generate large amounts of volumetric image data and even to record a specimen over time. In contrast to two-dimensional recordings of single sections of the specimen, volumetric recordings offer the possibility to observe the specimen as a whole. By recording several successive time-points, it is even possible to monitor its development.
However, the capability of human experts to observe dynamic processes in large amounts of volumetric data is strongly limited for the following reasons: First, in order to draw statistical conclusions, huge amounts of data have to be evaluated. Secondly, scientists usually already expect certain effects of their experiments on the data such that manual evaluations are potentially biased. Thirdly, if the biologically relevant information has to be found in three-dimensional space from volumetric recordings, the task of even analyzing a single dataset is very tedious. Most often, the data is then processed slice by slice by an expert who tries to combine the information from the current slice with the information gathered in neighboring slices. In contrast, an automated system is able to analyze volumetric information as such. High importance is therefore attached to computerized methods that either lead to a full automation of the analysis of the recorded data, or help the expert in analyzing the data in a semi-automated way. Crucial steps towards this automation are data preprocessing, such as denoising, deblurring, and registration of the data, object or event detection, and object segmentation.
This thesis is devoted to the automatic segmentation of small biological specimen from microscopic recordings. The specimen we are dealing with are cells, cell nuclei, and other cell organelles, i.e. we are working at the limit of the optical resolution. However, the exact knowledge of the individual cell or cell nucleus anatomy is crucial for the analysis of cellular and sub-cellular mechanisms. Depending on the concrete application, biologists for example want to measure the variation of size and shape of cells in a population, are interested in protein colocalizations, i.e. the co-occurrence of two or more proteins, or want to investigate the distribution of a certain protein in relation to the cell nucleus, nucleolus, mitochondria, or other structures. For these questions, it is usually necessary to record different data channels and relate them to one another. An example is given in figure 1.1. In this exemplary experimental setting, the cell nucleus is recorded in channel 0, the nucleolus in channel 1, and the protein under investigation in channel 2. From the segmentations of the cell nucleus and nucleolus and the detected protein clusters, a cell nucleus model can be built, that allows, for example, to measure the distances from the protein clusters to the nuclear membrane.

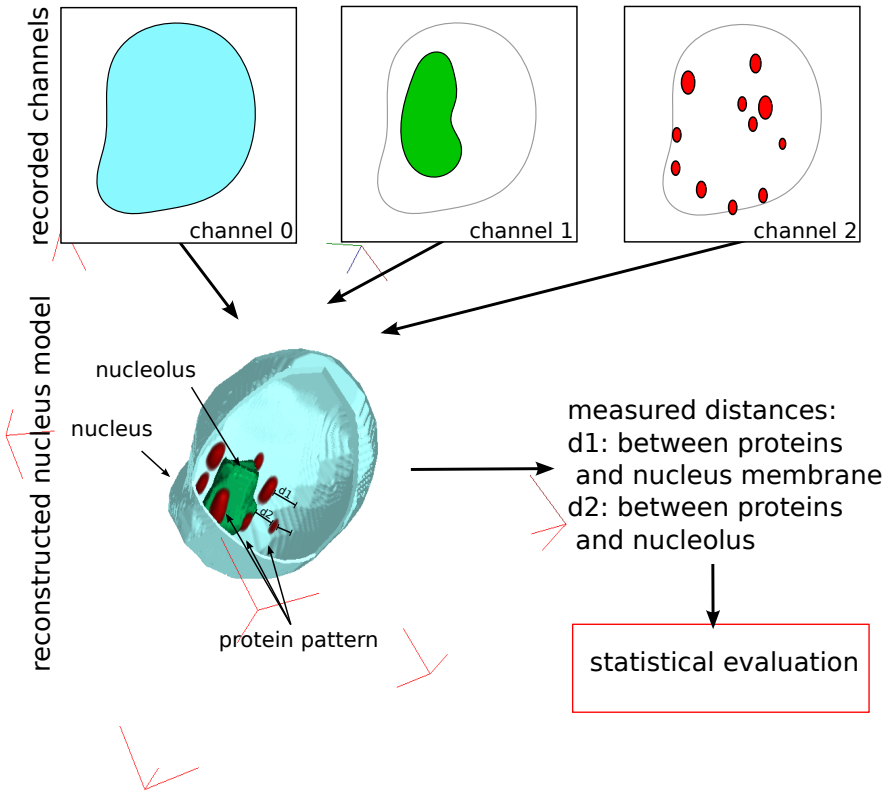


Figure 1.1.: Schematic sketch of a localization task. Data recorded in different channels need to be segmented and aligned in order to analyze protein patterns.

1.1. Segmentation in Microscopic Data

In the present work, we focus on the segmentation task, but we also give a short excursus to image deconvolution techniques that are in some cases a necessary tool for the processing of strongly blurred microscopic data. The image segmentation problem in general can be considered ill-posed. This is especially the case for the object segmentation task, even if full three-dimensional recordings are available. In object segmentation, the objective is not primarily to find regions homogeneous in color and texture but to segment whole semantic objects independently from the appearance of object parts. The question remains how to identify these objects as such. Good segmentations can only be achieved if sufficient prior information about the underlying object is given. Such information can be image-based or object-based. Under certain imaging conditions, the cell nucleus might for example yield higher image intensities than the cytoplasm such that the brightest region in the recording corresponds to the cell nucleus. Similarly, one often assumes homogeneous image intensities within single cell compartments. In this case, it makes sense to produce segmentations yielding high inter-region dissimilarity and high intra-region similarity. Often, edges in the image correspond to object boundaries. Object specific information can for example be given in form of topological knowledge. For example, we know that the cell nucleus is a closed region inside the cell and has, depending on the imaging conditions, a different appearance than the cytoplasm

but can have intensities similar to other cell organelles as for example mitochondria. Topological knowledge becomes especially important when parts of the object are not visible in the image. Even though we are not confronted with occlusions when working on full volumetric data, the information on the object boundaries is often not complete in fluorescence microscopic recordings.
One reason for deficient boundary information in fluorescence microscopic images is the blur that is introduced by the imaging system. Especially along the optical axis, the data is heavily blurred. Depending on the size of the recorded specimen and the recording system, it might therefore even be impossible to reconstruct the true boundary position in lower and upper object regions. Another reason for limited data quality is the potentially uneven fluorescent staining. Often, the stained protein is not evenly distributed in the specimen or the staining itself does not work perfectly. In these cases, the recorded object boundary has a strongly varying appearance even though it should theoretically be homogeneous. Due to absorption, bleaching, and scattering effects, boundary information can even be completely missing in some image regions.
To cope with these challenges and to facilitate the inclusion of topological knowledge, our algorithms for the segmentation in volumetric microscopic recordings are based on parametric active surfaces. Besides naturally incorporating topological information in the parameterization, they also allow for an object-based optimization of the surface by local image cues. Further constraints on the object surface like expected smoothness can easily be imposed. Since our data mainly consists of recordings from cell cultures, where isolated cells are fixed or floating in a growth medium, we can expect to find good initial estimates for the positions and sizes of our objects. The appropriate initialization of active surfaces should thus be accomplishable, such that good segmentations can be achieved by local optimizations. However, our main contributions lie in the creation of appropriate data force fields driving the surfaces towards the object boundaries. These force fields could as well be used in a convex optimization setting with non-parametric active surfaces if the incorporation of topological knowledge is not essential, and are therefore not limited to local optimizations.
A main issue we are facing is the fact that for fluorescence microscopic recordings, the measured image intensities can hardly be directly used as a cue for the object boundary. Blurring, bleaching and signal attenuation effects are too strong to allow for a direct evaluation of image intensities. Our approaches for the segmentation in fluorescence microscopic data are therefore based on the gradient information in the images.
Since in many cases the necessary information on the object we want to segment is not accessible at a local scale, we have to combine local and global image information in order to produce meaningful segmentations. To this end, we have developed a method based on the optimization of a hierarchical graphical model that allows for the automatic learning of topological information from ground truth annotations. This method was applied to the segmentation of two-dimensional recordings from electron microscopy into background, cytoplasm, and several cell organelles. Segmentations of these high resolution images are crucial to measure for example the cell or cell nucleus surface area that can only be roughly estimated from fluorescence microscopic recordings.

1.2. Contributions

Our contributions are in the field of image deconvolution and image segmentation.

In conventional fluorescence microscopic data, deconvolution is often needed especially for qualitative data analysis. It can also be used as preprocessing step for a later segmentation of the data. Common methods for **blind deconvolution** in conventional fluorescence microscopy use very complex prior assumptions on the blurring kernel. In [Keuper et al., 2012], we have proposed a very simple, image-based regularization method for the reconstructed blurring kernel, that nevertheless stabilizes the blind deconvolution scheme. Besides the presentation of this new kernel regularization, we also provide an evaluation of three commonly used methods for the regularization of the reconstructed data in [Keuper et al., 2012]. The method and its evaluation are described in section 4.7.1.

In [Koziolek, 2011], we have, together with Maja Temerinac-Ott, pursued the idea of computing the regularization of the blurring kernel not in the spatial domain but in the frequency space. This makes sense because in theory, the kernel should be well localized in the frequency domain and is infinitely extended in the spatial domain. A theoretic revision of the first attempts made in [Koziolek, 2011] and its evaluation will be proposed in section 4.7.2.

We have presented a new data term for the segmentation of cell nucleoli with active surfaces from volumetric multi-channel recordings: the **Channel Differential Structure** that will be described in section 5.3.1.

An **active surface model** has been proposed for the segmentation of cell nuclei that we have published together with the Channel Differential Structure in [Keuper et al., 2009]. The data term proposed in [Keuper et al., 2009] combines knowledge about the objects of interest with information from the object detection step in order to produce robust segmentations. The proposed method and its evaluation on a synthetic dataset as well as a dataset of fluorescence microscopic recordings of *Drosophila* S2 cell nuclei will be described in section 5.3.2.

Since volumetric data recorded with conventional fluorescence microscopy is strongly blurred, the correct object boundary in direction of the optical axis is hard to find. However, segmentation artifacts originating from the introduced blurring can be alleviated with the **Mean Shift Gradient Vector Flow** method we have proposed in [Keuper, 2010; Keuper et al., 2010b]. This method combines the standard Gradient Vector Flow computation with the Mean Shift method on gradient fields. The resulting vector field is not only less easily influenced by image blur but also more robust against noise. In [Chlap, 2010], we have used Mean Shift Gradient Vector Flow in order to produce robust Landmark candidates in recordings of Zebrafish embryos. In [Renz, 2010], we have tested the Mean Shift tracking algorithm on volumetric videos of cell nuclei in frog kidney. The derivation and evaluation of the Mean Shift Gradient Vector Flow method will be described in section 5.3.3.

The active surface data term presented in [Keuper et al., 2009] is adapted to a special, typical imaging setting, where the objects of interest, the cell nuclei, are entirely stained. If this is not given, the data term has to be newly adapted to the presented imaging setting. In [Keuper et al., 2010a], we have proposed a method for a **semi-automatic generation of edge filters** that perform this task. The method is based on a pre-clustering of radial boundary profiles. It has been evaluated on a set of

confocal microscopic recordings of tobacco protoplast cells in which different membrane proteins were stained. The proposed algorithm will be described in section 5.3.4.

For the optimization of active contours and surfaces, the adjustment of the weighting factors for data and regularization terms is crucial. In [Keuper et al., 2010c], we have proposed a method for the **automatic adjustment of weighting factors** for active surfaces. The method dynamically adapts active surface weighting factor depending on the data point currently covered by the surface. By dynamically increasing the regularization parameter, we can also handle low data quality with deficient boundary information. The proposed algorithm and its evaluation of *Drosophila* S2 cell recordings as well as on recordings of cell nuclei in the *Arabidopsis* root tip will be described in section 5.4.

Furthermore, we have developed a **hierarchical Markov Random Field** method that enables to use local and regional information for the segmentation of recordings into semantically different classes. The method has been applied to the segmentation of cells and several sub-cellular structures different in size from electron microscopic (EM) data, and was published in [Keuper et al., 2011]. A detailed description of the method is given in chapter 6. The hierarchical Markov Random Fields are built upon a region-wise texture classification. Appropriate features for this classification have been evaluated in [Huhn, 2009]. In [Morath et al., 2013], we have used semi-automatic segmentation, only using the first step of the method proposed in [Keuper et al., 2011] for the determination of cell and cell nucleus surface area in different kinds of human and mouse cells.

1.3. Publications Contributing to this Thesis

Peer Reviewed Conference Papers

M. Keuper, J. Padeken, P. Heun, H. Burkhardt, and O. Ronneberger. A 3D active surface model for the accurate segmentation of *Drosophila* Schneider cell nuclei and nucleoli. In *Int. Symp. on Visual Computing (ISVC), Part I*, LNCS, volume 5875, pages 865–874. Springer, 2009.

M. Keuper, R. Bensch, K. Voigt, A. Dovzhenko, K. Palme, H. Burkhardt, and O. Ronneberger. Semi-supervised learning of edge filters for volumetric image segmentation. In *Proc. of the 32nd DAGM Symp.*, LNCS, pages 462–471. Springer, 2010a.

M. Keuper, J. Padeken, P. Heun, H. Burkhardt, and O. Ronneberger. Mean shift gradient vector flow: A robust external force field for 3D active surfaces. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 2784–2787, 2010b.

M. Keuper, T. Schmidt, J. Padeken, P. Heun, K. Palme, H. Burkhardt, and O. Ronneberger. 3d deformable surfaces with locally self-adjusting parameters - a robust method to determine cell nucleus shapes. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 2254–2257, 2010c.

M. Temerinac-Ott, M. Keuper, and H. Burkhardt. Evaluation of a new point clouds registration method based on group averaging features. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 2452–2455, 2010.

M. Keuper, T. Schmidt, M. Rodriguez-Franco, W. Schamel, T. Brox, H. Burkhardt, and O. Ronneberger. Hierarchical Markov Random Fields for mast cell segmentation in electron microscopic recordings. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 973–978. 2011.

M. Keuper, M. Temerinac-Ott, J. Padeken, P. Heun, T. Brox, H. Burkhardt, and O. Ronneberger. Blind deconvolution with PSF regularization for widefield microscopy. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 1292–1295. 2012.

T. Schmidt, M. Keuper, T. Pasternak, K. Palme, and O. Ronneberger. Modeling of sparsely sampled tubular surfaces using coupled curves. In *Proc. of the 34nd DAGM Symp.*, LNCS. Springer, pages 83–92, 2012. (to appear).

Biological Applications (Journals)

C. S. Casas Delucchi, A. Brero, M. Keuper, O. Ronneberger, A. Wutz, H. Burkhardt, B. Joffe, I. Solovei, and T. Cremer. Chromosome territories and nuclear architecture: A study using native and transgenic Xist inactivation. *Cellular Oncology*, 30(3):251, 2008.

V. Morath* , M. Keuper* , M. Rodriguez-Franco, S. Deswal, G. Fiala, B. Blumenthal, D. Kaschek, J. Timmer, G. Neuhaus, O. Ronneberger, and W. Schamel. Semi-automatic determination of cell surface areas used in systems biology. Frontiers in Bioscience, 5, 533–545, Jan. 2013 , * authors contributed equally.

Miscellaneous

M. Keuper. Mean shifting gradient vector flow, an improved force field for active surfaces in wide-field microscopy. Technical Report 255, IIF-LMB, University of Freiburg, 2010.

P. Chlap. Detection and description of landmark candidates for the registration of 3D Zebrafish embryos, 2010. Master Thesis, Albert-Ludwigs-Universität Freiburg, Supervisor: Margret Keuper.

D. Huhn. Texture analysis for the segmentation of t-cells in recordings from electron microscopy, 2009. Student Research Project, Albert-Ludwigs-Universität Freiburg, Supervisor: Margret Keuper.

E. Koziolek. Evaluation of deconvolution methods in volumetric microscopic data, 2011. Student Research Project, Albert-Ludwigs-Universität Freiburg, Supervisor: Margret Keuper and Maja Temerinac-Ott.

K. Renz. Comparison of different tracking methods in 3D datasets, 2010. Master Thesis, Albert-Ludwigs-Universität Freiburg, Supervisor: Thorsten Schmidt and Margret Keuper.

1.4. Related Work

Even though the aim of this thesis is clearly defined as producing reliable segmentations for cells and sub-cellular structures in microscopic images, the methods we have developed to this end are quite diverse and range from traditional image segmentation methods to image deconvolution techniques.
Related work on the segmentation of cells and cell nuclei from microscopic images is very diverse. Countless methods have been published that are specialized for the segmentation of cells or cell nuclei from special recording settings. When focusing on work treating the segmentation problem in three-dimensional microscopic recordings, some works are to be mentioned as for example the three-dimensional Gaussian intensity models [Thomann et al., 2002; Heinzer et al., 2008] and more elaborate parametric models like the super-ellipsoidal intensity models [Wörz et al., 2010, 2007]. These methods focus, as our active surface approach, on the segmentation of whole objects. However, the resulting shape variations are limited and the segmentation result is strongly coupled to the measured image intensities that are, at least in our data, not sufficiently reliable. Region-based level sets, also strongly depending on image intensities, have for example been used in [Dufour et al., 2005] in the context of three-dimensional cell segmentation.
A different approach has been pursued for example in [Sommer et al., 2011; Fehr et al., 2005; Ronneberger et al., 2005; Kaster et al., 2011], where the object segmentation is built upon two or three-dimensional feature-based learning of the objects' textures or structures and a subsequent voxel-wise classification. These approaches are not object-based such that for example the delineation of touching cells can not always be solved. In [Lou et al., 2012], cell nuclei are segmented based on previously learned shape priors.
A detailed overview on state-of-the-art segmentation methods in general will be given in chapter 2. Furthermore, references to works closely related to our individual, proposed methods will be given in the respective chapters such that differences and similarities can be directly pointed out.

1.5. Structure of this Thesis

This thesis is structured as follows. First, we will give an overview on image segmentation techniques that have been developed in the field of natural image segmentation in chapter 2. These methods are relevant since they formalize general concepts that, with few adaptations, can be applied to three-dimensional data. Then, we will describe the image formation process of the microscopic data we use in this thesis in chapter 3. Chapter 4 is devoted to the description of methods for the blind deconvolution of three-dimensional recordings, where we contributed two regularization methods for the reconstructed kernel. In chapter 5, we will describe our active surface models with the data terms designed for the challenges in fluorescence microscopic data. In chapter 6, we will present our hierarchical Markov Random Field for the learning of topological information for image segmentation before we will finally conclude in chapter 7.

1.6. Mathematical Notation

$\mathbb{N}$	the space of natural numbers
$\mathbb{R}$	the space of real numbers
$\mathbb{C}$	the space of complex numbers
$\Omega \subseteq \mathbb{R}^d$	image domain in d-dimensional space
$\Omega_1 \sqcup \Omega_2$	union over open regions Ω_1 and Ω_2
$\Omega_1 \cup \Omega_2$	union over closed regions Ω_1 and Ω_2
$\mathbf{x}$	d-tuple of real coordinates
$\Gamma : [0,1] \to \Omega \subseteq \mathbb{R}^2$	a contour in $\mathbb{R}^2$
$\lvert\Gamma\rvert$	the length of contour Γ
$\kappa(s)$	the curvature of a contour Γ at position $s \in [0,1]$
$\mathbf{\Gamma} : [0,1] \times [0,1] \to \Omega \subseteq \mathbb{R}^3$	a contour in $\mathbb{R}^3$
$\mathbf{v}^T$	transposed vector $\mathbf{v}$
$\nabla I = \left(\frac{\partial I}{\partial x_1}, \ldots, \frac{\partial I}{\partial x_d}\right)^T$	the gradient operator on $I : \mathbb{R}^d \to \mathbb{R}$
$\operatorname{div}(I) = \frac{\partial I}{\partial x_1} + \cdots + \frac{\partial I}{\partial x_d}$	the divergence operator on $I : \mathbb{R}^d \to \mathbb{R}$
$\Delta I = \frac{\partial^2 I}{\partial x_1^2} + \cdots + \frac{\partial^2 I}{\partial x_d^2}$	the Laplace operator on $I : \mathbb{R}^d \to \mathbb{R}$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	undirected graphical model
$\mathcal{V} = \{v_1, \ldots, v_{\lvert\mathcal{V}\rvert}\}$	set of nodes
$\lvert\mathcal{V}\rvert$	cardinality of the set of nodes
$\mathcal{N}$	neighborhood system
$\mathcal{X}$	probability space
$\boldsymbol{\ell} \in \mathcal{X}$	label configuration
ℓ_i	label of vertex v_i
$\mathcal{C}$	set of all cliques in graph $\mathcal{G}$
$\Psi(\boldsymbol{\ell})$	potential functions
$\Phi : \mathbb{R}^d \to \mathbb{R}$	d-dimensional embedding function
$\langle \cdot, \cdot \rangle$	inner product
G_σ	a Gaussian normal distribution with standard deviation σ
$*$	the convolution operator
$p(\cdot)$	probability density function
$H(\cdot)$	Heavyside function
δ	the Dirac impulse
$\mathrm{TV}(\cdot)$	the TV norm
$\lVert \cdot \rVert$	the L_2 norm of a vector
$\lvert \cdot \rvert$	the absolute value
$\Pi_{\mathcal{S}}(\cdot)$	projection onto the set $\mathcal{S}$
$o : \Omega \to \mathbb{R}$	the objective function
$s : \Omega \to \mathbb{R}$	the specimen function
$h : \Omega \to \mathbb{R}$	the point spread function
$n : \mathbb{R} \to \mathbb{R}$	a voxel-wise noise function
n_G	zero mean Gaussian noise function

$\hat{f}$	estimate of function f
$f^m(\mathbf{x}) = f(-\mathbf{x})$	the mirrored function f
$\mathcal{F}(\cdot)$	the Fourier transform
$\mathcal{F}^{-1}(\cdot)$	the inverse Fourier transform
f'	derivative of a differentiable function f
x^*	the complex conjugate of $x \in \mathbb{C}$
J_0	zero order Bessel function of the first kind
$\imath$	the imaginary unit
P_l^m	associated Legendre polynomial of order m and degree l
Y_l^m	spherical harmonic basis function of order m and degree l

2. Overview on Image Segmentation

In this chapter, we give an overview on existing state-of-the-art segmentation methods in the field of natural image segmentation. Although these methods were initially proposed for the segmentation in two-dimensional images, they can mostly very easily be adapted to the three-dimensional case. Here, we are listing a large variety of methods, that range from discrete graphical models over continuously defined active contours to superpixel segmentation methods. These methods are in one or the other way the basis for our own segmentation methods that will be presented in chapters 5 and 6. Our methods for the segmentation of cells, cell nuclei, and nucleoli from volumetric recordings (see chapter 5) are based on parametric active contour models, which we will describe in sections 2.3.1-2.3.4. For the segmentation of natural images, these methods are widely believed to be outperformed by convex, region-based methods (see sections 2.3.6 and 2.3.8). On our microscopic data, we can, however, show that the use of edge-based, parametric active surfaces is beneficial. Discrete optimization of graphical models (see section 2.2) and a superpixel segmentation method (section 2.4.1) are important steps in our segmentation with hierarchical Markov Random Fields (chapter 6).

Image segmentation methods generally aim at subdividing an image into segments according to some predefined rules. For example, these segments could correspond to whole objects visible in the image, to object parts or just to regions of homogeneous color or texture. If the objective is to segment brighter from darker regions in an image, threshold-based methods can succeed. Otsu's threshold [Otsu, 1979] for example segments the image such that the gray-value variation is minimal inside the resulting regions whereas it is maximal between the regions. Obviously, these methods can not cope with strong background variations, i.e. high intra-class variations and they have to fail as soon as the same gray-value can belong to foreground and background. Segmentation problems involving noisy data and more complex structures to segment - for example whole objects - thus need a more evolved formulation of the objective.

It follows that a good object segmentation can only be achieved if sufficient prior knowledge about the underlying object is given. Otherwise, due to missing information in the recorded data, object segmentation has to be considered as an ill-posed problem. The most commonly used property is the continuity and smoothness of the object's contour, but more specific, priorly learned shape knowledge can be included in order to improve segmentation results [Leventon et al., 2000; Cootes et al., 2001; Cremers et al., 2002; Cremers, 2006; Brox et al., 2011]. Specially in the field of natural image segmentation, where one has to deal with 2D projections of 3D objects and object occlusions, this approach is very helpful. The prerequisite is that the shape variation inside one object class is limited or can be modeled (e.g. by poselets as done in [Brox et al., 2011]), and a sufficiently large number of fully labeled training examples is given, which is not always the case - especially when dealing with volumetric data.

An objective function that has often been considered in literature is the Mumford-Shah functional [Mumford and Shah, 1989]. Its minimization has been the aim of many later works in image segmentation and image denoising. In the following, we will briefly sketch the Mumford-Shah

functional (2.1). Then, we will describe how its discrete counterpart, the Potts model, has been optimized in the context of Markov Random Fields (section 2.2). Continuous deformable models will be described in section 2.3. The last section of this chapter is dedicated to describing superpixel segmentation methods. Superpixels can be used as basis for higher-level segmentation methods.

2.1. The Mumford-Shah Functional

Mumford and Shah introduced energy formulations for the image denoising and image segmentation problem, assuming that the image domain consists of segments that are separated by a contour $\Gamma : [0,1] \rightarrow \Omega \subseteq \mathbb{R}^2$ placed on the image. The first, most general energy proposed in [Mumford and Shah, 1989] is minimal if the image is decomposed into regions or segments such that it varies smoothly within each segment and there are strong variations between two segments. For this purpose, a function u is introduced that approximates the original image I defined on $\Omega \subseteq \mathbb{R}^2$, is differentiable within the disjoint, open regions Ω_i with $\Omega = \Omega_1 \sqcup \cdots \sqcup \Omega_n \sqcup \Gamma$, and may be discontinuous across the boundary Γ. Additionally, the resulting boundary length $|\Gamma|$ should be small. The functional is given by [Mumford and Shah, 1989]

$$E^{\text{Mumford-Shah}}(u,\Gamma) = \underbrace{\mu^2 \int_{\mathbf{x}\in\Omega} (u-I)^2 d\mathbf{x}}_{E_{\text{data}}(u)} + \underbrace{\int_{\mathbf{x}\in\Omega\setminus\Gamma} \|\nabla u\|^2 d\mathbf{x} + \nu|\Gamma|}_{E_{\text{int}}(u,\Gamma)}, \tag{2.1}$$

where μ and ν are constant weighting factors for the data driven energy term E_{data} and the *internal* energy term E_{int} depending only on the segmentation itself. ∇ is the nabla operator with $\nabla u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}\right)^T$. The optimal result given by the pair (u,Γ) is a smooth version of the original image I, where only the segment boundaries are drawn sharply [Mumford and Shah, 1989]. The resulting u is a smooth image with sharply drawn edges which is often considered the objective of the image denoising problem. The objective has, however, also been used for image segmentation as for example [Cremers et al., 2002]. A first approach to find approximate solutions to the functional (2.1) by variational methods has been proposed in [Ambrosio and Tortorelli, 1990]. In [Felzenszwalb and Huttenlocher, 1998], an objective similar to the one formulated in equation (2.1) is solved based on a graph-theoretic approach for image segmentation.
In [Mumford and Shah, 1989], the authors proposed two further energy functionals that are basically variations of $E^{\text{Mumford-Shah}}$ with a special choice for u, namely the piecewise constant model $E_0^{\text{Mumford-Shah}}$ and the geodesic model $E_\infty^{\text{Mumford-Shah}}$. The piecewise constant model is given by

$$E_0^{\text{Mumford-Shah}}(u,\Gamma) = \sum_{i=1}^{n} \int_{\mathbf{x}\in\Omega_i} \left(u_{\Omega_i} - I\right)^2 d\mathbf{x} + \nu_0|\Gamma|, \tag{2.2}$$

where $u_{\Omega_i} = \text{mean}_{\Omega_i}(I)$ should take on the mean value of the image I inside each open region Ω_i. The constant ν_0 is defined as $\nu_0 = \nu/\mu$. This functional behaves similarly to the discrete Potts model defined in equation 2.14 (see section 2.2.6), and is a typical objective for image segmentation problems. The optimal result is a "cartoon" of the original image, which is why this functional is also referred to as the *cartoon limit*. An important difference between the Mumford-Shah functional and the Potts model is the boundary length $|\Gamma|$ that is continuously modeled in the Mumford-Shah functional but can only take on discrete values in the Potts model.

The geodesic functional is given by

$$E_{\infty}^{\text{Mumford-Shah}}(\Gamma) = \int_0^1 \left(\nu_{\infty} - \left(\frac{\partial I}{\partial \mathbf{n}(s)} \left(\Gamma(s) \right) \right)^2 \right) ds, \tag{2.3}$$

where ν_{∞} is constant, $\mathbf{n}$ is the unit normal to Γ, and ds is an infinitesimal part of the curve Γ. This last functional given in [Mumford and Shah, 1989] is independent from u and seeks for a contour Γ that is as short as possible while, perpendicular to Γ, the image I varies most strongly. Obviously, this energy is minimal if Γ is the empty set, which, however, is not the desired solution of the image segmentation problem.
The optimization of these objective functions has been the basis of various discrete and continuous segmentation methods. Discrete methods can be theoretically embedded into the Markov Random Field framework whereas continuous optimization is done using variational approaches.

2.2. Discrete Graphical Models

Probabilistic graphical models are widely used to represent dependencies between sets of discrete random variables. In these graphical models, the nodes represent random variables and the edges encode their mutual dependencies. The most commonly used graphical models are, among others, Bayesian Networks and Markov Random Fields (MRFs). Bayesian Networks are directed acyclic graphs which means that there exists an underlying ordering of the nodes and the edges are leading from *parent* nodes to *child* nodes. As a consequence, nodes are conditionally independent of each other given their parents [Grenander and Miller, 2007]. Markov Random Fields are undirected graphical models and thus provide handling for Markov processes where no natural ordering of the random variables is given. Thus MRFs are a suitable tool for image processing tasks like image denoising or segmentation, where one usually wants to model the dependencies between neighboring pixels or voxels on a grid. They are the basis for many discrete image segmentation methods. In the following, we will give a brief introduction on MRFs. For a more detailed description, refer to [Winkler, 2003; Bishop, 2009] or to the original work of [Geman and Geman, 1984] in the context of image processing.

2.2.1. Markov Random Fields

A MRF is based on an undirected graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a discrete set of nodes with cardinality $|\mathcal{V}|$ and $\mathcal{E} \subseteq \{(v_i, v_j) | v_i, v_j \in \mathcal{V}, v_i \neq v_j\}$ with $|\mathcal{E}| \leq \binom{|\mathcal{V}|}{2}$ is a set of edges, i.e. of pairs of those nodes. These edges define a *neighborhood system* $\mathcal{N}$ that fulfills the following properties

- $\forall v_i \in \mathcal{V}$: $v_i \notin \mathcal{N}(v_i)$
- $\forall v_i, v_j \in \mathcal{V}$: $v_j \in \mathcal{N}(v_i) \iff v_i \in \mathcal{N}(v_j)$
- $v_j \in \mathcal{N}(v_i) \implies \exists e \in \mathcal{E}$ that connects v_i and v_j.

Each node $v_i \in \mathcal{V}$ is assigned a random variable X_i out of a finite set of random variables $\mathbf{X} = \{X_0, \ldots, X_{|\mathcal{V}|}\}$ in the probability space $\mathcal{X}$. In undirected graphical models, the joint probability

distribution factorizes over the maximal cliques of the graph

$$P(\mathbf{X} = \boldsymbol{\ell}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \Psi_c(\boldsymbol{\ell}), \tag{2.4}$$

where $\boldsymbol{\ell}$ is the configuration of $\mathbf{X}$, the $\Psi_c(\boldsymbol{\ell})$ are *potential functions* that only depend on those components of $\boldsymbol{\ell}$ that are in the clique c. The set of all cliques in $\mathcal{G}$ is denoted by $\mathcal{C}$, and Z is a normalization constant given by

$$Z = \sum_{\boldsymbol{\ell} \in \mathcal{X}} \prod_{c \in \mathcal{C}} \Psi_c(\boldsymbol{\ell}), \tag{2.5}$$

that is called *partition function*. In contrast to other probabilistic models, in MRFs the conditional probabilities depend only on the local neighborhoods. $\mathbf{X}$ defines a MRF on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if the joint probability distribution $P(\mathbf{X} = \boldsymbol{\ell})$ is positive on $\mathcal{X}$ and the so-called *local characteristics* $P(X_i = \ell_i | X_j = \ell_j, j \neq i)$ fulfill the local Markov property:

- $P(\mathbf{X} = \boldsymbol{\ell}) > 0$ for all possible configurations $\boldsymbol{\ell} \in \mathcal{X}$
- $P(X_i = \ell_i | X_j = \ell_j, i \neq j) = P(X_i = \ell_i | X_j = \ell_j, v_j \in \mathcal{N}(v_i))$.

A so defined MRF is specified by specifying all its local characteristics which is very complex and in most cases not possible in practice [Winkler, 2003]. It is the equivalence of MRFs to Gibbs distributions that provides a handling for specifying MRFs.

2.2.2. Gibbs Distributions

A *Gibbs distribution* is a probability distribution on the probability space $\mathcal{X}$ with

$$P(\mathbf{X} = \boldsymbol{\ell}) = \frac{1}{Z} e^{-U(\boldsymbol{\ell})/T}, \tag{2.6}$$

where the *partition function* is $Z = \sum_{\boldsymbol{\ell} \in \mathcal{X}} e^{-U(\boldsymbol{\ell})/T}$ and $U : \mathcal{X} \to \mathbb{R}$ is called *energy function*. The constant T controls the "width" of the probability distribution $P(\mathbf{X})$. In a Gibbs distribution $P(\mathbf{X})$ with respect to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the energy function U is of the form

$$U(\boldsymbol{\ell}) = \sum_{c \in \mathcal{C}} \Psi_c(\boldsymbol{\ell}). \tag{2.7}$$

$\mathcal{C}$ denotes the set of all cliques in $\mathcal{G}$, and the $\Psi_c : \mathcal{X} \to \mathbb{R}$ depend only of those components ℓ_i of $\boldsymbol{\ell}$ that are in c. The equivalence of MRFs to Gibbs distributions has been proved in several works (see e.g. [Grenander and Miller, 2007] and the references given there). Given this equivalence, it is possible to specify an MRF by specifying potentials $U : \mathcal{X} \to \mathbb{R}$.

2.2.3. Image Segmentation with MRFs

In image segmentation tasks, one usually considers nodes $v_i \in \mathcal{V}$ of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ representing the image pixels. The edges $e \in \mathcal{E}$ represent the mutual dependencies between two neighboring pixels. The image is thus described by a two dimensional lattice structure, in most cases with a four- or eight-pixel neighborhood. The extension to a three dimensional lattice structure for the

segmentation of volumetric data with six-, 18- or 26-voxel neighborhood is straightforward. In this framework, the segmentation is a labeling task. The true label configuration $\boldsymbol{\ell}$ is hidden and the aim is to estimate the ℓ_i for each node v_i, depending on a given observable feature vector $\mathbf{y}_i$ and the local neighborhood. A simple example for $\mathbf{y}_i$ would be the color information at position i. The energy function U for the $n \times n$ lattice of nodes $v_i \in \mathcal{V}$ for $1 \leq i, j \leq n^2$ with a four-pixel neighborhood is of the form

$$U(\boldsymbol{\ell}, \mathbf{y}) = \sum_{1 \leq i \leq n^2} \Psi_{\{i\}}(\mathbf{y}_i, \ell_i) \quad + \quad \sum_{\substack{1 \leq i,j \leq n^2 \\ v_j \in \mathcal{N}(v_i)}} \Psi_{\{i,j\}}(\ell_i, \ell_j) \tag{2.8}$$

The functions $\Psi_{\{i\}}(\mathbf{y}_i, \ell_i)$ are the node potentials and determine the influence of the observable data on the labeling, the $\Psi_{\{i,j\}}(\ell_i, \ell_j)$ are called edge potentials and determine the influence of the local neighborhood. The use of higher order potential functions is appealing, because they can model more complex dependencies. In recent years, ternary potential functions $\Psi_{\{i,j,k\}}(\ell_i, \ell_j, \ell_k)$ have come to be used in practice [Komodakis and Paragios, 2009; Andres et al., 2011].

2.2.4. MAP Estimation

The quality of an estimate $\hat{\boldsymbol{\ell}}$ for a labeling $\boldsymbol{\ell}$ given the feature vectors $\mathbf{y}_i$ for $1 \leq i \leq n^2$, and the local neighborhood system can be measured in different ways. A measure that many popular methods (e.g. [Boykov and Kolmogorov, 2004; Juan and Boykov, 2006; Kohli and Torr, 2007; Delong and Boykov, 2008]) seek to maximize is the maximum a posteriori probability (MAP) of the estimate

$$\hat{\boldsymbol{\ell}} = \operatorname*{argmax}_{\boldsymbol{\ell}} \prod_{i=1}^{n^2} \big(P(\ell_i|\mathbf{y}_i)\big) = \operatorname*{argmax}_{\boldsymbol{\ell}} \prod_{i=1}^{n^2} \left(\frac{P(\mathbf{y}_i|\ell_i)P(\ell_i)}{P(\mathbf{y}_i)} \right). \tag{2.9}$$

$P(\mathbf{y}_i)$ does not depend on ℓ_i and can be neglected for the maximization. The maximization of the MAP probability $P(\boldsymbol{\ell}|\mathbf{y})$ is equivalent to maximizing the energy $U(\boldsymbol{\ell}, \mathbf{y})$ [Winkler, 2003] and to minimizing the negative log-likelihood which is $-U(\boldsymbol{\ell}, \mathbf{y})$. Other estimates that have been used as an alternative to the MAP criterion are for example the Marginal Posterior Mode (MPME), the Posterior Minimum Mean Squares (MMSE) or the iterated conditional modes (ICM) [Besag, 1986] estimate. In [Winkler, 2003], some advantages and disadvantages of the different estimates are listed.

The exact inference of the label configuration $\boldsymbol{\ell}$ is not tractable in general, because of the computation of the partition function Z. In the worst case, the number of possible configurations can be growing exponentially with the number of nodes such that the computation of Z is an NP-hard problem. There are many algorithms that generate approximate solutions based on random sampling, like for example the Gibbs sampler (for generating an MMSE estimate) or Simulated Annealing [Geman and Geman, 1984]. Those are iterative methods, starting with an initial configuration $\boldsymbol{\ell}_0$, where in every sampling step only one of the random variables X_i can change its state. Message passing algorithms on grid-structured graphs like the Loopy Belief Propagation are deterministic, but there is, depending on the model, no guarantee that the algorithm converges to a good solution or even converges at all [Bishop, 2009]. While we can thus only find approximate solutions for the NP-hard multi-label problems, binary labeling tasks can be efficiently solved. For the special case of binary labeling problems (i.e. foreground-background segmentation), the *minimum cut* or *Graph Cut* algorithm [Greig et al., 1989] can generate the exact MAP estimation in polynomial time. In the following, we will briefly describe the basics of this algorithm, before we give in 2.2.6 more details on the multi-label problem and its approximate solutions.

2.2.5. Two-Label Image Segmentation and Minimal Cuts

The *minimum cut* (Min-Cut) or *Graph Cut* algorithm has been proposed by [Greig et al., 1989] for the reconstruction of binary images with priors similar to the famous *Ising model*. The Ising model was introduced in 1925 by E. Ising to explain ferromagnetism [Winkler, 2003], but it has also been studied in the field of binary image segmentation (e.g in [Geman and Geman, 1984]). In this model the configuration space $\mathcal{X}$ is binary, i.e. $\ell_i \in \{+1, -1\}$, homogeneous, and isotropic. The energy function U is given by

$$U(\boldsymbol{\ell}, \lambda) = \alpha \sum_{1 \leq i \leq n^2} \lambda_i \ell_i + \beta \sum_{\substack{1 \leq i,j \leq n^2 \\ v_j \in \mathcal{N}(v_i)}} \ell_i \ell_j \tag{2.10}$$

for constant parameters $\alpha > 0$, weighting the influence of the external field λ, and $\beta > 0$ encouraging labels of neighboring nodes to take on the same value. Maximizing this second term corresponds to minimizing the boundary length measured as $\sum_{\substack{1 \leq i,j \leq n^2 \\ v_j \in \mathcal{N}(v_i)}} (\ell_i - \ell_j)^2$. The optimal solution will therefore be biased towards boundaries in the directions of the image grid (grid bias).

The formulation of the MAP estimation for this binary labeling problem as *minimum cut* problem allowed Greig et al. [1989] to use the Ford-Fulkerson algorithm [Ford and Fulkerson, 1962] to find the exact solution in polynomial time [Winkler, 2003]. To do so, the MAP estimation is formulated as minimization of the negative log-likelihood $-\bar{U}$ that can be translated to a maximum flow problem.
Let the energy $-\bar{U}$ be defined for $\ell_i \in \{0, 1\}$ for $1 \leq i \leq n^2$ as

$$-\bar{U}(\boldsymbol{\ell}, \mathbf{w}) = \sum_{1 \leq i \leq n^2} w_i(\ell_i) + \sum_{1 \leq i,j \leq n^2} w_{ij}\left(\ell_i, \ell_j\right). \tag{2.11}$$

$\mathbf{w}$ consists of unary terms w_i that correspond to the external data and binary terms w_{ij} defining the interactions between neighboring vertices. Here, the unary terms w_i are defined with $w_i = \ln\left(P(\mathbf{y}_i|X_i = 0)/P(\mathbf{y}_i|X_i = 1)\right)$ and the binary terms are given as $w_{ij} > 0$ if $v_j \in \mathcal{N}(v_i)$ and $w_{ij} = 0$ otherwise. Note that for $w_i = -\alpha\lambda_i$ and $w_{ij}(\ell_i, \ell_j) = -\beta$ if $v_j \in \mathcal{N}(v_i)$ and $\ell_i = \ell_j$, $w_{ij}(\ell_i, \ell_j) = 0$ otherwise, the expressions (2.10) and (2.11) are equivalent.
Generally, it can be shown that binary labeling problems can be formulated as minimum cut problems if and only if the pairwise terms are *regular*, i.e *submodular* [Kolmogorov and Zabih, 2004]:

$$w_{ij}(1, 1) + w_{ij}(0, 0) \leq w_{ij}(1, 0) + w_{ij}(0, 1). \tag{2.12}$$

For these energies, a *flow* or *capacity network* can be defined. This *capacity network* is a directed graph with $n^2 + 2$ vertices, one vertex v_i for each of the n^2 image pixels and two additional vertices: a source s and a sink t. The influence of the external data, i.e. the unary potentials w_i, is encoded in these additional vertices: There are directed edges from s to every vertex v_i with the respective capacity $c_{si} = w_i$ and directed edges from every vertex v_i to the sink t with capacity $c_{it} = -w_i$. As before, two neighboring vertices v_i and v_j are connected by an undirected edge with capacity $c_{ij} = w_{ij}$. In this network, the task is now to determine the maximal flow that can pass from source to sink. As stated in [Ford and Fulkerson, 1962], the maximal flow is equivalent to the minimal cut separating source from sink, where a cut is defined as

$$\text{cut}\left(C \cup \{s\}, \bar{C} \cup \{t\}\right) = \sum_{v_i \in \bar{C}} \max(0, w_i) + \sum_{v_i \in C} \max(0, -w_i) + \sum_{v_i \in C, v_j \in \bar{C}} w_{ij}, \tag{2.13}$$

where $C \cup \bar{C} = \mathcal{V}$.
There have been several algorithms proposed in literature to find the minimal cut with growing efficiency: beginning with augmenting paths algorithms [Ford and Fulkerson, 1962; Dinic, 1970; Greig et al., 1989] and the push-relabel algorithm of [Goldberg and Tarjan, 1988] to more efficient algorithms that make use of the images' grid structure like [Boykov and Kolmogorov, 2004; Juan and Boykov, 2006; Kohli and Torr, 2007; Delong and Boykov, 2008] or use primal-dual problem formulations like [Komodakis et al., 2007] just to name few. In [Szeliski et al., 2008] an overview and comparison of different optimization methods in terms of performance and run-time is given.
While the binary labeling problem can be efficiently solved, this is, as we have mentioned before, not the case for the practically much more relevant problem of multi-label segmentation. In the following, we will briefly discuss the challenges that arise from multi-label problems and describe how the solutions found for the binary labeling task can be employed in order to find good solutions for the multi-label problem.

2.2.6. Multi-Label Problems

Multi-object segmentation corresponds to finding a partitioning of $\mathcal{V}$ into disjoint subsets $C_1, \ldots, C_k$. In order to model multi-object segmentations, we must allow the X_i to take on more than two states, i.e. $\ell_i \in \{1, \ldots, m\}$, where m is the number of objects. The natural generalization of the Ising model (see section 2.2.5) to more than two states is the *Potts model* [Potts, 1952].
In the *Potts model*, the energy has the form

$$U(\boldsymbol{\ell}, \mathbf{w}) = \sum_{1 \leq i \leq n^2} w_i \ell_i + \sum_{\substack{1 \leq i,j \leq n^2 \\ v_j \in \mathcal{N}(v_i)}} w_{ij} \delta_{\ell_i, \ell_j}, \tag{2.14}$$

where δ_{ℓ_i,ℓ_j} is the Kronecker symbol, i.e. $\delta_{\ell_i,\ell_j} = 1$ if $\ell_i = \ell_j$, $\delta_{\ell_i,\ell_j} = 0$ otherwise. In the standard Potts model, discontinuities between states are equally penalized regardless of the states, i.e. $w_{ij} > 0$ is constant. This model was first used in [Geman et al., 1990] in the context of image processing.
To allow for more flexibility in the choice of the energy to be optimized than offered by the Potts model, we are considering energies of the form

$$U(\boldsymbol{\ell}, \mathbf{w}) = \sum_{1 \leq i \leq n^2} w_i \ell_i + \sum_{\substack{1 \leq i,j \leq n^2 \\ v_j \in \mathcal{N}(v_i)}} w_{ij}(\ell_i, \ell_j), \tag{2.15}$$

where w_{ij} is a potential function depending on the states ℓ_i and ℓ_j. Energies of this form can account for the fact that some labels are more likely to co-occur than others.
In figure 2.1, an example of this multi-label problem is given. The already mentioned sampling methods like the Gibbs sampler and ICM as well as Loopy Belief Propagation can, among others, be used to find approximate solutions for this multi-object segmentation problem. However, there is no efficient method producing an exact solution in general as the problem is NP-hard [Boykov et al., 2001]. Here, the term *super-/submodularity* comes into play. A multi-label function is called *submodular* [Kolmogorov and Rother, 2007] if the pairwise terms satisfy

$$w_{ij}(a, b) + w_{ij}(c, c) \leq w_{ij}(a, c) + w_{ij}(c, b) \tag{2.16}$$

for all states or labels $a, b, c \in \{1 \ldots m\}$. If w_{ij} is submodular, $-w_{ij}$ is *supermodular*. The minimization of submodular energies (and the maximization of supermodular energies) of an ordered

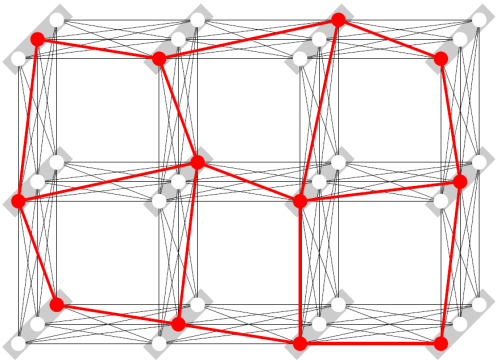


Figure 2.1.: Multi-label graph structure corresponding to the energy given in (2.15) for a 3×4 image grid and three labels. An example labeling is marked in red. The illustration is taken from [Werner, 2007].

set of labels can be done in polynomial time [Ishikawa, 2003; Kolmogorov and Zabih, 2004]. However, as pointed out in Kolmogorov and Zabih [2004], the available algorithms are still very slow. Furthermore, an ordering of the labels can not easily be provided for many applications as for example image segmentation. These facts explain the popularity of approximation methods.
An approximation method that is widely used for the MAP maximization of multi-label MRFs with supermodular energies are *Graph cuts with α-expansions* [Boykov et al., 2001]. The basic idea is to split the multi-label problem into a series of two-label segmentation problems that are submodular and can be easily solved using the *minimum cut* algorithm (see section 2.2.5). Beginning with an initial labeling, the algorithm iteratively applies so-called *α-expansion* operations. For every label α, the minimum cut is computed, i.e. every variable X_i can keep its old value or switch to the label α. The cut is applied if the energy decreases, rejected otherwise. The algorithm terminates when no α-expansion move leads to an energy decrease anymore. The algorithm has shown fast convergence and has a guaranteed approximation quality [Boykov et al., 2001]. In [Kolmogorov and Rother, 2007], the authors advertise the use of Graph cuts algorithms for the minimization of non-submodular (i.e. the maximization of non-supermodular) energies. The method reviewed in [Kolmogorov and Rother, 2007] for example produces a partial labeling for non-submodular energies.
In recent years, the MAX-SUM solver reviewed in [Werner, 2007] has become very popular. This solver directly tackles the multi-label segmentation and provides a solution for the linear programming relaxation of the MAP maximization of a multi-label MRF. It converges even for energies that are not supermodular [Werner, 2007]. As the problem is still NP-hard, the MAX-SUM solver can of course not provide the exact solution.
Although the *minimum cut* method and its multi-label counterparts are widely used for image segmentation, they do not always lead to the desired segmentation result. One of the major drawbacks of the *minimum cut* is that it actually favors cutting isolated pixels from the image [Wu and Leahy, 1993]. If there is a pixel or a small region with low capacities to its neighbors, the cut separating this small region will be extremely cheap, because the cost increases with the number of edges to be cut. This is of cause also depending on the unary terms of the graph. The problem will not arise for nodes with strong unary costs. If , however, there are no strong affinity of a node to a certain label, one has to carefully choose the edge capacities in order to avoid such "cost-free" cuts [Wu and Leahy, 1993]. There have been several methods proposed to solve this problem on a graph-

theoretic basis as for example the algorithms proposed in [Hagen and Kahng, 1992; Shi and Malik, 2000; Ding et al., 2001]. The next section is devoted to these balanced cuts.

2.2.7. Balanced Cuts

In general, it is not well defined what the ideal segmentation algorithm should do. However, there is an agreement that the algorithm should not have a preference for cutting off small areas corresponding to small sets of isolated nodes in the graph. Generally, there are two objectives pursued in literature: either, the segmentation is preferred to be independent of the size of the resulting regions. This is the case for *Ratio cuts* that have been proposed in [Hagen and Kahng, 1992] (see section 2.2.7.1) and for *Normalized cuts* [Shi and Malik, 2000] (see section 2.2.7.2). Or, the resulting pixel subsets are preferred to be similar in size. This is the case for *Min-max cuts* (see section 2.2.7.3) [Ding et al., 2001] as well as for *Ratio Cheeger cuts* and for *Normalized Cheeger cuts* [Buehler and Hein, 2009; Hein and Buehler, 2010; Szlam and Bresson, 2010]. Unfortunately, finding the exact solution of these cuts is NP-complete [Shi and Malik, 2000; Ding et al., 2001; Hein and Buehler, 2010]. This puts these balanced cuts in the context of stochastic relaxation methods and spectral clustering (see section 2.2.7.4).

2.2.7.1. Ratio Cuts

For a partitioning of the graph nodes into two disjoint subsets C and $\bar{C}$ with $C \cup \bar{C} = \mathcal{V}$ where we have no preference on which set should be foreground and which should be background, let a cut be defined as

$$\mathrm{cut}(C, \bar{C}) = \sum_{v_i \in C, v_j \in \bar{C}} w_{ij}. \tag{2.17}$$

The *Ratio cut* [Hagen and Kahng, 1992] is then defined as

$$\mathrm{RCut}(C, \bar{C}) = \frac{\mathrm{cut}(C, \bar{C})}{|C|} + \frac{\mathrm{cut}(C, \bar{C})}{|\bar{C}|}, \tag{2.18}$$

i.e. the *Ratio cut* value is normalized by the cardinality of the subsets. As a consequence, the global optimum of the *Ratio cut* is independent of the size of the resulting partitions. As pointed out by [Shi and Malik, 2000], it is also independent of the similarities within the resulting groups C and $\bar{C}$. However, for image segmentation, we usually want to optimize not only for low inter group similarity but also for high intra group similarity. As shown in [Shi and Malik, 2000], these properties are fulfilled by the *Normalized cut*.

2.2.7.2. Normalized Cuts

The *Normalized cut* [Shi and Malik, 2000] is defined as

$$\mathrm{NCut}(C, \bar{C}) = \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(C)} + \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(\bar{C})}, \tag{2.19}$$

where

$$\mathrm{vol}(C) = \sum_{v_i \in C v_j \in \mathcal{V}} w_{ij} \tag{2.20}$$

denotes the volume of C. The volume of a subset C describes the total connection from nodes in C to all nodes in $\mathcal{V}$. This normalization leads to cuts with high intra-class similarity and high

inter-class dissimilarity. *Normalized cuts* are an important step in one of the most popular state of the art segmentation methods [Arbeláez et al., 2009, 2011]. There it helps to introduce global image knowledge into the segmentation. We will describe the method of [Arbeláez et al., 2009, 2011] with some more details in section 2.4.1.

2.2.7.3. Min-Max Cuts

The *Min-Max cut* [Ding et al., 2001] is defined as

$$\mathrm{MCut}(C, \bar{C}) = \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(C) - \mathrm{cut}(C, \bar{C})} + \frac{\mathrm{cut}(C, \bar{C})}{\mathrm{vol}(\bar{C}) - \mathrm{cut}(C, \bar{C})}. \tag{2.21}$$

As shown in [Ding et al., 2001], the normalization used in *Min-max cut* favors cuts that yield equally large partitions. This is different from the *Ratio cut* and *Normalized cut* objectives, that do not have a preference for the cluster sizes. Ding et al. [2001] argue that this is generally an important property, as it prevents unstable results. For image segmentation tasks, however, it depends on the concrete application if the assumption holds that the desired image partitions have the same size.

Ratio cuts, *Normalized cuts* and *Min-Max cuts* can be estimated using spectral clustering algorithms. In the following, we will give a brief introduction to the spectral clustering algorithms presented in [Hagen and Kahng, 1992; Shi and Malik, 2000; Ding et al., 2001]. For a detailed introduction to spectral clustering, please refer to [von Luxburg, 2007].

2.2.7.4. Spectral Clustering

The basic idea of spectral clustering is to make use of the spectrum of a matrix encoding the graph structure of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in an appropriate way. Let us go back to the two-label segmentation problem in equation (2.11) and assume all unary terms $w_i = 1$. The according *adjacency matrix* W is a $n^2 \times n^2$ matrix with $W(i,j) = w_{ij}$ for $i, j = 1 \dots n^2$. Note that by definition of the binary terms $w_{ij} = w_{ji}$, i.e. W is symmetric. The *degree* of a vertex $v_i \in \mathcal{V}$ is defined as $d_i = \sum_{j=1}^{n^2} w_{ij}$. The diagonal $n^2 \times n^2$ matrix D of all d_i is called *degree matrix*. Then $L = D - W$ is called the *graph Laplacian*. The graph Laplacian L and its normalized versions are the basis of spectral clustering. L has n^2 non-negative eigenvalues that will be denoted by $\lambda_1 \dots \lambda_{n^2}$. The smallest strictly positive eigenvalue of L is called the *spectral gap*. Since the number of eigenvalues $\lambda_i = 0$ equals the number of connected components in the graph [von Luxburg, 2007], in fully connected graphs (like this should be the case for images) the spectral gap is given by λ_2. Now, we come to the connection between spectral clustering and *Ratio cuts*, *Normalized cuts* and *Min-Max cuts*.
The optimal partitioning according to the *Ratio cut* criterion is given by the Rayleigh quotient [von Luxburg, 2007]

$$\min_{\boldsymbol{\ell}} \mathrm{RCut}(\boldsymbol{\ell}) = \min_{\mathbf{y}} \frac{\mathbf{y}^T (D - W)\mathbf{y}}{\mathbf{y}^T \mathbf{y}}, \tag{2.22}$$

and the optimal partitioning according to the *Normalized cut* criterion by

$$\min_{\boldsymbol{\ell}} \mathrm{NCut}(\boldsymbol{\ell}) = \min_{\mathbf{y}} \frac{\mathbf{y}^T (D - W)\mathbf{y}}{\mathbf{y}^T D \mathbf{y}}. \tag{2.23}$$

Here, $\boldsymbol{\ell}$ is an indicator vector with $\ell_i = 1$ if node i belongs to the subset C and -1 otherwise and $\mathbf{y} = (\mathbf{1} + \boldsymbol{\ell}) - \frac{k}{1-k}(\mathbf{1} - \boldsymbol{\ell})$, where k is depending on the degree matrix D and defined as $k = \frac{\sum_{\ell_i > 0} d_i}{\sum_i d_i}$. If the y_i are relaxed, i.e. if we allow $y_i \in \mathbb{R}$, equations (2.22) and (2.23) can be minimized by solving the eigenvalue system of the *graph Laplacian* L and the *normalized graph Laplacian* $D^{-1/2}LD^{-1/2}$ respectively [Chung, 1997]:

$$(D - W)\mathbf{y} = \lambda \mathbf{y} \tag{2.24}$$

$$D^{-1/2}(D - W)D^{-1/2}\mathbf{y} = \lambda \mathbf{y}. \tag{2.25}$$

The eigenvector $\mathbf{v}^{(2)}$ to the second smallest eigenvalue λ_2 (the spectral gap) is the global optimizer for the relaxed problem (2.22) and (2.23) respectively [von Luxburg, 2007]. For multi-label problems, i.e. for a k-way partitioning of the graph, it can be shown [von Luxburg, 2007] that the relaxed solution is given by the k eigenvectors to the k smallest non-zero eigenvalues.
The *Min-Max cut* can also be expressed in terms of the Rayleigh quotient [Ding et al., 2001] as

$$\min_{\boldsymbol{\ell}} \mathrm{MCut}(\boldsymbol{\ell}) = \min_{\mathbf{y}} \frac{J_N(\mathbf{y})}{1 - J_N(\mathbf{y})/2} \Rightarrow \min_{\mathbf{y}} J_N(\mathbf{y}) \tag{2.26}$$

where

$$J_N(\mathbf{y}) = \frac{\mathbf{y}^T(D - W)\mathbf{y}}{\mathbf{y}^T D \mathbf{y}}. \tag{2.27}$$

As a result, the solution of the relaxed *Min-Max cut* problem is given by the same eigenvalue system of the *normalized graph Laplacian* (equation (2.25)) as it is for the *Normalized cut*.
Obviously, the solution of the relaxed problem does not directly induce a partitioning of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, since the values of the eigenvector $\mathbf{v}^{(2)}$ tend to vary smoothly. A solution can for example be found by thresholding $\mathbf{v}^{(2)}$ such that the thresholding result minimizes the respective cut objective [Ding et al., 2001; Shi and Malik, 2000]. For multi-label problems, [Hagen and Kahng, 1992; Shi and Malik, 2000] propose two possibilities. Either, the graph can be recursively partitioned until the desired number of regions is reached, or one can perform a k-means clustering on the k eigenvectors belonging to the k smallest non-zero eigenvalues in order to partition the graph into k segments.

2.3. Active Contour Models

Active (or deformable) contours are a widely used concept for image segmentation. The basic idea of active contours is that a contour $\Gamma : [0,1] \to \Omega \subseteq \mathbb{R}^2$ placed on the image is drawn towards the object's contour by image and contour forces. Most often, these forces optimize a previously defined energy that is typically of the form

$$E(\Gamma) = E_{\text{data}}(\Gamma) + E_{\text{int}}(\Gamma), \tag{2.28}$$

where $E_{\text{data}}(\Gamma)$ is the *data term*, that is minimal if Γ lies exactly on the object contours in the image, and where $E_{\text{int}}(\Gamma)$ is the *internal* contour term that imposes some prior constraints on Γ. While MRFs formulate the segmentation problem inside a probabilistic framework, continuous active contour models start directly with the formulation of the energy to be minimized. The Mumford-Shah functional (see section 2.1) has often been the basis for the active contour energy formulation.
The number of publications in the field of deformable contours is so large that an extensive description is beyond the scope of this thesis and we can only give a brief overview. We start with parametric active contours, i.e. contours that are explicitely represented by a function Γ (section 2.3.1-2.3.4). Those parametric active contours are mostly edge driven (2.3.1-2.3.3) and hence related to the functional (2.3) proposed by Mumford and Shah. An implicitly represented edge driven active contour model, Geodesic Active Contours, will be briefly described in 2.3.4. The implicit representation of active contours by embedding functions, however, has the advantage that it can naturally incorporate regional information in the contour evolution such that they are eligible for the optimization of the cartoon model (2.2). Region based active contours will be described in section 2.3.5 - 2.3.8 for two-label and multi-label segmentation tasks.

2.3.1. Parametric Active Contours – Snakes

Parametric active contours or *snakes* were first proposed by Kass et al. [1988]. In their work, the contour Γ is represented by a spline. Splines are functions that are composed of piecewise concatenated polynomial functions. Linear splines, for example, consist of concatenated line segments that are represented by control points. Higher order splines like quadratic or cubic splines are piecewise composed of quadratic and cubic polynomials respectively. At the connection points of these higher order splines, one usually claims smooth transitions. For example cubic C^2 splines are twice differentiable [de Boor, 1978].
In [Kass et al., 1988], the *controlled continuity* framework from Terzopoulos [1986] is used to enforce smooth curves. This regularization is a generalization of the quadratic Tikhonov regularization [Tikhonov and Arsenin, 1987] that we will also discuss in the context of image deblurring (see section 4.6.2). The internal energy of the contour γ parameterized by $s \in [0,1]$ is given by

$$E_{\text{int}}(\Gamma) = \frac{1}{2}\int_0^1 \alpha \left|\frac{\partial \Gamma}{\partial s}(s)\right|^2 + \beta \left|\frac{\partial \Gamma^2}{\partial^2 s}(s)\right|^2 ds, \tag{2.29}$$

where α and β are steering the influence of the first and second order terms that are controlling the stretching and the bending of the contour Γ respectively.
For the data term $E_{\text{data}}(\Gamma)$, Kass et al. [1988] propose three different functionals, drawing the contour to lines, edges and terminations in the image

$$E_{\text{data}}(\Gamma) = w_{\text{line}} E_{\text{line}}(\Gamma) + w_{\text{edge}} E_{\text{edge}}(\Gamma) + w_{\text{term}} E_{\text{term}}(\Gamma). \tag{2.30}$$

For the functional attracting the contour to lines, the authors propose to use the image intensity itself with $E_{\text{line}}(\Gamma) = \int_0^1 I(\Gamma(s))ds$. The sign of the weighting parameter w_{line} determines whether the contour is attracted to dark or to bright lines.
The edge functional is maybe the most commonly used. Its objective is related to (2.3). Assuming the contour Γ should be attracted to large image gradients, it is natural to set

$$E_{\text{edge}}(\Gamma) = \int_0^1 -\left\|\nabla I\left(\Gamma(s)\right)\right\|^2 ds. \tag{2.31}$$

In order to increase the distance from which the snake is attracted to the desired contour (the so-called *capture range*), Kass et al. [1988] propose a scale-space approach where the snake is first drawn towards blurred image edges. The result is used as initialization on less strongly blurred image edges, and so on.
The termination functional aims for pulling the contour to line terminations, using the curvature of level lines. The curvature of level lines is given by

$$E_{\text{term}}(\Gamma) = \int_0^1 \frac{\frac{\partial^2 I}{\partial x_2^2}\left(\frac{\partial I}{\partial x_1}\right)^2\left(\Gamma(s)\right) - 2\frac{\partial^2 I}{\partial x_1 \partial x_2}\frac{\partial I}{\partial x_1}\frac{\partial I}{\partial x_2}\left(\Gamma(s)\right) + \frac{\partial^2 I}{\partial x_1^2}\left(\frac{\partial I}{\partial x_2}\right)^2\left(\Gamma(s)\right)}{\left(\frac{\partial I}{\partial x_1}^2\left(\Gamma(s)\right) + \frac{\partial I}{\partial x_2}^2\left(\Gamma(s)\right)\right)^{3/2}} ds, \tag{2.32}$$

where x_1 and x_2 are the components of the coordinate vector $\mathbf{x}$. Usually, this functional is computed on a smoothed image [Kass et al., 1988]. The minimization of the energy $E(\Gamma)$ from (2.28) is done using the calculus of variations. For minima of $E(\Gamma)$, the first variation must vanish, i.e. $\frac{\partial E(\Gamma+\epsilon\bar{\Gamma})}{\partial \epsilon}\Big|_{\epsilon=0} = 0$ for any scalar ϵ close to zero and test function $\bar{\Gamma}$ (test contour). This leads to a Euler-Lagrange equation that can be considered as a force balance system [Xu and Prince, 1998]:

$$\mathbf{F}_{\text{data}} + \mathbf{F}_{\text{int}} = 0. \tag{2.33}$$

For example, for a contour that should be attracted to image edges, i.e. $E(\Gamma) = E_{\text{edge}}(\Gamma) + E_{\text{int}}(\Gamma)$, the Euler-Lagrange equation is given by

$$-\nabla\left\|\nabla I\left(\Gamma(s)\right)\right\|^2 - \alpha\frac{\partial^2\Gamma}{\partial s^2}(s) + \beta\frac{\partial^4\Gamma}{\partial s^4}(s) = 0. \tag{2.34}$$

A minimum can be found by gradient descent. A time variable t is introduced and one assumes that at convergence, the contour Γ does not change anymore over time, i.e. $\frac{\partial\Gamma}{\partial t} = 0$.
One of the disadvantages of snakes is that the global optimum of the energy functional is actually the empty set with a contour length $|\Gamma| = 0$. Therefore, one hopes to find the local minimum corresponding to the desired solution. As a consequence, the snakes need to be initialized closely to the sought image contour. A further problem is that, even if the snake is initialized sufficiently close to the object contour, it may get caught in isolated points with strong gradients in the image that are caused by noise, or, if the forces are too strong, it can jump over the sought contour, i.e. it easily gets caught in a local minimum that is not the desired solution.
One of the big advantages of snakes is that the data term $E_{\text{data}}(\Gamma)$ can easily be modified and furthermore, one can even directly manipulate the resulting data force. This idea was also the basis for *Balloon Snakes* [Cohen, 1991] and *Gradient Vector Flow (GVF) snakes* [Xu and Prince, 1997, 1998], that tackle many of the aforementioned problems.

2.3.2. Balloon Snakes

An extension to the original snake method of Kass et al. [1988] are *balloons* that have been proposed by Cohen [1991]. In [Cohen, 1991] two specific problems of the snake model that are addressed are listed:

- *"If the curve is not close enough to an edge, it is not attracted by it."*
- *"If the curve is not submitted to any forces, it shrinks on itself."*

In order to alleviate these problems, the author proposes to modify the data forces, that were originally defined as $\mathbf{F}_{\text{data}} = \nabla \left\|\nabla I\right\|^2$. The proposed data forces are

$$\mathbf{F}_{\text{data}}^{\text{balloon}}(s) = k_1 \mathbf{n}(s) + k \frac{\nabla \left\|\nabla I\right\|^2}{\left\|\nabla \left\|\nabla I\right\|^2\right\|} \left(\Gamma(s)\right), \tag{2.35}$$

where the k_1 and k are constant weighting parameters. The first term is a force with amplitude k_1 that pulls the curve in the direction of its unit normal vector $\mathbf{n}$. The effect of this force is thus an *inflation* of the curve that counteracts the shrinking effect of the internal contour force. The second term is the normalized original data force. The normalization ensures that the curve moved according to the force does not eventually "jump" over the sought object contour, i.e. it ensures that the steps are small enough. Since the shrinking force is curvature dependent, the choice of k and k_1 is critical. In general, it is important that k is chosen larger than k_1. Otherwise, the edge forces are not able to stop the curve at the object contour.

2.3.3. Gradient Vector Flow

In [Xu and Prince, 1997, 1998], the authors propose to compute a vector diffusion on the gradient field, called *gradient vector flow* (GVF) in order to produce data force fields for snakes $\mathbf{F}_{\text{data}}$. They show that, compared to directly using the gradients $\nabla \left|\nabla I(\Gamma(s))\right|^2$, GVF increases the capture range and makes it possible to drive the contours into concavities. Additionally the contours are less prone to noise.
GVF aims for providing a smooth vector field in regions where no edges are given in the data, and for keeping the original gradient information in regions with strong edges. Thus, when the vector field is used as an external force field e.g. for active contours, the capture range is much larger than it would be, if the original gradient information was used.
The two dimensional gradient vector flow field is the vector field $\mathbf{v} : \Omega \to \mathbb{R}^2, \mathbf{v}(\mathbf{x}) = \left(u(\mathbf{x}), v(\mathbf{x})\right)^T$ that minimizes the energy functional

$$E^{\text{GVF}}(\mathbf{v}) = \int_\Omega \mu(\|\nabla \mathbf{v}\|^2) + \|\nabla M\|^2 \|\mathbf{v} - \nabla M\|^2 d\mathbf{x}, \tag{2.36}$$

where the first term $\mu(\|\nabla \mathbf{v}\|^2) = \mu\left(\|\nabla u(\mathbf{x})\|^2 + \|\nabla v(\mathbf{x})\|^2\right)$ determines the smoothness of the vector field and the second term is minimal for

$$\mathbf{v} = \nabla M, \tag{2.37}$$

i.e. it preserves the original information for strong gradients. M denotes an edge map and is typically set to $M = \left|\nabla I\right|$. The parameter μ regularizes the trade-off between the first and the

second term. The energy is minimized by the solution of the following Euler-Lagrange equations:

$$\mu\nabla^2 u - \left(u - \frac{\partial M}{\partial x_1}\right)\left(\left(\frac{\partial M}{\partial x_1}\right)^2 + \left(\frac{\partial M}{\partial x_2}\right)^2\right) = 0$$
$$\mu\nabla^2 v - \left(v - \frac{\partial M}{\partial x_2}\right)\left(\left(\frac{\partial M}{\partial x_1}\right)^2 + \left(\frac{\partial M}{\partial x_2}\right)^2\right) = 0 \tag{2.38}$$

The solution can be found by gradient descent.

2.3.4. Geodesic Active Contours

In [Caselles et al., 1997], the authors aim at finding minimal distance paths based on an image induced metric. The *geodesic active contour* model is derived from the following energy functional that is related to the edge based functional given by Kass et al. [1988]:

$$E_1^{\text{GAC}}(\Gamma) = \alpha \int_0^1 \left|\frac{\partial \Gamma}{\partial s}(s)\right|^2 ds - \nu \int_0^1 \left\|\nabla I\left(\Gamma(s)\right)\right\| ds \tag{2.39}$$

with constant weighting factors α and ν, that is optimized by a geodesic curve in a Riemannian space induced from the image I [Caselles et al., 1997]. The minimization of equation (2.39) is, for some strictly decreasing function $g : [0, +\infty[\to \mathbb{R}^+$ with $g(x) \to 0$ for $x \to +\infty$, equivalent to minimizing [Caselles et al., 1997]

$$E_2^{\text{GAC}}(\Gamma) = \int_0^1 g\left(\left\|\nabla I\left(\Gamma(s)\right)\right\|\right)\left|\frac{\partial \Gamma}{\partial s}(s)\right| ds, \tag{2.40}$$

As for the original snakes model, the global optimum of this energy is the empty set with a contour with zero length. This is not a desired result. A local minimum of equation (2.40) can be found by gradient descent. The Euler-Lagrange equation is given by

$$g\left(\left\|\nabla I\left(\Gamma(s)\right)\right\|\right)\kappa(s)\mathbf{n}(s) - \left\langle \nabla g\left(\left\|\nabla I\left(\Gamma(s)\right)\right\|\right), \mathbf{n}(s)\right\rangle \mathbf{n}(s) = 0, \tag{2.41}$$

where $\mathbf{n}(s)$ is the unit normal of the contour Γ in s, κ is the curvature, and $\langle .\,, .\rangle$ denotes the inner product.

In [Caselles et al., 1997], the optimization of the contour is computed using a *level set* representation. The paper thus connects the snake model (see section 2.3.1) to level sets that had been previously introduced by Osher and Sethian [1988] in the context of modeling solid-liquid interfaces and are thus based on the theory of curve evolution.

Level sets provide a handling for implicitly representing contours using so-called *embedding functions*. A contour Γ in a two dimensional image I is represented by a 2D embedding function $\Phi : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}$, such that the actual contour is given by the zero level set of the embedding function $\Gamma = \{\mathbf{x} \in \Omega \subseteq \mathbb{R}^2 | \Phi(\mathbf{x}) = 0\}$. For $\Phi(\mathbf{x}) > 0$, the position $\mathbf{x}$ is inside the segmented region, for $\Phi(\mathbf{x}) < 0$ the position $\mathbf{x}$ is outside the segmented region. The contour Γ can be evolved by evolving the embedding function Φ over time, introducing an additional time variable t. Then, the contour at time t is given by the zero-level set of $\Phi(t, \mathbf{x})$.

The movement of the contour Γ in the direction of its normal vector $\frac{\partial \Gamma(s,t)}{\partial t} = k_1 \mathbf{n}(s)$ (compare equation (2.35)) corresponds to a front propagation movement of the embedding function (a proof is given in [Caselles et al., 1997]). The front propagation movement with curvature dependent speed is given by the following differential equation [Osher and Sethian, 1988]

$$\frac{\partial \Phi}{\partial t} = \|\nabla \Phi\| \text{div}\left(\frac{\nabla \Phi}{\|\nabla \Phi\|}\right), \tag{2.42}$$

where $\nabla\Phi$ denotes only the derivatives in the spatial directions and $\kappa = \text{div}\left(\frac{\nabla \Phi}{\|\nabla \Phi\|}\right)$ is the *mean curvature* of the contour. The mean curvature dependent front propagation, the so-called *mean curvature motion*, moves the contour in its normal direction and thus reduces its length. The initial contour is defined by the set $\{\mathbf{x}|\Phi(0,\mathbf{x}) = 0\}$. Level sets have also been used before in the context of image segmentation. For example the *geometric active contours* proposed in [Caselles et al., 1993] and the model proposed in [Malladi et al., 1995] have a similar objective as the later, more evolved geodesic active contours, but in these works, the problem not defined as an energy minimization.
Compared to parametric active contours, the implicit representation of level sets has the advantage that topological changes can be naturally handled. A modeling of region properties of the segmented object and background is very easy since Φ represents the whole enclosed and exclosed region. Furthermore, the extension to higher dimensions is straightforward since there is no underlying control point structure that would have to be translated to higher dimensions.
For the contour Γ represented by the embedding function Φ, the gradient descent resulting from equation (2.41) is given by

$$\begin{aligned} \frac{\partial \Phi}{\partial t} &= \|\nabla \Phi\| \text{div}\left(g\left(\left\|\nabla I\right\|\right)\frac{\nabla \Phi}{\|\nabla \Phi\|}\right) \\ &= g\left(\left\|\nabla I\right\|\right)\|\nabla \Phi\| \text{div}\left(\frac{\nabla \Phi}{\|\nabla \Phi\|}\right) + \left\langle \nabla g\left(\left\|\nabla I\right\|\right), \nabla \Phi \right\rangle. \end{aligned} \tag{2.43}$$

The approach is not limited to a certain edge detector function $g\left(\left\|\nabla I\right\|\right)$ but can be used with any function $g(I)$ that stops the contour at the desired boundary. This function could for example be defined as

$$g(I) = \frac{1}{1 + \left|\nabla (G_\sigma * I)\right|^p}, \tag{2.44}$$

where $p = 1$ or 2, G_σ is a Gaussian normal distribution with standard deviation σ and $*$ is the convolution operator. The concept of geodesic length minimization has been used in many works on image segmentation [Goldenberg et al., 2001; Paragios et al., 2001; Paragios and Deriche, 1999a,b, 2000]. For example in the works of Paragios and Deriche [1999a,b, 2000], *Geodesic Active Regions* have been proposed, combining geodesic active contours with region-based contour evolution for image segmentation and motion tracking.

2.3.5. Region-based Active Contours

The level set method described in the above section does not use any regional information in order to perform a segmentation, but only rely on boundary information. Paragios and Deriche [1999a,b,

2000] combine the edge-based *geodesic active contours* with a region term, in which the region properties are modeled, as proposed before in [Zhu and Yuille, 1996], by the conditional probabilities $p_1(I(\mathbf{x})) = p(I(\mathbf{x})|\mathbf{x} \in \Omega_1)$ and $p_2(I(\mathbf{x})) = p(I(\mathbf{x})|\mathbf{x} \in \Omega_2)$, where Ω_1 and Ω_2 are the two regions separated by Γ with $\Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_1 \cap \Omega_2 = \emptyset$. The resulting *geodesic active regions* are defined by the objective function [Paragios and Deriche, 1999a]

$$\begin{aligned} E^{\text{GAR}}(\Gamma) &= (1-\alpha)E_2^{\text{GAC}}(\Gamma) - \\ &\quad \alpha\left(\int_{\Omega_1} \log\left(p_1(I(\mathbf{x}))\right) d\mathbf{x} + \int_{\Omega_2} \log\left(p_2(I(\mathbf{x}))\right) d\mathbf{x}\right). \end{aligned} \tag{2.45}$$

Actually, in the original papers [Paragios and Deriche, 1999a,b, 2000], the edge terms in $E_2^{\text{GAC}}(\Gamma)$ are formulated in a general way by a *conditional boundary density function* $g\left(p_B\left(I\left(\Gamma\right)\right)\right) = g\left(p\left(I\left(\mathbf{x}\right)|\mathbf{x} \in B\right)\right)$ instead of $g\left(\left\|\nabla I\left(\Gamma(s)\right)\right\|\right)$, measuring the conditional probability of an observed image pixel intensity given the pixel lies on the correct boundary B. The gradient-based edge indicator function $\left\|\nabla I\left(\Gamma(s)\right)\right\|$ is a common choice for $p_B\left(I\left(\Gamma\right)\right)$. The first term in equation (2.45) thus measures the geodesic length for an edge detector function. The second term of E^{GAR} measures the negative log-likelihood of an observed image I given the regions Ω_1 and Ω_2. In [Paragios and Deriche, 1999a,b, 2000], all probabilities are described using multivariate Gaussian normal distributions. The minimization on the energy E^{GAR} is done using gradient descent. The Euler-Lagrange equation computed from E^{GAR} is

$$\begin{aligned} (1-\alpha)\left(g\left(\left\|\nabla I\left(\Gamma(s)\right)\right\|\right)\kappa(s)\mathbf{n}(s) - \left\langle \nabla g\left(\left\|\nabla I\left(\Gamma(s)\right)\right\|\right), \mathbf{n}(s)\right\rangle \mathbf{n}(s)\right) + \\ \alpha\left(\log\left(\frac{p_2(I(\Gamma))}{p_1(I(\Gamma))}(s)\right)\mathbf{n}(s)\right) = 0, \end{aligned} \tag{2.46}$$

where κ is the curvature of Γ and $\mathbf{n}$ is the unit normal pointing towards region Ω_1. The first part of this equation equals the Euler-Lagrange equation derived from E_2^{GAC} given in equation (2.41) weighted by $1-\alpha$. As for geodesic active contours, the representation of the contour is changed to a level set function Φ for the curve evolution.

A purely region-based active contour method based on level sets with a simpler region model has been proposed in [Chan and Vese, 2001]. They formulate the region-based energy functional with the help of the *Heavyside function*

$$H : \mathbb{R} \to \{0,1\}, \qquad H(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.47}$$

as

$$\begin{aligned} E^{\text{region-based}}\left(\Phi\right) &= \int_\Omega \|\nabla H\left(\Phi(\mathbf{x})\right)\| d\mathbf{x} + \lambda \int_\Omega |H\left(\Phi(\mathbf{x})\right)| d\mathbf{x} \\ &\quad + \nu_1 \int_\Omega \left(I(\mathbf{x}) - \mu_1\right)^2 H\left(\Phi\left(\mathbf{x}\right)\right) d\mathbf{x} \\ &\quad + \nu_2 \int_\Omega \left(I(\mathbf{x}) - \mu_2\right)^2 \left(1 - H\left(\Phi\left(\mathbf{x}\right)\right)\right) d\mathbf{x}, \end{aligned} \tag{2.48}$$

where $\frac{\partial H}{\partial x} = \delta$ is the Dirac impulse such that $\int_\Omega \|\nabla H\left(\Phi(\mathbf{x})\right)\| \, d\mathbf{x}$ is the contour length, $\int_\Omega |H\left(\Phi(\mathbf{x})\right)| \, d\mathbf{x}$ measures the size of the region inside the contour and actually results in a shrinking force. This term is mostly neglected in practice. λ, ν_1, and ν_2 are constant weighting factors, and μ_1 and μ_2 are the averages of I inside and outside the contour respectively:

$$\mu_1(\Phi) = \frac{\int_\Omega I(\mathbf{x}) H\left(\Phi\left(\mathbf{x}\right)\right) d\mathbf{x}}{\int_\Omega H\left(\Phi\left(\mathbf{x}\right)\right) d\mathbf{x}} \tag{2.49}$$

and

$$\mu_2(\Phi) = \frac{\int_\Omega I(\mathbf{x}) \left(1 - H\left(\Phi\left(\mathbf{x}\right)\right)\right) d\mathbf{x}}{\int_\Omega \left(1 - H\left(\Phi\left(\mathbf{x}\right)\right)\right) d\mathbf{x}}. \tag{2.50}$$

The energy $E^{\text{region-based}}$ corresponds to the cartoon limit of the Mumford-Shah energy functional (2.2). The minimization is done using gradient descent. The Euler-Lagrange equation is deduced assuming fixed μ_1 and μ_2 and replacing H in equation (2.48) by a C^2 regularized function H_ϵ with $\delta_\epsilon = \frac{\partial H_\epsilon}{\partial x}$. $\delta_\epsilon(\Phi)$ takes on positive values closely around the region boundary and is zero everywhere else. The resulting Euler-Lagrange equation is

$$\delta_\epsilon(\Phi) \left(\text{div}\left(\frac{\nabla\Phi}{\|\nabla\Phi\|}\right) - \nu_1(I-\mu_1)^2 + \nu_2(I-\mu_2)^2 - \lambda \right) = 0. \tag{2.51}$$

In [Chan and Vese, 2001], the average values μ_1 and μ_2 are updated as the contour evolves and the gradient descent update on the evolving contour only acts locally. The result thus strongly depends on the initialization.

2.3.6. Convex Relaxation

For fixed values μ_1 and μ_2 it is possible to formulate the region-based energy minimization as a convex optimization problem using convex relaxation [Chan et al., 2006] and the *total variation* (TV) norm. The TV norm of a function $u : \Omega \to \mathbb{R}$ is generally defined as

$$\text{TV}(u) = \sup_{v \in \mathcal{D}^{\text{TV}}} \int_\Omega u \, \text{div} v \, d\mathbf{x} \tag{2.52}$$

with the dual variable v from

$$\mathcal{D}^{\text{TV}} = \{v \in C^1(\Omega)^2 | \quad \|v(\mathbf{x})\| \leq 1 \quad \forall \mathbf{x} \in \Omega\}. \tag{2.53}$$

For continuously differentiable functions, the TV norm equals

$$\text{TV}(u) = \int_\Omega \|\nabla u\| d\mathbf{x}. \tag{2.54}$$

original image initialization 50 iterations 100 iterations final result

Figure 2.2.: Result of the convex region-based segmentation. The mean values $\mu 1$ and $\mu 2$ were set to 0.2 and 0.55 (on a scale of from 0 to 1) respectively, λ was set to 0.017.

According to the *coarea* formula, the TV norm of a function with bounded variation (i.e. $\mathrm{TV}(u) < \infty$) equals the sum of the contour lengths of all regions that can be generated by thresholding u [Strang, 1983]. The TV norm of functions with bounded variation is thus a measure for the contour length.
To allow for a convex formulation of the region-based two-label segmentation problem, the level set function Φ is replaced in [Chan et al., 2006] by a binary function $u : \Omega \to \{0, 1\}$ which makes the *Heavyside function* dispensable. The energy then reads

$$E^{\text{convex}}(u) = \int_\Omega \|\nabla u\| \, d\mathbf{x} + \nu \int_\Omega \Big(\big(I - \mu_1\big)^2 - \big(I - \mu_2\big)^2 \Big) \, u \, d\mathbf{x}, \tag{2.55}$$

where the TV norm measures the contour length and ν is a constant weighting factor. This restriction of u to binary functions is then relaxed to the interval $[0, 1]$ to make the energy convex. The according Euler-Lagrange equation is

$$\operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|} \right) - \nu \Big((I - \mu_1)^2 - (I - \mu_2)^2 \Big) = 0. \tag{2.56}$$

The condition that $0 \leq u(\mathbf{x}) \leq 1$ is important because otherwise the function u would tend to $+\infty$ whenever it is positive and to $-\infty$ whenever it is negative [Chan et al., 2006]. The condition can for example be kept by reprojecting u to the valid interval after every iteration. Since the energy is convex, the globally optimal relaxed function u can be found by gradient descent. Faster implementations can be done using a primal-dual optimization scheme [Chambolle, 2004; Aujol and Chambolle, 2005]. An alternative is to linearize the equation with a *lagged diffusivity* fixed point iteration scheme [Vogel and Oman, 1996]. For the resulting linear system, the optimization can be done efficiently using Successive over-relaxation as for example in [Klodt et al., 2008]. The solution of the original segmentation problem can be found by thresholding the optimal u. In this very special case, all thresholds between zero and one lead to a globally optimal solution of the original segmentation problem [Chan et al., 2006]. Recall that the labeling problem for two labels could also be solved exactly using discrete optimization (see section 2.2.5). An advantage of the continuous optimization is that the contour length is modeled accurately and there is no metrication error (i.e. no grid bias) in the segmentation result [Klodt et al., 2008; Cremers et al., 2011]. An example of a segmentation result with the convex method of Chan et al. [2006] on a natural gray value image is displayed in figure 2.2. The optimization was done as described in [Klodt et al., 2008] with a lagged diffusivity scheme and Successive over-relaxation.
Based on this approach of Chan et al. [2006], Bresson et al. [2005, 2007] have proposed a convex relaxation method for the optimization of snakes and geodesic active contours. Instead of the TV norm as used in [Chan et al., 2006], a weighted TV norm is used in [Bresson et al., 2005] as a

original image initialization 50 iterations 100 iterations final result

Figure 2.3.: Result of the convex geodesic active contour segmentation. The mean values $\mu 1$ and $\mu 2$ were set to 0.2 and 0.55 (on a scale from 0 to 1) respectively, λ was set to 2.5.

measure of the geodesic contour length. The objective function proposed by Bresson et al. [2005] for two-label image segmentation is given by

$$E^{\text{GAC convex}}(u) = \int_\Omega g(I)\|\nabla u\|\, d\mathbf{x} + \nu \int_\Omega \left(\left(I-\mu_1\right)^2 - \left(I-\mu_2\right)^2 \right) u\, d\mathbf{x}, \tag{2.57}$$

with $0 \leq u(\mathbf{x}) \leq 1$. Here, the weighted TV norm minimizing the geodesic contour length is used in combination with a region-based data term in order to render the geodesic active contour model convex. The functional is thus not strictly convex - its convexity depends on the choice of the parameter $\nu \geq 0$. The functional corresponds to the piece-wise constant Mumford-Shah functional (equation 2.2). The minimization can again be done using gradient descent. The Euler-Lagrange equation is

$$\text{div}\left(g(I)\frac{\nabla u}{\|\nabla u\|} \right) - \nu \left((I-\mu_1)^2 - (I-\mu_2)^2 \right) = 0. \tag{2.58}$$

A primal-dual optimization scheme for the above functional has been proposed in [Bresson et al., 2007]. A result of the convex geodesic active contour segmentation with the piece-wise constant data assumption implemented with a lagged diffusivity scheme and Successive over-relaxation is displayed in figure 2.3. The contour length is weighted by $g(I) = \dfrac{1}{\sqrt{\left(\nabla(G_\sigma * I)\right)^2 + 1}}$.
For the model proposed in [Bresson et al., 2005], the convexification of the snake energy is done by adding a region term which in some way requires that the intensities inside the object differ from the intensities outside. For natural images, this is usually no problem. However, when looking at microscopic images with for example a membrane staining, this constraint can not always be met.

2.3.7. Multi-Label Segmentation with Level Sets

Different implementations of so-called *multi-phase* level sets have been proposed for example in the works of [Zhao et al., 1996; Vese and Chan, 2002; Paragios and Deriche, 2002; Chung and Vese, 2005; Brox and Weickert, 2006; Lellmann and Schnörr, 2011; Pock et al., 2009; Chambolle et al., 2012]. The extension to multiple regions is not straightforward. One of the major challenges for the multi-label level set segmentation is that one must ensure that every image pixel is attributed to exactly one region, i.e. $\Omega_1 \cup \cdots \cup \Omega_N = \Omega$ and $\Omega_i \cap \Omega_j = \emptyset$ for all $i \neq j$. For binary partitioning problems, this condition is always fulfilled. For more regions, different approaches to solve the problem have been proposed.
The work of Vese and Chan [2002], builds upon the binary region-based level set segmentation [Chan and Vese, 2001] presented in the previous section (section 2.3.5). The image is recursively

subdivided by level set functions such that the segmentation of an image into $N = 2^M$ regions is computed using M level set functions. This is a suitable concept if the number of desired regions is a power of two. Every point is attributed to exactly one region automatically. Unfortunately, this representation entails further problems if the desired number of regions is not a power of two. Then, empty regions have to be generated. An additional disadvantage is that, during the evolution, label transfers are only possible between certain labels.
Chung and Vese [2005] propose to generate a multi-label segmentation by evaluating several level lines of one level set function. Unfortunately, some cases can not be modeled this way, as for example triple junctions [Chung and Vese, 2005]. In this case, a further level set function has to be used for further subdivision. For example when using two level set functions with two levels each, nine regions can be generated with this approach.
In the approaches of Zhao et al. [1996], Paragios and Deriche [2002] and Brox and Weickert [2006], one level set function is used for each of the N regions to be generated. This entails the problem that pixels can either be attributed to multiple regions, to one region, or to none of the regions.
Here, we omit the details on the different solutions to this problem that have been proposed in the papers cited above and come directly to the more recent convex relaxation methods for multi-label segmentation problems. In the non-convex models, the segmentation result is heavily depending on the initialization. No guarantee can be given for the quality of the segmentation result. However, even for multi-label problems where finding the exact segmentation is known to be NP-hard (compare section 2.2.6), it is possible to provide solutions within a bound of the optimal solution using convex relaxation [Cremers et al., 2011].

2.3.8. Convex Relaxation Methods for Multi-Phase Level Sets

Convex relaxation methods for multi-phase level sets have been proposed for example in [Zach et al., 2008; Pock et al., 2009; Cremers et al., 2011; Chambolle et al., 2012; Lellmann et al., 2009; Lellmann and Schnörr, 2011]. A quite general formulation can be found in [Lellmann and Schnörr, 2011].
Recall that for the two-label case, the convex relaxation methods are based on the TV norm of the indicator function u. In order to measure the length of the contours represented by an indicator function $u : \Omega \to \{1, \dots, n\}$ where $u(\mathbf{x}) = \ell$ if and only if $\mathbf{x} \in \Omega_\ell$, the TV norm has to be transferred to the multi-label case.
The perhaps most straightforward extension of the convex relaxation methods seen before (section 2.3.6) to the multi-label case was proposed in [Zach et al., 2008]. The indicator function u is represented by a vector of binary indicator functions u_i with $\sum_{i=1}^n u_i = 1$ such that $u(\mathbf{x}) = \ell \Leftrightarrow u_\ell(\mathbf{x}) = 1$. The energy proposed to minimize is

$$E_{\text{Zach}}^{\text{ML convex}} = \int_\Omega \nu \sum_{\ell=1}^{n} \|\nabla u_\ell\| + \big\langle u(\mathbf{x})\,,\, s(\mathbf{x})\big\rangle \, d\mathbf{x}, \quad \lambda > 0 \tag{2.59}$$

where the data term is linear in u and determined by $s(\mathbf{x}) = (s_1(\mathbf{x}), \dots, s_n(\mathbf{x}))^T$. By analogy to the two label case, the s_i can for example be chosen as $s_i(\mathbf{x}) = (I(\mathbf{x}) - \mu_i)^2$, where μ_i is the mean value of region i.
The u_i are relaxed to the convex set $C := \{u : \Omega \to \mathbb{R}^n | u_i(\mathbf{x}) \geq 0, \sum_{i=1}^n u_i(\mathbf{x}) = 1\}$ to allow for a convex optimization. The final labeling is determined by assigning each point $\mathbf{x}$ the label ℓ of the maximum in $\{u_i(\mathbf{x}) | i = 1, \dots, n\}$.
A different relaxation based on convex constraints on the dual variable is proposed in [Pock et al.,

2009]. In this work, the labeling function u is represented by $n-1$ binary functions $\theta_1(\mathbf{x}), \dots \theta_{n-1}(\mathbf{x})$ where

$$\theta_\ell(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad u(\mathbf{x}) > \ell \\ 0 & \text{otherwise} \end{cases} \tag{2.60}$$

such that the labeling function u can be expressed by $u(\mathbf{x}) = 1 + \sum_{\ell=1}^{n} \theta_\ell(\mathbf{x})$. The objective function in [Pock et al., 2009] is defined based on this label representation and the dual representation of the TV norm

$$\min_{\boldsymbol{\theta}\in\mathcal{B}} \sup_{\boldsymbol{\xi}\in\mathcal{K}} \sum_{\ell=1}^{n-1} -\nu \int_\Omega \theta_\ell \, \mathrm{div}\xi_\ell d\mathbf{x} + \int_\Omega |\theta_\ell(\mathbf{x}) - \theta_{\ell+1}(\mathbf{x})| s_{\ell+1} d\mathbf{x}. \tag{2.61}$$

where the vector $\boldsymbol{\theta}$ of θ_ℓ is from the set

$$\mathcal{B} = \left\{ \boldsymbol{\theta} = (\theta_1, \dots, \theta_{n-1}) | \, \Omega \to \{0,1\}^{n-1} \right\}, \tag{2.62}$$

with $\theta_0 = 1$ for all $\mathbf{x}$, $\theta_n = 0$ for all $\mathbf{x}$, and the set $\mathcal{K}$ is defined as

$$\mathcal{K} = \left\{ \boldsymbol{\xi} = (\xi_1, \dots, \xi_{n-1}) | \, \Omega \to \mathbb{R}^{2\times(n-1)}, \left| \sum_{\ell_1 \le \ell \le \ell_2} \xi_\ell(\mathbf{x}) \right| \le 1, \forall \mathbf{x} \in \Omega, 1 \le \ell_1 \le \ell_2 \le n-1 \right\}. \tag{2.63}$$

The objective function is made convex by relaxing the θ_i to the convex set

$$\mathcal{R} = \left\{ \boldsymbol{\theta} = (\theta_1, \dots \theta_{n-1}) | \, \Omega \to [0,1]^{n-1} \right\}. \tag{2.64}$$

Because of the absolute value, the objective function in (2.61) is not continuously differentiable. For the minimization, an additional dual variable is introduced in [Pock et al., 2009] such that (2.61) transforms to

$$\min_{\boldsymbol{\theta}\in\mathcal{B}} \sup_{\substack{\boldsymbol{\xi}\in\mathcal{K} \\ \boldsymbol{\eta}\in\mathcal{W}}} \sum_{\ell=1}^{n-1} -\nu \int_\Omega \theta_\ell \, \mathrm{div}\xi_\ell d\mathbf{x} + \int_\Omega (\theta_\ell(\mathbf{x}) - \theta_{\ell+1}(\mathbf{x})) \eta_{\ell+1}(\mathbf{x}) d\mathbf{x}. \tag{2.65}$$

where

$$\mathcal{W} = \left\{ \boldsymbol{\eta} = (\eta_1, \dots \eta_n) | \, \Omega \to \mathbb{R}^n, |\eta_\ell(\mathbf{x})| \le s_\ell(\mathbf{x}), \forall \mathbf{x} \in \Omega, 1 \le i \le n \right\}. \tag{2.66}$$

The primal-dual optimization is done by alternating gradient descent in the primal variable $\boldsymbol{\theta}$ and gradient ascent in the dual variables $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$. The resulting update scheme is [1]

$$\begin{aligned} \theta_\ell^{k+1} &= \Pi_{\mathcal{R}} \left(\theta_\ell^k - \tau_p \left(-\nu \mathrm{div}\xi_\ell - \left(\eta_\ell + \eta_{\ell+1} \right) \right) \right) \\ \xi_\ell^{k+1} &= \Pi_{\mathcal{K}} \left(\xi_\ell^k + \tau_d \cdot \nu \left(\nabla \theta_\ell^{k+1} \right) \right) \\ \eta_\ell^{k+1} &= \Pi_{\mathcal{W}} \left(\eta_\ell^k + \tau_d \left(\theta_{\ell-1} - \theta_\ell \right) \right). \end{aligned} \tag{2.67}$$

[1] The listed update scheme differs from the one given in [Pock et al., 2009] in some details, but follows directly from computing the derivatives of (2.65).

The Π denote projections onto the respective convex sets $\mathcal{R}$, $\mathcal{K}$ and $\mathcal{W}$, the τ denote step-widths for the gradient descent. Pock et al. [2009] propose to compute the projection of $\boldsymbol{\xi}$ onto the convex set $\mathcal{K}$ using Dykstra's algorithm [Boyle and Dykstra, 1986], whereas the projections onto $\mathcal{R}$ and $\mathcal{W}$ can be done by truncation.
In the works of [Lellmann et al., 2009; Lellmann and Schnörr, 2011], a vector-valued formulation of the TV norm was proposed. As in [Zach et al., 2008], the indicator function u is represented by a vector of binary indicator functions u_i with $\sum_{i=1}^{n} u_i = 1$ such that $u(\mathbf{x}) = \ell \Leftrightarrow u_\ell(\mathbf{x}) = 1$. These functions are combined with a vector valued TV formulation based on the Euclidean metric [Lellmann et al., 2009; Lellmann and Schnörr, 2011]

$$\mathrm{TV}(u) = \sup_{v \in \mathcal{D}^{\mathrm{TV}}} -\sum_{j=1}^{n} \int_{\Omega} u \,\mathrm{div} v^{(j)} \, d\mathbf{x} \tag{2.68}$$

where $v^{(j)}$ is the j-th column of the $2 \times n$ matrix v from

$$\mathcal{D}^{\mathrm{TV}} = \{v \in C^{\infty}(\Omega)^{2\times n} | \quad \|v(\mathbf{x})\| \leq 1 \quad \forall \mathbf{x} \in \Omega\}, \tag{2.69}$$

where C^k is the space of k-times continuously differentiable functions on Ω and $\|v\|$ is the Frobenius norm of the matrix v with $\|v\| = \sqrt{\sum_{i=1}^{2} \sum_{j=1}^{n} |v_{ij}|^2}$. For continuously differentiable vector-valued functions, the TV norm equals

$$\mathrm{TV}(u) = \int_{\Omega} \sqrt{\|\nabla u_1\|^2 + \cdots + \|\nabla u_n\|^2} d\mathbf{x}. \tag{2.70}$$

The resulting multi-phase level set energy in [Lellmann et al., 2009] is

$$E^{\text{ML convex}}_{\text{Lellmann}} = -\int_{\Omega} \left\langle u(\mathbf{x}) \,,\, s(\mathbf{x}) \right\rangle d\mathbf{x} + \nu \mathrm{TV}(u), \quad \nu > 0, \tag{2.71}$$

where the data term determined by s is defined as before.
This original problem is relaxed in order to compute the optimization on a convex set $C := \{u : \Omega \to \mathbb{R}^n | u_i(\mathbf{x}) \geq 0, \sum_{i=1}^{n+1} u_i(\mathbf{x}) = 1\}$. The Euler-Lagrange equation to equation (2.71) is

$$\frac{\partial E}{\partial u_i} = -s_i - \nu \,\mathrm{div}\left(\frac{\nabla u_i}{\sqrt{\sum_{i=1}^{n} \|\nabla u_i\|^2}} \right) = 0. \tag{2.72}$$

In [Lellmann and Schnörr, 2011], a primal-dual optimization algorithm is proposed to solve this Euler-Lagrange equation. Alternatively, as in the work of [Klodt et al., 2008] for the convex two label segmentation, a lagged diffusivity scheme can be developed for the optimization[2]. We briefly sketch this numerical scheme here, because it is the one used in order to produce the example segmentation in figure 2.4. The source of non-linearity in equation (2.72) is the diffusivity $d = \frac{1}{\sqrt{\sum_{i=1}^{n} \|\nabla u_i\|^2}}$ which is depending on the solution u. It follows that equation (2.72) can be linearized by a fixed point iteration scheme, transforming (2.72) into a sequence of linear equations that can be solved efficiently. The optimization can be done by alternating the computation of the diffusivities for the current u and solving the linear system of equations for fixed diffusivities. As in

[2]The use of this optimization scheme for the energy formulated by [Lellmann et al., 2009; Lellmann and Schnörr, 2011] was suggested by Thomas Brox in his lecture on Computer Vision.

original image | Pock et al. [2009] | Lellmann et al. [2009]

Figure 2.4.: Result of the multi label level set segmentation with different convex relaxation methods and energy formulations. The segmentation has been computed for four labels with fixed mean color values in rgb color space with $\lambda = 25$.

[Klodt et al., 2008], the linear systems of equations can be solved using Successive over-relaxation. The resulting update for the u_i is given by

$$u_i^{k+1}(\mathbf{x}) = (1-\omega)u_i^k(\mathbf{x}) + \omega \frac{\sum\limits_{\substack{\mathbf{y}\in\mathcal{N}(\mathbf{x}) \\ y_1\leq x_1 \wedge y_2 \leq x_2}} d_{\mathbf{x}\sim\mathbf{y}} u_i^{k+1}(\mathbf{y}) + \sum\limits_{\substack{\mathbf{y}\in\mathcal{N}(\mathbf{x}) \\ y_1> x_1 \vee y_2 > x_2}} d_{\mathbf{x}\sim\mathbf{y}} u_i^{k}(\mathbf{y}) - s_i(\mathbf{x})/\nu}{\sum\limits_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} d_{\mathbf{x}\sim\mathbf{y}}}, \tag{2.73}$$

where

$$d_{\mathbf{x}\sim\mathbf{y}} = \frac{d_{\mathbf{x}} + d_{\mathbf{y}}}{2}. \tag{2.74}$$

The diffusivities are computed as

$$d_{\mathbf{x}} = \frac{1}{\sqrt{\sum_{j=1}^{n+1} \|\nabla u_j(\mathbf{x})\|^2 + \epsilon}} \tag{2.75}$$

with a sufficiently large ϵ in order to avoid numerical problems for $\sum_{j=1}^{N+1} \|\nabla u_j(\mathbf{x})\|^2$ close to zero. $\omega \in (0,2)$ is the relaxation parameter. For $\omega = 1$, the scheme in equation (2.73) equals the Gauss-Seidel method. After each iteration, the u_i must be reprojected into the convex set, such that $\sum_{i=1}^{n} u_i(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \Omega$.
From the optimal relaxed u_i a final segmentation can be generated by assigning each point $\mathbf{x}$ the label ℓ of the maximum in $\{u_i(\mathbf{x}) | i = 1, \ldots, n\}$. The reprojection to discrete labels generally does not have the globally optimal energy, but is within bounds of the optimal solution [Lellmann and Schnörr, 2011].

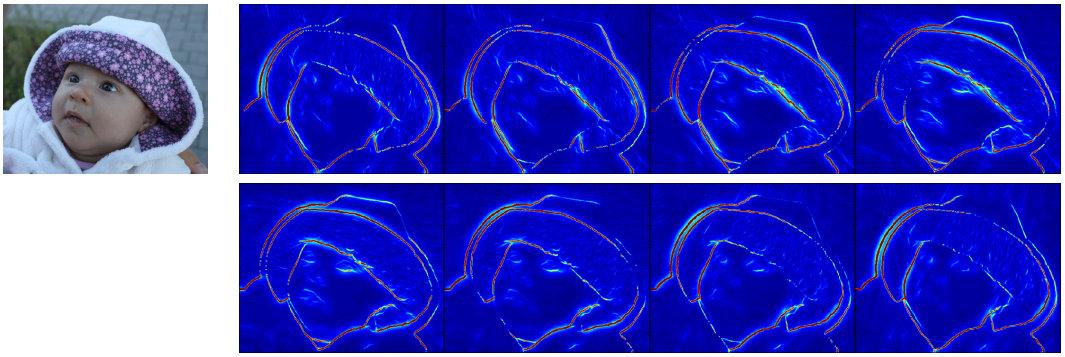
Segmentation results with the convex relaxation methods proposed in [Pock et al., 2009] and [Lellmann et al., 2009] with five labels are shown in figure 2.4. The segmentation with the convex relaxation proposed in [Lellmann et al., 2009] has been implemented using the SOR scheme (equation 2.73). The results are comparable in huge parts of the image. It is however interesting to observe that the eye area is assigned to different labels by the two methods. The difference in the results can be due to the fact that the relaxation used in [Pock et al., 2009] is tighter than the one proposed in [Lellmann et al., 2009].

2.4. Superpixel Generation

For many higher level image processing applications like object detection [Fulkerson et al., 2009], video segmentation [Ochs and Brox, 2011], or semantic image segmentation [Keuper et al., 2011; Yang et al., 2010; Lim et al., 2009; Fulkerson et al., 2009], it is beneficial to group the image pixels into small regions of homogeneous color and texture. This grouping leads to a reduced complexity and the resulting regions build a natural domain for the computation of features [Arbeláez et al., 2011]. These regions are also referred to as *superpixels* [Arbeláez et al., 2011]. There are many methods available for generating such superpixels [Vincent and Soille, 1991; Felzenszwalb and Huttenlocher, 1998; Comaniciu and Meer, 2002; Felzenszwalb and Huttenlocher, 2004; Najman and Schmitt, 1996; Mičušík and Pajdla, 2007; Arbeláez et al., 2011; Liu et al., 2011] or their volumetric analogon [Kaster et al., 2011]. A very simple and robust method from the field of mathematical morphology is the *watershed transform* [Vincent and Soille, 1991; Najman and Schmitt, 1996; Roerdink and Meijster, 2000]. In its basic form, it builds upon the gray-level values of an image that are interpreted as a topographic relief, i.e. the intensity of every point is interpreted as its elevation. Starting from the local minima of this topographic map, the regions are defined by the catchment basins. A review of different algorithms for the computation of the watershed transform is given in [Roerdink and Meijster, 2000].
Various modifications of the watershed transform have been published to make it directly applicable to image segmentation tasks, where whole objects should be segmented from the image. The marker-based watershed transform for example allows for predefined catchment basins [Beucher, 1990; Najman and Schmitt, 1996] such that there is only one watershed region per object. Hierarchical segmentations based on the watershed transform aim at weighting the resulting boundaries with the importance of the contour [Beucher, 1990; Najman and Schmitt, 1996; Arbeláez et al., 2011] such that the desired segmentation can be found by thresholding the weighted contours.
In the following, we will describe two very different algorithms for the generation of superpixels that have been used in some of our own contributions [Keuper et al., 2011; Morath et al., 2013; Keuper et al., 2010b]. The first method *gPb-OWT-UCM* of Arbelaéz et al. [Arbeláez et al., 2011, 2009] has been used for the segmentation in electron microscopic images in our publications [Keuper et al., 2011; Morath et al., 2013]. It combines *Normalized cuts* (see section 2.2.7.2) and a new watershed based algorithm in order to produce region hierarchies. The second method *Mean shift segmentation* of Comaniciu and Meer [2002] only evaluates local image information in order to create regions. The fundamentals of this method have been used in our work on robust gradient vector fields for the segmentation with active surfaces [Keuper et al., 2010b].

2.4.1. gPb-OWT-UCM

The method *gPb-OWT-UCM* proposed by [Arbeláez et al., 2011, 2009] generates a hierarchical segmentation of an image. Contours in higher levels of the resulting hierarchy usually belong to more important structures of the image whereas contours in the lowest level of hierarchy define *superpixels*, i.e. small regions of homogeneous color and texture. These smallest superpixels should contain all visible edge information of the image and are most often smaller than the smallest object of interest in the image.
The first part of the *gPb-OWT-UCM* algorithm is the contour detection that has first been proposed in [Martin et al., 2004]. The contour detection uses brightness, color and texture information: the original image is split into four separate feature channels, where the first three contain the image's brightness and color information and the fourth channel represents the image specific texture in-



original image *gPb*

Figure 2.5.: Result of the *gPb* contour detection for the different orientations θ for an example color image. The results are visualized with a jet colormap, i.e. dark red corresponds to high magnitudes whereas dark blue corresponds to low magnitudes.

formation. For these four channels, oriented gradients are computed: for every image pixel, the circular neighborhood within a certain radius σ is split into two halves at an angle θ. The gradient magnitude for the orientation θ is computed as the χ^2 distance of the histograms of the two half-disks. Multiscale information is gained by linearly combining the gradient responses for spherical pixel neighborhoods with different radii σ. The result is called mPb. The last step of the contour detection is what the authors call *globalization* [Maire et al., 2008]. An adjacency matrix W is built based on the boundary information in *mPb*. For every pair of pixels within a certain neighborhood, $W(i,j)$ is computed from the maximal value of *mPb* on the line $\bar{ij}$ connecting the two pixels i and j as

$$W(i,j) = \exp\left(-\max_{p\in\bar{ij}} \max_{\theta} \frac{mPb(p,\theta)}{\rho}\right), \tag{2.76}$$

where ρ is a constant. For W, the generalized eigenvalue problem is solved [Shi and Malik, 2000] (see section 2.2.7.2). On the resulting eigenvectors, gradients are computed with Gaussian derivative filters in the same orientations as the color and texture gradients before. The filter responses then contain the spectral contour information *sPb*. The linear combination of the oriented gradients and *sPb* is called global *Pb* (*gPb*). The *gPb* for an example image can be seen in figure 2.5.

The second part of the algorithm is the *Oriented Watershed Transform (OWT)* [Arbeláez et al., 2009, 2011]. On the maximal response of the contour detector over the orientations $\max_\theta gPb$, an initial watershed segmentation is computed that produces an oversegmentation of the image. For the resulting watershed boundary lines, the orientations are computed and binned such that they can be compared with the orientations of the detected contour *gPb* with a simple look-up. Every contour pixel is weighted with the strength of the contour detector output *gPb* in the according orientation. In the last step, the average of these weights is computed for every watershed contour arc. The result of the *OWT* is a set of closed, non self-intersecting weighted contours.

The last part of the algorithm consists in computing an *Ultrametric Contour Map (UCM)* [Arbeláez, 2006] in order to define the duality between these contours and a hierarchy of regions. From the result of the *OWT*, a graph is constructed containing the regions as nodes and the watershed contour arcs as edges. The edges are weighted with the weights of the *OWT*. In this graph, nodes that are connected with the weakest edges are subsequently merged. The result is a tree of regions where the original regions are the leaves and the root is the entire image. In the upper levels of the hierarchy,

original image *OWT* *UCM*

Figure 2.6.: Result of the *OWT* and *UCM* for an example color image. The result has been dilated for the visualization.

only the most dominant structures of the image are present, whereas the lower levels contain many details. The *OWT* and *UCM* for the example from figure 2.5 can be seen in figure 2.6.
In [Arbeláez et al., 2011] and [Arbeláez et al., 2009], a final segmentation was obtained by thresholding the *UCM* at an automatically chosen level. The threshold was chosen according to the best F-measure, which is defined as the harmonic mean of precision and recall. However, the authors also provide a framework for the interactive refinement of the segmentation by user-specified annotations. This framework was used by an expert in [Morath et al., 2013] to generate final segmentations from transmission electron microscopic recordings of cells. In [Keuper et al., 2011], we used the hierarchic segmentation result as input for our multi-label segmentation method with hierarchic MRFs (compare section 6).

2.4.2. Mean Shift

The mean shift procedure as presented in [Cheng, 1995] aims at analyzing the density distribution of arbitrary feature spaces and can be used to detect density maxima or to perform filtering or clustering of feature points. The theoretical basis for the application of the procedure to images has been presented in [Comaniciu and Meer, 2002]. The mean shift algorithm is based on the kernel density estimation also known as Parzen Window Technique. In the following, we summarize the derivation of the mean shift algorithm described in [Comaniciu and Meer, 2002]. Superpixels can be generated using the mean shift segmentation procedure, which will be described in section 2.4.2.5.

2.4.2.1. Kernel Density Estimation

For n independent data points $\mathbf{x}_i$, $i = 1, \ldots, n$ in d-dimensional space $\mathbb{R}^d$, that are generated by an unknown probability distribution f, a *kernel density estimator* with the kernel bandwidth parameter b is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{nb^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{b}\right), \tag{2.77}$$

where $K(\mathbf{x})$ is a d-variate kernel that integrates to one. For radially symmetric kernels, the multivariate *kernel density estimate* can be given by

$$\hat{f}_{b,K}(\mathbf{x}) = \frac{c_{k,d}}{nb^d} \sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{b}\right\|^2\right), \tag{2.78}$$

where the function $k(x)$ is the radial profile of the kernel $K(\mathbf{x})$:

$$K(\mathbf{x}) = c_{k,d} k(\|\mathbf{x}\|^2). \tag{2.79}$$

The function $k(x)$ with $k(x) \geq 0$ only needs to be defined for $x \geq 0$. The constant $c_{k,d}$ with $c_{k,d} > 0$ ensures, that $K(\mathbf{x})$ integrates to one. While the algorithm derived in [Comaniciu and Meer, 2002] can work with arbitrary kernels fulfilling the above listed properties, the authors specify two kernels used in practice: the Epanechnikov kernel and the normal kernel. The Epanechnikov kernel has the profile

$$k_E(x) = \begin{cases} 1 - x, & \text{if } 0 \leq x \leq 1 \\ 0, & \text{if } x > 1. \end{cases} \tag{2.80}$$

The radially symmetric kernel is given by

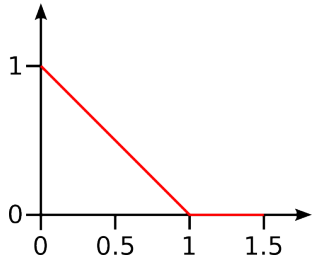
$$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1 - \|\mathbf{x}\|^2), & \text{if } \|\mathbf{x}\| \leq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{2.81}$$

where c_d is the volume of the d-dimensional unit sphere. k_E is not differentiable for $x = 1$, for $x \neq 1$, $k'_E(x)$ is uniform. For a visualization, see figure 2.7.
The profile of the normal kernel is given by

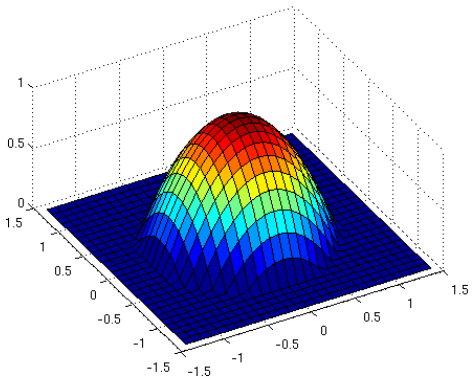
$$k_N(x) = \exp\left(\left(-\frac{1}{2}x\right)\right), \qquad x \geq 0. \tag{2.82}$$

The multivariate normal kernel is then given by

$$K_N(\mathbf{x}) = (2\pi)^{-d/2} \exp\left(\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\right). \tag{2.83}$$



(a) Radial profile k_E of the Epanechnikov kernel.



(b) Surface plot of the 2D Epanechnikov kernel K_E.

Figure 2.7.: Epanechnikov kernel and the according profile.

2.4.2.2. Density Gradient Estimation

The main step of the mean shift procedure is to determine the positions of local maxima (modes) in this density distribution of the feature space. Thus, zero positions of the actual density gradient $\nabla f(\mathbf{x}) = 0$ are estimated by gradient ascent of the estimate $\hat{f}_{b,K}(\mathbf{x})$. For differentiable profiles k, the gradient of $\hat{f}_{b,K}(\mathbf{x})$ is given by [Comaniciu and Meer, 2002]

$$\begin{aligned}\nabla \hat{f}_{b,K}(\mathbf{x}) &= \frac{2c_{b,d}}{nb^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{b} \right\|^2 \right) \\ &= \frac{2c_{k,d}}{nb^{d+2}} \left[\sum_{i=1}^{n} k' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{b} \right\|^2 \right) \right] \left[\mathbf{x} - \frac{\sum_{i=1}^{n} \mathbf{x}_i k' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{b} \right\|^2 \right)}{\sum_{i=1}^{n} k' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{b} \right\|^2 \right)} \right],\end{aligned}$$

where the first term is proportional to the density estimate with kernel $G(\mathbf{x}) = c_{g,d} g(\|\mathbf{x}\|^2)$ with $g(x) = -k'(x)$ and the second term is the mean shift

$$\mathbf{m}_{b,G}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{b} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{b} \right\|^2 \right)} - \mathbf{x}, \tag{2.84}$$

from which the gradient ascent iteration is derived as:

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i g \left(\left\| \frac{\mathbf{y}_j - \mathbf{x}_i}{b} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{\mathbf{y}_j - \mathbf{x}_i}{b} \right\|^2 \right)}. \tag{2.85}$$

In each iteration, the points are shifted towards the next local density modes.

2.4.2.3. Mean Shift in the joint Spatial-Range Domain

In [Comaniciu and Meer, 2002], the authors show how the mean shift procedure can be employed as a discontinuity preserving image filter. This is done by interpreting the image as a two-dimensional lattice of p-dimensional vectors ($p = 1$ in the gray level case, $p = 3$ for color images). The lattice then forms the spatial domain, the color information forms the range domain of a $d = p + 2$ dimensional feature vector in the joint spatial-range domain. The mean shift is performed using a multivariate kernel [Comaniciu and Meer, 2002]

$$K_{b_s,b_r}(\mathbf{x}) = \frac{C}{b_s^2 b_r^2} k \left(\left\| \frac{\mathbf{x}^s}{b_s} \right\|^2 \right) k \left(\left\| \frac{\mathbf{x}^r}{b_r} \right\|^2 \right), \tag{2.86}$$

defined as a product of two radially symmetric kernels with the common profile $k(x)$. $\mathbf{x}^s$ is the spatial part, $\mathbf{x}^r$ the range part of the feature vector. These bandwidth parameters b_s and b_r control the size of the kernel in the spatial and range domain and thus determine the resolution of the density maximum detection. c is the corresponding normalization constant.

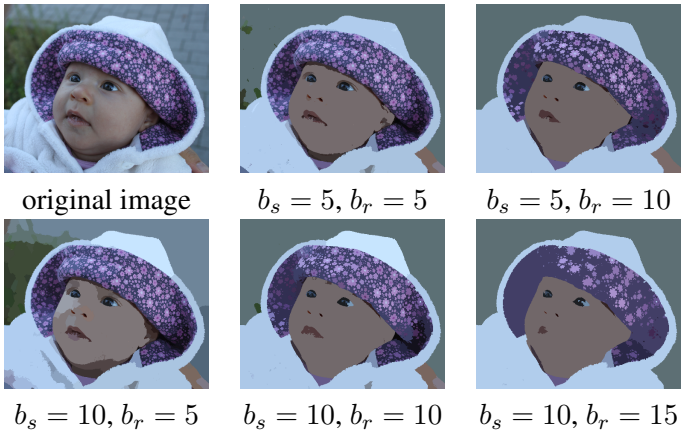


Figure 2.8.: Influence of the bandwidth parameters b_s and b_r on the resulting superpixels with the mean shift segmentation method for an example color image. The minimal regions size M has been set to 10. The colors have been interpreted as features in the *L*u*v** color space [Comaniciu and Meer, 2002].

2.4.2.4. Mean Shift Filtering

The mean shift filtering is computed on the d-dimensional feature vectors $\mathbf{x}_i, i = 1 \dots n$ in the joint spatial-range domain. The algorithm given in [Comaniciu and Meer, 2002] is as follows.
For each pixel:

1. Initialize $j = 1$ and $\mathbf{y}_{i,1} = \mathbf{x}_i$.
2. Compute $\mathbf{y}_{i,j+1}$ according to equation (2.85) until convergence, $\mathbf{y} = \mathbf{y}_{i,c}$.
3. Assign $\mathbf{z}_i = (\mathbf{x}_i^s, \mathbf{y}_{i,c}^r)^T$.

$\mathbf{z}_i$ are pixels of the resulting filtered image. The subscripts s and r denote the spatial and range components of the vectors respectively. In the resulting image, the pixels at position $\mathbf{x}_i^s$ are assigned the range component of the convergence point $\mathbf{y}_{i,c}^r$.

2.4.2.5. Mean Shift Clustering

In [Comaniciu and Meer, 2002], the clustering of image pixels using mean shift in the joint spatial-range domain is also used to compute image segmentations into homogeneous regions. The segmentation algorithm given in [Comaniciu and Meer, 2002] is based on the mean shift filtering on the d-dimensional feature vectors $\mathbf{x}_i, i = 1 \dots n$ in the joint spatial-range domain. The $\mathbf{z}_i, i = 1 \dots n$ are the filtered image pixels and ℓ_i are the labels of the pixels. The algorithm is given in [Comaniciu and Meer, 2002] as follows:

1. Run the mean shift filtering procedure for the image and store all information about the d-dimensional convergence point in $\mathbf{z}_i$, i.e. $\mathbf{z}_i = \mathbf{y}_{i,c}$.
2. Delineate in the joint spatial-range domain the clusters $\{C_p\}_{p=1\dots m}$ by grouping together all $\mathbf{z}_i$ which are closer than b_s in the spatial domain and b_r in the range domain.
3. For each image pixel $i = 1 \dots n$, assign $\ell_i = \{p | \mathbf{z}_i \in C_p\}$.

4. Optional: Eliminate spatial regions containing less than M pixels.

The result of the mean shift clustering of image pixels is a segmentation of the image. However, it only results in closed regions if the value M is chosen sufficiently large. In the experiments given in [Comaniciu and Meer, 2002], M is chosen between 2 and 6 times larger than b_r. The size and homogeneity of the regions resulting from the mean shift segmentation depends on the choice of the parameters b_s and b_r. The influence of these parameters on the resulting regions can be seen in figure 2.8. The mean shift was computed with an Epanechnikov kernel using the EDISON library[3].

[3] http://coewww.rutgers.edu/riul/research/code.html

3. Microscopic Data

In this work, we mainly deal with the analysis of single cells and sub-cellular structures from microscopic recordings. Our data is mainly recorded using widefield fluorescence microscopy but also standard confocal microscopy, spinning disk confocal microscopy, and electron microscopy are employed. Depending on the employed imaging technique, different problems arise for the analysis of the recorded data. Therefore, knowledge about the effects of the different imaging modalities is required when tackling the segmentation in microscopic datasets. The special challenges present in these datasets mostly cause state-of-the-art image segmentation techniques (see chapter 2) to fail, when they are applied without further adaptations. A brief overview on the image acquisition with different microscopy techniques is given in section 3.1. We focus only on those techniques that were actually used for the acquisition of our data. Especially data recorded with widefield microscopy suffers from strong blurring. For many image analysis tasks, these data have to be reconstructed using deconvolution techniques (see chapter 4).

3.1. Image Formation with Fluorescence Microscopy

As the name implies, fluorescence microscopes use the phenomenon of fluorescence: Light with a certain wavelength excites the fluorescent molecules of the specimen. These excited, fluorescent molecules then emit light with a longer wavelength that can be observed and recorded with a camera or a detector [Sarder and Nehorai, 2006]. The most important part of fluorescence microscopes is the dichroic (or dichromatic) mirror. It reflects the excitation light and directs it to the specimen while it is transparent to light in the emission wavelength, such that the emitted light can pass the mirror and be observed by a detector. Thus, it is possible to observe exclusively the light emitted by the specimen [Spring, 2003; Lichtman and Conchello, 2005].
In order to record a specimen with fluorescence microscopy, the parts of interest must be fluorescent. This can be achieved by marking these parts with fluorescent stains (for example DAPI (4',6-diamidino-2-phenylindole) marks the DNA). Specific proteins can be marked using immunofluorescence, where antibodies are bound to specific antigens. These "primary" antibodies are either fluorescent themselves or are bound to a "secondary" antibody, conjugated to a fluorophore [Fritschy and Härtig, 2001; Lichtman and Conchello, 2005]. In immunofluorescence, the Green Fluorescent Protein (GFP) plays an important role [Chalfie et al., 1994], because it can be introduced into an organism and used as a marker of gene expression. Last but not least, some biological specimen are autofluorescent, as for example mitochondria and lysosomes that contain amino acids [Menter, 2006], but also chlorophyll shows autofluorescence. Autofluorescence is not always a desired effect for fluorescence microscopy, because it possibly interferes with the detection of other fluorescent stains.

3.1.1. Widefield Fluorescence Microscopy

For the analysis of living cells, conventional (i.e. widefield) fluorescence microscopy still plays an important role, because it is prevalently available and, compared to confocal microscopy, has some advantages concerning temporal resolution and phototoxicity. Usually a mercury lamp that emits white light is used as light source. From this white light, the wavelength of the excitation light is selected by optical filters. The dichroic mirror directs the excitation light to the specimen such that the fluorescent molecules of the specimen are excited. These fluorescent molecules emit light of a certain emission wavelength spherically in all directions. A part of the emitted light passes through the dichroic mirror and is again filtered by a barrier filter suppressing the remaining light of the excitation wavelength [Spring, 2003]. This emitted light can be directly observed or recorded by a conventional charged coupled device (CCD) camera [Spring, 2003]. See figure 3.1 for an illustration and [Spring, 2003; Abramowitz, 1993] for more details.

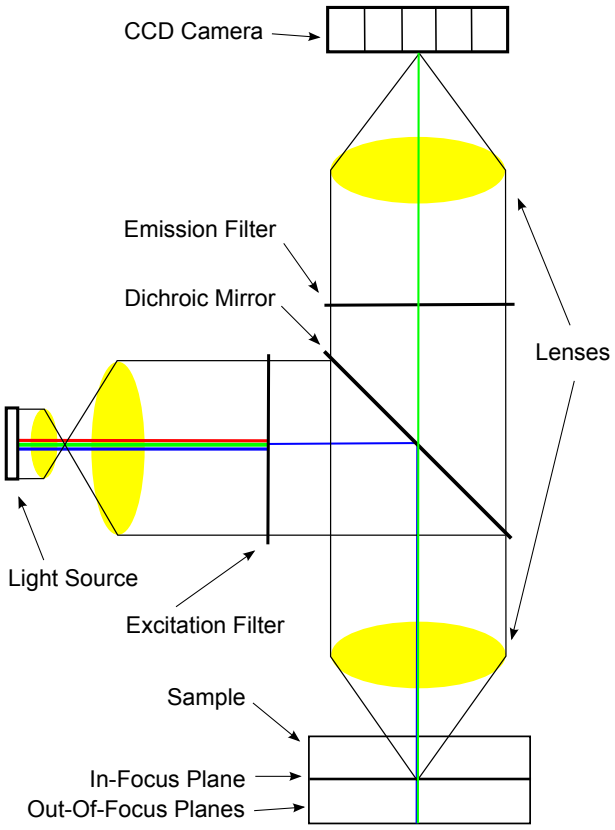


Figure 3.1.: Schematic sketch of a widefield microscope.

The 3D information of the specimen is recorded in a set of 2D images. Since the whole specimen is illuminated for every image, there is always light from in-focus and from out-of-focus planes recorded. Thus, the resolution along the optical axis (the resolution in z-direction) is heavily reduced [Sarder and Nehorai, 2006]. The recorded out-of-focus light also determines the impulse response, the point-spread function (PSF) of the imaging system. A detailed description of the factors determining the PSF and methods for the estimation of the PSF of an imaging system will be given in section 3.1.3. Besides the PSF, there are several factors limiting the recording qual-

ity [Sarder and Nehorai, 2006] as the spatial fluctuation of illumination in the specimen and lamp flicker. Self absorption, bleaching effects, and scattering can lead to the attenuation of certain image areas and the quantum nature of light causes Poisson noise [Sarder and Nehorai, 2006; Boyd, 1983]. Simplified, the image formation can be described by a convolution of the specimen function s with the microscope specific PSF h and a voxelwise noise function n.

$$o = n(s * h), \tag{3.1}$$

where

- $\Omega \subseteq \mathbb{R}^3$ is the image domain,
- $o : \Omega \to \mathbb{R}$ denotes the objective function, i.e. the recorded image,
- $s : \Omega \to \mathbb{R}$ is the specimen function,
- $h : \mathbb{R}^3 \to \mathbb{R}$ is the point spread function (PSF),
- '$*$' is the convolution operator,
- $n : \mathbb{R} \to \mathbb{R}$ is a voxelwise noise function, and
- h integrates to one: $\int_{\mathbb{R}^3} h(\mathbf{x}) d\mathbf{x} = 1$.

3.1.2. Confocal Microscopy

Compared to widefield microscopes, confocal laser scanning microscopes (CLSM) can produce images with a higher resolution in the direction of the optical axis. They do so by point-wise scanning through the sample using a laser beam [Lichtman and Conchello, 2005], such that the laser is focused onto exactly one point at a time. As a result, very little fluorescence is emitted from out of focus points. The remaining out-of-focus light is discarded using a pinhole that is placed in front of the detector, in most cases a photomultiplier. Due to the limited sensitivity of the photomultiplier compared to a CCD, confocal microscopes are far less light sensitive, thus resulting in a poorer signal to noise ratio [Sarder and Nehorai, 2006; Toomre and Pawley, 1996]. In order to produce good image quality, it is required to scan slowly through the specimen and expose it for longer time. The enhanced image quality thus comes with negative side effects that are photobleaching and high phototoxicity. In figure 3.3, a sketch of a confocal microscope is given. For more details refer to [Wilhelm et al., 2012].
Since it becomes more and more necessary to image living specimen, the long scanning times of CLSM are problematic because they limit the temporal resolution. This problem is overcome by the use of spinning disk confocal microscopes. In spinning disk microscopes, the excitation light is directed through multiple spinning microlenses and a spinning pinhole disk (Nipkow Disk) onto the sample such that multiple points are excited at the same time. The emission light from these points passes through the same pinhole disk and is directed to a CCD camera. For more details on spinning disk microscopy refer for example to [Graef et al., 2005].

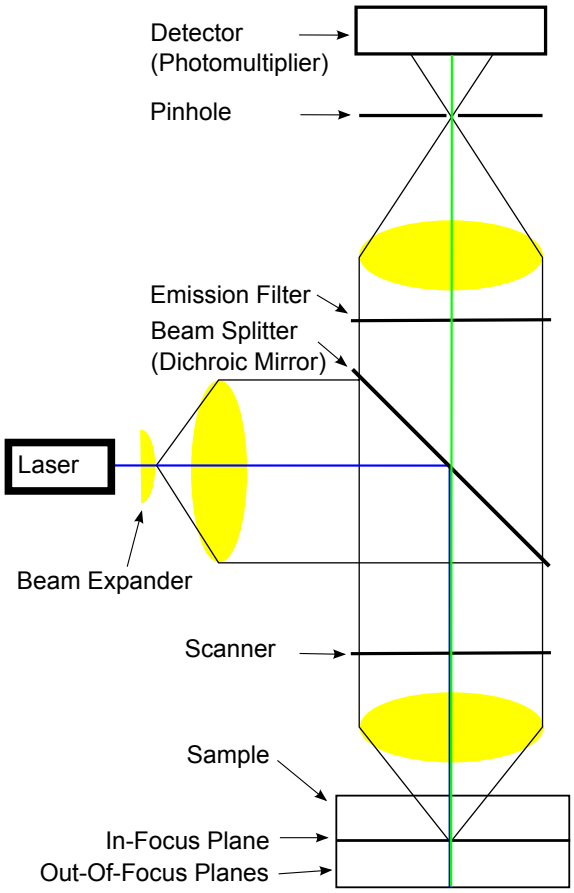
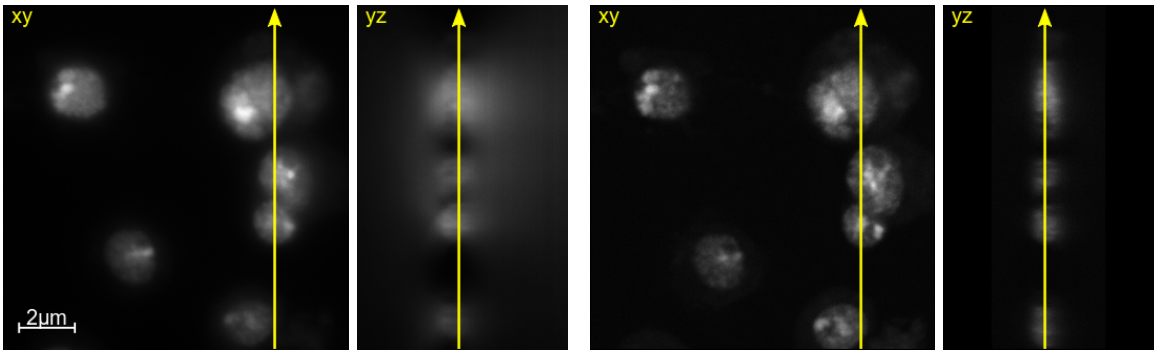


Figure 3.2.: Schematic sketch of a confocal laser scanning microscope.

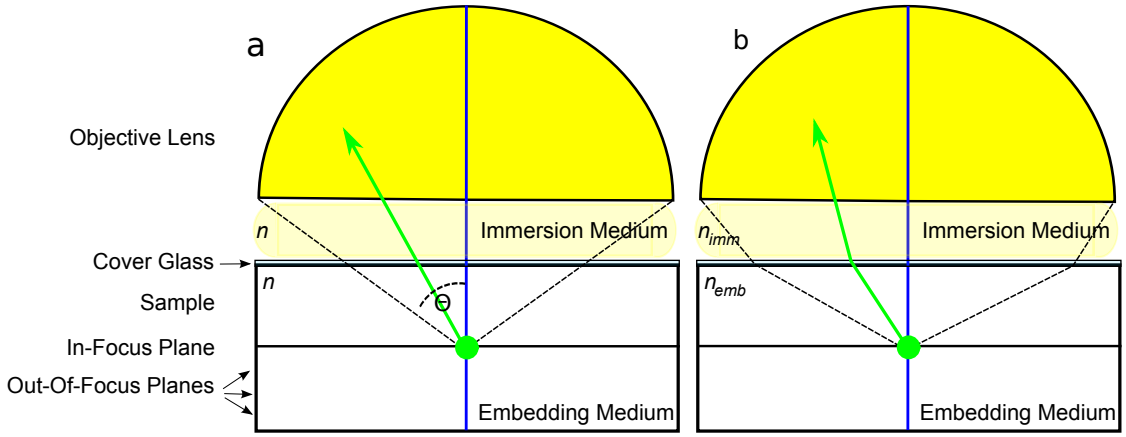


(a) Widefield recording. (b) Spinning disk recording.

Figure 3.3.: Exactly the same sample of *Drosophila* S2 cell nuclei recorded with a widefield and a spinning disk confocal microscope. The volumetric data is displayed in two orthogonal views. The arrows indicate the position of the cutting plane. Especially the resolution along the optical axis is greatly reduced in the widefield recording. The sample preparations and recordings have been provided by Jan Padeken. The registration of the images has been computed using normalized cross-correlation.

3.1.3. PSF Estimation

The PSF of a fluorescence microscope depends on several facts as for example the Numerical Aperture (NA), the emission wavelength λ, and the position of the recorded object. The NA is proportional to the sine of one half of the angular aperture Θ of the objective and the refractive index of the immersion medium n_{imm}: $\text{NA} = n_{imm} \cdot \sin\Theta$ [Abramowitz, 1993]. Most importantly, the finite lens aperture introduces diffraction ring patterns in the recorded xy-sections [Sarder and Nehorai, 2006], the so called Airy pattern (see figure 3.5(a)), that limits the resolution of the recording system [1]. A second important effect is caused by refractive index mismatch. In an ideal recording setting, the refractive indices n of immersion medium, specimen embedding medium, and specimen are the same. Differences between these refractive indices cause spherical aberrations[2] (see figure 3.4). Because of the Airy pattern, the PSF of widefield microscopes does not have a compact support



(a) Aberration-free imaging. The refractive index of the immersion medium and the refractive index of the embedding medium are equal.

(b) Imaging with spherical aberration. The refractive index n_{imm} of the immersion medium and the refractive index n_{emb} of the embedding medium are different.

Figure 3.4.: Sketch showing the imaging of a bead with and without refractive index mismatch.

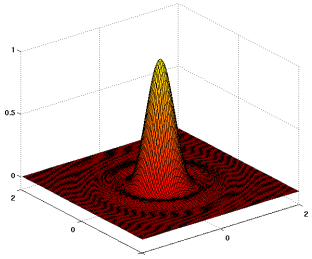
[Frisken-Gibson and Lanni, 1992; Born and Wolf, 1999; Goodman, 1996; Streibl, 1985; Stokseth, 1969; Gustafsson et al., 1995] and usually has values of an unneglectable range in the whole image domain Ω.

In order to analyze or to alleviate the blur introduced in the recording by the PSF, it is crucial to estimate the PSF as good as possible (see section 4). PSF estimates can be generated either analytically by simulating the recording system or experimentally by measuring the impulse response of a point light source [Agard, 1984]. A third possibility is to estimate the PSF from the recording of an object using blind deconvolution techiques (see section 4).

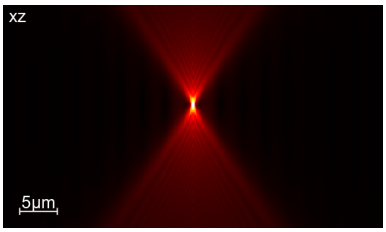
[1] According to the Abbe diffraction limit, the lateral resolution is limited to $d_{xy} = \frac{\lambda}{2\text{NA}}$ and the axial resolution is limited to $d_z = \frac{2\lambda}{\text{NA}^2}$ [Lipson et al., 2011].

[2] Spherical aberration means that the light rays are focused depending on the radial distance from the optical axis. Ideally, all light should be focused to one point, independently from the distance to the optical axis at which it enters the lens.

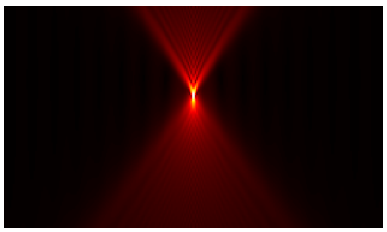
3.1.3.1. PSF Simulation



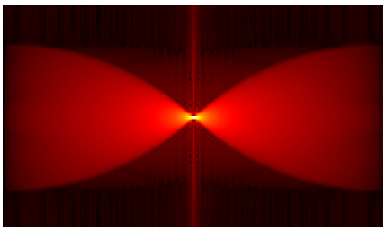
(a) Airy Pattern



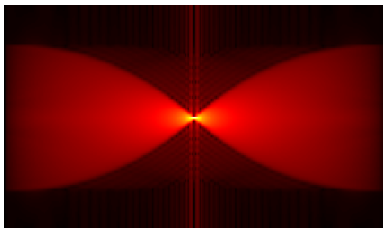
(b) Symmetric PSF



(c) Asymmetric PSF



(d) OTF of the symmetric PSF



(e) OTF of the asymmetric PSF

Figure 3.5.: The Airy Pattern (a) and yz-sections of simulated 3D widefield PSFs with the Gibson and Lanni model [Frisken-Gibson and Lanni, 1992] without (b) and with (c) spherical aberration. The magnitude of the according OTFs are shown in (d) and (e). The PSF's and OTF's values have been γ-corrected with $\gamma = 1/3$ for the visualization and are displayed with a hot color map.

There are several methods to theoretically model the recording system and thus model the PSF [Stokseth, 1969; Streibl, 1985; Frisken-Gibson and Lanni, 1992; Born and Wolf, 1999] or its Fourier domain equivalent, the object transfer function (OTF) [Frieden, 1967; Nakamura and Kawata, 1990; Sheppard and Gu, 1994; Philip, 2009; Schönle and Hell, 2002]. A complete presentation of all methods is beyond the scope of this thesis and only the most common methods are shortly presented. The well-known model from Born and Wolf [Born and Wolf, 1999; Frisken-Gibson and Lanni, 1989] simulates the optical system with some restrictions. The model describes the diffraction that occurs when the observed fluorophore particle is in the focal plane. The PSF simulated with [Born and Wolf, 1999] thus does not depend on the position of the simulated particle.
This is different for the method of Gibson and Lanni [Frisken-Gibson and Lanni, 1992] that takes the different refractive indices of the lens immersion medium and the specimen embedding medium into account. This way, the resulting spherical aberration can be modeled. Even further refractive index-mismatches for example of the specimen and the embedding medium can be modeled by [Frisken-Gibson and Lanni, 1992]. The simulated particle can be located at any point in the sample. The resulting PSF depends on the z-position of the recorded point. The computation is based on the Kirchhoff diffraction integral and given by

$$h(x,y,z) = \frac{C}{z}\left|\int_0^1 J_o\left(\frac{k\mathrm{NA}\rho\sqrt{x^2+y^2}}{z}\right)e^{\imath W(\rho)}\rho d\rho\right|^2, \tag{3.2}$$

where C is a constant, J_0 is the zero order Bessel function of the first kind, $k = 2\pi/\lambda$ is the wave

number, NA the numerical aperture, $\imath$ denotes the imaginary unit, and $W(\rho)$ is the phase aberration at the normalized radius ρ, which is in fact depending on the different refractive indices and the depth of the in-focus plane. Figure 3.5 shows xz-sections of two simulated PSF recordings with different refractive indices, generated with the ImageJ[3] tool described in [Kirshner et al., 2011] using the model of Gibson and Lanni [Frisken-Gibson and Lanni, 1992]. The refractive index of the immersion medium is 1.5 in both cases. For the PSF with spherical aberration, we have used a refractive index of 1 for the specimen. For the remaining parameters we took the default values (NA = 1.4, λ = 500nm, particle position in z-direction = 2000nm, resolution in xy-direction = 0.25μm and in z-direction 0.5μm).

In the model of Richards and Wolf [Richards and Wolf, 1959; Aguet, 2009], the phase aberration is described as in [Frisken-Gibson and Lanni, 1992], but the electric field components are considered independently, such that the vector-valued diffraction can be modeled. However, this only offers small improvements in the accuracy compared to the scalar-valued diffraction model of [Frisken-Gibson and Lanni, 1992] at higher computational costs (see [Pankajakshan et al., 2009])[4].

Due to the nature of optical systems, it often makes sense to directly analyze the Fourier domain properties of the imaging system which has been done for example in [Frieden, 1967; Nakamura and Kawata, 1990; Sheppard and Gu, 1994; Philip, 2009; Schönle and Hell, 2002; Gustafsson et al., 1995]. In [Frieden, 1967], the three-dimensional OTF for systems with small apertures was derived. This approximation is not valid for lenses with high aperture [Schönle and Hell, 2002; Philip, 2009], such that improved approximations need to be considered for fluorescence microscopes. In [Nakamura and Kawata, 1990], the authors deduce the support of the widefield OTF, the finite region where it has non-zero values. The values of the widefield OTF are not determined in [Nakamura and Kawata, 1990] since the focus of the paper is the computation of the confocal OTF. In [Philip, 2009] and [Schönle and Hell, 2002], the computation of the widefield OTF is deduced, defined by the autocorrelation of a spherical cap of the Ewald Sphere. [Schönle and Hell, 2002] derive the OTF as a line integral also modeling refractive index mismatches, whereas [Philip, 2009] give an algorithm to directly compute the OTF-values for aberration-free systems.

In this thesis, we only use knowledge about the support of the widefield OTF, since it determines the resolution of an imaging system. Frequencies outside the OTF support are cut off during the imaging process. The OTF support of a widefield microscope is depicted in figure 3.6. The recorded light field is assumed to be nearly monochromatic with constant wavelength λ. The corresponding wave-number is $k = 2\pi/\lambda$ for the whole field, which means that the Fourier transform of the field amplitude has non-zero values only on a shell with radius $2\pi/\lambda$ (figure 3.6(a)). Since only the light from a limited angle Θ can be recorded (figure 3.6(b)), the recorded information is limited to a "cap" of the spherical shell in the Fourier space (figure 3.6(c)). The autocorrelation of this spherical cap is the region from which intensity information is accessible, i.e. the support of the OTF. Compare [Gustafsson et al., 1995; Philip, 2009] for more detailed information.

[3] ImageJ is a public domain image processing software written in Java. The program and documentation can be downloaded from `http://rsbweb.nih.gov/ij/`.

[4] Recently, the method of Richards and Wolf [Richards and Wolf, 1959] has also been implemented in ImageJ [Kirshner et al., 2011].

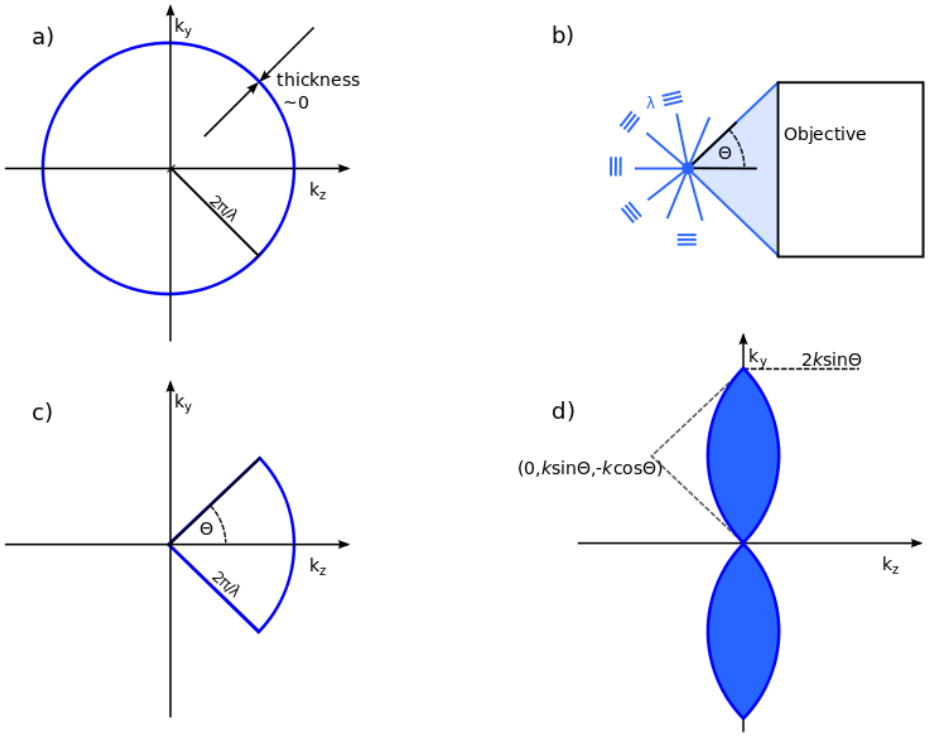


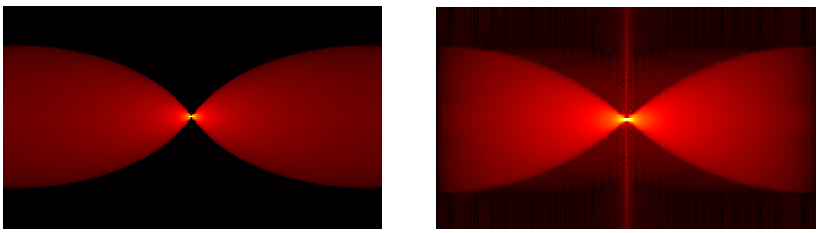
Figure 3.6.: Illustration of the support of the widefield OTF. (a) The spherical shell on which the Fourier transform of the monochromatic light field amplitudes is non-zero. (b) The objective lens with a finite acceptance angle Θ. (c) The spherical cap containing the recorded amplitude information. (d) The resulting OTF support. (Compare [Gustafsson et al., 1995; Philip, 2009] for similar visualizations)

The OTF is rotation symmetric around the $\mathrm{k_z}$ frequency axis. Therefore, it is sufficient to compute the OTF for the positive frequencies. Furthermore, the OTF value at frequency position $\xi = (\mathrm{k_x}, \mathrm{k_y}, \mathrm{k_z})^T$ equals the value at position $\bar{\xi} = \left(0, \sqrt{{\mathrm{k_x}}^2 + {\mathrm{k_y}}^2}, \mathrm{k_z}\right)^T$. The OTF at frequency $\xi = (0, \mathrm{k_y}, \mathrm{k_z})^T$ can be computed as [Philip, 2009]:

$$\mathcal{F}(h)(\xi) = \text{const} \cdot \frac{1}{\|\xi\|} \sin^{-1}\left(\frac{k^2 \sin^2\Theta - \left(\left({\mathrm{k_y}}^2 + {\mathrm{k_z}}^2 + 2k\mathrm{k_z}\cos\Theta\right)/2\mathrm{k_y}\right)^2}{k^2 - \|\xi\|^2/4} \right)^{1/2} . \quad (3.3)$$

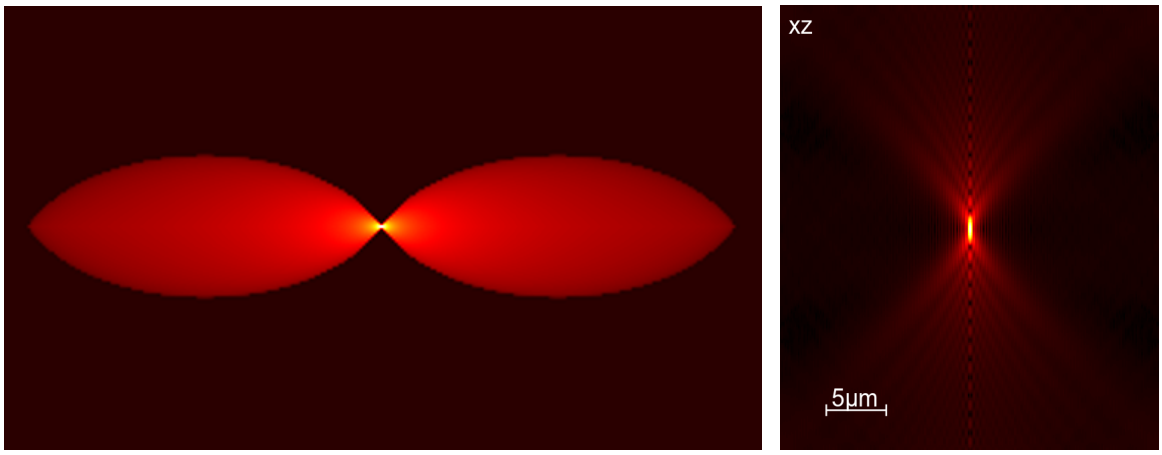
It can be observed that in $\mathrm{k_z}$-direction, the OTF drops to zero very quickly. The conical region where the OTF is zero is called *missing cone* and explains the significant amount of blur in z-direction. Here, even low frequency information is cut off. This missing cone can also be seen in the OTFs computed from the simulated PSFs (compare figure 3.5 (d) and (e)). However, in the OTFs computed from the simulated PSFs, there are artifacts originating from sampling and from the fact that the PSFs are cut off at some point in the spatial domain. For comparison, we have directly simulated the OTF with the same parameters as the symmetric PSF in figure 3.5(b) with the simulation method proposed by Philip [2009]. The result is displayed in figure 3.7. For this

parameter setting, the computed OTF is also cut off in $\mathrm{k_x k_y}$-direction. For a higher spatial resolution ($0.125 \times 0.125 \times 0.25\mu\mathrm{m}^3$), the simulated OTF and the symmetric PSF computed from this OTF are shown in figure 3.8. With this resolution, the frequencies are not cut off anymore. Small artifacts in the resulting PSF can be seen around the center. It follows that when using simulation methods for the widefield PSF, one has to decide for either a good simulation in the spatial domain with artifacts in the frequency domain or for a good representation in the frequency domain but artifacts in the spatial domain. Furthermore, to the best of our knowledge, there is no method published that allows for the direct computation of the OTF values of a spherically aberrated system.



(a) Simulated OTF. (b) OTF computed from simulated PSF.

Figure 3.7.: Comparison between the simulated OTF and the same OTF computed from the simulated PSF from figure 3.5(b) for the parameters NA = 1.4, λ = 500nm, $n_{imm} = n_{emb} = 1.5$, and a spatial resolution of $0.25 \times 0.25 \times 0.5\mu\mathrm{m}^3$.



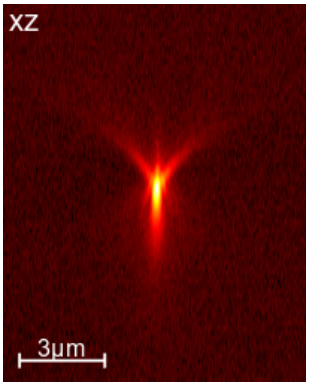
(a) Simulated OTF. (b) PSF computed from OTF.

Figure 3.8.: A $\mathrm{k_x k_z}$-section of the simulated OTF for the parameters NA = 1.4, λ = 500nm, $n_{imm} = n_{emb} = 1.5$, and a spatial resolution of $0.125 \times 0.125 \times 0.25\mu\mathrm{m}^3$ and an xz-section of the PSF computed from this OTF.

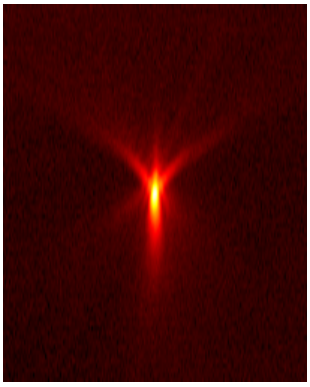
The PSF of confocal laser scanning microscopes is related to the widefield PSF. Using the Helmholtz reciprocity theorem, the authors of [Pankajakshan et al., 2009] write the CLSM PSF as

$$h_{\mathrm{CLSM}}(x,y,z) = c \cdot \left|h(x,y,z;\lambda_{ex})\right|^2 \cdot \int_{r^2 \leq \frac{D^2}{4}} \left|h(x-x_1, y-y_1, z;\lambda_{em})\right|^2 dx_1 dy_1, \qquad (3.4)$$

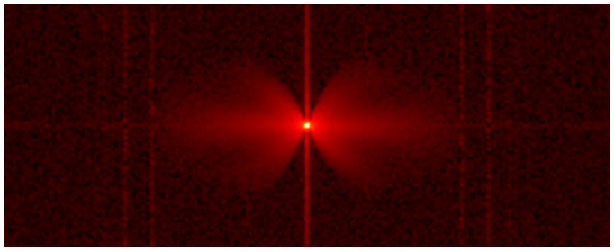
where h is the widefield PSF, λ_{ex} and λ_{em} are the excitation and emission wavelengths, $r^2 = x_1^2 + y_1^2$ and D is the back-projected diameter of the circular pinhole. The computation of the confocal



(a) Bead recording.



(b) Averaged bead over four recordings.



(c) OTF from measured PSF.

Figure 3.9.: Measured PSF from bead recording. xz-sections through the center of the bead are shown with a γ-correction with $\gamma = 1/3$. The section of the OTF is visualized with $\gamma = 1/4$. The recordings have been taken by Jan Padeken.

OTF is given in [Nakamura and Kawata, 1990], where it is also shown that the confocal OTF has no missing cone. In [Zhang et al., 2006], the authors show that the confocal PSF can be well approximated by a 3D Gaussian function.

3.1.3.2. PSF Estimation using Bead Recordings

The PSF of a recording system can be measured by recording fluorescent microspheres, so called beads. In order to have a punctual light source, it is important that the recorded bead is smaller than the resolution. Furthermore, all recording parameters must be exactly the same as in the specimen recording (NA, wavelength λ, refractive indices of immersion and embedding medium, etc.). A remaining problem is the recorded noise. In order to compute a PSF estimation from a recorded image, the recorded background noise can be estimated by fitting a Poisson distribution to the image intensity distribution. The underlying assumption is that the recorded bead is small in relation to the recorded background and the photon noise is Poisson distributed [Boyd, 1983]

$$P_\Lambda(X = k) = \frac{\Lambda^k}{k!} e^{-\Lambda}. \tag{3.5}$$

A common method to reduce noise in the measured PSF is to record multiple beads and build the average over the recordings. The resulting measurement contains less noise. However, the recordings of the beads have to be properly aligned. Otherwise, fine structures are lost and undesired blur is introduced.

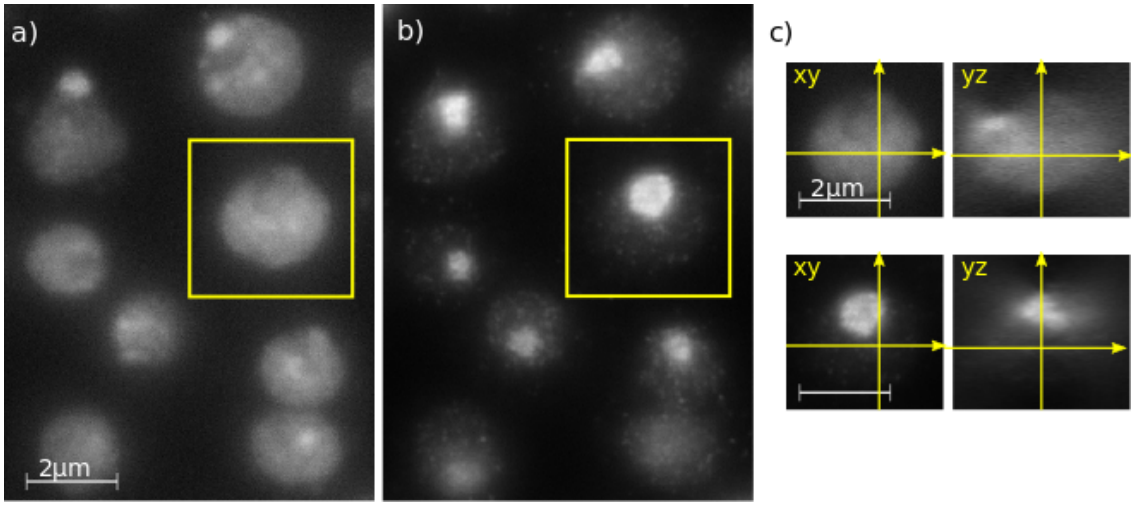


Figure 3.10.: *Drosophila* S2 cell nucleus recordings. a) Channel 0: The DAPI stained chromatin. b) Channel 1: The GFP stained fibrillarin of the nucleoli inside the nuclei. c) Orthogonal views of one of the cell nuclei for channel 0 (top row) and channel 1 (bottom row). In the yz-view, the strong blurring caused by the widefield PSF is visible.

In figure 3.9, the result of a PSF measurement is shown. For this measurement, a TetraSpeck 0.2μm bead was recorded four times[5]. The used parameters are NA = 1.4, λ = 358, $n_{imm} = 1.516$, $n_{emb} = 1.45$, and a spatial resolution is $0.0642 \times 0.0642 \times 0.2\mu\text{m}^3$. The measured PSF is highly asymmetric and noisy even after averaging. Vertical lines in the computed OTF from this measurement indicate that the values are cut off in z-direction.

3.2. Fluorescence Microscopic Datasets

The segmentation methods developed in this thesis have been applied to several different datasets recorded with fluorescence microscopy. Three of these datasets on which we show results in this thesis will be briefly described in the following.

3.2.1. *Drosophila* S2 Cell Nuclei

The first dataset (see figure 3.10) used in this thesis consists of volumetric recordings of *Drosophila* S2 cell nuclei imaged with widefield microscopy. The S2 or Schneider cell line is a commonly used cell line originally derived from *Drosophila* embryos. In 45 recordings, 440 cells have been fully imaged. Some additional cells were lying on the boundary of the recording and were therefore useless for biological evaluation. The data originates from a project initiated by Patrick Heun and Olaf Ronneberger and was recorded by Jan Padeken in the Max-Planck-Institute for Immunobiology and Epigenetic in Freiburg. For the biological question at hand, it was our goal to provide exact knowledge about the sub-nuclear anatomy, i.e. the exact position of the nucleolus[6] within the cell nucleus. Usually, there were three channels recorded, where the DAPI staining inside the cell nucleus is imaged in channel 0, channel 1 shows the GFP stained fibrillarin inside the nucleolus and

[5]The recording has been done together with Jan Padeken.
[6]The nucleolus is a structure inside the nucleus, that is important for the transcription of ribosomal RNA.

the third channel shows a protein to be investigated. Our work focuses on the segmentation of cell nuclei and nucleoli from the first two channels. DAPI stains the chromatin inside the nucleus such that bright regions in channel 0 probably belong to the nucleus. However, the nucleolus inside the nucleus does not contain DNA. It is therefore not stained by DAPI and causes a "hole" in the nucleus recording. The fibrillarin stained in channel 1 is certainly inside the nucleolus but the density of the staining near the nucleolus boundaries is inhomogeneous. The true nucleolus boundary thus has to be searched in between the stained bright regions in channel 1 and the dark region inside the nucleus in channel 0. A sketch illustrating the task we want to approach within these data is given in figure 3.11. However, we only give an example of a possible biological evaluation. The exact biological question is not mentioned here, because it has not been published so far.

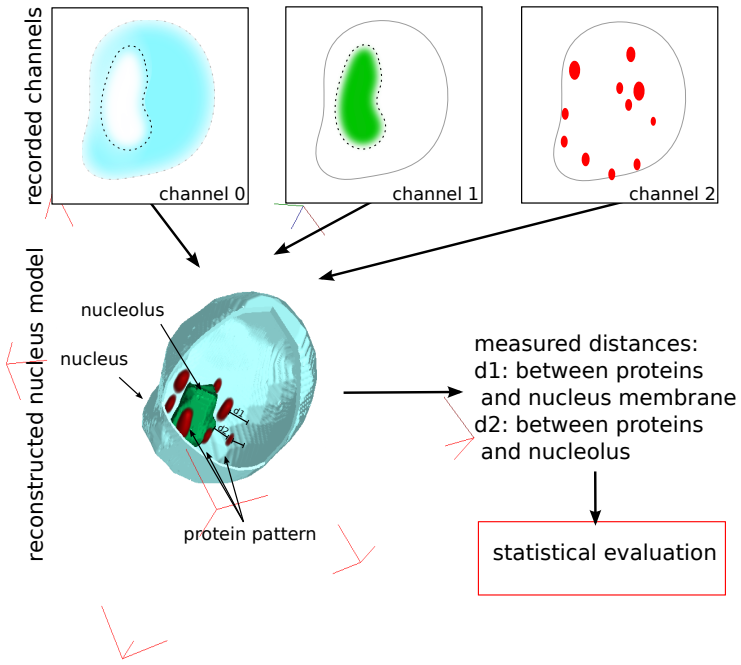


Figure 3.11.: Schematic sketch of the task given in the *Drosophila* S2 cell nucleus dataset. The sought nucleolus boundary needs to be extracted from information given in the DAPI (recorded in channel 0) and the fibrillarin staining (recorded in channel 1).

The voxel size in xy-direction is 0.064μm in z-direction 0.2μm. The recordings were taken with a Deltavision Microscope using a 60x oil immersion objective with NA = 1.4. The excitation wavelength for DAPI is λ = 358. The used embedding medium is SlowFade Gold from Invitrogen with a refractive index $n_{emb} = 1.45$ and the immersion medium had a refractive index $n_{imm} = 1.516$.
In figure 3.10, an example slice from the first two channels is displayed, as well as orthogonal slices of one of the nuclei. The main challenges in this dataset are the strong blurring in z-direction as well as the inhomogeneous staining. For example in channel 0, regions of dense chromatin cause bright "blobs" in the recorded data. Dark regions can either correspond to less dense chromatin, to the nucleolus, or to background.

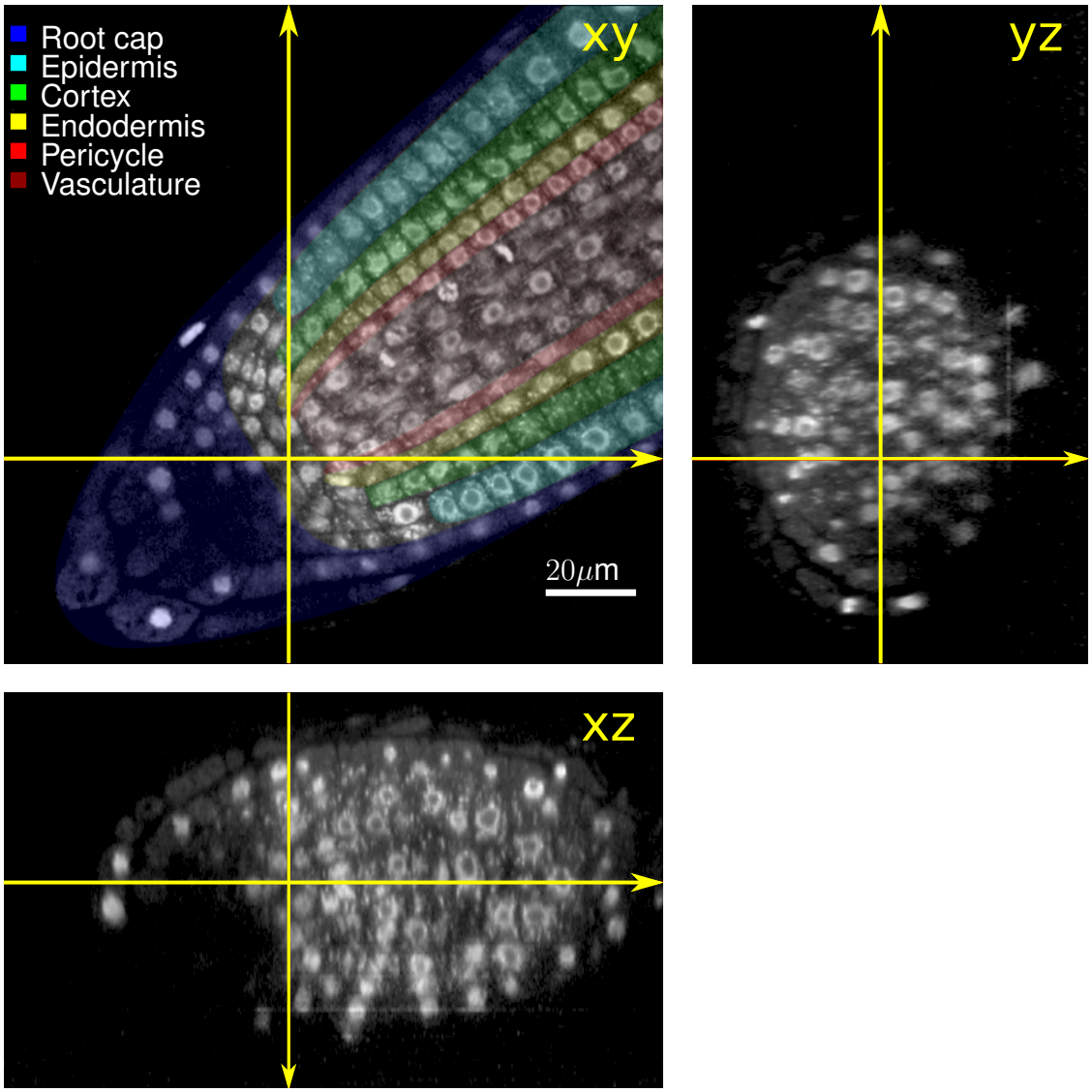


Figure 3.12.: Orthogonal views of a sample root tip after fixation and DAPI staining. Bright spherical structures correspond to the cell nuclei. The differentiated nuclei of the root tip and the dividing cells can be clearly distinguished from the undifferentiated interphase cells. The different layers of the root have been manually colorized in the xy-plane. The figure has been created by my colleague Thorsten Schmidt who kindly let me use it in this thesis for visualization.

3.2.2. Cell Nuclei in the *Arabidopsis Thaliana* Root Tip

The second dataset consists of volumetric CLSM recordings of *Arabidopsis thaliana* root tips. The recording, staining, and fixation have been done by Taras Pasternak at the Institute of Biology II, University of Freiburg. As for the *Drosophila* S2 cells, DAPI staining was used that binds to DNA. Bright regions thus correspond to nuclear chromatin. The nucleoli inside the nuclei are visible as dark "holes". Smaller bright spots around the nuclei correspond to mitochondria. In figure 3.12, orthogonal views of a root tip are shown. In the xy-section, the different layers in the root are

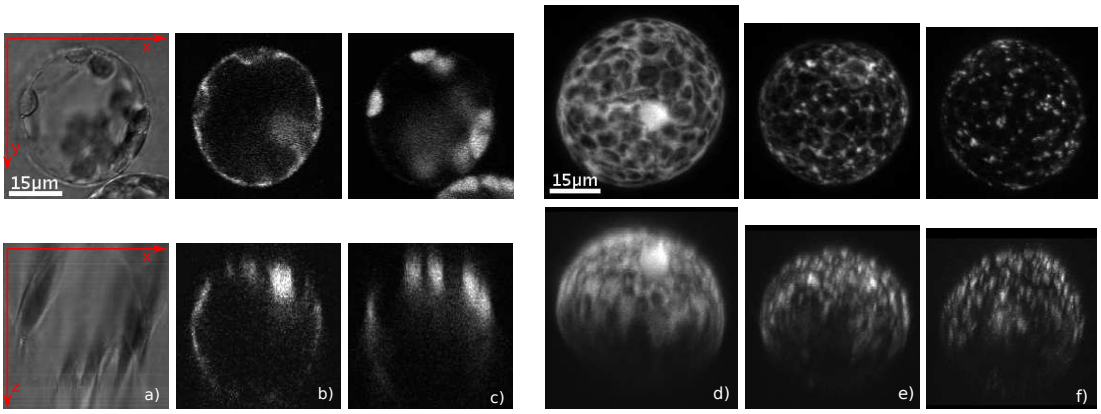


Figure 3.13.: (a-c) Slices of the recorded channels: a) differential interferent contrast (DIC) image. b) protein pattern (Cyt). c) chloroplasts. (d-e) maximum intensity projections in z-direction (top row) and in y-direction (bottom row) of the protein stainings: d) Cyt. e) ER. f) GA [8].

annotated. The size and shape of the nuclei varies with the layer and differentiation state. Since the specimen are quite thick, a signal attenuation due to absorption and light scattering can be observed in z-direction. In the visualization in figure 3.12, this attenuation has been alleviated with a γ-correction. In the tip, however, a dark region in the lower left part of the xz-section is still visible that originates from signal attenuation. For each root, two partially overlapping image stacks were recorded and stitched together using xuvTools [Emmenlauer et al., 2009][7]. These data have already been investigated for mitosis analysis in [Schulz et al., 2006; Skibbe et al., 2012], where detection algorithms are presented.

3.2.3. Tobacco Protoplast Cells

This dataset consists of volumetric CLSM recordings of living tobacco leaf protoplasts that were prepared and recorded by my colleague Robert Bensch and Karsten Voigt at the Institute of Biology II, University of Freiburg. Protoplasts are cells whose cell walls have been enzymatically removed. Plant protoplasts can be used to study the function of the plasma membrane, cellular reprogramming or cellular development [Dovzhenko et al., 1998]. An exact segmentation of the cells is needed for various applications, such as the description of the cell anatomy itself or of developmental processes in a meaningful anatomical coordinate system. Each recorded sample contains one to three single cells. The recorded resolution varies and is either $0.28 \times 0.28 \times 0.4\mu\text{m}^3$ or $0.28 \times 0.28 \times 0.5\mu\text{m}^3$. Besides the recorded noise, the most limiting fact for the recording quality was the absorption, resulting in a signal intensity attenuation in the z-direction (see bottom row in fig. 3.13). The cells have been recorded on three successive days. Directly after preparation, the cells are usually roundish with the chloroplasts mainly located on the bottom side. With time, they develop more elongated and complex shapes and the chloroplasts redistribute inside the cell. Three channels have been recorded, a transmitted light channel and two confocal fluorescence channels, one showing the auto-fluorescence of the chloroplasts and another showing the fluorescence of the

[7] The stitchings have been done by Thorsten Schmidt.
[8] This visualization of the data has been generated by Robert Bensch who kindly let me use it in this thesis.

tagged protein (see fig. 3.13a-c). The position of the chloroplasts is important because it influences the protein pattern.
Three different protein patterns have been recorded which all have a different appearance (see fig. 3.13d-f). While the cytoplasm (Cyt, 55 cells in the database) fills the whole space between the chloroplasts, the endoplasmatic reticulum (ER, 46 cells) forms a mesh structure and the golgi apparatus (GA, 86 cells) is organized in spot-like structures. All protein patterns are located in the outer shell of the cell, i.e. between outer cell membrane and inner vacuole membrane (cf. fig. 3.13b).

3.3. Transmission Electron Microscopy

The resolution of fluorescence microscopes is physically limited by the wavelength λ of the emitted photons and the NA (compare section 3.1). Imaging with electrons instead of light allows for much higher resolutions because the electrons can be accelerated to much shorter wavelengths. However, for transmission electron microscopes (TEM), the recording principles are quite similar to light microscopes and are only briefly summarized here. A detailed description of the functionalities of TEM is given in [Fultz and Howe, 2007]. In TEM, an electron emission source generates an electron beam that travels inside a vacuum. The lenses, consisting of electromagnets, direct the electron beam through the sample, such that only very thin samples, usually about 100nm thick, can be imaged. The observed image is the integration over the whole depth of the sample. The electron beam passing through the sample is made visible on a phosphor screen or recorded by a CCD camera. Constrast is usually formed by absorption of electrons in the sample (mass-thickness contrast). Thicker regions or regions with higher atomic number appear darker than regions with lower atomic number. Other mechanisms to produce contrast in the electron microscope are *diffraction contrast* and *phase contrast*.

3.4. TEM Data

The TEM dataset used in this thesis consists of 27 two-dimensional recordings of bone marrow-derived mast cells (BMMC). For the recording, the cells were cut into thin slices of approximately 90 nm thickness. The image size is 1024×1024 pixels with a resolution of $11.6 \times 11.6\text{nm}^2$ per pixel. Two example recordings are shown in figure 3.14. Fine details of the cell like membrane protrusions (called microvilli) or the lamella structure of mitochondria are visible. The recordings have been taken by Marta Rodriguez-Franco at the Faculty of Biology, University of Freiburg. For this dataset, ground truth annotations are available, segmenting the images into the five different classes background, cytoplasm, nucleus, mitochondria, and other vesicles.

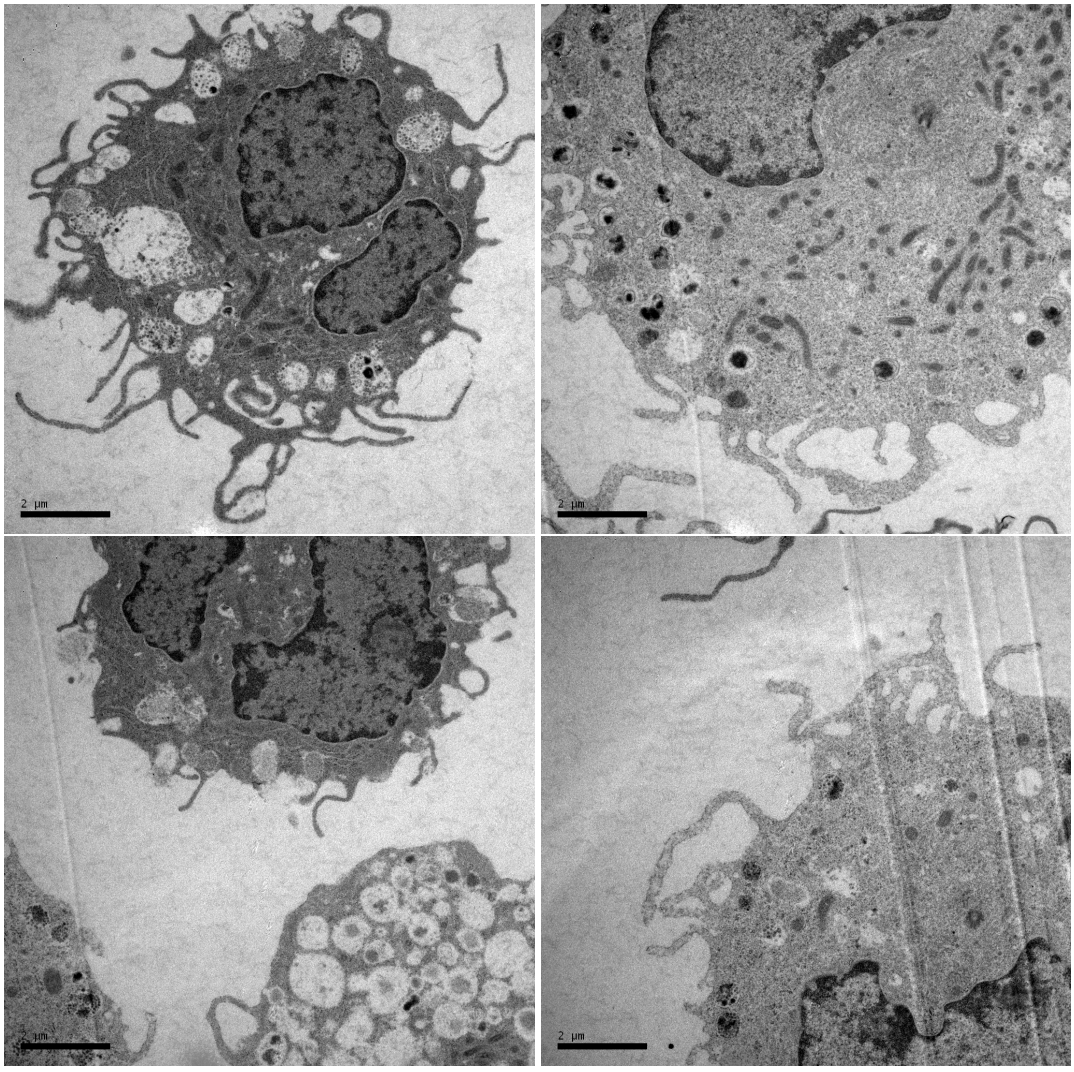


Figure 3.14.: TEM recordings of BMMC cells. The extended dark regions inside the cell correspond to the nucleus. Smaller, elongated dark structures are mitochondria.

4. Deconvolution

Recall that the image formation of fluorescence microscopes can be approximated by the equation

$$o = n(s * h), \tag{4.1}$$

where s is the specimen function, h the microscope specific PSF, and n a voxelwise noise function. Deconvolution techniques seek to reconstruct the specimen s from the image o under certain assumptions on the noise function n. Blind deconvolution methods do so by estimating the PSF at the same time, while non-blind deconvolution methods rely on a given PSF estimate (see section 3.1.3).

It is easy to see that the deconvolution of 3D widefield data is ill-posed because a part of the information is already lost even if the correct PSF was given (see again section 3.1.3). In blind deconvolution, the situation is even worse, because the number of variables to be estimated from the measured signal o increases drastically. However, blind deconvolution techniques can cope with deficient initial PSF estimates. This is beneficial, because the PSF depends on many parameters, is subjected to changes in the recording system, like thermal expansions, aberrations, or an unexpected behavior of the recorded specimen, and is therefore hard to estimate a priori.

For the deconvolution of widefield microscopic data, Maximum Likelihood Expectation Maximization (MLEM) deconvolution methods have proved to be very efficient [Sarder and Nehorai, 2006; Holmes, 1992; Conchello and McNally, 1996; Kenig et al., 2010]. However, they do not converge if no regularization is implemented [Sarder and Nehorai, 2006]. Blind deconvolution methods even tend to converge to the trivial solution, where the deconvolution kernel (the estimated PSF) collapses to the δ-impulse and the reconstructed specimen equals the recorded image. Thus, it is necessary to regularize either the specimen function or the deconvolution kernel or both.

In the following, we describe the blind MLEM deconvolution framework with different specimen function regularizations (see sections 4.2 - 4.5). For the regularization of the deconvolution kernel in the blind MLEM deconvolution framework, we propose two different methods. First, a zero order Tikhonov penalizer is employed for the regularization (see section 4.7.1). This approach has been evaluated with different specimen function regularizations together with Maja Temerinac-Ott and published in [Keuper et al., 2012]. With this very general penalty term, we can easily avoid the trivial solution and stabilize the blind MLEM scheme without introducing a complex PSF model. Secondly, we propose to use a regularization of the Fourier transform of the deconvolution kernel, employing its frequency space properties (see section 4.7.2). A preliminary implementation of this idea has been the subject of a student project co-supervised by Maja Temerinac-Ott [Koziolek, 2011]. Here I present a much improved version.

4.1. Related Work

Image deconvolution is a vast topic with applications in the fields of microscopic image reconstruction, motion deblurring, or the deblurring of astronomic images. The algorithms depend on

the assumptions on the noise function n. Assuming n is a function adding Gaussian noise to the recording, i.e.

$$o = s * h + n_G, \tag{4.2}$$

linear deconvolution methods such as the Wiener Filter [Wiener, 1949], the Tikhonov-Miller filter [Miller, 1970] or a simple pseudo-inverse filter can be employed. However, in diffraction-limited systems where certain frequencies are cut off during the recording process, linear methods have no means to recover any of these frequencies since zero multiplied by any number remains zero [Carasso, 1999]. Furthermore, linear methods provide no control over the range of the restored data. This is why non-linear image deconvolution methods based on statistic considerations are widely believed to outperform conventional linear methods [Carasso, 1999; Sarder and Nehorai, 2006], coming at the cost of higher computational complexity since generally, no direct computation of the solution is possible. A further advantage of non-linear methods is that more complex noise models as for example Poisson noise can be modeled. In [Sarder and Nehorai, 2006], an overview of the advantages and disadvantages of several linear and non-linear deconvolution methods for widefield microscopy is given. In [Carasso, 1999] the derivation of several nonlinear deconvolution methods as the Lucy-Richardson method [Lucy, 1974; Richardson, 1972], the Maximum Likelihood Expectation Maximization (MLEM) method [Shepp and Vardi, 1982], the Poisson maximum a posteriori method, and a maximum entropy method [Nuñez and Llacer, 1990] are described in detail.

The MLEM and the Lucy-Richardson method both lead to the same deconvolution algorithm although the derivations are different. MLEM was first proposed in [Shepp and Vardi, 1982] for the reconstruction of positron emission tomography recordings and is based on Poisson statistics. In the papers of Richardson [1972] and Lucy [1974], no assumptions are made on the specific statistical model for the noise process in the recording.

MLEM deconvolution methods have proved to be very efficient for image restoration and particularly for deconvolution of widefield microscopic data [Kenig et al., 2010; Markham and Conchello, 2001; Preza and Conchello, 2004]. Without using a prior model of the data to be reconstructed, MLEM tends to amplify noise and to introduce ringing artifacts [Sarder and Nehorai, 2006; Shaw, 1996]. However, using pre-denoised images, it has shown to be possible to use MLEM deconvolution methods without regularization [de Monvel et al., 2001]. More common methods use prior models such as the Tikhonov-Miller (TM) penalizer [vanKempen et al., 1997] or Total Variation (TV) regularization [Dey et al., 2004b,a, 2006], enforcing smoothness in the deconvolved data and thus reducing noise. In [Conchello and McNally, 1996], the authors use an intensity penalization for the data term, hindering the data from taking too high values. However, in [Conchello and McNally, 1996] not the maximum likelihood but the information divergence is optimized.

MLEM based blind deconvolution methods estimate not only the specimen function but also the PSF from the recorded data and an initial guess of the PSF [Markham and Conchello, 1999; Kenig et al., 2010]. In [Markham and Conchello, 1999; Kenig et al., 2010] the PSF update is regularized by a projection of the current estimate onto a given PSF model. In [Markham and Conchello, 1999], the parameters of the PSF model given in [Frisken-Gibson and Lanni, 1992] are estimated in each step, in [Kenig et al., 2010], the current estimate is projected onto the space spanned by all simulated PSFs using a kernel PCA. In [Kenig et al., 2010], the PSF simulation is done using the method of [Stokseth, 1969] which is similar to [Frisken-Gibson and Lanni, 1992]. [Kenig et al., 2010] proposes to additionally use a residual denoising technique for data and PSF restoration as done before in [Starck et al., 2002] for the deconvolution of astronomical images. In [Hom et al., 2007], an adaptive image deconvolution algorithm (AIDA) is presented: a blind (myopic) deconvo-

lution framework for 2D and 3D data. The basic assumption is that the PSF of the optical system is approximately known and either given in the form of the object transfer function (OTF, the Fourier transform of the PSF) or as several OTFs of which the true OTF is expected to be a linear combination. During the deconvolution process, the PSF estimation is bound to these given OTFs by the L_2-norm. Also the data fidelity term in [Hom et al., 2007] is based on the L_2 and in contrast to MLEM does not assume Poisson statistics. As [Kenig et al., 2010], [Pankajakshan et al., 2009] use the PSF model of [Stokseth, 1969] in the blind MLEM deconvolution framework. Similarly to [Hom et al., 2007], they bind the estimated PSF with a quadratic term to the simulated PSF.

4.2. Maximum Likelihood Expectation Maximization Deconvolution

Even though the Lucy-Richardson deconvolution and the MLEM deconvolution approach lead to the same update scheme, we present our methods in the MLEM framework, because it easily allows for extensions. Priors on the specimen function as well as on the deconvolution kernel can be incorporated in a natural way using Bayes' theorem.
For the derivation of the MLEM algorithm we follow [Carasso, 1999]. The aim of the MLEM algorithm is to find the specimen function estimate that maximizes the conditional probability of the measured objective function:

$$\hat{s} = \arg\max_{s}\{p(o|s,h)\}, \tag{4.3}$$

where o denotes the objective function, s the specimen function, and h the PSF from equation (3.1). Since photon noise is Poisson distributed [Boyd, 1983], the recorded objective function $o(\mathbf{x})$ at voxel $\mathbf{x}$ can be viewed as a sample of a Poisson distribution with mean $(s * h)(\mathbf{x})$ [Carasso, 1999], such that

$$p(o(\mathbf{x})|s,h) = \frac{((s*h)(\mathbf{x}))^{o(\mathbf{x})}\exp\left(-(s*h)(\mathbf{x})\right)}{o(\mathbf{x})!}. \tag{4.4}$$

Since the intensity at each voxel in o is drawn from an independent Poisson process, the objective function o given the specimen s and the PSF h can be expressed in an ML way as the product

$$p(o|s,h) = \prod_{\mathbf{x}\in\Omega\subseteq\mathbb{R}^3} \frac{((s*h)(\mathbf{x}))^{o(\mathbf{x})}\exp\left(-(s*h)(\mathbf{x})\right)}{o(\mathbf{x})!}, \tag{4.5}$$

where $\Omega \subseteq \mathbb{R}^3$ is the recording domain. Instead of maximizing (4.5), it is equivalent to minimize the negative log-likelihood function

$$-\log(p(o|s,h)) = \int_{\Omega} -o(\mathbf{x})\cdot\log(s*h)(\mathbf{x}) + (s*h)(\mathbf{x}) + \log(o(\mathbf{x})!)d\mathbf{x}. \tag{4.6}$$

The term $\log(o(\mathbf{x})!)$ is constant relative to s and can thus be ignored in the optimization. The functional that has to be minimized is

$$J_{\text{MLEM}}(s) = \int_{\Omega} (s*h)(\mathbf{x}) - o(\mathbf{x})\cdot\log(s*h)(\mathbf{x})d\mathbf{x}. \tag{4.7}$$

The minimization of 4.7 is equivalent to finding a solution to $\nabla J(s) = 0$, where $\nabla J(s)$ is found using the calculus of variation (see appendix A). $\nabla J(s) = 0$ yields the Euler-Lagrange equation

$$\int_{\Omega} h^m(\mathbf{x})d\mathbf{x} - \left(h^m * \frac{o}{(h*s)}\right)(\mathbf{x}) = 0, \tag{4.8}$$

where $h^m(\mathbf{x}) = h(-\mathbf{x})$ is the mirrored PSF and where the PSF h is assumed to integrate to one: $\int_\Omega h^m(\mathbf{x})d\mathbf{x} = 1$. This equals

$$\left(h^m * \frac{o}{(h * s)}\right)(\mathbf{x}) = 1. \tag{4.9}$$

Given this equation, two different optimization schemes can be derived, namely an additive scheme and a multiplicative scheme. The additive scheme uses the fact that at convergence, the difference $\hat{s}_{k+1} - \hat{s}_k = 0$, yielding the gradient descent algorithm

$$\hat{s}_{k+1}(\mathbf{x}) = \hat{s}_k(\mathbf{x}) + \delta t \left(\left(h^m * \frac{o}{(h * \hat{s}_k)}\right)(\mathbf{x}) - 1\right), \tag{4.10}$$

where δt denotes the timestep.
In order to derive the multiplicative scheme, the property that at convergence $\frac{\hat{s}_{k+1}}{\hat{s}_k} = 1$ is employed. The resulting iterative MLEM scheme is the same as in the Lucy-Richardson algorithm:

$$\hat{s}_{k+1}(\mathbf{x}) = \hat{s}_k(\mathbf{x}) \cdot \left(\left(h^m * \frac{o}{(h * \hat{s}_k)}\right)(\mathbf{x})\right). \tag{4.11}$$

This update scheme has some desired properties such as the conservation of the positivity and the conservation of the radiant flux [Carasso, 1999]. However it does not converge if no regularization is implemented [Sarder and Nehorai, 2006] and thus, stopping criteria are usually employed. Furthermore, one has to take care for the case that the denominator becomes zero. To avoid numerical problems, we set $\hat{s}_{k+1}(\mathbf{x}) = 0$ if $(h * \hat{s}_k)(\mathbf{x}) = 0$. As an alternative, one could also add an epsilon to the denominator.

4.3. Blind EM Deconvolution

In the blind deconvolution framework, we seek to estimate both the specimen function $\hat{s} = \arg\max_s\{p(o|s,h)\}$ and the PSF $\hat{h} = \arg\max_h\{p(o|s,h)\}$. We thus have to differentiate equation (4.7) with respect to s and h and set the derivative to zero. The derivative with respect to h is given by [Kenig et al., 2010]

$$\left(s^m * \left(\frac{o}{(h * s)} - 1\right)\right)(\mathbf{x}) = 0, \tag{4.12}$$

where $s^m(\mathbf{x}) = s(-\mathbf{x})$ is the mirrored specimen function. It follows that

$$\left(s^m * \frac{o}{(h * s)}\right)(\mathbf{x}) = \int_\Omega s(\mathbf{y})\, d\mathbf{y}. \tag{4.13}$$

The additive update scheme for estimating the PSF h is thus

$$\hat{h}_{k+1}(\mathbf{x}) = \hat{h}_k(\mathbf{x}) + \delta t \left(\hat{s}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)}\right)(\mathbf{x}) - \int_\Omega \hat{s}_k(\mathbf{y})\, d\mathbf{y}. \tag{4.14}$$

and the multiplicative update scheme is given by

$$\hat{h}_{k+1}(\mathbf{x}) = \frac{\hat{h}_k(\mathbf{x})}{\int_\Omega \hat{s}_k(\mathbf{y})d\mathbf{y}} \cdot \left(\hat{s}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)(\mathbf{x}). \tag{4.15}$$

The blind MLEM deconvolution algorithm is given by alternately updating $\hat{h}$ with equation (4.15) and $\hat{s}$ with

$$\hat{s}_{k+1}(\mathbf{x}) = \hat{s}_k(\mathbf{x}) \cdot \left(\hat{h}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)(\mathbf{x}). \tag{4.16}$$

4.4. Residual Denoising

Since the measured objective function o is noisy, this update scheme can lead to amplified noise in the deconvolved data. This effect can be reduced by denoising the residual R_k defined by

$$o = \hat{s}_k * \hat{h}_k + R_k \leftrightarrow R_k = o - \hat{s}_k * \hat{h}_k \tag{4.17}$$

Since the function h is considered to be a blurring kernel and $\hat{h}_k$ is an approximation of h, $\hat{s}_k * \hat{h}_k$ should be noise-free such that all the noise of o is contained in R_k [Kenig et al., 2010]. R_k can be denoised with any denoising function as wavelet denoising [Starck et al., 2002] or median filtering [Kenig et al., 2010]

$$\bar{R}_k = \text{denoise}(R_k). \tag{4.18}$$

We decided to follow [Kenig et al., 2010] in using a $3 \times 3 \times 3$ median filter. With this, the residual denoised objective function $\bar{o} = \hat{s}_k * \hat{h}_k + \bar{R}_k$ can be computed. $\bar{o}$ is used instead of o in the update scheme.
The multiplicative MLEM scheme with residual denoising is then:

$$\hat{s}_{k+1} = \hat{s}_k \cdot \left(\hat{h}_k^m * \frac{\hat{s}_k * \hat{h}_k + \bar{R}_k}{(\hat{h}_k * \hat{s}_k)} \right) \tag{4.19}$$

$$\hat{h}_{k+1} = \frac{\hat{h}_k}{\int_\Omega \hat{s}_k(\mathbf{y})d\mathbf{y}} \cdot \left(\hat{s}_k^m * \frac{\hat{s}_k * \hat{h}_k + \bar{R}_k}{(\hat{h}_k * \hat{s}_k)} \right) \tag{4.20}$$

4.5. Regularization

A possibility to introduce prior knowledge on the specimen function or the PSF into the deconvolution process is the penalized MLEM algorithm [Green, 1990a,b]. Penalized MLEM does not optimize the maximum likelihood $p(o|s,h)$ but the maximum a-posteriori probability (MAP). In the blind deconvolution case, we thus seek to find

$$\hat{s}, \hat{h} = \arg\max_{s,h} \{p(s,h|o)\}. \tag{4.21}$$

From Bayes rule, we know that $p(s,h|o) = \frac{p(o|s,h)p(s,h)}{p(o)}$. Since s and h can be assumed to be statistically independent:

$$p(s,h|o) = \frac{p(o|s,h)p(s)p(h)}{p(o)}. \tag{4.22}$$

Again, we maximize $p(s,h|o)$ by minimizing the negative log-likelihood

$$-\log(p(s,h|o)) = -\log(p(o|s,h)) - \log(p(s)) - \log(p(h)) + \log(p(o)), \tag{4.23}$$

where $-\log(p(o|s,h))$ is given by equation (4.6). As [Kenig, 2009], we define the general prior probability functions

$$p(s) = \exp(-\lambda_{\mathrm{D}} P_{\mathrm{D}}(s))$$

for the data and

$$p(h) = \exp(-\lambda_{\mathrm{K}} P_{\mathrm{K}}(h)) \tag{4.24}$$

for the kernel. $P_{\mathrm{D}}(s)$ and $P_{\mathrm{K}}(h)$ can be arbitrary penalty functionals, returning high values for inputs that do not match the prior knowledge and low values for inputs that do. λ_{D} and λ_{K} are positive constants, that serve as weights for the penalty terms. Since $p(o)$ does not depend on s and h, the last term of equation (4.23) can be omitted for the optimization, finally yielding the penalized MLEM functional:

$$J_{\mathrm{PMLEM}}(s,h) = J_{\mathrm{MLEM}}(s,h) + \lambda_{\mathrm{D}} P_{\mathrm{D}}(s) + \lambda_{\mathrm{K}} P_{\mathrm{K}}(h). \tag{4.25}$$

Usually, the optimization is implemented using Greens multiplicative *one step late* (OSL) algorithm [Green, 1990b], even though it may fail to converge if there occur oscillations in the derivative of the penalty term. The OSL update is given by

$$\hat{h}_{k+1}(\mathbf{x}) = \frac{\hat{h}_k(\mathbf{x})}{\int_\Omega \hat{s}_k(\mathbf{y})d\mathbf{y} + \lambda_{\mathrm{K}} \frac{\partial}{\partial h} P_{\mathrm{K}}(h)|_{h_k}(\mathbf{x})} \left(\hat{s}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)(\mathbf{x}). \tag{4.26}$$

and

$$\hat{s}_{k+1}(\mathbf{x}) = \frac{\hat{s}_k(\mathbf{x})}{1 + \lambda_{\mathrm{D}} \frac{\partial}{\partial s} P_{\mathrm{D}}(s)|_{s_k}(\mathbf{x})} \left(\hat{h}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)(\mathbf{x}). \tag{4.27}$$

In the following, we present some common penalty terms based on image statistics as well as two new methods for the kernel regularization.

4.6. Specimen Function Regularization

Many statistical priors for the specimen function have been investigated in literature, from simple intensity regularizations preventing the reconstructed specimen function from collapsing [Conchello and McNally, 1996] to more elaborate priors as the Gaussian Prior [Verveer and Jovin, 1997] that can be considered as a generalization of the Tikhonov-Miller regularization [vanKempen et al., 1997], the entropy prior [Verveer and Jovin, 1997], that minimizes the entropy in the intensity distribution, or Total Variation regularization [Dey et al., 2004a, 2006]. While the intensity regularization adds very little prior knowledge to the functional, most other terms are based on the assumption that the specimen function is in some way continuous or smooth.

4.6.1. Intensity Penalty

For example in [Conchello and McNally, 1996] the zero order Tikhonov stabilizer is used as regularization for the specimen estimation. This simple regularizer avoids that the specimen function

takes on too high values and leaves the range of the recorded image. The energy functional is given by

$$J_{\text{IP}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_{\text{IP}} \int_{\Omega} s(\mathbf{x})^2 d\mathbf{x} + \lambda_{\text{K}} P_{\text{K}}(h) \tag{4.28}$$

where λ_{IP} is the weighting for the intensity penalty regularization. The according Euler-Lagrange equation for the specimen function optimization is

$$1 - \left(h^m * \frac{o}{(h * s)} \right)(\mathbf{x}) + 2\lambda_{\text{IP}} s(\mathbf{x}) = 0, \tag{4.29}$$

and the resulting, regularized OSL update for the specimen function is

$$\hat{s}_{k+1}(\mathbf{x}) = \frac{\hat{s}_k(\mathbf{x})}{1 + 2\lambda_{\text{IP}} \hat{s}_k(\mathbf{x})} \cdot \left(\hat{h}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)(\mathbf{x}). \tag{4.30}$$

4.6.2. Tikhonov-Miller

In order to enforce smoothness in the reconstructed image, the quadratic TM regularizer can be employed [vanKempen et al., 1997]. The energy functional becomes

$$J_{\text{TM}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_{\text{TM}} \int_{\Omega} \|\nabla s(\mathbf{x})\|^2 d\mathbf{x} + \lambda_{\text{K}} P_{\text{K}}(h) \tag{4.31}$$

where λ_{TM} is the weighting factor for the TM regularization. The Euler-Lagrange equation is given by

$$1 - \left(h^m * \frac{o}{(h * s)} \right)(\mathbf{x}) - 2\lambda_{\text{TM}} \Delta s(\mathbf{x}) = 0, \tag{4.32}$$

where $\Delta s = \frac{\partial^2 s}{\partial x_1^2} + \frac{\partial^2 s}{\partial x_2^2} + \frac{\partial^2 s}{\partial x_3^2}$ is the Laplace operator.
The TM regularized specimen estimation update scheme is:

$$\hat{s}_{k+1}(\mathbf{x}) = \frac{\hat{s}_k(\mathbf{x})}{1 - 2\lambda_{\text{TM}} \Delta \hat{s}_k(\mathbf{x})} \cdot \left(\hat{h}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)(\mathbf{x}). \tag{4.33}$$

The discrete computation of the Δs is given in the appendix B.1.

4.6.3. Total Variation

An alternative is the Total Variation (TV) regularizer [Dey et al., 2004a, 2006]. TV does not enforce smoothness but piecewise constancy. Thus, it depends on the specimen which of the two models performs better. The energy functional with a TV regularizer on the data term is:

$$J_{\text{TV}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_{\text{TV}} \int_{\Omega} \|\nabla s(\mathbf{x})\| d\mathbf{x} + \lambda_{\text{K}} P_{\text{K}}(h), \tag{4.34}$$

where λ_{TV} is the weighting for the TV regularization. The Euler-Lagrange equation is given by:

$$1 - \left(h^m * \frac{o}{(h * s)} \right)(\mathbf{x}) - \lambda_{\text{TV}} \text{div} \left(\frac{\nabla s(\mathbf{x})}{\|\nabla s(\mathbf{x})\|} \right) = 0, \tag{4.35}$$

where $\text{div}(s) = \frac{\partial s}{\partial x_1} + \frac{\partial s}{\partial x_2} + \frac{\partial s}{\partial x_3}$ is the divergence. The TV regularized specimen estimation update scheme is:

$$\hat{s}_{k+1}(\mathbf{x}) = \frac{\hat{s}_k(\mathbf{x})}{1 - \lambda_{\text{TV}} \text{div}\left(\frac{\nabla \hat{s}_k(\mathbf{x})}{\|\nabla \hat{s}_k(\mathbf{x})\|}\right)} \cdot \left(\hat{h}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)}\right)(\mathbf{x}). \tag{4.36}$$

For the computation of the divergence term $\text{div}\left(\frac{\nabla s(\mathbf{x})}{\|\nabla s(\mathbf{x})\|}\right)$, we follow [Dey et al., 2006] in using the numerically stable scheme presented in [Rudin et al., 1992] for two-dimensional image denoising extended to three dimensions (see appendix B.2).

4.7. Kernel Regularization

In order to regularize the deconvolution kernel for the blind deconvolution, we have followed two different approaches. The first approach is based on a very simple intensity based regularization, preventing the kernel from collapsing. This approach has been evaluated together with different specimen function regularizations in [Keuper et al., 2012]. In [Cho and Lee, 2009] an intensity penalization has been used for the kernel regularization in a motion deblurring application. However [Cho and Lee, 2009] as well as [Hom et al., 2007] use the L_2 norm and a Gaussian noise model for the data fitting term resulting in a different optimization scheme than the one we use.
A second, more elaborate approach uses information on the frequency spectrum of the PSF in order to regularize the deconvolution kernel.

4.7.1. Kernel Intensity Penalty

To stabilize the deconvolution kernel, we add a zero order Tikhonov stabilizer to the energy functional, that penalizes too high intensities in the kernel. With this regularization, we can avoid the trivial solution, where h is the δ-impulse and the reconstructed specimen function equals the objective function. The penalized MLEM functional with kernel intensity penalization (KIP) is

$$J_{\text{KIP}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_{\text{D}} P_{\text{D}}(s) + \lambda_{\text{KIP}} \int_\Omega h(\mathbf{x})^2 d\mathbf{x}, \tag{4.37}$$

where λ_{KIP} is the weighting for the intensity penalty regularization of the kernel. The resulting Euler Lagrange equation is

$$\int_\Omega s(\mathbf{y}) d\mathbf{y} - \left(s^m * \frac{o}{(h * s)}\right)(\mathbf{x}) + 2 \cdot \lambda_{\text{KIP}} h(\mathbf{x}) = 0. \tag{4.38}$$

The update scheme for the kernel is given by

$$\hat{h}_{k+1}(\mathbf{x}) = \frac{\hat{h}_k(\mathbf{x})}{\int_\Omega \hat{s}_k(\mathbf{y}) d\mathbf{y} + 2 \cdot \lambda_{\text{KIP}} \hat{h}_k(\mathbf{x})} \cdot \left(\hat{s}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)}\right)(\mathbf{x}). \tag{4.39}$$

4.7.2. TV in the Frequency Domain

For widefield microscopic recordings, we know that the PSF is actually not smooth (compare section 3.1.1), such that a straight forward regularization of the kernel by imposing smoothness is not reasonable.
Therefore, we want to employ the properties of the PSF in the frequency domain for regularization. We decompose the Fourier transform of our PSF h into its amplitude and phase (or argument) ϕ:

$$\mathcal{F}(h) = \underbrace{|\mathcal{F}(h)|}_{\text{amplitude}} \cdot \underbrace{\frac{\mathcal{F}(h)}{|\mathcal{F}(h)|}}_{e^{\imath\phi}}. \tag{4.40}$$

$\mathcal{F}(h)$ denotes the Fourier transform of h. Now, from section 3.1.1, we know that the OTF (the Fourier transform of the PSF) fulfills several properties: Its support is limited and inside its domain, its values vary smoothly. Therefore, we formulate an energy with frequency domain TV regularization of the deblurring kernel (KFTV) as

$$J_{\text{KFTV}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_{\text{D}} P_{\text{D}}(s) + \lambda_{\text{KFTV}} \int \Big\| \nabla |\mathcal{F}(h)|(\xi) \Big\| d\xi. \tag{4.41}$$

It is easy to see that the regularization term equals a TV regularization of the amplitudes of the kernel in the frequency domain. The optimum of the TV energy term, a constant amplitude of the OTF, does not account for the fact that the support of the OTF is limited. We therefore ensure a limited support by generating a mask for the OTF. Outside this mask, its values have to be zero. This idea of setting certain PSF frequencies to zero has before been formulated in [Holmes, 1992] and has been tested on two-dimensional data.
In order to make sure that our mask does not cut off any valid frequencies, we compute the largest theoretically possible OTF support for a widefield microscope according to [Philip, 2009]. The OTF is largest for the smallest possible emission wavelength and the largest possible $\sin(\Theta)$ where Θ is the angle of the maximum cone of light that can enter the objective lens (see section 3.1.3, figure 3.6). We thus compute the maximal OTF-support with wavelength $\lambda = 380$nm and $\Theta = \frac{\pi}{2} - \epsilon$. See figure 4.1 for a visualization. Values outside this support are set to zero in every iteration of the deconvolution update. This operation can introduce negative values in the current PSF estimate. In order to reestablish the non-negativity of the PSF estimate, [Holmes, 1992] proposed to cut off negative PSF values and set them to zero. This non-linear operation caused numerical problems in our data. It reintroduces frequencies outside the OTF support. Additionally, PSF values that are cut off will not be able to take on any value different from zero in the multiplicative update scheme. Instead of setting negative values to zero, we normalize the PSF to a valid range to avoid of these problems.
The minimization of functional (4.41) is again done using the calculus of variations. Since the Fourier transform is linear, a variation in the spatial domain can be written in the Fourier domain as

$$\int \Big\| \nabla |\mathcal{F}(h + \epsilon g)|(\xi) \Big\| d\xi = \int \Big\| \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)|(\xi) \Big\| d\xi. \tag{4.42}$$

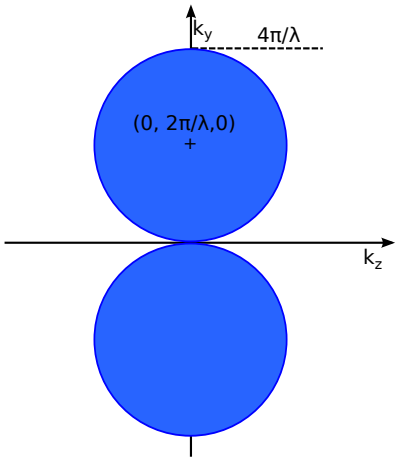


Figure 4.1.: For $\Theta \to \frac{\pi}{2}$, the OTF support tends to a horn torus with radius $\frac{2\pi}{\lambda}$.

The gradient of the kernel regularization term $\int \left\| \nabla |\mathcal{F}(h)|(\xi) \right\| d\xi$ can be computed as described in appendix C, yielding the following Euler-Lagrange equation

$$\int_\Omega s(\mathbf{y}) d\mathbf{y} - \left(s^m * \left(\frac{o}{(h * s)} \right) \right) - \lambda_{\text{KFTV}} \mathcal{F}^{-1} \left(\text{div} \left(\frac{\nabla |\mathcal{F}(h)|}{\left\| \nabla |\mathcal{F}(h)| \right\|} \right) \cdot \underbrace{\frac{\mathcal{F}(h)}{|\mathcal{F}(h)|}}_{e^{\imath \cdot \arg(\mathcal{F}(h))}} \right) = 0. \quad (4.43)$$

The resulting multiplicative update scheme for the deconvolution kernel is:

$$\hat{h}_{k+1} = \frac{\hat{h}_k \cdot \left(\hat{s}_k^m * \frac{o}{(\hat{h}_k * \hat{s}_k)} \right)}{\int_\Omega \hat{s}_k(\mathbf{y}) d\mathbf{y} - \lambda_{\text{KFTV}} \mathcal{F}^{-1} \left(\text{div} \left(\frac{\nabla |\mathcal{F}(\hat{h}_k)|}{\left\| \nabla |\mathcal{F}(\hat{h}_k)| \right\|} \right) \cdot \frac{\mathcal{F}(\hat{h}_k)}{|\mathcal{F}(\hat{h}_k)|} \right)}. \quad (4.44)$$

Implementation Since, for efficiency reasons, we want to compute the convolutions using the Fast Fourier transform [Cooley and Tukey, 1965] [1], there are some issues to mention here. If during the recording, the Nyquist criterion is fulfilled, i.e. the recorded resolution is high enough, the true sampled OTF completely fits into the recorded array. The largest possible OTF however might be larger, i.e. allow for higher frequencies. As a consequence, our method, as any unregularized method, allows for high frequencies in the reconstructed PSF. As the multiplication in the spatial domain corresponds to a convolution in frequency domain, these high frequencies potentially cause problems. They are wrapped with other frequencies (aliasing) in the circular convolution computed by the FFT. These aliasing effects can be avoided by choosing a sufficient spatial resolution. In our experiments, a resolution of $0.2 \times 0.064 \times 0.064 \mu\text{m}^3$, which is commonly used in practice, was sufficient to avoid artifacts.

[1] We use the FFTW implementation `http://www.fftw.org`.

A similar problem arises from the PSF in the spatial domain. Since the OTF has a limited support, the true PSF does not. Of course, its values can only be reconstructed in a finite array. If this array is chosen too small, the reconstructed PSF can have non-zero values at the array boundaries such that zero-padding for the convolutions introduces artifacts. In our implementation, we initially extend the array by padding it with zeros, assuming that the initial PSF estimate drops to zero at the array boundaries. During the deconvolution process, we update the PSF in the whole, extended array, even though we know that this might again lead to aliasing effects after applying a circular convolution, if the PSF grows too large. A possible remedy for this problem would be to iteratively increase the array size. However, since our implementation performed well during the experiments, we did not try this potentially very time and memory consuming alternative.

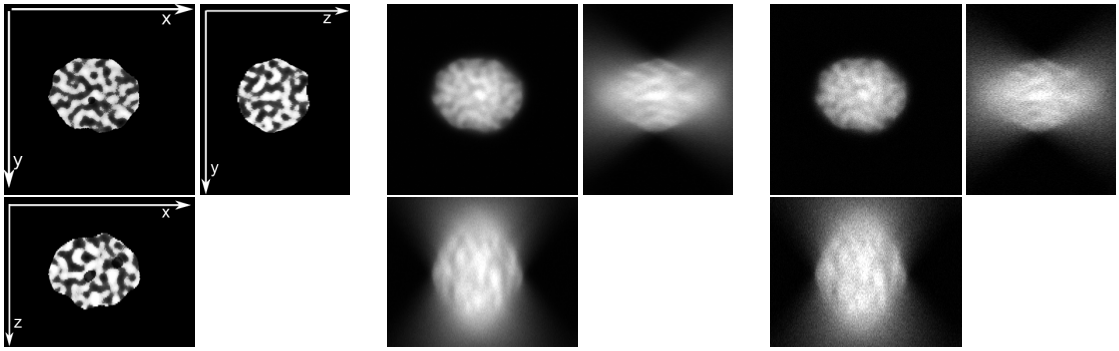
4.8. Test Datasets

Two datasets were used for the quantitative and qualitative evaluation of the presented blind deconvolution methods. The first dataset is purely synthetic and thus allows to give reliable numbers for the method evaluation. This dataset will be described in section 4.8.1. Since synthetic datasets obey the underlying model of the imaging process, it is always requisite to have a look at real microscopic recordings, as well. The second dataset used for evaluation consists thus of two microscopic recordings of the identical specimen: one taken with a widefield microscope and one taken with a spinning disk microscope. For lack of the true ground truth, we use the spinning disk recording instead, although we know that this dataset is itself blurred by a small PSF. This dataset will be described in 4.8.2.

4.8.1. Recording Simulation

For an authentic specimen simulation, we used the HeLa Cell Nucleus simulation tool from [Svoboda et al., 2009]. A slice view of the simulated cell nucleus is shown in figure 4.2(a). For the recording, we assumed a background fluorescence of 1.2% of the maximum specimen fluorescence. The specimen was blurred with an ideal widefield PSF without spherical aberration downloaded from [2]. The PSF was generated assuming a refractive index of 1.518, numerical aperture NA $= 1.4$, wavelength $\lambda = 530$nm, spatial resolution $\delta r = 64.5$nm and an axial resolution $\delta z = 160$nm in a volume of total size $16.4 \times 16.4 \times 16 \mu\text{m}^3$.
The simulated HeLa Cell nucleus was first blurred with this PSF. Then, we added two different levels of Poisson noise, resulting in *signal to noise ratios* (SNR) of 18.588 and 27.552 (where the blurred PSF is considered the signal). The SNRs from the final simulations to the ground truth PSF are 1.628 (SIM 1) and 1.6135 (SIM 2) respectively.
For non-blind deconvolution, the PSF of the recording must be given in advance, but also for blind deconvolution, it is beneficial to have a good initial estimate of the PSF. Especially for the blind deconvolution of widefield data, where the PSF is large, a good initial estimate is essential. For our experiments on real data, we determined an initial PSF estimate by measuring bead recordings. For the simulation, we also simulated this PSF estimation step. The bead recording was simulated by convolving a small sphere with a radius of 90nm with the simulated PSF. A background fluorescence of 1% of the maximum intensity value L was added and a volume of $6.5 \times 6.5 \times 8.3 \mu\text{m}^3$ was cropped around the simulated bead in order to simulate a finite bead recording. Then we added

[2] `http://bigwww.epfl.ch/deconvolution/?p=bead`.



(a) Ground Truth (b) SIM 1 (c) SIM 2

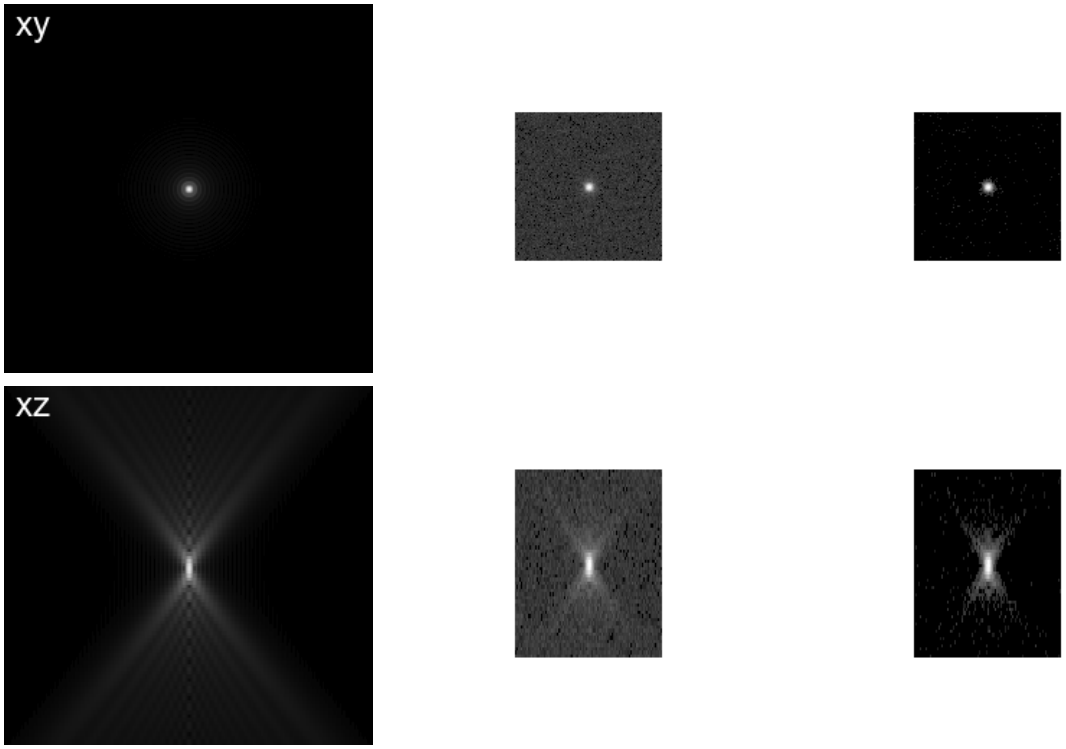
Figure 4.2.: Simulated HeLa cell nucleus and the simulated recordings with different noise levels displayed in three orthogonal views. In every view, the central slice is shown.

Poisson noise. The resulting SNR was 2.9856 in the cropped volume. Since the blind deconvolution only seeks to reconstruct the PSF in a finite, pre-defined volume, we considered the cropped original PSF inside the same volume ($6.5 \times 6.5 \times 8.3 \mu\text{m}^3$) as ground truth.
The estimated background fluorescence was 1.08%. Assuming a Poisson distribution p with $\lambda_p = 0.0108 \cdot L$, we fit the scaling of the Poisson distribution to a stepsize of $s_p = 0.0500$. Based on this distribution, we set a threshold such that it is larger than 95% of the resulting samples. Everything smaller than this threshold was estimated as background.

4.8.2. Real Data

The microscopic recordings were taken from a fixed DAPI stained *Drosophila* S2 cell nucleus. For the widefield microscope, a voxel size of $0.0642 \times 0.0642 \mu\text{m}^2$ in xy direction and 0.2μm in z-direction was used. With the same microscope settings and voxel size, a TetraSpeck 0.2μm bead was recorded four times. The average of the bead recordings served us as initial PSF estimate. In figure 3.9, we have shown an xz-section of one of the bead recordings and the initial PSF estimate, as an example for the PSF measurement (compare section 3.1.3.2).
The identical *Drosophila* S2 cell nucleus was recorded with a spinning disk confocal microscope for comparison. The spinning disk recording was taken with a voxel size of $0.1 \times 0.1 \mu\text{m}^2$ in xy direction and 0.2μm in z-direction. The preparations and recordings were done by Jan Padeken.
Since the PSF of the spinning disk microscope is very small, this data serves us as ground truth for the widefield deconvolution. The two recordings were registered using affine registration from the Matlab tool SPM[3]. The central slice and an xz-section of the spinning disk recording after registration is shown in figure 4.8(a), the same views of the widefield recording are displayed in figure 4.8(b). The SNR computed between the recordings from the different modalities is 0.9229.

[3] `http://www.fil.ion.ucl.ac.uk/spm/software/`. The registration was computed by my colleague Maja Temerinac-Ott.



(a) Simulated PSF (b) Simul. Bead Recording (c) Initial PSF Estimate

Figure 4.3.: Simulated PSF, bead recording and result after background subtraction (initial PSF estimate for blind deconvolution). For all three data sets, the central xy and xz slices are shown. The figures were made using a γ-correction with $\gamma = 1/3$ to make low intensities visible.

4.9. Evaluation

The evaluation of deconvolution methods is intricate. Assuming the ground truth is given, quantitative results can be reported in terms of different qualtiy or error measures, as there are for example: the *mean squared error* (MSE), the *root mean squared error* (RMSE), the *signal to noise ratio* (SNR), or the *peak signal to noise ratio* (PSNR). These methods build upon the MSE:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \big(\text{GT}(i) - \text{estimate}(i)\big)^2 . \tag{4.45}$$

However, it is easy to see that the MSE of identical images with a high difference in only one pixel might be higher than the MSE of two images that differ in many positions. Thus, numbers resulting from these quality measures have to be handled carefully. A qualitative inspection of the deconvolution results is indispensable. A further important fact one has to keep in mind is that all these measures depend on the range of the data.
For our experimental evaluation, we choose to use the *signal to noise ratio* (SNR), computed on datasets normalized by their mean value as

$$\text{SNR} = 10 \log_{10} \frac{\sum_{i=1}^{N} \big(\text{GT}(i) - \mu\big)^2}{\sum_{i=1}^{N} \big(\text{GT}(i) - c \cdot \text{estimate}(i)\big)^2}, \tag{4.46}$$

with

$$c = \frac{\sum_{i=1}^{N} \mathrm{GT}(i)}{\sum_{i=1}^{N} \mathrm{estimate}(i)}. \tag{4.47}$$

where GT denotes the ground truth and μ is its mean value. The normalization by c accounts for different scalings of the data [Kenig et al., 2010].
In [Kenig et al., 2010], the results are given in terms of *root mean squared error* (RMSE). For the comparison of KFTV against the [Kenig et al., 2010], we thus also report the RMSE of our results, computed as in [Kenig et al., 2010] according to

$$\mathrm{RMSE} = \left(\frac{1}{N} \sum_{i=1}^{N} \left(\mathrm{GT}(i) - c \cdot \mathrm{estimate}(i) \right)^2 \right)^{\frac{1}{2}}. \tag{4.48}$$

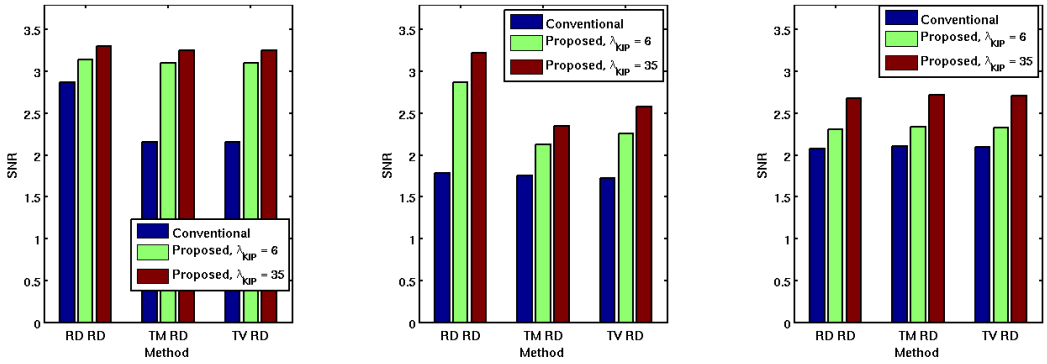
4.10. Experiments and Results

First, we want to evaluate the kernel intensity penalizer (KIP) together with possible regularizations for the specimen function in section 4.10.1. As a baseline, we first compute the deconvolution of our test datasets using the commonly used specimen function regularizations Tikhonov-Miller (TM), and Total Variation (TV) and, for comparison, the residual denoising (RD) technique. For these baseline experiments, the deconvolution kernel is updated using RD without any further regularization term. The results of these experiments are compared to the results that can be achieved using the kernel intensity penalty (KIP) for the regularization of the deconvolution kernel.
Since results with TV and TM regularization are expected to yield comparable results depending on the properties of the recorded specimen, the kernel frequency Total Variation (KFTV) method is only evaluated in conjuction with the TV regularization for the specimen function and the RD update scheme. Additionally to the evaluation on our own test datasets, we evaluate KFTV on the datasets used in [Kenig et al., 2010] for evaluation. The results of KFTV will be presented in 4.10.3.
For variational methods, the choice of appropriate weighting parameters is crucial. Unfortunately, the different parameters for the specimen function and for deconvolution kernel regularization influence each other. For efficiency reasons, we nevertheless tested the parameters for the specimen functions in the non-blind deconvolution setting, where we used the initial PSF estimate as deconvolution kernel, and kept the optimal values fixed for the blind deconvolution setting without further optimization.
We evaluated the different weightings $\lambda_{\mathrm{TV}} = 5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 0.001, 0.005, 0.01$ for the TV regularization. For both noise levels, $\lambda_{\mathrm{TV}} = 0.001$ yielded the best results for the non-blind deconvolution. For the TM regularization of the data term in non-blind deconvolution, we tested $\lambda_{\mathrm{TM}} = 10^{-7}, 5 \cdot 10^{-7}, 10^{-6}, 5 \cdot 10^{-6}, 10^{-5}, 5 \cdot 10^{-5}$. The best results on SIM 1 were achieved with $\lambda_{\mathrm{TM}} = 10^{-7}$ and $\lambda_{\mathrm{TV}} = 5 \cdot 10^{-4}$. For SIM 2, $\lambda_{\mathrm{TM}} = 5 \cdot 10^{-6}$ and $\lambda_{\mathrm{TV}} = 10^{-3}$ yielded the best results. The parameters that performed best in the non-blind deconvolution were used in the blind deconvolution setting.
Since the specimen function is not normalized, the regularization parameters for the reconstruction of the deconvolution kernel have to be adapted to the range of the specimen function. To facilitate the comparison of the results with kernel regularization, we weight the regularizer by the integral of the objective function, i.e. by $\lambda_{\mathrm{KIP}} \cdot \int_\Omega o(\mathbf{x})\, d\mathbf{x}$ and $\lambda_{\mathrm{KFTV}} \cdot \int_\Omega o(\mathbf{x})\, d\mathbf{x}$ respectively.

4.10.1. Results with Kernel Intensity Penalty

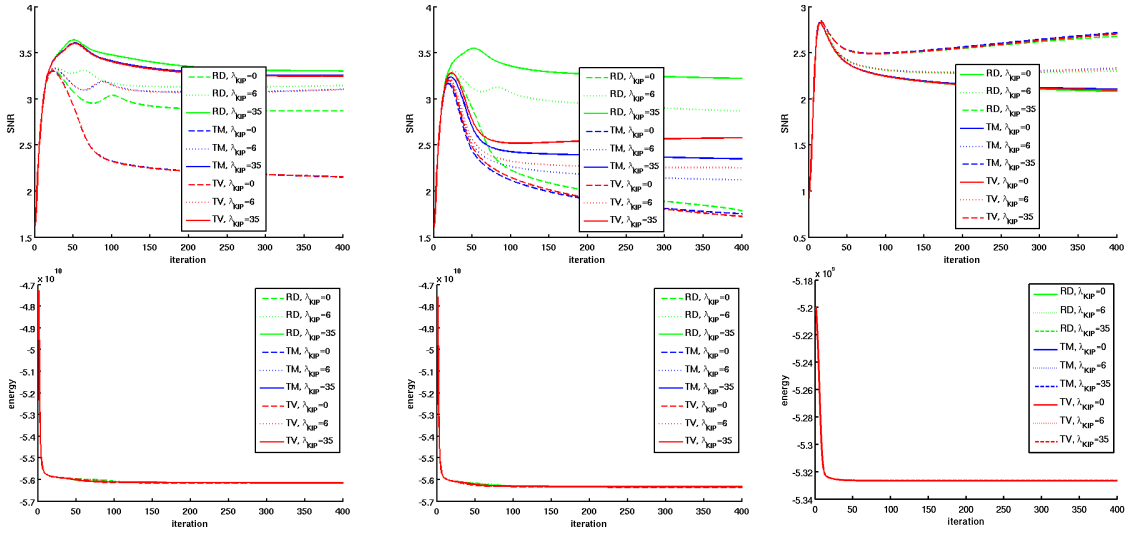
KIP, was evaluated for the parameters $\lambda_{\text{KIP}} = 6$ and $\lambda_{\text{KIP}} = 35$ for the real and synthetic datasets. We always computed 400 iterations to be sure that the energy converged. The resulting SNR of the reconstructed data with and without KIP is plotted in figure 4.4.
All methods were improved by the new regularization term. The results on simulated datasets SIM 1 and SIM 2 can be seen in figure 4.6 (c)-(e) and figure 4.7 (c)-(e) respectively in orthogonal slice views. The results on real data are displayed in figure 4.8(c), (d), and (e). In every case, the results with kernel intensity penalization outperform the standard algorithm. Without the kernel penalizer, spurious edges from out-of-focus planes are visible in SIM 1, that can be eliminated with our new approach. Additionally, the contrast within the nucleus is much better with our method. Especially when looking at the results for SIM 2, it can be seen that the noise is amplified if the kernel is not regularized.

In terms of SNR, the results with the higher KIP weighting $\lambda_{\text{KIP}} = 35$ are better than the results with the lower weighting $\lambda_{\text{KIP}} = 6$. Figure 4.5 shows the development of the SNR over the iterations for the three datasets with the given parameters. Below the SNR, the development of the energy J_{MLEM} (equation (4.7)) is displayed. It is interesting to see that the SNR generally has a maximum after 15-25 iterations, before the energy has converged. Without regularization of the deconvolution kernel, the SNR decreases afterwards, while the energy converges to a minimum. Sarder and Nehorai [2006] have also noticed this effect. With a regularization of the deconvolution kernel, this effect can be attenuated such that the methods become stable. For SIM 1, the SNR converges already with $\lambda_{\text{KIP}} = 6$. When looking at the resulting data (figure 4.6), it can be seen that the main advantage of the higher weighting lies in the upper and lower image regions, where the strongest blur has to be removed. If the focus lies on the central slices however, one might prefer the result of the lower KIP weighting. On the noisier dataset, we need a higher weighting λ_{KIP}. The methods are stable with $\lambda_{\text{KIP}} = 35$, but still, the results in terms of SNR are quite far below the maximum for TM and TV regularization. Therefore, we have tested the methods on SIM 2 with higher weightings λ_{KIP}. The results in terms of SNR are given in table 4.1. With higher weightings λ_{KIP} the SNR increases further for residual denoising but the TM and TV regularized schemes are not further improved. On the real data, the SNR converges as for SIM 1 already with a weighting $\lambda_{\text{KIP}} = 6$.
One conclusion of these observations is that the weighting λ_{KIP} should be chosen in order to stabilize the result. Since the OSL optimization scheme is not guaranteed to be stable for too high weightings of the penalization, we refrain from choosing the highest possible value even though it could yield a higher SNR.
For the simulated data, we achieved the best results without any statistical model on the data term but with the RD technique. On the noisier dataset, the TV regularization worked better than the TM regularization, whereas they performed equally on the dataset SIM 1 with little noise. On the actual microscopic data, the SNR of the TM regularized results are a little better than the result of RD but all three methods show comparable results in terms of SNR. The good result of the TM regularization may be a result of the fact that our ground truth is a spinning disk confocal microscope recording. Usually, confocal recordings are blurred with a small PSF that can well be approximated by a Gaussian (see section 3.1.3). In the xz-section in figure 4.8(a), it can be seen that the recorded specimen is rather flat. However the high frequencies in z-direction have not been recovered by any of the methods.



(a) SIM 1 (b) SIM 2 (c) MIC

Figure 4.4.: Resulting SNR for blind MLEM deconvolution with and without KIP. KIP improves the deconvolution result for all three specimen function regularization settings.

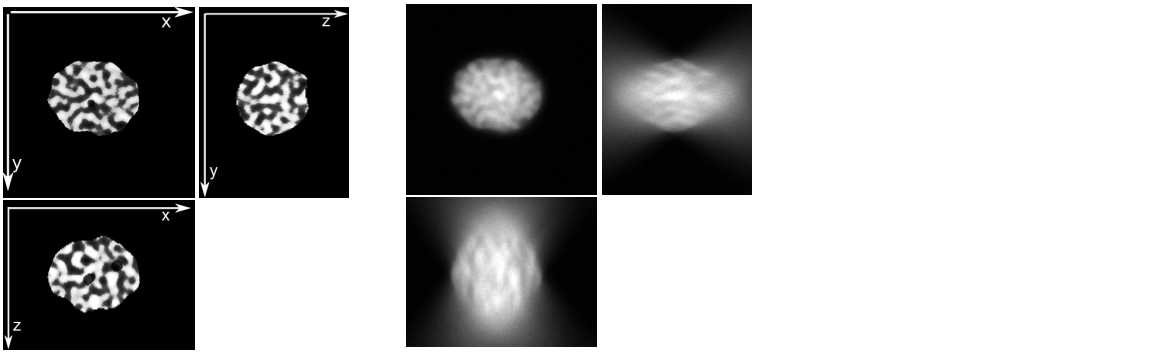


(a) SIM 1 (b) SIM 2 (c) MIC

Figure 4.5.: Development of the SNR (top row) and the energy (bottom row) of the reconstructed data over the iterations for blind MLEM with $\lambda_{\text{KIP}} = 6$ and $\lambda_{\text{KIP}} = 35$.

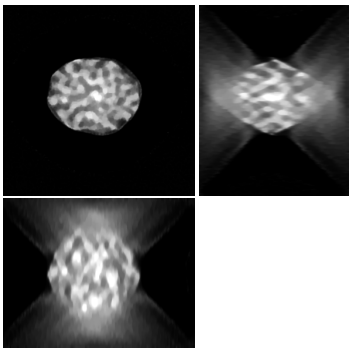
Table 4.1.: SNR of the reconstructed data with blind MLEM deconvolution with KIP for higher weightings λ_{KIP}.

	λ_{KIP}	RD	TM	TV
SIM 2	35	3.22309	2.35007	2.5793
SIM 2	50	3.27016	2.34216	2.5692
SIM 2	100	**4.6812**	2.2427	2.4471
MIC	35	2.67976	**2.71775**	2.7056

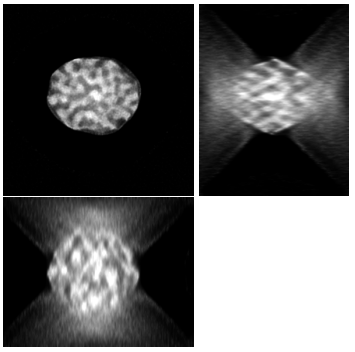


(a) Ground Truth

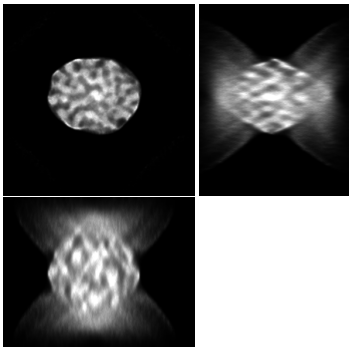
(b) SIM 1



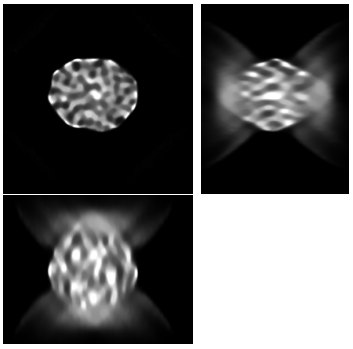
(c) TV RD, SNR=2.156



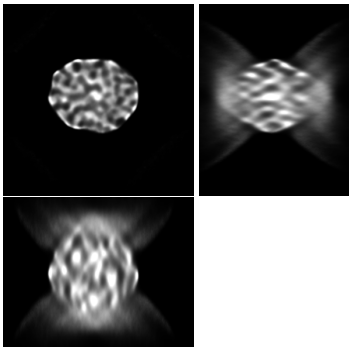
(d) TM RD, SNR=2.153



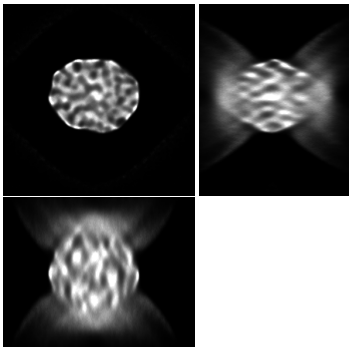
(e) RD RD, SNR=2.872



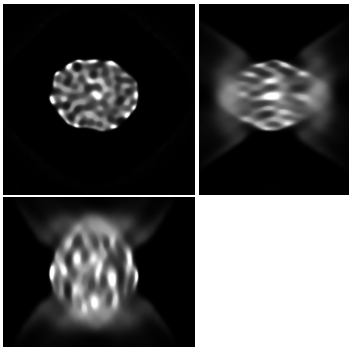
(f) TV RD+KIP, $\lambda_{\text{KIP}} = 6$
SNR=3.102



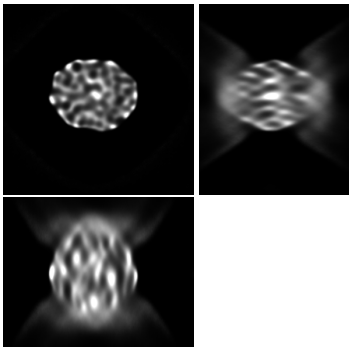
(g) TM RD+KIP, $\lambda_{\text{KIP}} = 6$
SNR=3.105



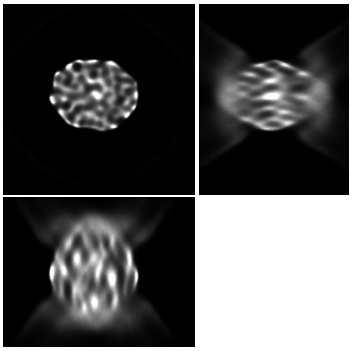
(e) RD RD+KIP, $\lambda_{\text{KIP}} = 6$
SNR=3.146



(f) TV RD+KIP, $\lambda_{\text{KIP}} = 35$
SNR=3.248

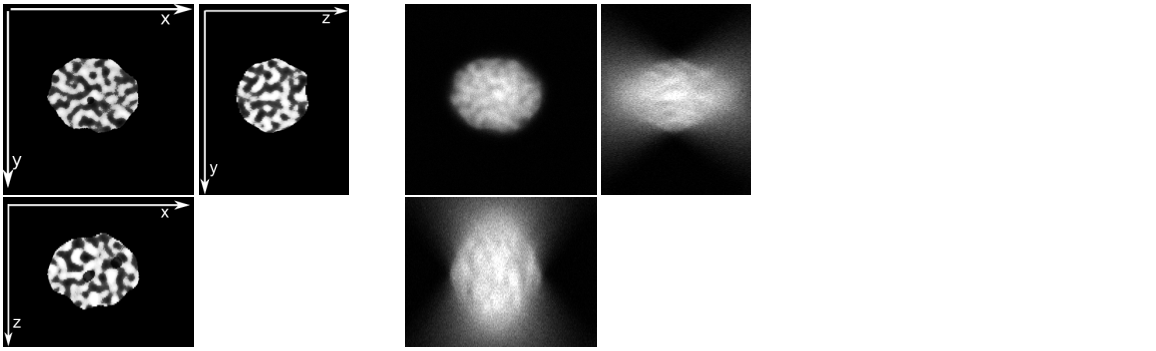


(g) TM RD+KIP, $\lambda_{\text{KIP}} = 35$
SNR=3.255



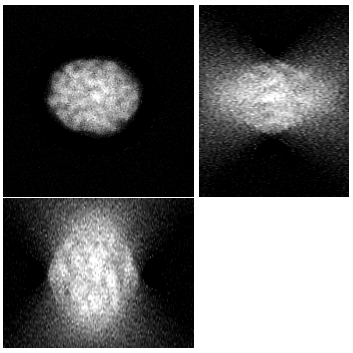
(e) RD RD+KIP, $\lambda_{\text{KIP}} = 35$
SNR=3.303

Figure 4.6.: KIP blind deconvolution results on SIM 1 displayed in three orthogonal views.

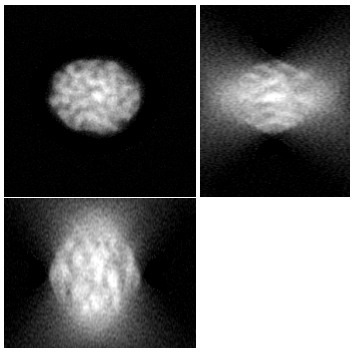


(a) Ground Truth

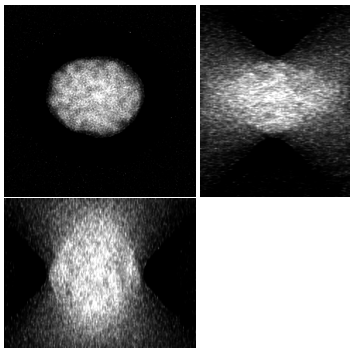
(b) SIM 2



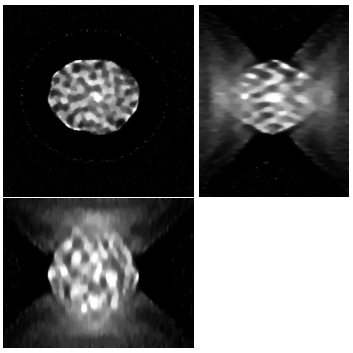
(c) TV RD, SNR=1.726



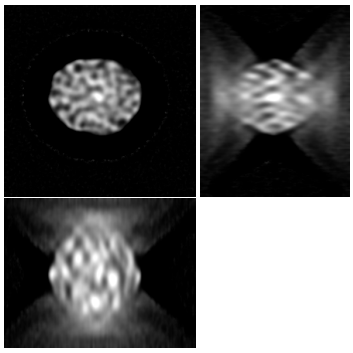
(d) TM RD, SNR=1.754



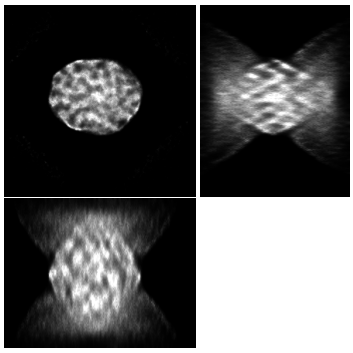
(e) RD RD, SNR=1.786



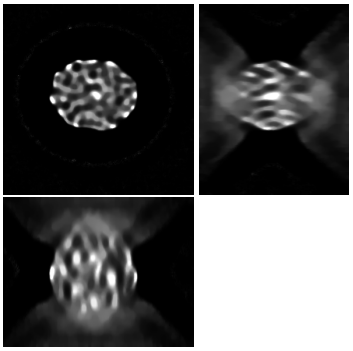
(f) TV RD+KIP, $\lambda_{\text{KIP}} = 6$
SNR=2.254



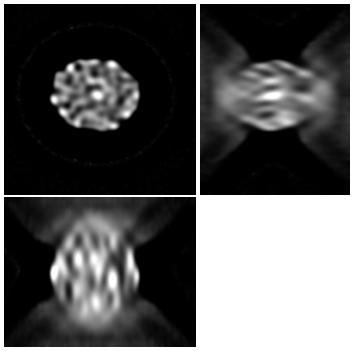
(g) TM RD+KIP, $\lambda_{\text{KIP}} = 6$
SNR=2.122



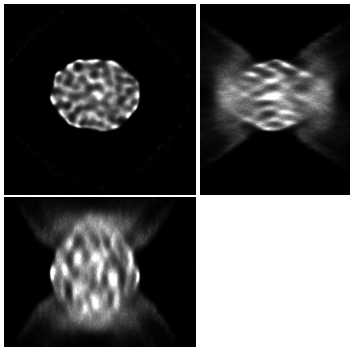
(e) RD RD+KIP, $\lambda_{\text{KIP}} = 6$
SNR=2.868



(f) TV RD+KIP, $\lambda_{\text{KIP}} = 35$
SNR=2.579

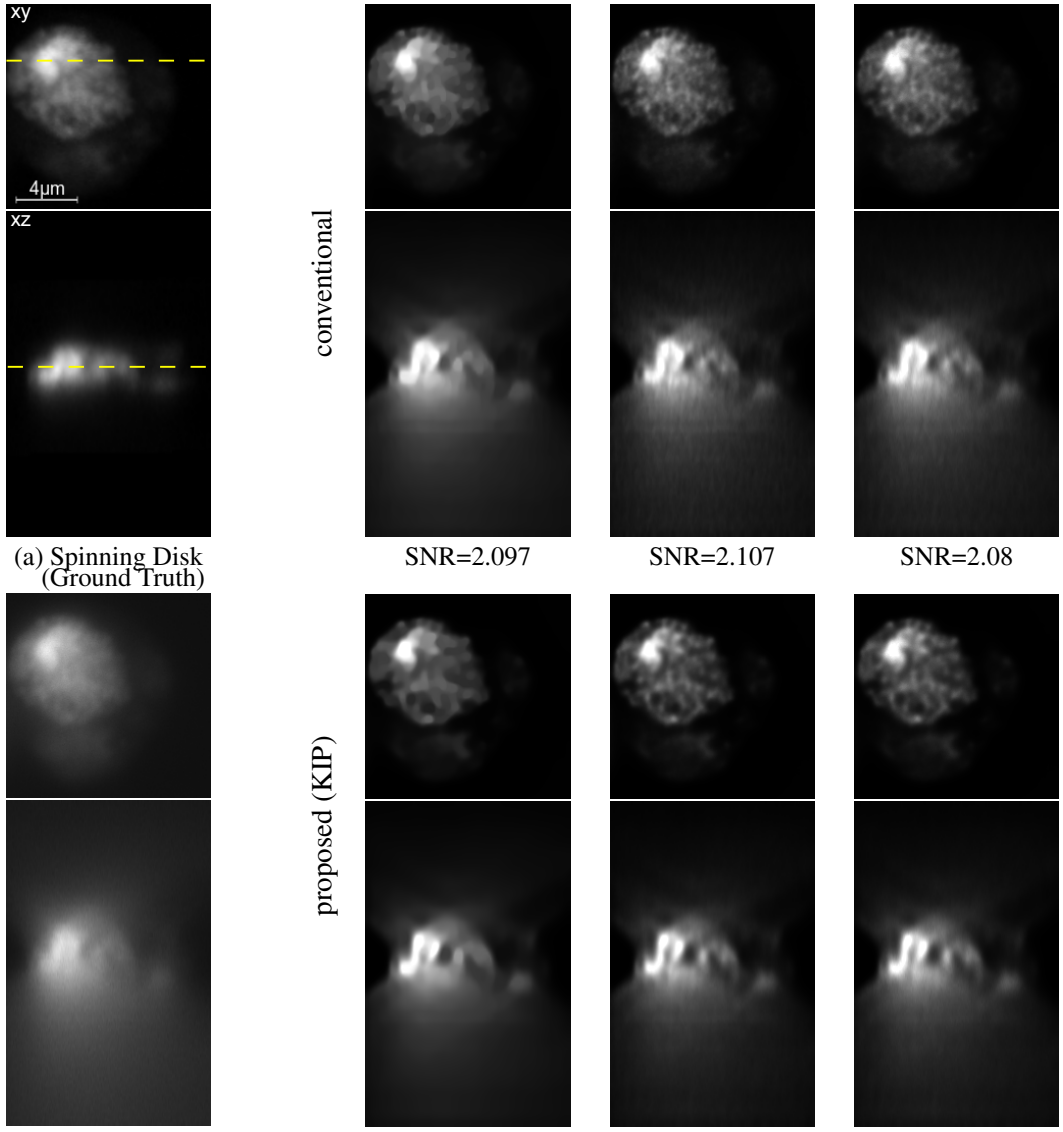


(g) TM RD+KIP, $\lambda_{\text{KIP}} = 35$
SNR=2.35



(e) RD RD+KIP, $\lambda_{\text{KIP}} = 35$
SNR=3.223

Figure 4.7.: KIP blind deconvolution results on SIM 2 displayed in three orthogonal views.



(a) Spinning Disk (Ground Truth)

(b) Widefield

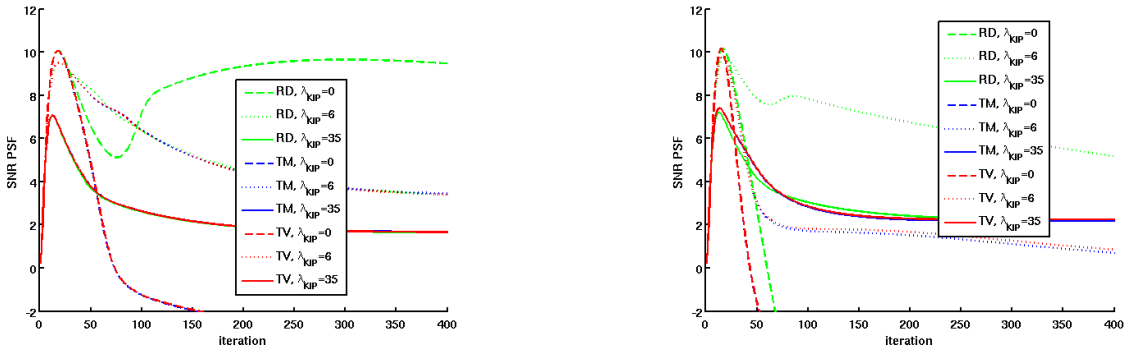
(c) TV RD

(d) TM RD

(e) RD RD

Figure 4.8.: KIP blind deconvolution results on a *Drosophila* S2 cell nucleus recording. The result is shown in two orthogonal views

Figure 4.9 shows the development of the SNR of the estimated deconvolution kernel compared to the ground truth for the simulated data. Without regularization, the SNR drops below zero. However, it is interesting to see, that the best SNR of the reconstructed PSF does not necessarily correspond to the best reconstruction of the specimen function. This can be due to the fact that the real PSF is cut off at some point since it does not have a finite support such that even a perfect reconstruction of the PSF inside a small domain is maybe not the best deconvolution kernel. In [Kenig et al., 2010], an interesting effect is reported in this context: in one of their two examples for quantitative analysis, the deconvolution results achieved with their blind deconvolution method are better in terms of root mean squared error (RMSE) than the results that could be achieved with the original blurring kernel.



(a) SIM 1 (b) SIM 2

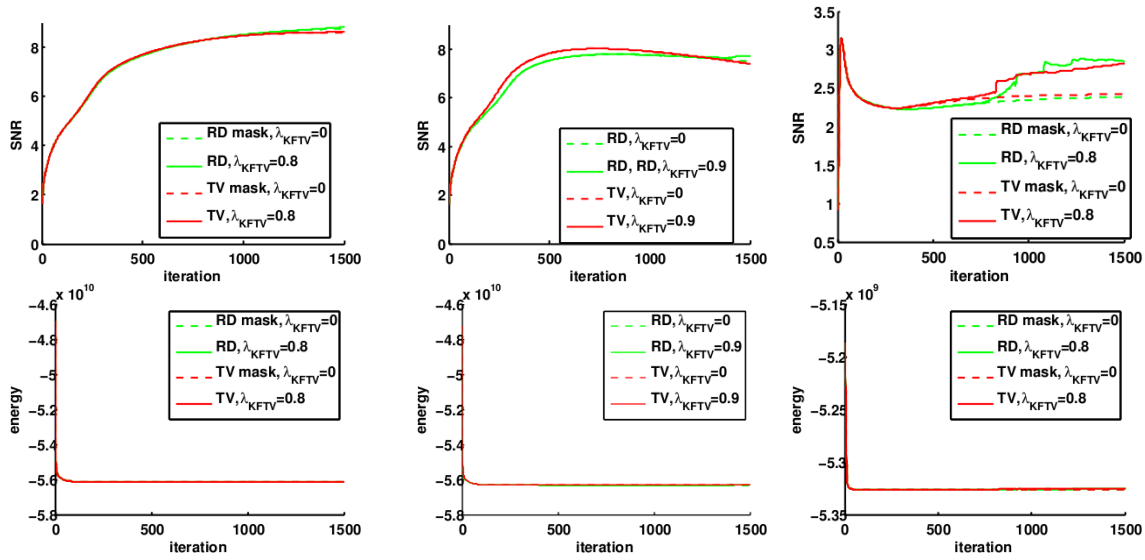
Figure 4.9.: Development of the SNR of the reconstructed PSF over the iterations for blind MLEM with KIP for the values $\lambda_{\text{KIP}} = 6$ and $\lambda_{\text{KIP}} = 35$.

4.10.2. Discussion

The proposed KIP method greatly improves the blind deconvolution results on real and synthetic data. In contrast to other methods, no complex PSF model is needed. The SNR can be stabilized. However when considering the deconvolution results in the xz-sections, one can see that there is still room for improvement. While higher KIP weightings yield higher SNR values and slightly better deconvolution results along the optical axis, the structures in the central layer visually appear clearer with lower values, that are just strong enough to stabilize the result. Whether high weightings of KIP are useful or not thus depends on the application.

4.10.3. Results with KFTV

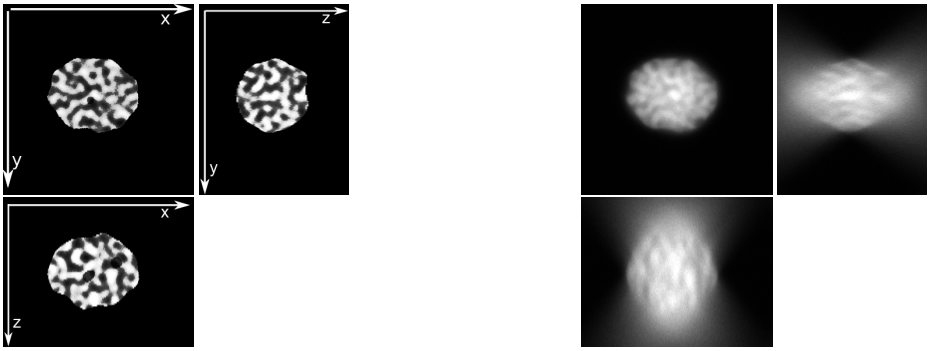
In terms of SNR, the results with kernel frequency mask (KF mask) with and without KFTV are much better than the results with KIP and of course better than without regularization. The development of the SNR of the reconstructed specimen function and the energy J_{MLEM} (equation (4.7)) over the iterations is plotted in figure 4.10. We computed 1500 iterations until the SNR stabilized for all data sets. The energy converges earlier. On all our datasets, the SNR with KFTV is higher than only with KF mask, although the results on the toy datasets can only be improved by a little as they are already quite good with KF mask. On SIM 2, the SNR without KFTV tends to drop again after about 900 iterations while it is stable with a KFTV regularization and resdual denoising. On the microscopic data set, the SNR with KFTV is significantly higher than with the kernel frequency mask only. For all three data sets, the energy converges rapidly to a minimum that is in the same range as it was before with KIP.



(a) SIM 1 (b) SIM 2 (b) MIC

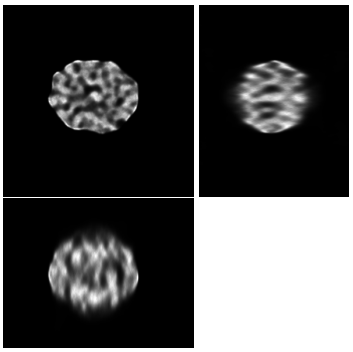
Figure 4.10.: Development of the SNR (top row) and the energy (bottom row) of the reconstructed data over the iterations for blind MLEM with the kernel frequency mask and KFTV regularization.

After 1500 iterations, the SNR on the synthetic datasets is in the range of 8 while with KIP, the best results we could achieve had an SNR of about 4.7. In figure 4.11 and 4.12, the deconvolved datasets SIM 1 and SIM 2 are shown in three orthogonal views. While the deconvolution result in the central layer is comparable to the best results with KIP (compare figure 4.6 and 4.7), the main advantage of the frequency domain regularization seems to be in the deblurring along the optical axis, where the results with kernel frequency mask and KFTV are much clearer. On the real microscopic data set, the improvement is not in the same range of SNR as on the synthetic data. Nevertheless, the deconvolved data is also much clearer in direction of the optical axis (compare figure 4.13).

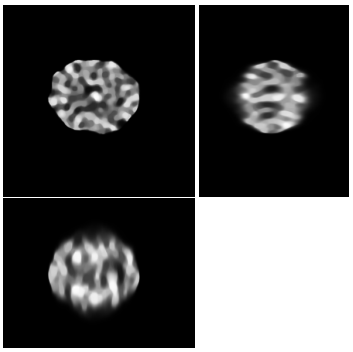


(a) Ground Truth

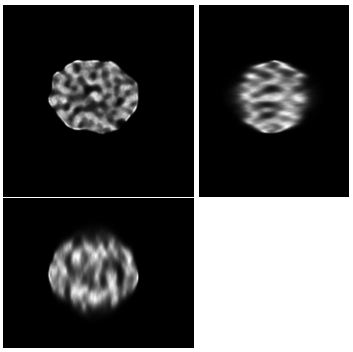
(b) SIM 1



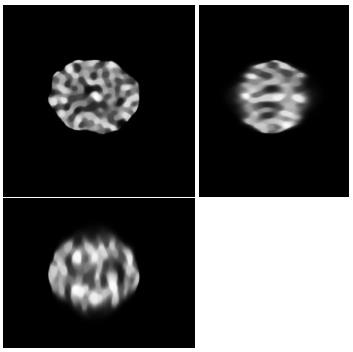
(c) RD KF mask, SNR=8.75428



(d) TV KF mask, SNR=8.59922



(3) RD KFTV, $\lambda_{\text{KFTV}} = 0.8$
SNR=8.8155

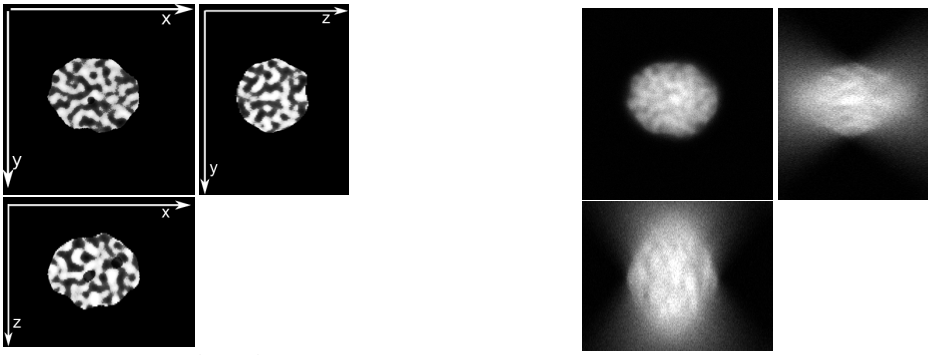


(g) TV KFTV, $\lambda_{\text{KFTV}} = 0.8$
SNR=8.6112

Figure 4.11.: KFTV blind deconvolution results on SIM 1 displayed in three orthogonal views. Especially in z-direction, the resulting data is much sharper than with KIP (compare figure 4.6).

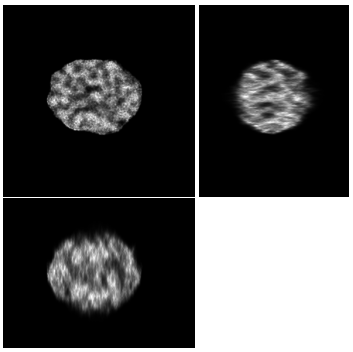
For a comparison to state-of-the-art methods that are using complex PSF models for blind deconvolution, we additionally evaluated our method on the test data used in [Kenig et al., 2010][4]. Kenig et al. [2010] perform blind MLEM deconvolution with RD in the specimen function and in the kernel update. In their work, the space of possible kernel functions is learned a priori. It is spanned using kernel PCA computed on a synthetically generated library of PSFs. In every iteration, the current kernel estimate is projected onto this space. The synthetic datasets used in [Kenig et al., 2010] are $129 \times 129 \times 129$ voxel volumes with a voxel size of $0.1\mu\text{m}^3$ that are blurred with synthetically generated PSFs and corrupted by Poisson noise. The recorded specimen is a set of three tori with

[4]Thanks to Tal Kenig who kindly let us use his data for evaluation.

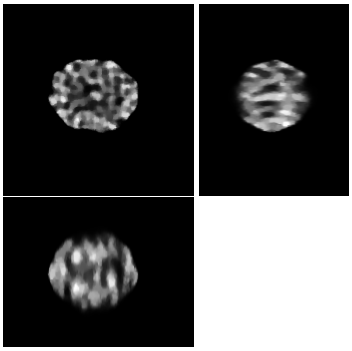


(a) Ground Truth

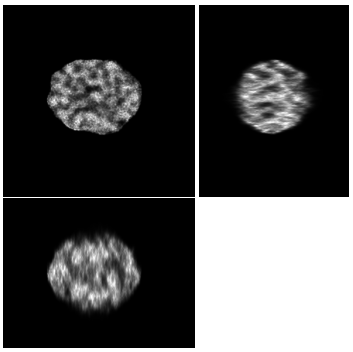
(b) SIM 2



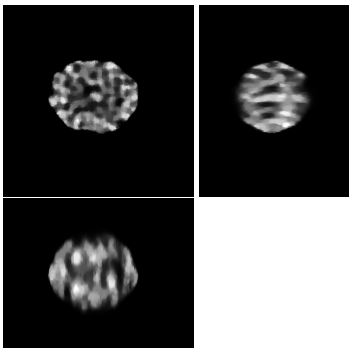
(c) RD KF mask, SNR=7.46086



(d) TV KF mask, SNR=7.392



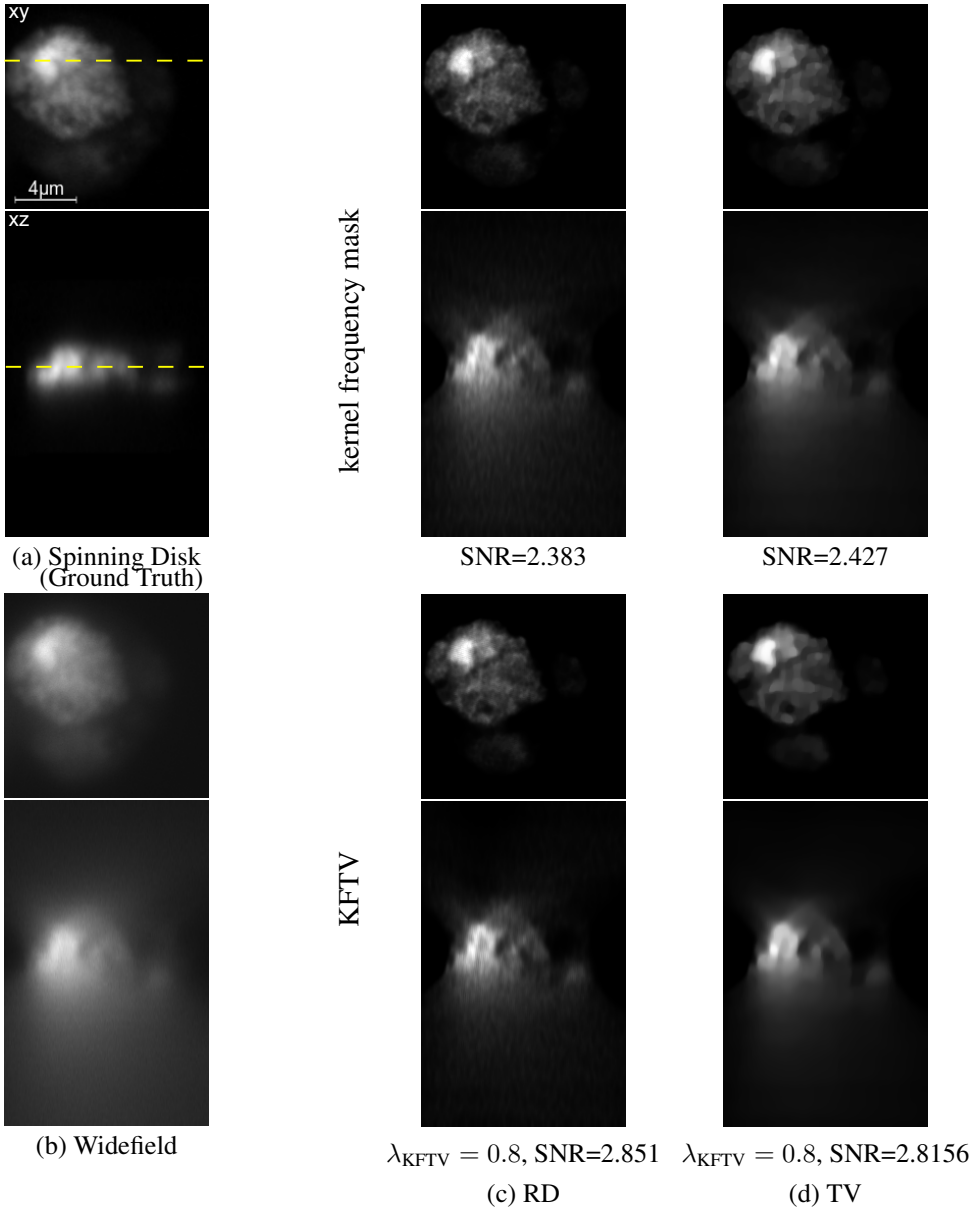
(3) RD KFTV, $\lambda_{\text{KFTV}} = 0.9$
SNR=7.7175



(g) TV KFTV, $\lambda_{\text{KFTV}} = 0.8$
SNR=7.3924

Figure 4.12.: KFTV blind deconvolution results on SIM 2 displayed in three orthogonal views. Especially in z-direction, the resulting data is much sharper than with KIP (compare figure 4.7).

different size and intensities. This specimen is blurred with a symmetric and with an asymetric PSF. All synthetic PSFs are generated using the method of [Stokseth, 1969]. The initialization is done using a symmetric PSF support computed according to [Kenig, 2009]. The results are reported in terms of *root mean squared error* (RMSE) and compared against [Hom et al., 2007].
On the real data Kenig et al. [2010] only show qualitative results. For the comparison to [Kenig et al., 2010] and [Hom et al., 2007], we also use RD for the specimen function and the kernel update. Our initialization is done similar to [Kenig et al., 2010] with the symmetric PSF support. First we compare our results on the torus datasets from [Kenig et al., 2010] blurred with symmetric and an asymmetric PSF. The results are displayed in figure 4.14. In terms of RMSE the result with KFTV



(a) Spinning Disk (Ground Truth)

SNR=2.383

SNR=2.427

(b) Widefield

$\lambda_{\text{KFTV}} = 0.8$, SNR=2.851

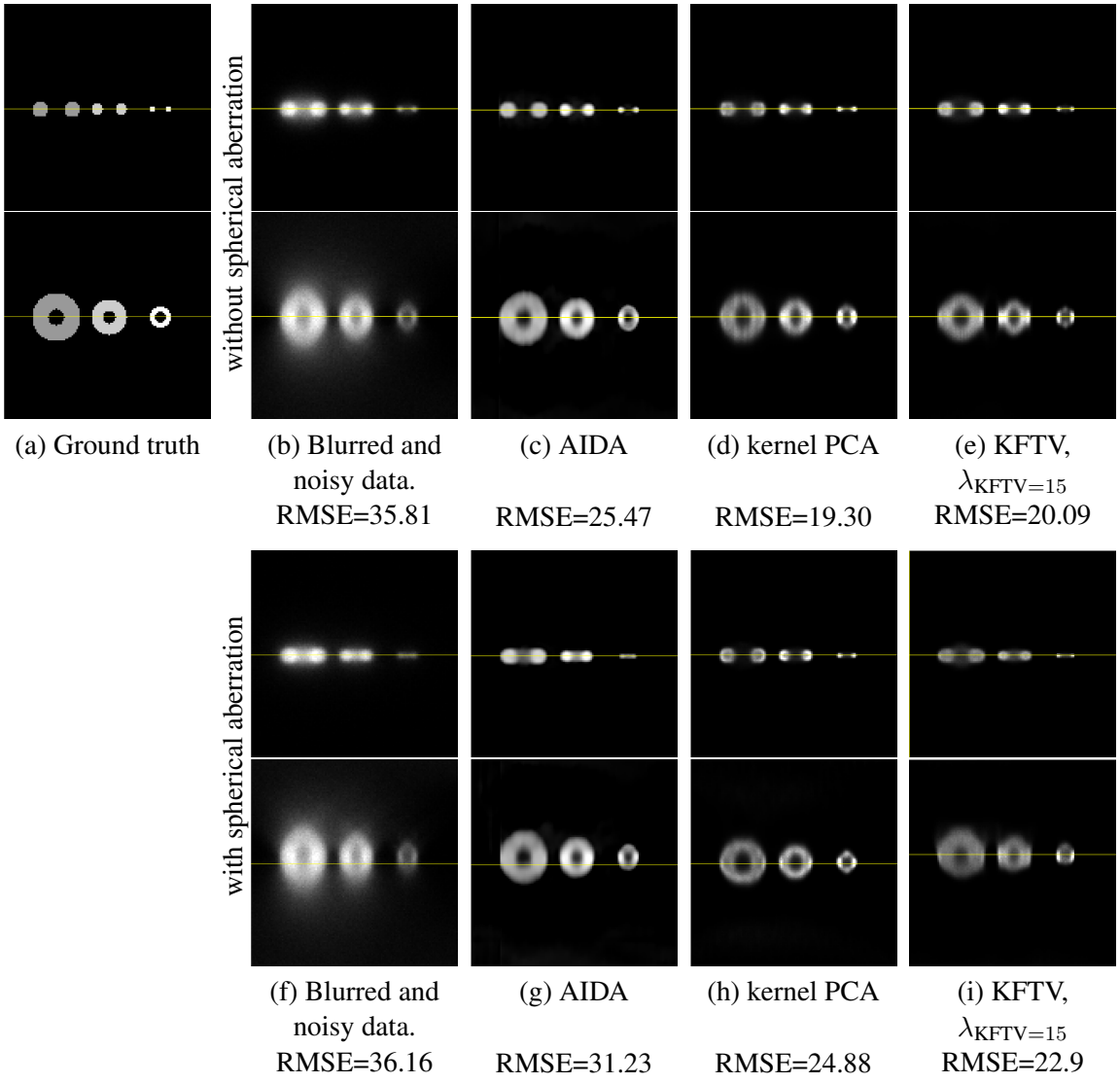
(c) RD

$\lambda_{\text{KFTV}} = 0.8$, SNR=2.8156

(d) TV

Figure 4.13.: KFTV blind deconvolution results on a *Drosophila* S2 cell nucleus recording. The result is shown in two orthogonal views.

ranks better than the results with [Hom et al., 2007] and similar to [Kenig et al., 2010] on the dataset blurred with the symmetric PSF. For the highly asymmetric data, we chose a high KFTV weighting with $\lambda_{\text{KFTV}} = 15$. On the dataset blurred with the aberrated PSF, our algorithm algorithm can not estimate the shift that results from the aberration. We always assume that the maximum of the PSF is in the center. The RMSE value we report for the torus dataset with spherically aberrated PSF is thus after correcting for this shift, which has been done manually. With this correction, our method shows a better RMSE value for this second dataset than both other methods. Qualitative evaluation, however, shows that the smallest, rightmost torus is not very well reconstructed by our method. The lower part in z-direction is not preserved. This is better with both of the other methods that like



(a) Ground truth | (b) Blurred and noisy data. RMSE=35.81 | (c) AIDA RMSE=25.47 | (d) kernel PCA RMSE=19.30 | (e) KFTV, $\lambda_{\text{KFTV}=15}$ RMSE=20.09

(f) Blurred and noisy data. RMSE=36.16 | (g) AIDA RMSE=31.23 | (h) kernel PCA RMSE=24.88 | (i) KFTV, $\lambda_{\text{KFTV}=15}$ RMSE=22.9

Figure 4.14.: Toy data used in [Kenig et al., 2010] and the deconvolution results with the different methods AIDA [Hom et al., 2007], kernel PCA [Kenig et al., 2010], and our proposed KFTV. Two orthogonal views are displayed: the xy-view (top row) and the xz-view (bottom row). Sub-figures (a) - (d) and (f) - (h) and the respective RMSE values are taken from [Kenig et al., 2010].

our method find stronger intensities in the central layer but preserves the topology of this smallest structure also in z-direction in the spherically aberrated data.

For the microscopic recordings, we initialized the blind deconvolution with our measured PSF (compare section 4.8.2). For these complex data, we reached convergence only after 2500 iterations. The deconvolution results of our method, of AIDA [Hom et al., 2007], and the kernel PCA method [Kenig et al., 2010] on the real data are displayed in figure 4.15. While with AIDA and kernel PCA, some of the structures are suppressed, our deconvolution preserves most of the relevant structures in the central plane. However, in the upper most example (dataset 1), some PSF artifacts are still

visible after deconvolution in the xz-view in the upper and lower dataset regions. For dataset 2 and dataset 3, this is not the case.

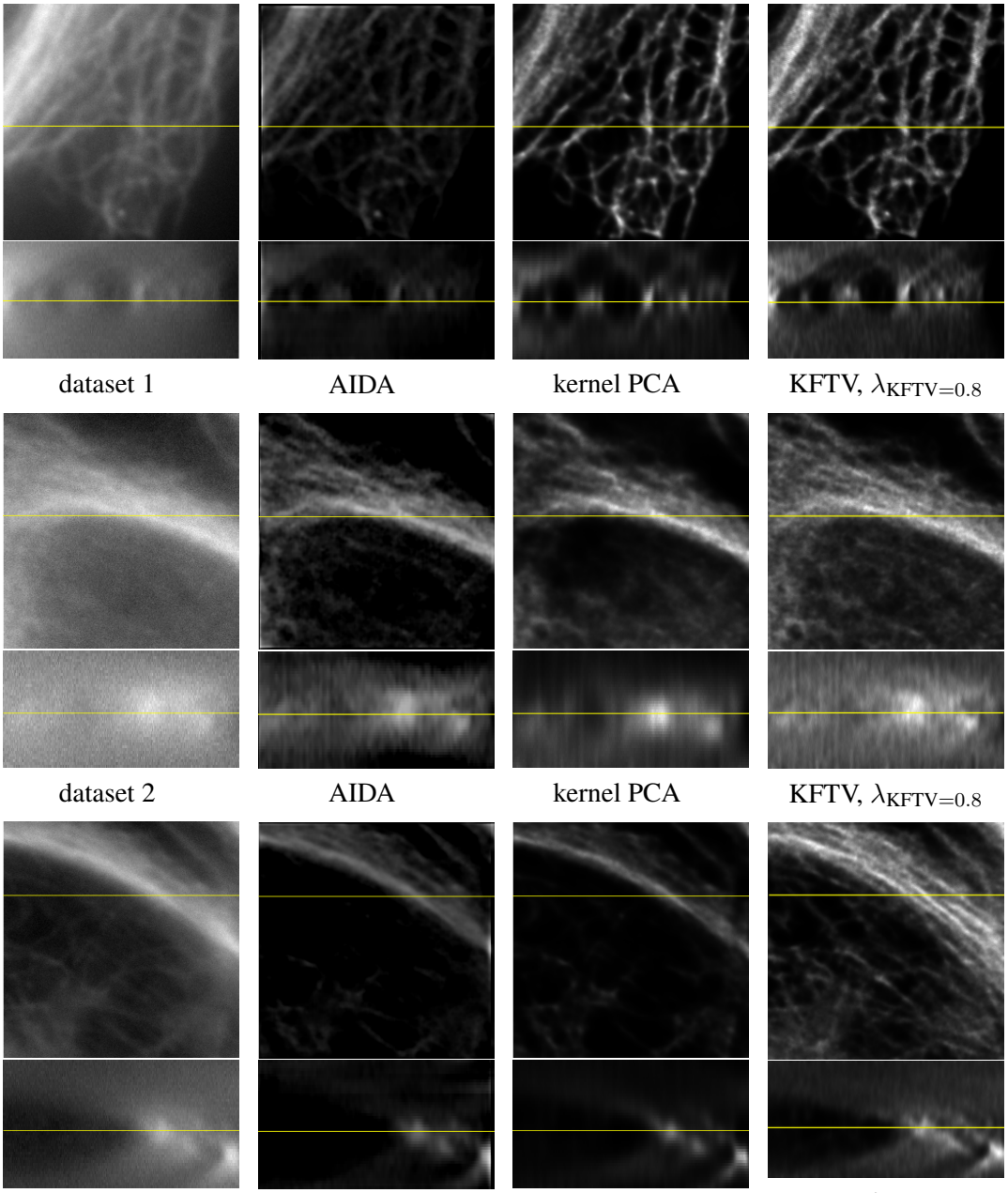


Figure 4.15.: Microscopic datasets used in [Kenig et al., 2010] and the deconvolution results with the different methods AIDA [Hom et al., 2007], kernel PCA [Kenig et al., 2010], and our proposed KFTV. Two orthogonal views are displayed: the xy-view (top row) and the xz-view (bottom row). Sub-figures (a) - (c) are taken from [Kenig et al., 2010].

4.10.4. Discussion

On our own test data and on the test data provided by Tal Kenig, our proposed KFTV method with kernel frequency mask shows competitive results. In contrast to other state-of-the-art methods, our proposed algorithm does not need to learn the space of valid PSFs a priori. In our opinion, this is a great advantage since learned PSF models can only be as good as the given training data. Given the fact that widefield PSFs have no compact support and state-of-the-art PSF simulation methods are nothing but approximations, such learned "PSF spaces" might even limit the deconvolution quality. In our KFTV method, the deconvolution kernel is projected onto a subspace of possible solutions by masking out frequencies that certainly have to be zero. As a result, we can be sure that the original PSF is contained in the subspace of valid solutions. The mask only has to be computed once and can be used for the deconvolution of images from widefield microscopy with very different recording settings. On our real dataset as well as on the synthetic data provided by Tal Kenig, the KFTV regularization greatly improves the results. Our impression is that with increasing asymmetry of the specimen function, KFTV becomes more important. This, however, remains to be proved.

4.11. Conclusion

For widefield microscopic data, deconvolution potentially yields greatly improved data quality. We have evaluated several state-of-the-art specimen function regularizations in the blind MLEM deconvolution scheme on our specimen. The experiments show that the residual denoising technique performs slightly better than standard regularization methods in the variational setting. We have presented two regularizers for the reconstruction of the deconvolution kernel. Both regularization terms are independent of a specific PSF model as this is the case for other state-of-the-art blind deconvolution methods. We show that already the pure kernel intensity penalization regularization stabilizes the blind deconvolution scheme such that it converges to a solution with high SNR. Even better results are achieved with the kernel regularization in the frequency domain. With a kernel frequency mask that only masks out frequencies that are certainly not inside the OTF support and the KFTV regularization scheme that is derived in a variational framework, our solution performs similar to state-of-the-art methods in terms of RMSE on synthetic data and showed convincing results on real widefield microscopic recordings without the need of a complex, priorly learned PSF model.

5. Active Surface Models for Volumetric Microscopy

The segmentation of objects from volumetric microscopic images comes with some challenges compared to the segmentation of natural images. Signal attenuations, noise, uneven staining, and bleaching effects restrain the direct use of the recorded intensities for the segmentation. Even though occlusions are not an issue when whole volumetric data is recorded, these same effects may result in invisible object boundaries in whole image regions. Depending on the staining, the object of interest also might not be entirely marked, for example, if a membrane staining has been recorded and the object enclosed by this membrane is to be segmented. Last but not least, blur introduced in the recording by the PSF always leaves an uncertainty about the real boundary position even for human experts.
Generally, the use of gradient information instead of intensity information can help in these scenarios, since gradients are at least invariant against additive signal changes.
If a threshold-based segmentation is possible, the gradients can be used to find an appropriate threshold as done in [Morath et al., 2013] for the segmentation of different cells and cell nuclei from spinning disk microscopic data and before in [Keuper, 2007] for the segmentation of the nuclei in mouse embryonic fibroblast and human fibroblast cells. The intensity gradient of the recorded data set I is denoted by ∇I, the gradient magnitude by $\|\nabla I\|$. The optimal threshold τ for data set I maximizes the gradient magnitude on the resulting contour:

$$\max_{\tau} \frac{1}{\int_{\Omega} \delta(\mathrm{I}(\mathbf{x}) - \tau) d\mathbf{x}} \int_{\Omega} \delta(\mathrm{I}(\mathbf{x}) - \tau) \|\nabla I\|(\mathbf{x}) d\mathbf{x}. \tag{5.1}$$

δ is the Dirac function. The term $\delta(\mathrm{I}(\mathbf{x}) - \tau)$ thus gives a contribution wherever the data I has exactly the threshold value τ. This method can obviously only be applied if the recorded intensities are higher inside the object than on the boundary. In [Morath et al., 2013] and [Keuper, 2007], we have accounted for this by applying a Gaussian smoothing to the data such that the few points with low intensity inside the nucleus were smoothed away.
Whenever threshold methods are not sufficient to produce good segmentations we need more elaborate models for the object surfaces.
In our case, these objects are either cells, cell nuclei or nucleoli. For all of these objects, we can assume that they have a star-shaped surface, i.e. there exists a center $\mathbf{c}$ such that every line from $\mathbf{c}$ to the surface lies completely inside the object. For the segmentation of these objects from volumetric data, we have decided to use active surfaces. Then, we again had to choose between implicit, non-parametric active surfaces and explicit, parametric active surfaces. We have decided to use parametric active surface models for several reasons. First, the topology of the object we want to segment is known a priori. The ability of implicit deformable models to change their topology is therefore not desired. Second, since the objects of interest are fluorescently marked, it is usually

possible to detect them and roughly estimate their size. Therefore, we can expect good initializations of the contours. Convex optimization techniques as they are available for implicit deformable models are not required. Third, depending on the surface representation, the evolution of parametric active surfaces can be computed rapidly even for large volumetric datasets.
Most of our contributions that will be presented in this chapter are, however, the data terms we have developed, accounting for the challenges given in our volumetric biological data. These data terms could in theory as well be used in an implicit active surface implementation in a similar way as GVF is, for example, used in [Xu et al., 2000].
In the following, we will theoretically describe three-dimensional active surface models, and explain how the regularity of these models can be measured and optimized (section 5.1). The method we are using for the initialization of the active surfaces is described in section 5.2. Our contributions in the field of data term design are presented in section 5.3. The first presented method is a preprocessing step, combining the information from two recording channels in order to allow for a good segmentation of the nucleoli. Then, we describe how the result of the object detection step can be used to derive a prior for the objects' boundaries [Keuper et al., 2009]. The third method we propose is a modification of the standard GVF algorithm that accounts for the strong blur in our data due to the PSF [Keuper et al., 2010b; Keuper, 2010]. The fourth presented method was developed in order to learn specific, new object boundary appearances by user-interaction, such that these learned boundary appearances can be used to automatically segment large amounts of data [Keuper et al., 2010a]. In section 5.4, we will present a new method for the dynamic adjustment of the parameters steering the regularization of the surfaces [Keuper et al., 2010c].

5.1. Parametric Active Surface Models

By analogy to the two dimensional case (see section 2.3), a three-dimensional active surface can be described as a function $\mathbf{\Gamma} : [0,1] \times [0,1] \rightarrow \Omega \subseteq \mathbb{R}^3$ which is placed on a potentially multi-channel dataset $I : \Omega \rightarrow \mathbb{R}^n$. As in the two dimensional case (compare equation (2.28)), the active surface energy consists of a *data term* and an *internal energy term*:

$$E(\mathbf{\Gamma}) = E_{\text{data}}(\mathbf{\Gamma}) + E_{\text{int}}(\mathbf{\Gamma}). \tag{5.2}$$

The data term seeks to drive $\mathbf{\Gamma}$ towards the object's surface and the internal energy term usually has the purpose of regularizing the shape of the active surface in terms of smoothness and continuity. The adaption of the active surface to the underlying data takes place by minimizing this energy. This can be done by gradient descent. The negative gradients of $E_{\text{data}}(\mathbf{\Gamma})$ and $E_{\text{int}}(\mathbf{\Gamma})$ can be considered as data and internal forces $\mathbf{F}_{\text{data}}$ and $\mathbf{F}_{\text{int}}$ respectively. In the following, we will briefly sketch how the internal energy can be defined in three-dimensional space.

5.1.1. Internal Energy

In order to prevent the surface from stretching and bending too much, the continuity and curvature energies have to be formulated for surfaces. The straightforward translation of the energies defined

by Kass et al. [1988] to surfaces in three-dimensional space is (e.g. in [Mishra et al., 2011])

$$\begin{aligned} E_{\text{int}}(\mathbf{\Gamma}) &= \frac{1}{2}\int_{\mathbf{\Gamma}} \alpha \left(\left(\frac{\partial \mathbf{\Gamma}}{\partial s_1}(\mathbf{s}) \right)^2 + \left(\frac{\partial \mathbf{\Gamma}}{\partial s_2}(\mathbf{s}) \right)^2 \right) \\ &\quad + \beta \left(\left(\frac{\partial^2 \mathbf{\Gamma}}{\partial {s_1}^2}(\mathbf{s}) \right)^2 + \left(\frac{\partial^2 \mathbf{\Gamma}}{\partial {s_2}^2}(\mathbf{s}) \right)^2 + 2 \left(\frac{\partial^2 \mathbf{\Gamma}}{\partial s_1 \partial s_2}(\mathbf{s}) \right)^2 \right) d\mathbf{s}, \end{aligned} \tag{5.3}$$

where the contour $\mathbf{\Gamma}$ is partially derived with respect to the components s_1 and s_2 of $\mathbf{s} \in [0,1] \times [0,1]$, and α and β are constant weighting factors for the continuity and curvature regularization respectively. The minimization of this energy $E_{\text{int}}(\mathbf{\Gamma})$ leads to an Euler-Lagrange equation with the second and forth derivative of the contour, the elasticity and the rigidity force

$$\mathbf{F}_{\text{ela}}(\mathbf{s}) := -\left(\frac{\partial^2 \mathbf{\Gamma}}{\partial {s_1}^2}(\mathbf{s}) + \frac{\partial^2 \mathbf{\Gamma}}{\partial {s_2}^2}(\mathbf{s}) \right), \tag{5.4}$$

and

$$\mathbf{F}_{\text{rig}}(\mathbf{s}) := \frac{\partial^4 \mathbf{\Gamma}}{\partial {s_1}^4}(\mathbf{s}) + \frac{\partial^2}{\partial s_1^2}\frac{\partial^2 \mathbf{\Gamma}}{\partial s_2^2}(\mathbf{s}) + \frac{\partial^4 \mathbf{\Gamma}}{\partial {s_2}^4}(\mathbf{s}). \tag{5.5}$$

The Euler-Lagrange equation is then given by

$$\alpha \mathbf{F}_{\text{ela}} + \beta \mathbf{F}_{\text{rig}} = 0. \tag{5.6}$$

Although this formulation often works fine in practice [Mishra et al., 2011], it theoretically has the drawback that the energy (and its minimization) depends on the surface parameterization. This is why it is often preferable to base the regularization on the local mean curvature. The local mean curvature of a surface $\mathbf{\Gamma}$ is defined as $H = -\frac{1}{2}\text{div}(\mathbf{n}) = \frac{1}{2}(\kappa_1 + \kappa_2)$, where div(.) is the divergence operator, $\mathbf{n}$ is the surface unit normal and κ_1 and κ_2 are the principal curvatures [1]. For continuity and curvature regularization, the energy functional can be given as in [Hsu et al., 1992] by the following integral over the surface area

$$E_{\text{int}}(\mathbf{\Gamma}) = \int_{\mathbf{\Gamma}} (\alpha + \beta H^2(\mathbf{s}) + \gamma K(\mathbf{s}))\, d\mathbf{s}, \tag{5.7}$$

where α, β, and γ are constant weighting factors for the continuity and curvature, and $K = \kappa_1 \cdot \kappa_2$ is the Gaussian curvature of the surface $\mathbf{\Gamma}$. Since the Gaussian curvature of closed spherical surfaces is a topological constant [Hsu et al., 1992] with

$$\int_{\mathbf{\Gamma}} K(\mathbf{s})\, d\mathbf{s} = 2\pi\chi(\mathbf{\Gamma}), \tag{5.8}$$

where $\chi(\mathbf{\Gamma})$ is the Euler characteristic of $\mathbf{\Gamma}$, it is sufficient to consider the following energy for the regularization:

$$E_{\text{int}}(\mathbf{\Gamma}) = \int_{\mathbf{\Gamma}} (\alpha + \beta H^2(\mathbf{s}))\, d\mathbf{s}. \tag{5.9}$$

[1] The principle curvatures of a surface point are the maximum and minimum values of the curvature at this point. They correspond to the eigenvalues of the *shape operator* [Grenander and Miller, 2007].

The Euler-Lagrange equation, the solution of which is minimizing this functional is highly nonlinear and given by [Xu and Zhang, 2006; Hsu et al., 1992]

$$\Delta H + 2H(H^2 - (K - \alpha/\beta)) = 0, \tag{5.10}$$

where Δ is the Laplace operator on $\mathbf{\Gamma}$. *Minimal surfaces*, fulfilling the equation $H = 0$, are a special class of surfaces solving also the above Euler-Lagrange equation. They can be found by computing the mean curvature flow of a surface

$$\frac{\partial \mathbf{\Gamma}}{\partial t}(\mathbf{s}) = H(\mathbf{s})\mathbf{n}(\mathbf{s}), \tag{5.11}$$

where $\mathbf{n}(\mathbf{s})$ is the unit normal of $\mathbf{\Gamma}$ in $\mathbf{s}$.

5.1.2. Surface Parameterization

The first step of the active surface implementation is the choice of an appropriate surface parameterization. This parameterization can be performed for example by defining a surface mesh [Baumgart, 1972; Mantyla, 1988; Delingette, 1999; Slabaugh and Unal, 2005] (these are also called polyhedral surfaces), by using splines [Coons, 1974; Terzopoulos, 1986; Terzopoulos et al., 1987] (as an extension of polyhedral surfaces to curved surfaces), or as functions on the sphere [Ballard and Brown, 1981; Schudy and Ballard, 1978, 1979] (by longitudinal and latitudinal coordinates). The longitudinal and latitudinal coordinates can also be connected in order to form e.g. a triangular or quadrangular mesh. For parameterizations on the sphere, one can further distinguish between spatial and frequency representations of the surfaces.
As we assume a roughly spherical shape of the objects to detect, it is intuitive to initialize the active surface with a spherical mesh. In the following, two possibilities for a mesh specification with spherical topology are described.

5.1.2.1. Parameterization in Euclidean Coordinates

In the ideal case, the surface sampling should be equidistant. However, equidistant sampling of the sphere can only be done for a very limited number of sampling points (4, 6, 8, 12, and 20). More nodes can not be equidistantly placed on a sphere such that approximations become necessary [Cundy and Rollett, 1961]. It is a common choice to build up the surface parameterization based on an icosahedron, the regular Platonic solid with 12 vertices. The nodes in the icosahedron are the surface sample points controlling the shape and the edges define their neighborhood. Subdivisions of an icosahedron still provide a nearly equidistant sampling with a higher resolution, leading to a more accurate segmentation [Ahlberg, 1996; Mille et al., 2006]. The icosahedron subdivision is also represented by nodes $v_i \in \mathcal{V}$ and edges $e_{ij} \in \mathcal{E}$ defining a neighborhood system $\mathcal{N}(v_i)$. In the following, the position of vertex v_i in $\mathbb{R}^3$ is denoted by $\mathbf{x}_{v_i}$.

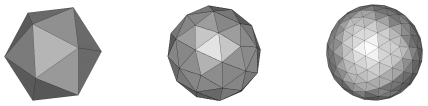


Figure 5.1.: The sampling of the spherical surface can be done equidistantly by using an icosahedron. Icosahedron subdivisions yield a nearly equidistant sampling.

This parameterization allows for arbitrary shapes with spherical topology. Since the distances of the nodes change during surface evolution, it is important to implement a shape regularization with this parameterization. On the one hand, neighboring control points should not be torn apart, on the other hand the surface should remain smooth (since smoothness can usually be assumed for surfaces of biological objects).
However due to the data term, the surface may be stretched despite the regularization. In this case, we need node insertion and node deletion operations.

5.1.2.2. Surface Resampling

To ensure that during the evolution of the active surface, the resolution of the mesh is always high enough, splitting and merging operations are needed. For our work on the segmentation of *Drosophila* S2 cell nuclei and nucleoli [Keuper et al., 2009], we implemented the following splitting and merging scheme: After each iteration, the length l of every edge is checked. Edges longer than a threshold t_u are split and a new vertex is inserted. This new vertex has to be connected to all its neighboring vertices. If edges are shorter than threshold t_l, they are deleted and the corresponding vertices are merged. See figure 5.2 for an example.

Figure 5.2.: (left) If the length $l > t_u$, the edge is split. (right) Edges for which $l < t_l$ are merged.

Different methods for the surface mesh optimization were for example proposed in [Hoppe et al., 1993] and [Mishra et al., 2011].

5.1.2.3. Parameterization in Spherical Coordinates

For the segmentation of star-shaped objects, an easy way of parameterizing a shape is to describe the Euclidean distance r of a surface point to the shape center $\mathbf{c}$ as a function on the sphere

$$S^2 = \{(\theta, \phi) \in \mathbb{R}^2 | 0 \leq \theta \leq \pi, 0 \leq \phi < 2\pi\}. \tag{5.12}$$

The resulting parameterized surface is [Miller et al., 1994; Ballard and Brown, 1981]

$$\mathbf{\Gamma} : S^2 \rightarrow \mathbb{R}, (\theta, \phi) \rightarrow r(\theta, \phi). \tag{5.13}$$

This corresponds to a representation in spherical coordinates $(\theta, \phi, r(\theta, \phi))$ with the latitude θ and longitude ϕ and $r(\theta, \phi)$ being the radius in direction (θ, ϕ). Active surfaces based on this parameterization have also been termed "normal active surfaces" [Miller et al., 1994; Grenander and Miller, 2007]. The spherical parameterization is quite restrictive: only star-shaped objects can be represented without loss. However, this is not limiting for our applications, since our cell and cell nucleus datasets fulfill this property. A further important issue is the surface sampling. While the icosahedron parameterization aims for providing an equidistant sampling of the surface, the equidistant sampling of the latitudes and longitudes does not result in an equidistant surface sampling. The result of an equidistant sampling of the sphere in longitudes ϕ and latitudes θ is displayed in figure 5.3. As a consequence, the sampling has to be chosen such that the sampling density at the equator is high enough for representing the finest surface structures. If we want to compute the surface curvature on the discrete grid, we additionally have to take care for the right scaling of the curvature depending on the sampling point distances.

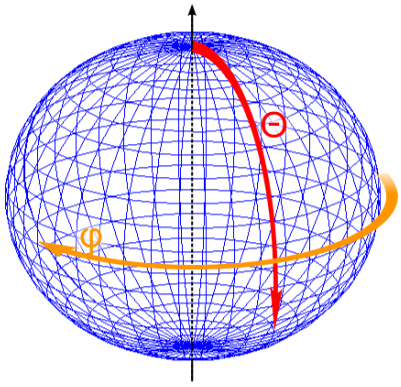


Figure 5.3.: Radial parameterization of the sphere with equidistant sampling in longitudes and latitudes. Close to the poles, the sampling points are closer to one another than at the equator.

5.2. Initialization

Since parametric active surfaces can only be locally optimized, the initialization of the surface is quite important. In order to get a satisfying segmentation result, the initial surface should not be much smaller than the object but it should not be infinitely large either: most importantly, it should not include more than one object. In this context, the initialization is mainly a detection task.
Since this work is focused on the handling of roundish objects like cells, cell nuclei or other subcellular structures, the method used for the detection of these objects in the raw data is also adapted to this kind of roundish objects.
Generally, the detection of spherical objects can be done with the generalized Hough transform for spherical objects [Ballard and Brown, 1981] as presented in [Schulz et al., 2006]. The main idea is to let each voxel vote for possible positions $\mathbf{c}$ of sphere centers with specific sphere radii r.
First, most datasets have to be smoothed for this purpose, in order to reduce noise. Then, one has to select a weighting function that determines the contribution of each image voxel to the voting scheme. This function can simply be determined using a threshold on the intensity values or on the gradient magnitude depending on the nature of the recorded data. Then, all voxels with intensity higher than the determined threshold would vote equally. Another possibility is to directly make use of the image intensities or gradient magnitudes as voting functions. Then, voxels with high intensity (or high gradient magnitude respectively) contribute stronger than others to the voting.
In [Schulz et al., 2006], the authors proposed to use the gradient direction to determine the position of the votes. This turned out to make the voting process much more stable. Finally, the votes are combined by integration. Formally, the four-dimensional voting space P of a function $I : \mathbb{R}^3 \to \mathbb{R}$ is computed as follows:

$$P(\mathbf{c}, r) = \int_{\mathbb{R}^3} \delta_\epsilon \left(\mathbf{c} - r \frac{\nabla (G_\sigma * I)(\mathbf{x})}{||\nabla (G_\sigma * I)(\mathbf{x})||} \right) ||\nabla (G_\sigma * I)(\mathbf{x})|| d\mathbf{x}, \tag{5.14}$$

where G_σ is the 3D Gaussian function with isotropic standard deviation σ. δ_ϵ is an indicator function giving contribution only if the argument is nearby zero and can also be considered as a truncated Gaussian with very small standard deviation. The detection is then done by determining the local maxima of the voting space P.

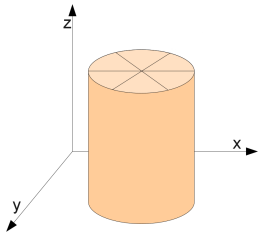


Figure 5.4.: Used bins for the voting directions.

An improvement of this method has been proposed by [Ronneberger et al., 2008]. The authors show that the voting becomes even more robust if the voting directions are also taken into account. Votes from only one direction are not as clearly an indicator for an object as votes from all directions are. Thus, in [Ronneberger et al., 2008] the vote directions for each center are binned. In practice, this results in k voting spaces P_i for a quantization of the voting directions into k bins. The final voting space P is then computed as the median over the P_i at every point $(\mathbf{c}, r)$.
The local maxima of the voting space indicate the centers $\mathbf{c}$ and the approximate radii r of the detected objects. Given these values, we initialized our active surfaces with spheres around $\mathbf{c}$ with radius r.

5.2.1. Evaluation

The cell detection was evaluated on the *Drosophila* S2 cell nucleus dataset described in section 3.2.1. For this dataset we used an isotropic Gaussian smoothing with $\sigma = 2\mu$m. Given the recorded resolution, the resulting estimation for the radii r is therefore imprecise, but as these radii are only used for initialization, these rough estimations are sufficient. For the voting, we choose to use all voxels and weight the votes with the respective gradient magnitude values. A threshold would not make sense, because the intensity variations of the nuclei within one dataset are too strong. The voxel-wise gradient magnitude and direction is used to place the votes. The detection is then done by determining the local maxima of the voting space P.
For the detection of the nuclei, we searched for spheres with diameters between 3.6 and 6.42μm. For the detection of the nucleoli, we searched for spheres with diameters between 1.28 and 3.21μm. The resulting voting space had a size of $50 \times 640 \times 640 \times 22$ and $50 \times 640 \times 640 \times 15$ respectively, where $50 \times 640 \times 640$ is the dataset size in voxels. The nucleus segmentation masks were used for the detection of the nucleoli, such that we can ensure that every nucleolus lies inside a nucleus. Only the Hough-votes within the nucleus were evaluated and exactly one nucleolus was searched inside each nucleus, since for healthy cells, there should only be one nucleolus. We computed the detection with and without the vote direction binning for comparison. For the direction binning, we distinguished between six voting directions, all projected onto the xy-plane (see figure 5.4). As the final voting result, we took the value of the third smallest bin, i.e. we require votes from four out of six directions. Votes from below and above are not distinguished because the data is strongly blurred in z-direction.

5.2.2. Results

Although a variety of deformations of the nuclei can be observed, a quite reliable detection of the nuclei was possible. Nuclei clearly lying on the border of the captured dataset were left out of the

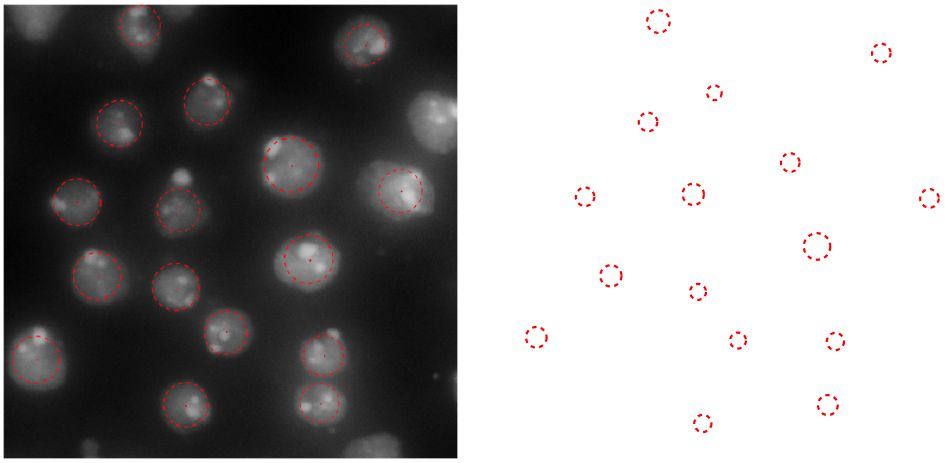


Figure 5.5.: (left) Detection results of nuclei displayed in the maximum intensity projection (MIP) of channel 0. (right) Detection results of nucleoli in the MIP of channel 1. We only detected one nucleolus per cell nucleus.

evaluation.
With the basic method without vote binning, we correctly detected 437 cell nuclei out of 440, 3 nuclei have not been detected. There were 16 false positives: 8 nuclei were detected in the background and 8 dead cells were erroneously detected as nuclei. Due to the relatively strong smoothing, the estimated radii were too small for some of the nuclei (compare figure 5.5).
Correct nucleolus positions were found for all cells. However, despite the fact that this might not be the norm for healthy cells, there are cells in the datasets containing more than one clearly defined nucleolus. In these cases due to the design of the evaluation, one of the nucleoli was missed.
Detection results for an example dataset can be seen in figure 5.5. With the vote binning method, the results could even be slightly improved. As without vote binning, 437 out of 440 nuclei were detected but only 11 false positives were found: 6 on dead cells and 5 in background regions. As before, for every correctly detected cell nucleus, we have correctly detected one nucleolus.

5.3. Data Terms for the Segmentation of Cells and Sub-cellular Structures

When implementing edge-based, parametric active surfaces, a standard choice for the data term in the energy functional 5.2 is the gradient magnitude at the position of the surface, i.e. $E_{\text{data}}(\mathbf{\Gamma}) = \int_{\mathbf{\Gamma}} -\|(\nabla I)(\mathbf{\Gamma})\|^2 d\mathbf{s}$. The resulting data force field $\mathbf{F}_{\text{data}}$ consists of the gradient vectors $-\nabla\|\nabla I\|^2$. This naive force field can be improved by computing the gradient vector flow (GVF) on the gradient vectors (see section 2.3.3).

Especially when dealing with 3D microscopic data, one has to handle several challenges as there are blurred object edges due to the PSF (see section 3.1.3) of the recording system as well as artifacts caused by inhomogeneities in the fluorescent stains. Especially for widefield microscopic data, the PSF strongly blurs the recording and causes artifacts.

In the following we present methods for the generation of data force fields that we have developed to handle the different challenges in the segmentation of cells and sub-cellular structures, most notably the cell nucleolus and nucleus, from volumetric fluorescence microscopy. These methods are to a certain level specifically adapted to our data (see sections 3.2.1, 3.2.2, and 3.2.3). The segmentation of the *Drosophila* S2 cells data turned out to be especially tricky. In these data, we have recorded three channels. From these three channels, we want to segment two cell organelles, the nucleus and the nucleolus. The final, biological task is to compute the colocalizations of the protein pattern recorded in the third channel to these organelles [2]. For this data, we are dealing with a variety of problems inherent in the biological setting as well as with problems originating from the widefield microscopy technique. The challenges given in this dataset have inspired most of the proposed techniques.

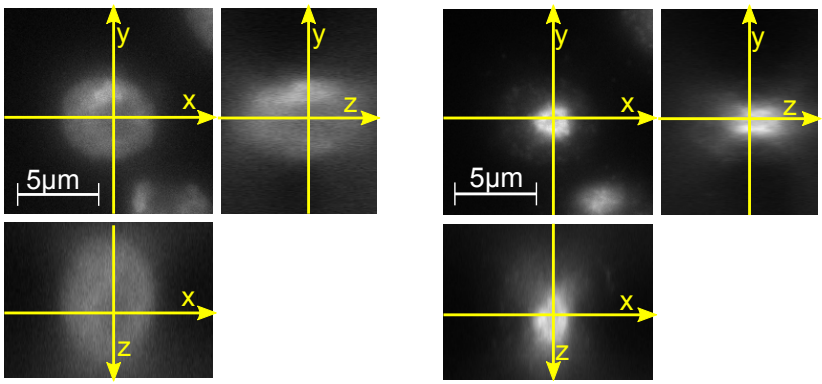
Although this data qualitatively looks much nicer when deconvolution techniques are applied, we have computed the segmentation on non-deconvolved data. Especially the information along the z-axis of the deconvolved data is not reliable because of the missing cone in the frequency domain. Deconvolution therefore produces artifacts in regions at the upper and lower nucleus and nucleolus boundaries. As we want to be sure to reliably locate the recorded protein pattern, we preferred to use the original data for all channels.

5.3.1. Channel Differential Structure

The segmentation of the nucleoli from the *Drosophila* S2 cell recordings appears to be an easy task at first glance, since the fibrillarin staining recorded in channel 1 yields clear bright structures. However, there are some challenges due to the widefield PSF that causes severe blurring and artificial, pointed structures in z-direction. Additionally, as mentioned in section 3.2.1, the true boundary has to be expected somewhere in between the boundary of the bright region in channel 1 and the "hole" in the nuclear staining recorded in channel 0 (compare figure 5.6). Basing the segmentation directly on gradients or the intensity values of channel 1 leads to elongated shapes that can even seem to protrude from the nucleus which, from a biological point of view, makes no sense. The method we proposed in [Keuper et al., 2009] to alleviate this problem is based on the idea to combine the information from the two recorded channels. The resulting method is called Channel Differential Structure (CDS).

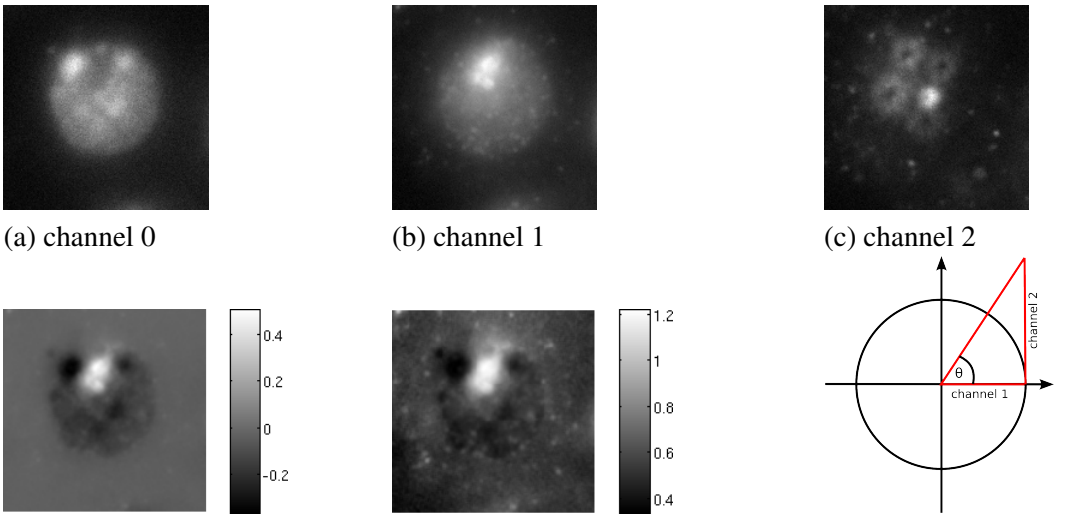
The CDS is inspired by the Color Differential Structure defined in [Geusebroek et al., 2003]. In [Geusebroek et al., 2003], this Color Differential Structure is computed as the convolution of the

[2] The concrete biological application is unpublished in the biological field and will not be a topic within this thesis.



(a) channel 0 (b) channel 1

Figure 5.6.: Three orthogonal views of the recording of a DAPI stained nucleus and GFP-stained fibrillarin inside the nucleolus.



(a) channel 0 (b) channel 1 (c) channel 2

(d) $I_{ch0} - I_{ch1}$ (e) $\arctan\left(\frac{I_{ch1}}{I_{ch0}}\right)$ (f) $\theta = \arctan\left(\frac{I_{ch1}}{I_{ch0}}\right)$

Figure 5.7.: (a)-(c) Three recorded data channels. (d)-(e) Measures to express the relation between recorded data channels. (f) The arctangent of the ratio between the channels corresponds to the angle between adjacent side and hypotenuse in a right triangle.

spectral color information with a Gaussian derivative to detect gradients between complementary colors.

For our task, not the perceptual color difference is important, which heavily depends on human color perception, but the relation between recorded channels in a multi-channel data set $I : \Omega \subseteq \mathbb{R}^3 \rightarrow \mathbb{R}^n_{>0}$. When for example considering the recording in figure 5.7 (a)-(c), one can clearly see that there is a strong correlation between the information of channel 0 and channel 1. In order to perform a segmentation of the nucleolus, it is therefore useful to point out regions where the two channels differ most, i.e. to compute some kind of differential of the two channels. Figure 5.7 (d) shows the difference between the two channels, that could directly be used to segment the nucleolus. Nevertheless, we chose a different measure to express the relation between the two

channels, namely the arctangent of their ratio, yielding a better representation of the nucleolus (figure 5.7 (e)). Positions, where the ratio of the channels has large values are likely to correspond to the nucleolus. The ratio of the intensities inside the nucleus are small, such that the boundaries of nucleus and nucleolus can not be confounded. The arctangent of the ratio corresponds to the angle θ in a right triangle (compare figure 5.7 (f)).
The $\mathrm{CDS}(I)$ of channel $i+1$ is finally computed as

$$\mathrm{CDS}(I) = \nabla \left(\frac{2}{\pi} \arctan \left(\frac{I_{i+1}}{I_i} \right) \right), \tag{5.15}$$

where $\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right)$ is the gradient operator and $\arctan \left(\frac{I_{i+1}}{I_i} \right)$ is scalar valued between 0 and $\frac{\pi}{2}$ since all measured intensities in I are ≥ 0.
For the segmentation of the nucleoli in [Keuper et al., 2009] in the *Drosophila* S2 cells, $\mathrm{CDS}(I)$ is used instead of the gradient on channel 1 ∇I_{ch1} in order to compute $\mathbf{F}_{\text{data}}$ with the standard GVF.

5.3.1.1. Active Surface Implementation

For the segmentation of the nucleoli, we chose the active surface parameterization based on a subdivided icosahedron mesh (compare section 5.1.2.1). Thus, the initial mesh we used had 162 vertices v_i organized in a triangle mesh. The icosahedron mesh was chosen, because it can represent arbitrary shapes with spherical topology. While we are quite certain that the nucleolus surfaces can in theory be well represented by star-shaped surfaces, we are not certain that we can find the right center point from which all surface points can be reached in the detection step for every nucleolus.

Regularization Since the computation of the principal curvatures on an arbitrary mesh is intricate, we base the approximation of the curvatures on the weighted first and second derivatives of the surface, which corresponds to minimizing the classical elasticity and rigidity energies [Kass et al., 1988] given in equation (5.3). The Euler-Lagrange equation for the active surface energy (equation (5.2)) is then given by

$$\underbrace{\alpha \mathbf{F}_{\text{ela}}}_{\text{minimizes the surface area}} + \underbrace{\beta \mathbf{F}_{\text{rig}}}_{\text{minimizes the total curvature}} - \nabla E_{\text{data}} = 0, \tag{5.16}$$

where the internal forces $\mathbf{F}_{\text{ela}}$ and $\mathbf{F}_{\text{rig}}$ prevent the surface from stretching and bending too much. Since $\mathbf{F}_{\text{ela}}$ minimizes the control point distances, it also has the sometimes undesired effect of shrinking the surface. To avoid shrinking effects, we allow $\mathbf{F}_{\text{ela}}$ only to pull the surface vertices v_i on a sphere around the object's center $\mathbf{c}$ with radius $\|\mathbf{\Gamma}(\mathbf{s}) - \mathbf{c}\| = \|\mathbf{x}_{v_i} - \mathbf{c}\|$ (see figure 5.8 for a two-dimensional visualization). A similar projection method for $\mathbf{F}_{\text{ela}}$ has been proposed in [Ahlberg et al., 2000].

The computation of the internal forces of the active surface have to be adapted for the three-dimensional mesh implementation. Since we want to preserve the equidistant sampling, the approximation proposed in [Ahlberg, 1996] for arbitrary polyhedral surfaces can be written as

$$\mathbf{F}_{\text{ela}}(v_i) = \frac{1}{c^2} \left(\sum_{v_j \in N(v_i)} \frac{\mathbf{x}_{v_j} - \mathbf{x}_{v_i}}{|\mathcal{N}(v_i)|} \right) \tag{5.17}$$

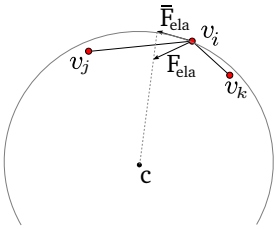


Figure 5.8.: The elasticity force that acts on a vertex v_i (red dot) is projected onto a sphere with center $\mathbf{c}$ at point $\mathbf{x}_{v_i} + \mathbf{F}_{\text{ela}}$.

for the elasticity force, where $\mathcal{N}(v_i)$ is the set of all neighbors of vertex v_i and c is the average distance between neighboring vertices and

$$c = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}(v_i)} \frac{\|\mathbf{x}_{v_j} - \mathbf{x}_{v_i}\|}{|\mathcal{N}(v_i)|}, \tag{5.18}$$

where $|\mathcal{V}|$ is the cardinality of the set of vertices. The rigidity force corresponding to the fourth derivative can by analogy be approximated as

$$\mathbf{F}_{\text{rig}}(v_i) = \frac{1}{c^4} \frac{1}{|\mathcal{N}(v_i)|(|\mathcal{N}(v_j)| - 1)} \sum_{v_j \in \mathcal{N}(v_i)} \sum_{\substack{v_k \in \mathcal{N}(v_j) \\ k \neq i}} (4\mathbf{x}_{v_j} - \mathbf{x}_{v_k} - 3\mathbf{x}_{v_i}). \tag{5.19}$$

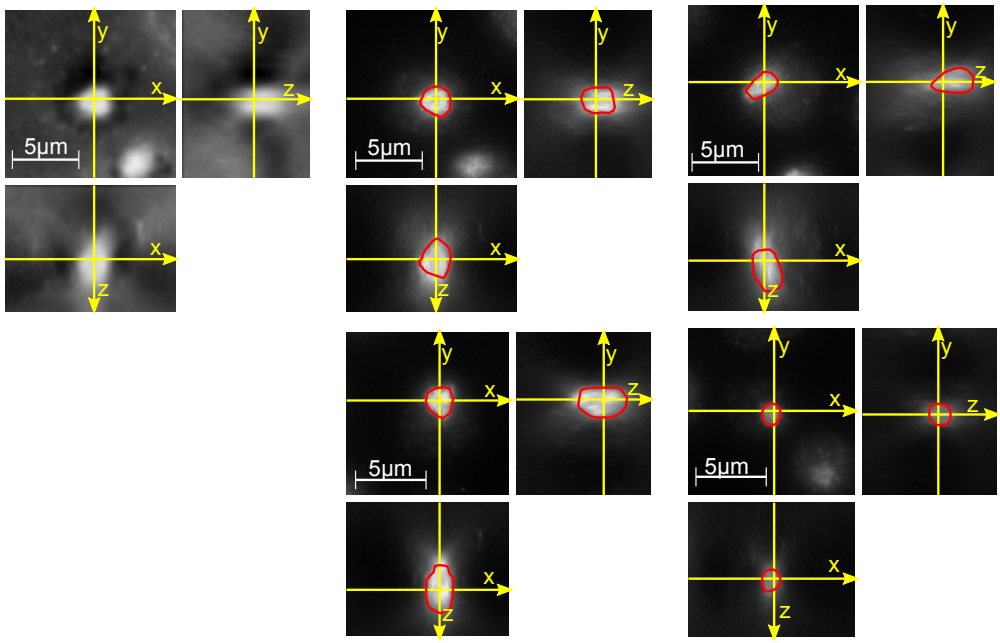
Slightly different implementations for curvature forces, that are only taking into account direct neighbors of v_i, have for example been proposed in [Zhang and Braun, 1997; Mille et al., 2006].
The surface points were initialized on a sphere around the automatically detected nucleoli with 1.5 times the estimated radius. The GVF on the three-dimensional gradient field was computed as described in appendix E. The weighting factors for internal energy term and data term were set to $\alpha = 0.1$, $\beta = 2$ and $\gamma = 0.5$.

5.3.1.2. Results

Some example results are shown in figure 5.9. It can be seen that the segmentation is good in the central xy-slices, and the elongated, pointed parts in z-direction are left out of the segmentation which is probably correct. Unfortunately, there is no ground truth available for this data which makes it hard to quantitatively judge the quality of the segmentation.

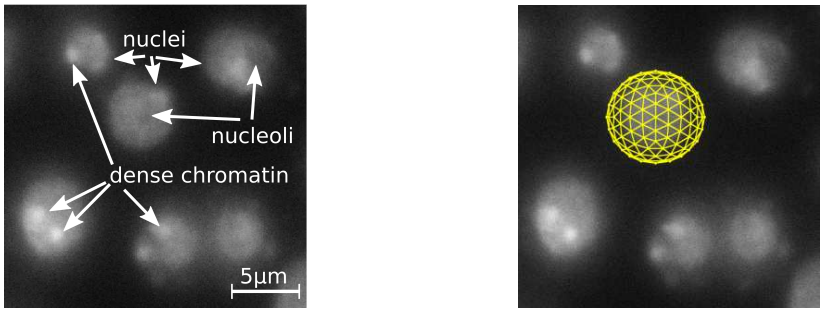
5.3.2. Detection-based Radial Prior for Cell Nucleus Segmentation

For the segmentation of the *Drosophila* S2 cell nuclei, we want to make use of the information we have gained in the detection step. This is necessary, because although all the chromatin inside the *Drosophila* S2 cell nuclei has been stained, the recorded nucleus intensities show strong inhomogeneities. The bright spots correspond to dense chromatin, low intensity regions are for example found in the nucleolus position, where there is no chromatin. If the nucleolus lies on the nucleus boundary, this leads to non-closed boundaries in the recording. This and the fact that there are typically touching cells in the dataset make the segmentation of the nuclei challenging. Figure 5.10(a) shows an xy-section of an example nucleus recording.



(a) $\arctan\left(\frac{I_{ch1}}{I_{ch0}}\right)$ (b) Segmentation results

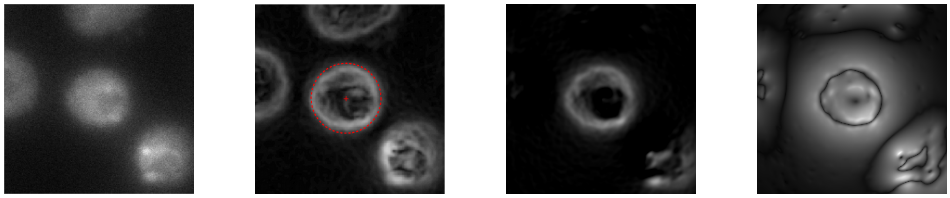
Figure 5.9.: $\arctan\left(\frac{I_{ch1}}{I_{ch0}}\right)$ computed for the dataset from figure 5.6 and four example results for the nucleolus segmentation with CDS.



(a) *Drosophila* S2 cell nuclei. (b) The same slice with the initial active surface grid.

Figure 5.10.: The cell nuclei stained in DAPI. Brighter regions are caused by denser chromatin. In the location of the nucleoli, no signal is recorded in this channel, thus causing a hole.

We assume the nuclei to have a star-shaped surface, i.e. there exists a point $\mathbf{c}$ such that each line segment connecting c to the object's boundary lies completely within the shape and we further assume that we have found a valid center $\mathbf{c}$ in the detection step. As proposed in [Ronneberger et al., 2008] for the segmentation of pollen grains from two-dimensional recordings, we used a projection of the dataset gradients onto radial vectors pointing away from the detected center $\nabla I_{\text{radial}}(\mathbf{x}) = \left\langle \nabla I(\mathbf{x}) \, , \, \frac{\mathbf{x}-\mathbf{c}}{\|\mathbf{x}-\mathbf{c}\|} \right\rangle$, thus reducing the influence of gradient vectors pointing in other directions. Additionally, as done in [Ronneberger et al., 2008], vectors originating from darker inner structures and thus pointing outwards were set to zero length. The resulting gradient image contains by far less gradients corresponding to structures other than the nucleus, but the vectors set



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

Figure 5.11.: (a) xy-slice from the original dataset (b) xy-slice from the gradient magnitude of the data with the estimated center and radius (c) xy-slice from the magnitude of the projected gradients $s\left(|\nabla I_{\text{radial}}\right)$, (d) xy-slice from the magnitudes of the gradient vector flow RGP GVF$(I) = \|\text{GVF}(\mathbf{A})\|$.

to zero length still cause problems in the next step.
Instead of applying the Canny edge detector as done in [Ronneberger et al., 2008], we directly use the resulting gradient magnitude as edge information. We compute the gradients of this edge image and, to get rid of the gradients caused by the zero-magnitude regions, we use the radial projection of these gradients. This results in a vector valued function $\mathbf{A} : \Omega \to \mathbb{R}^3$ with

$$\mathbf{A}(\mathbf{x}) = \left\langle \nabla s\left(\nabla I_{\text{radial}}\right)(\mathbf{x}),\ \frac{\mathbf{x}-\mathbf{c}}{\|\mathbf{x}-\mathbf{c}\|} \right\rangle \cdot \frac{\mathbf{x}-\mathbf{c}}{\|\mathbf{x}-\mathbf{c}\|}, \tag{5.20}$$

where $\langle .\,,\, . \rangle$ is the scalar product and $s(x)$ is defined as

$$s(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.21}$$

The data force field was finally found as the weighted sum of the GVF computed on $\mathbf{A}$ and the radially projected gradients, pulling the surface outside the object.

$$\mathbf{F}_{\text{data}} = -\gamma \text{GVF}(\mathbf{A}) + \eta \nabla I_{\text{radial}}, \tag{5.22}$$

where γ and η are constant weighting factors. The second term of $\mathbf{F}_{\text{data}}$ counteracts the shrinking effect of the internal active surface forces. If the internal forces are implemented without shrinking bias, for example with the projection of the internal forces onto the tangent plane (see section 5.3.1.1), η can be set to 0. In our original implementation used in [Keuper et al., 2009] however, the internal forces had a shrinking bias such that we used $\eta = 0.7$ in the original setting. The different steps leading to $\mathbf{F}_{\text{data}}$ are shown in figure 5.11 for an example cell nucleus in the central xy-section. The data force field based on radial gradient projections (RGP GVF) has some major advantages compared to standard gradient based force fields. On the one hand, the projection onto the radial vectors promotes stellar shapes, on the other hand, these projections and the deletion of gradient vectors pointing in the wrong direction has the effect that the capture range of the resulting force field is much larger. This is important, because of the touching cells in the dataset.

5.3.2.1. Evaluation

Since we do not have any ground truth labeling for the *Drosophila* S2 cell nuclei, we evaluated the new data force field (RGP GVF) on a synthetic dataset in order to produce quantitative results. On

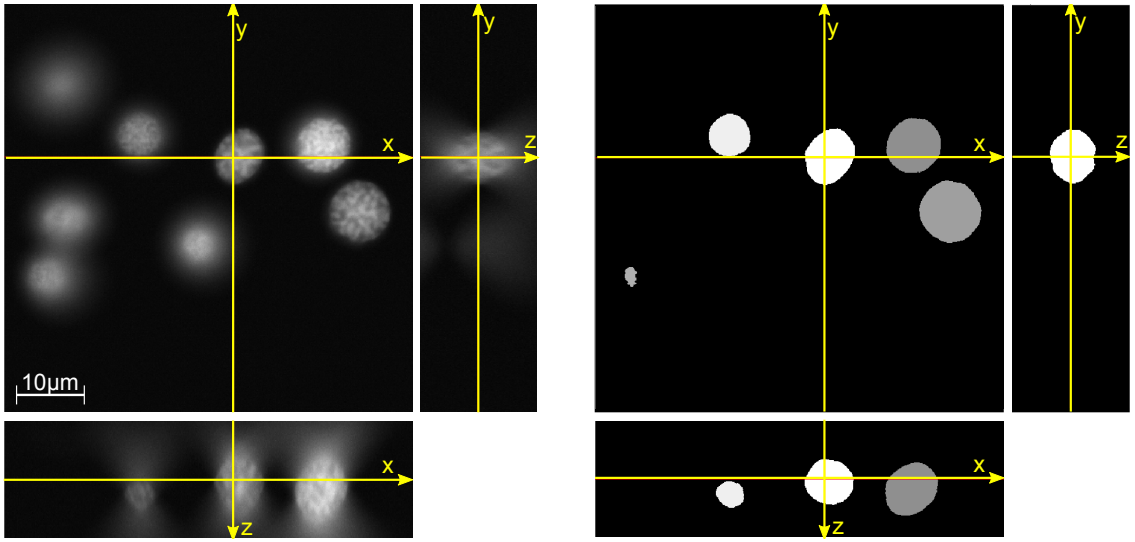


Figure 5.12.: An example of the synthetic data used for quantitative evaluation. In the ground truth data, each nucleus has a unique label that is indicated by the gray level.

the real microscopic recordings, we can only show examples of the segmentation results to give an impression of the segmentation quality.
The synthetic dataset was produced using the CytoPacq toolbox [Svoboda et al., 2009, 2007][3]. With this toolbox, we have simulated volumetric specimen, HeLa cell nuclei similar to the one used in sections 4.7.1 and 4.7.2 for deconvolution. In order to produce a segmentation scenario similar to the *Drosophila* S2 cell nucleus recordings, we simulated 19 recordings of eight cell nuclei randomly positioned in $20\times70\times70\mu$m^3 volumes, yielding 152 nuclei altogether. For the recording simulation, we used the optics simulation toolbox from CytoPacq with the ideal widefield PSF without spherical aberration downloaded from [4]. An example of our simulated recordings together with the original specimen mask that we will use as ground truth is given in figure 5.12 in three orthogonal views.

We segment these synthetic cell nucleus recordings using active surfaces with control points on a icosahedron subdivision grid with 162 initial vertices v_i, that were positioned on a sphere around the cell nuclei with an automatically estimated radius using the detection as described in section 5.2. The evolution is directly performed in the spatial domain as described in section 5.3.1.1. Since we also project the internal forces onto the tangent plane, the active surfaces do not show a shrinking bias. We prefer this implementation as it releases us from adapting the additional parameter η.The method is evaluated for slight variations of the weighting parameters α, β, and γ that weight the stretching, bending, and data consistency of the surface respectively. To make sure that we are getting comparable results, the data force fields GVF and the proposed RGP GVF were both normalized by the maximal vector length. In both cases, the GVF was computed with $\mu = 0.15$. The results are compared to the spherical initialization and to results that can be achieved with active surfaces using standard GVF force fields. Additionally, we report results we achieved with convex region based level sets [Chan et al., 2006] as described in section (2.3.6). The results

[3]The specimen and recording simulation toolbox is available for download under `http://cbia.fi.muni.cz/projects/cytopacq-a-simulation-toolbox_2.html`.
[4]`http://bigwww.epfl.ch/deconvolution/?p=bead`.

are given in terms of voxel precision, recall, and accuracy

$$\begin{aligned} \text{precision} &= \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}} \\ \text{recall} &= \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}} \\ \text{accuracy} &= \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}+\mathrm{fn}}, \end{aligned}$$

where tp, fp, and fn are the true positives, false positives and false negatives respectively. As we want to correctly locate protein patterns relative to the segmentation mask, the accuracy is the most important measure.
Since the image of a point is strongly elongated in z-direction due to the imaging technique, it remains doubtful if the original mask of the specimen is a good measure for the segmentation quality in the upper and lower image regions. Based on the image intensities, a slightly elongated segmentation visually would appear more plausible.
Since giving a correct measure for the segmentation quality in the upper and lower image regions is not straightforward, we remain with the pixel accuracy compared to the original specimen mask as a lower bound and focus more on recall than on precision for the evaluation on the whole volume. We additionally report the segmentation accuracy for the central xy-section of each cell, where the original specimen mask can without any doubt be used as Gold Standard.
The segmentation of the *Drosophila* S2 cell nuclei was done exactly the same way as the segmentation of the synthetic data. The initial active surface grid laid over one of the nuclei is shown in figure 5.10(b). Our results with RGP GVF on the real data will be compared to the results that can be achieved with an active surface implementation using a classical GVF force field and to a standard state-of-the-art method: the user-guided level set implementation ITK-SNAP [Yushkevich et al., 2006]. For the active surface segmentation, we manually tested some weighting parameter sets for three example nuclei and then used the best parameters for the segmentation of all the nuclei. Finally, we used $\alpha = \beta = 0.2$, and $\gamma = 0.7$ as for the synthetic data, but the method turned out to be very robust against small parameter variations. With those parameters, satisfying results could be achieved for the segmentation of all nuclei and nucleoli of the dataset. Some randomly chosen example results can be seen in figure 5.15.
The standard GVF force field was computed directly from the image gradients as $\mathrm{GVF}(\nabla\|\nabla I\|)$. The ITK-SNAP segmentation implements two algorithms: 3D geodesic active contours, where the internal forces are based on the gradient magnitude in the dataset, and a region competition method, based on voxel probability maps, which are estimated by manually adjusted intensity thresholds.

5.3.2.2. Results

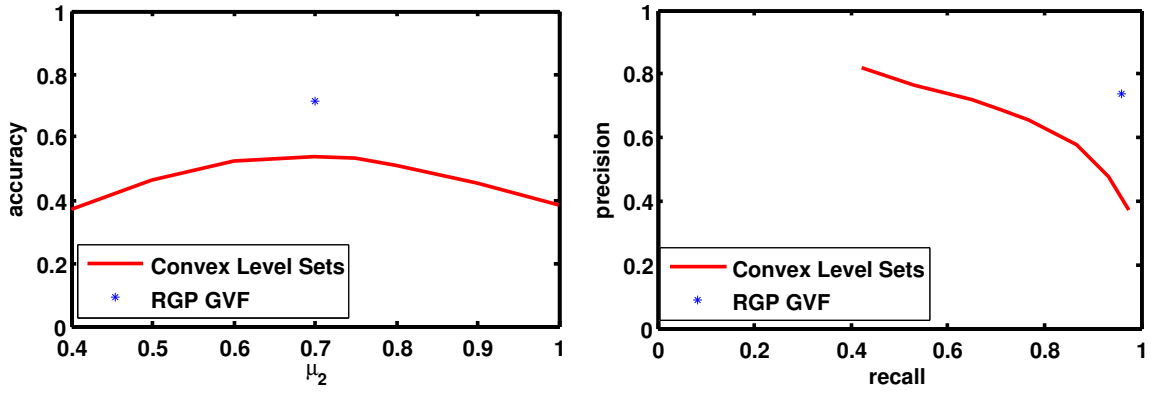
For the synthetic data, the overall precision, recall, and accuracy are given in table 5.1. The results on the central xy-sections can be seen in table 5.2. The accuracy of the proposed force field is significantly higher than the accuracy that could be achieved with standard GVF. It can also be observed that the method is not very susceptible to the parameter variations for the active surfaces. Although the diffused gradient fields from both methods GVF and RGP GVF are in the same range, the result with GVF is best if the GVF force field has least influence on the active surface evolution. On the whole volume, the GVF performs worse than the baseline where we directly take the spherical initialization as segmentation.
As we expected, region based level sets can not cope very well with our nucleus segmentation problem because of the unreliability of the measured pixel intensities. The results are given in figure

Table 5.1.: Segmentation results on the synthetic dataset evaluated in the whole volume.

	weights	precision	recall	accuracy
Sphere		82.13%	80.70%	66.13%
GVF	$\alpha = \beta = 0.2, \gamma = 0.5$	97.47%	60.01%	58.65%
GVF	$\alpha = \beta = 0.2, \gamma = 0.7$	98.07%	52.22%	51.35%
GVF	$\alpha = \beta = 0.3, \gamma = 0.8$	98.28%	52.13%	51.39%
RGP GVF	$\alpha = \beta = 0.2, \gamma = 0.5$	73.63%	95.51%	71.31%
RGP GVF	$\alpha = \beta = 0.2, \gamma = 0.7$	73.63%	96.02%	71.44%
RGP GVF	$\alpha = \beta = 0.3, \gamma = 0.8$	73.64%	95.85%	71.35%
Level Set	automatically determined μ_1 and μ_2	43.23%	95.73%	42.38%
Convex Level Set	$\mu_1 = 0, \mu_2 = 0.7$	65.54%	76.66%	53.94%

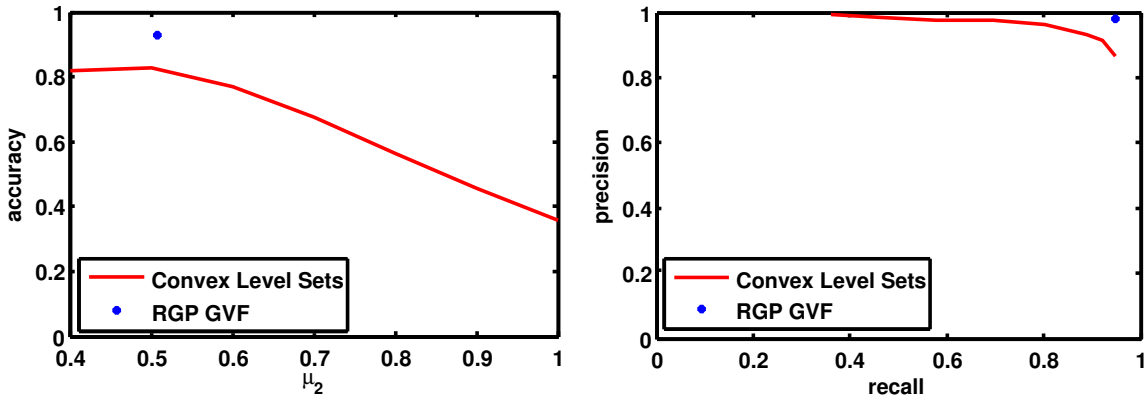
Table 5.2.: Segmentation results on the synthetic dataset evaluated in the central xy-section.

	weights	precision	recall	accuracy
Sphere		93.38%	83.95%	78.07%
GVF	$\alpha = \beta = 0.2, \gamma = 0.5$	99.22%	84.73%	84.15%
GVF	$\alpha = \beta = 0.2, \gamma = 0.7$	99.22%	79.15%	78.64%
GVF	$\alpha = \beta = 0.3, \gamma = 0.8$	99.19%	78.55%	78.03%
RGP GVF	$\alpha = \beta = 0.2, \gamma = 0.5$	98.13%	94.52%	92.83%
RGP GVF	$\alpha = \beta = 0.2, \gamma = 0.7$	98.14%	94.83%	93.13%
RGP GVF	$\alpha = \beta = 0.3, \gamma = 0.8$	98.25%	94.47%	92.88%
Level Set	automatically determined μ_1 and μ_2	91.31%	92.21%	84.20%
Convex Level Set	$\mu_1 = 0, \mu_2 = 0.7$	97.5%	69.47%	67.52%



(a) (b)

Figure 5.13.: (a) Accuracy of the convex level set implementation on the whole synthetic data, depending on the chosen mean value μ_2. The lower mean value μ_1 was set to zero. (b) Resulting precision-recall curve for the same parameters.



(a) (b)

Figure 5.14.: (a) Accuracy of the convex level set implementation evaluated only on the central xy-slice of the cells on the synthetic data, depending on the chosen mean value μ_2. The lower mean value μ_1 was set to zero. (b) Resulting precision-recall curve for the same parameters.

5.13 and figure 5.14. We have computed the segmentation on the whole synthetic recordings without the nucleus detection step. The intensities were normalized between zero and one. Depending on the chosen mean values for foreground and background, we either miss large parts of the nuclei or overestimate their shape (see figure 5.13). Since the segmentation heavily depends on the chosen mean values, we also tried to automatically estimate appropriate mean values by estimating them in every iteration from the current segmentation. For this experiment, the means were initialized with $\mu_1 = 0$ and $\mu_2 = 1$. The resulting precision, recall , and accuracy are given in table 5.1 and 5.2.
On the real *Drosophila* dataset, we compared our results to ITK-SNAP. For the methods implemented in ITK-SNAP, it was not possible to find weighting parameters that worked for the whole dataset. For four example cells, we manually adapted the ITK-SNAP parameters for each nucleus as well as possible, but even with this manual adaptation of the parameters to each cell we could not get good results for all of the cells. The 3D geodesic active contours did not work at all, because of the blurring in z-direction. It was not possible to find parameters that prevented the contour from running out of the object in the upper and lower dataset regions before filling the nucleus' volume in the center, where there are in fact stronger gradients. For the region competition method, the adjustable parameters are: Gaussian smoothing of the data, lower intensity threshold, upper intensity threshold, the initialization and the number of iterations. Event though we adjusted these parameters for every cell individually, the segmentation also grows slightly out of the nucleus in the upper and lower dataset regions. For a comparison of the standard GVF and the region competition method to our method, see figure 5.15. Although the region competition method from ITK-SNAP in most cases yields acceptable results if the manual thresholds are carefully adjusted, our method worked best for all of the nuclei. However, slight elongation of the surface due to the blurring by the PSF can still be observed.

5.3.2.3. Discussion

We have presented a method for the generation of data force fields for shapes with stellar topology. Based on the result of the automatic nucleus detection, we can eliminate misleading gradients from

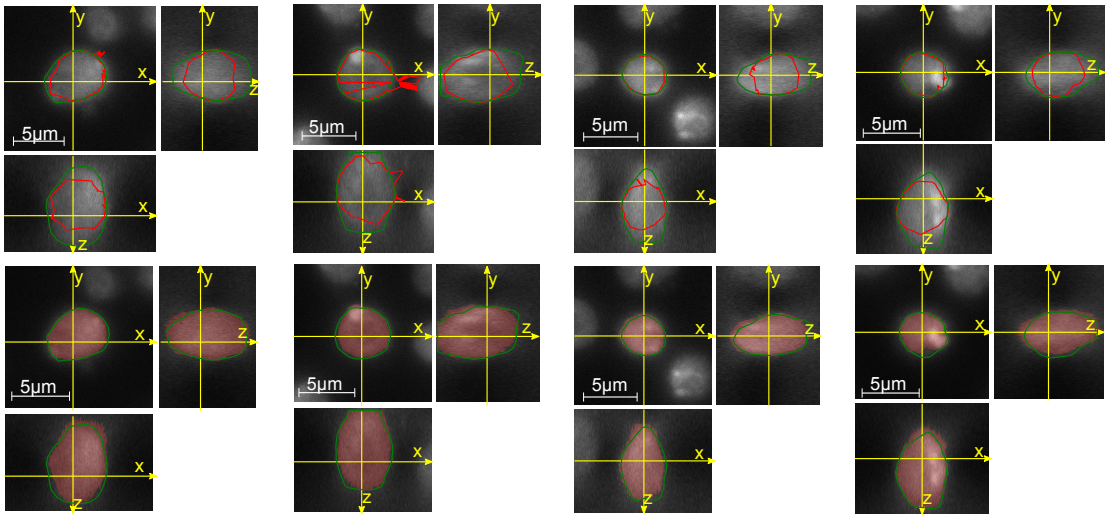
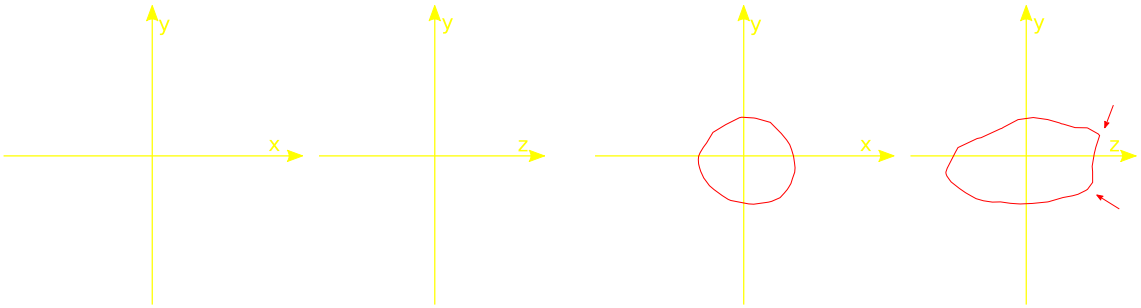


Figure 5.15.: (top row) Orthogonal views of the segmentation results of four nuclei. The red contours mark the segmentation result with the standard GVF force field whereas the proposed segmentation with RGP GVF is marked in green. It can be clearly seen that with GVF the contours underestimate the object in z-direction and tend to be attracted to neighboring cell nuclei and inner nuclear structures. This is not the case for our segmentation. (bottom row) Segmentation results for the same nuclei segmented with our method (green), and the region competition method from ITK-SNAP (red). The red masks are quite good, but elongated in z-direction, which is caused by the blurring in the dataset. In the last example, the mask leaves out a part of the nucleus where probably the nucleolus is located.

inner cell nucleus structures and neighboring cell nuclei. The resulting active surfaces outperform the surfaces based on standard GVF force fields and are quite robust to parameter changes. In contrast to the region-based segmentation with ITK-SNAP, where the parameters had to be manually adjusted for every nucleus, our method yields satisfying results with one fixed parameter set. A remaining challenge is, however, given by slight elongation of the resulting surfaces due to the PSF of the recording system.

5.3.3. Mean Shift Gradient Vector Flow

In section 5.3.2, we described our novel data force fields for the segmentation of DAPI-stained nuclei in a *Drosophila* S2 cell line. This method indeed leads to good contours but still, segmentation artifacts can be observed. One of the major problems is that the resulting active surfaces seem to be elongated and subject to PSF artifacts compared to the expected round shape. See figure 5.16 for an example of the data in two orthogonal views and the PSF of the used microscopy setting in the xz-view. In 5.3.2, we used GVF to smooth the projected gradient field before adapting the active surfaces. This leads to some desired effects like a large capture range for the active surface model as well as smooth gradient fields. On the other hand, the weak but important gradients in upper and lower image regions are smoothed away by the classical GVF.
To get rid of the artifacts caused by the dataset blurring, we have developed a vector diffusion method that not only preserves strong image gradients but also preserves weak gradients if they are



(a) Drosophila S2 nucleus in two orthogonal views

(b) The resulting segmentation with PSF artifacts.

Figure 5.16.: Two orthogonal views of a segmented cell nucleus. The pointed elongation of the segmentation in z-direction is a PSF artifact.

dense i.e. if all gradients in a neighborhood have similar direction and length [Keuper et al., 2010b; Keuper, 2010]. This method is the result of combining the basic GVF with the mean shift procedure [Cheng, 1995; Comaniciu and Meer, 2002]. In the classic GVF energy functional, the regularization term ensures that strong gradients (i.e. gradients with large magnitude) are not altered. When introducing the mean shift criterion into the energy optimization, we also ensure, that gradients are pulled towards their density modes. As a consequence, large gradients originating from small structures are preserved but do not diffuse as strongly as in GVF, i.e. have less impact on the overall force field. On the other hand, if in a certain region there are many weak gradient vectors pointing in the same direction, these valuable gradients will not be altered by larger neighboring gradients (as it would happen in GVF), but all gradients in the same neighborhood will converge to the mean direction and length.

5.3.3.1. Mean Shift on Gradient Vectors

The fundamentals of the mean shift procedure have been presented in section 2.4.2. In [Comaniciu and Meer, 2002], it is shown how the mean shift procedure can be employed as a discontinuity preserving image filter. This is done by interpreting the image as a two-dimensional lattice of p-dimensional vectors ($p = 1$ in the gray level case, $p = 3$ for color images). The lattice then forms the spatial domain, the color information forms the range domain of a $d = p+2$ dimensional feature vector in the joint spatial-range domain. The mean shift is performed using a multivariate kernel

$$K_{b_s,b_r}(\mathbf{x}) = \frac{C}{b_s^2 b_r^2} k\left(\left\|\frac{\mathbf{x}^s}{b_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^r}{b_r}\right\|^2\right), \tag{5.23}$$

defined as a product of two radially symmetric kernels with the common profile $k(x)$. $\mathbf{x}^s$ is the spatial part, $\mathbf{x}^r$ the range part of the feature vector. The Euclidean metric allows the use of a single bandwidth parameter in each domain, b_s and b_r respectively. These bandwidth parameters control the size of the kernel and thus determine the resolution of the density mode detection. C is the corresponding normalization constant.
An extension of the algorithm presented in [Comaniciu and Meer, 2002] to the three-dimensional space is straightforward. In the case of a three-dimensional vector field $\nabla M : \Omega \subseteq \mathbb{R}^3 \rightarrow \mathbb{R}^3$, where M is a 3D scalar field (the edge map), the mean shift is computed on the $d = 3 + 3$ dimensional feature vector, where the first three components contain the spatial information, the last three

dimensions contain the corresponding vector information.

$$K_{b_s,b_r}(\mathbf{x}) = \frac{C}{b_s^3 b_r^3} k\left(\left\|\frac{\mathbf{x}^s}{b_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^r}{b_r}\right\|^2\right), \tag{5.24}$$

As in the two-dimensional case, the Epanechnikov kernel (compare equation 2.81) performs satisfactorily.

When applying this edge preserving mean shift filter to gradient vector fields, the result is quite similar to what we know from 2D color images. Depending on the chosen spatial and range bandwidth parameters, the discontinuities at the dataset gradients are preserved and the vectors converge to equal length and direction in the neighborhood defined by the spatial bandwidth parameter.

See figure 5.17 for a toy example. The 2D toy data depicted in figure 5.17 (a)(120×120 pixels) is designed to simulate the blurred recordings of spherical objects with three undesired spots (spot 1-3) that should influence the overall force field as little as possible. The edge image in figure 5.17 (b) is the magnitude of the smoothed gradients of the toy data and the gradient field is displayed in figure 5.17 (c). The colors indicate the vector direction. In figure 5.17 (d), the GVF field (300 iterations with $\mu = 0.15$) of the toy data is displayed. In figure 5.17 (e) we have displayed the result of applying the 3D mean shift filter to our toy example (with parameters $b_s = 6$, $b_r = 0.4$). Because of the round shape of the object, the gradient vectors converge to the mean direction in their neighborhood, i.e. all vectors are pointing towards the edges. The result is depending on the filter bandwidth. Vectors with $\|\mathbf{v}_i - \mathbf{v}_j\| \leq b_r$ converge to equal length if they are spatially close to one another. Maxima in the gradient magnitude are not preserved. Below the actual gradient image in figure 5.17 (e), we have plotted the profile of the original gradient field (blue) and the mean shifted field (red) along the red line in figure 5.17 (a).

5.3.3.2. Mean Shift GVF

With the GVF, an isotropic vector diffusion is performed. The regularization only depends on the vector length disregarding the significance of the vectors in their local neighborhood. In regions, where the gradients are generally weaker, the Laplace term dominates equation 2.36 and a strong smoothing is performed. In our example of the microscopic recordings of cell nuclei, this is not always desirable. First, we do not want to promote strong gradients, if they are originating from small structures. Second, in the strongly blurred upper and lower dataset regions, we want to keep the gradients pointing towards the cell nuclei even though they are weak.

Generally, we want large gradients originating from small structures to be preserved but not to diffuse as strongly as in GVF, i.e. to have less impact on the overall force field. Then again, if in a certain region many weak gradient vectors point into the same direction, these valuable gradients shall not be altered by larger neighboring gradients (as it would happen in GVF), but all gradients in the same neighborhood shall converge to their mean direction and length. This is why we are using the advantages of the mean shift filtering method and the GVF simultaneously, performing a minimization of the GVF energy functional from section 2.3.3, equation (2.36) and a maximization of the kernel density estimate $\hat{f}_{\mathbf{b},K}(\mathbf{v})$ at the same time, which we call mean shift gradient vector flow (MSGVF).

Note that the density estimate is depending on two bandwidth parameters b_s and b_r with $\mathbf{b} := \left(b_s, b_r\right)^T$ for the spatial and range domain respectively. We thus want to find a vector field $\mathbf{v}$ that makes the best compromise between optimizing

$$\min_{\mathbf{v}} \int_{\Omega \subseteq \mathbb{R}^3} \mu\left(\left\|\nabla\left(\mathbf{v}\right)\right\|^2\right) + \left\|\nabla M\right\|^2 \left\|\mathbf{v} - \nabla M\right\|^2 d\mathbf{x} \tag{5.25}$$

and

$$\max_{\mathbf{v}} \int_{\Omega \subseteq \mathbb{R}^3} \hat{f}_{\mathbf{b},K}(\mathbf{v}) d\mathbf{x}. \tag{5.26}$$

This means, that we are diffusing large vectors and at the same time pulling the vectors towards their density maxima. This second objective, the kernel density maximization, leads in fact to a discontinuity preserving filtering. Equation 5.25 and 5.26 are optimized by alternating the gradient descent for the optimization of equation (5.25) and the gradient ascent for the optimization of (5.26):

$$\begin{aligned} \bar{\mathbf{v}}^k &= \mathbf{v}^k + \mu \nabla^2 \mathbf{v}^k - \left(\mathbf{v}^k - \nabla M\right) \left\|\nabla M\right\|^2 \\ \mathbf{v}^{k+1} &= \mathbf{m}_{\mathbf{b},G}(\bar{\mathbf{v}}^k) - \left(\bar{\mathbf{v}}^k\right), \end{aligned} \tag{5.27}$$

where $\mathbf{m}_{\mathbf{b},G}(\mathbf{v}(\mathbf{x}))$ is the mean shift of the vector $\mathbf{v}(\mathbf{x})$ with kernel G and bandwidths $\mathbf{b}$ defined as in section 2.4.2. Unlike for GVF, the optimization can not be done for each channel separately, because $\mathbf{m}_{\mathbf{b},G}(\mathbf{v})$ is depending on all channels.
An alternative, theoretically more appealing approach incorporating the kernel density into the GVF framework is to formulate a new energy functional

$$E(\mathbf{v}) = \int_{\mathbb{R}^3} \mu \left(\left\|\nabla \left(\mathbf{v}\right)\right\|^2 \right) + \left\|\nabla M\right\|^2 \left\|\mathbf{v} - \nabla M\right\|^2 - \nu \hat{f}_{\mathbf{b},K}(\mathbf{v}) d\mathbf{x}, \tag{5.28}$$

similar to the original GVF functional with one additional term penalizing low kernel density estimates with a constant weighting ν. This energy functional is minimized by the following Euler-Lagrange equation:

$$\mu \nabla^2(\mathbf{v}) - \left(\mathbf{v} - \nabla M\right) \left\|\nabla M\right\|^2 + \nu \hat{f}_{\mathbf{b},G}(\mathbf{v}) \cdot \mathbf{m}_{\mathbf{b},G}(\mathbf{v}) = 0. \tag{5.29}$$

This yields the following iteration scheme

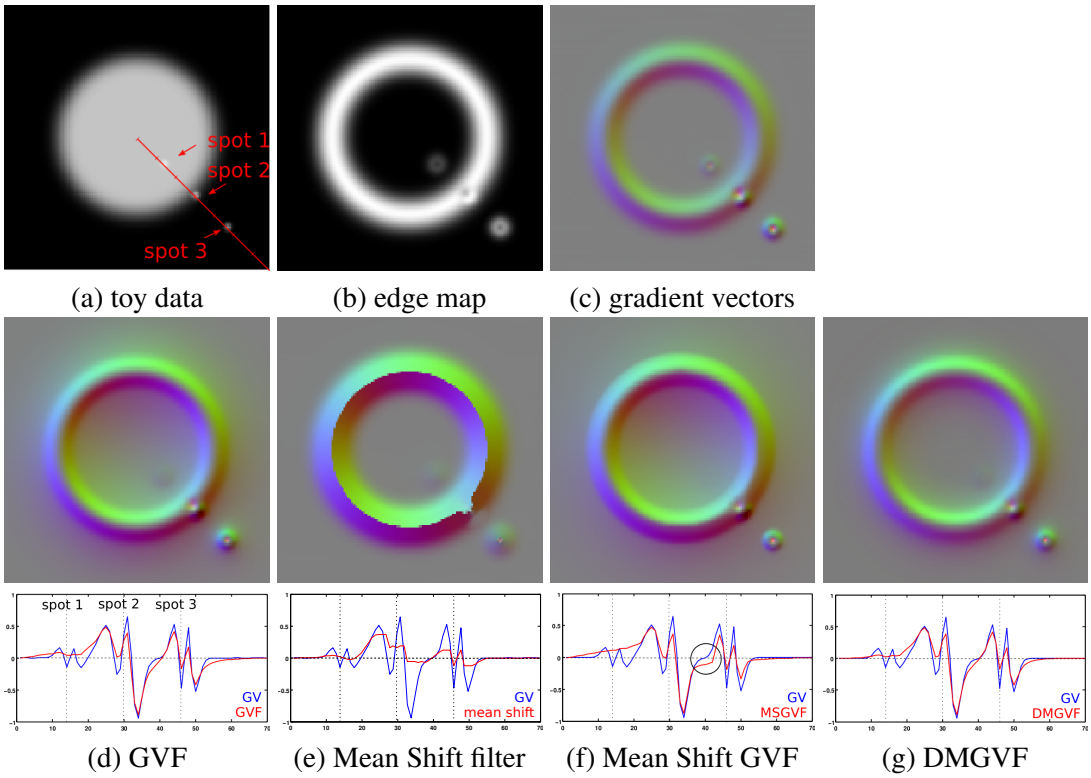
$$\begin{aligned} \mathbf{v}^{k+1}(\mathbf{x}) =& \mu \nabla^2 \mathbf{v}^k(\mathbf{x}) - \left(\mathbf{v}^k(\mathbf{x}) - \nabla M(\mathbf{x})\right) \left\|\nabla M(\mathbf{x})\right\|^2 + \\ & \nu \hat{f}_{\mathbf{b},G}\left(\mathbf{v}^k(\mathbf{x})\right) \cdot \mathbf{m}_{\mathbf{b},G}\left(\mathbf{v}^k(\mathbf{x})\right), \end{aligned} \tag{5.30}$$

which is in fact quite similar to the update scheme given in equation (5.27). The difference to (5.27) is that here we are using the mean shift computed from the previous iteration instead of computing it from the current vector field and weight it by the current kernel density and a constant weighting factor ν. We will refer to this second method as density maximization gradient vector flow (DMGVF).
The result of the MSGVF and DMGVF can be seen in figure 5.17 (f) and (g), for 100 iterations with $\mu = 0.15$, $b_s = 6$ and $b_r = 0.2$, and $\nu = 2$ for DMGVF. As when applying GVF in figure 5.17 (d), the discontinuities are preserved and long gradients corresponding to object edges are unaltered. As opposed to GVF, with MSGVF, gradients resulting from smaller objects have less impact on the overall result. This can be seen when looking at the profile in figure 5.17 (f). The small spot (spot 1) inside the object does not change the vector direction at all.
For a visualization of the vector directions, also compare the streamline plots in figure 5.18. When looking at the spot outside the object (spot 3) one can observe that strong gradients are caused by this structure, that are preserved by all three methods. Only the diffusion behavior is different. In the case of MSGVF, the zero crossing, indicating the change of vector direction, is shifted towards the small spot. Thus the capture range for our large object is larger and, even though we have

computed less iterations than for GVF, the magnitude of the vectors outside the object are larger. With DMGVF, the differences to GVF are only marginal. As with MSGVF, spot 1 inside the object has less impact on the force field but the larger capture range around spot 3 can not be observed. The exact results with DMGVF depend on the chosen weighting ν. In order to be able to compare the results to MSGVF, we kept the weighting reasonably small with $\nu = 2$. In MSGVF, the mean shift step is always weighted by one.

5.3.3.3. Evaluation and Results

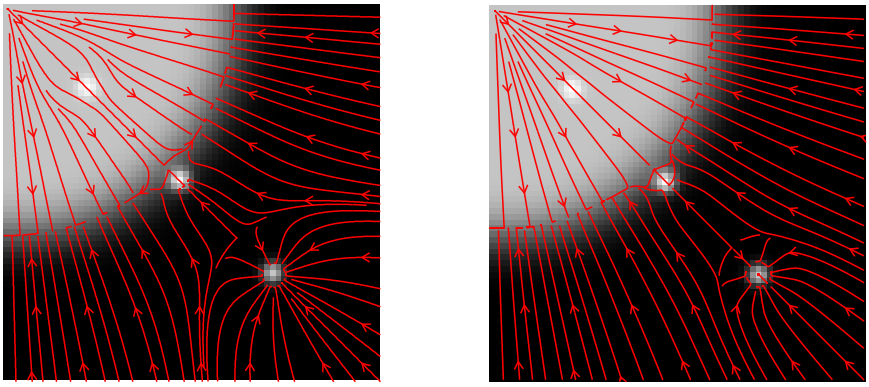
Both methods MSGVF and DMGVF have first been evaluated on the synthetic dataset described in section 5.3.2.1. We have computed the vector diffusion with both methods on the radially projected gradient field proposed in section 5.3.2 with the parameters $\mu = 0.15$, $b_s = 2$ and $b_r = 0.02$, and $\nu = 1$ for DMGVF. In [Keuper et al., 2009], the last step of the RGP GVF data force field generation is applying GVF to the vector field. For the segmentation, we only replace this last step by the proposed MSGVF and DMGVF method. Otherwise, the segmentation was done with the same active surface setting as in section 5.3.2.1. The quantitative results are given in tables 5.3 and 5.4. Since the artifacts we are looking for have a quite large effect on the shape but a rather small effect on the volume, the quantitative results show only a small improvement on the overall accuracy. As we already expected from the toy data, MSGVF yields slightly better results than



(a) toy data (b) edge map (c) gradient vectors

(d) GVF (e) Mean Shift filter (f) Mean Shift GVF (g) DMGVF

Figure 5.17.: The different vector diffusion and filter methods on toy data. The colors indicate the vector direction. Below the gradient field, the profile of the gradient fields along the red line in (a) is plotted for comparison.



(a) Streamline plot of the GVF field in the ROI.

(b) Streamline plot of the MSGVF field in the ROI.

Figure 5.18.: Especially when looking at the small spot outside the large object (spot 3), one can see that it has strong impact on the vector field, and this impact is significantly larger, when applying GVF (a) than when applying MSGVF (b). Thus, if we want to segment the main object, our initial surface must be closer to the object boundary with GVF than with MSGVF. Also, when applying MSGVF, spot 1 has no impact on the vector direction at all and even the capture range of spot 2 is less strong. So, both spots will not significantly alter the segmentation result.

Table 5.3.: Segmentation results on the synthetic dataset evaluated in the whole volume. The active surfaces have been computed with $\alpha = \beta = 0.2$ and $\gamma = 0.7$.

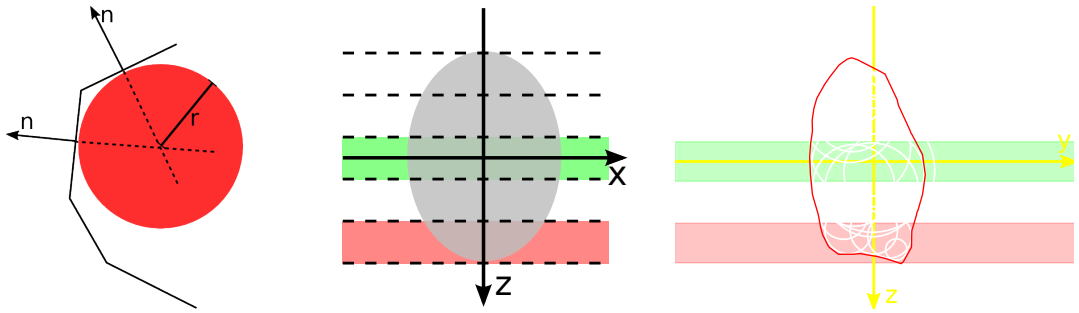
	precision	recall	accuracy
RGP GVF	73.63%	96.02%	71.44%
MSGVF	75.43%	96.03%	73.13%
DMGVF	74.11%	95.98%	71.87%

DMGVF.

This is why we have applied the new MSGVF method to generate force fields for the segmentation of *Drosophila* S2 cell nuclei. To allow for comparison with the previous method, all further steps of the active surface procedure are the same as in 5.3.2.

To measure the susceptibility of the segmentation surface to artifacts, we are locally approximating the surface by spheres and estimate their radii (see figure 5.19(a)). For the fitting, we use the fact that each two triangles in the surface mesh uniquely define a sphere by their normal vectors and centers of gravity. At the best, most radii should be similar to the radius of the cells that is somewhere near 2μm. The more artifacts we have in a surface, the more very small radii can be found. As we expect to find these artifacts in the lower dataset regions, we compute the measure of the lower 20% of the segmentation mask and compare it to the 20% in the center (see figure 5.19(b)).

The result of the evaluation can be seen in figure 5.20. The results show that the surfaces still are more strongly curved in the lower regions than in the center. Though, compared to GVF, the new method works clearly better. For GVF, the ratio of the number of triangles forming an angle in a reasonable range (radii of $1 - 3\mu$m) and those forming too small angles (radii $< 1\mu$m) is at 0.72, for MSGVF it is at 0.92. This can also be read from the slope of the curves in figure 5.20 (b). For



(a) Sphere fitting in the surface mesh.

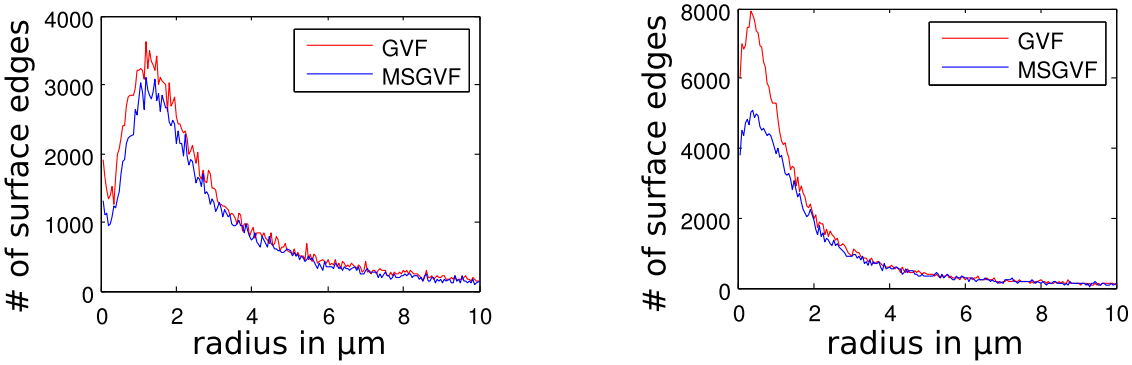
(b) Compared surface regions.

(c) overlay with a real dataset.

Figure 5.19.: Illustration of the surface evaluation method. Each two triangles in the surface mesh uniquely define a sphere by their normal vectors and centers of gravity.

Table 5.4.: Segmentation results on the synthetic dataset evaluated in the central xy-section. The active surfaces have been computed with $\alpha = \beta = 0.2$ and $\gamma = 0.7$.

	precision	recall	accuracy
RGP GVF	98.14%	94.83%	93.13%
MSGVF	98.16%	94.89%	93.22%
DMGVF	98.18%	94.80%	93.13%



(a) Histogram of the radii in the center.

(b) Radii in the lower 20% of the surface.

Figure 5.20.: Radius distribution of the spheres fitted in the triangle mesh.

GVF, the slope is much steeper than for MSGVF. Furthermore, the curves in figure 5.20 (b) show that the number of triangles in the lower dataset regions is drastically increased with GVF, which is evidence for more degenerate surfaces in this region. Having a closer look at the data (see figure 5.21 and 5.22, the red contour has been computed with GVF the green one with MSGVF), one can clearly see that in the central z-slices, the segmentation is very similar. Only in the more difficult lower regions, the result found by MSGVF is smoother and shows fewer artifacts. Additionally, one can see that most of the resulting surfaces are less elongated than with GVF. This results from the fact that in MSGVF, the vectors in the lower dataset regions that are pointing towards the surface

are generally better preserved. Sometimes, when applying GVF, this information is completely lost.

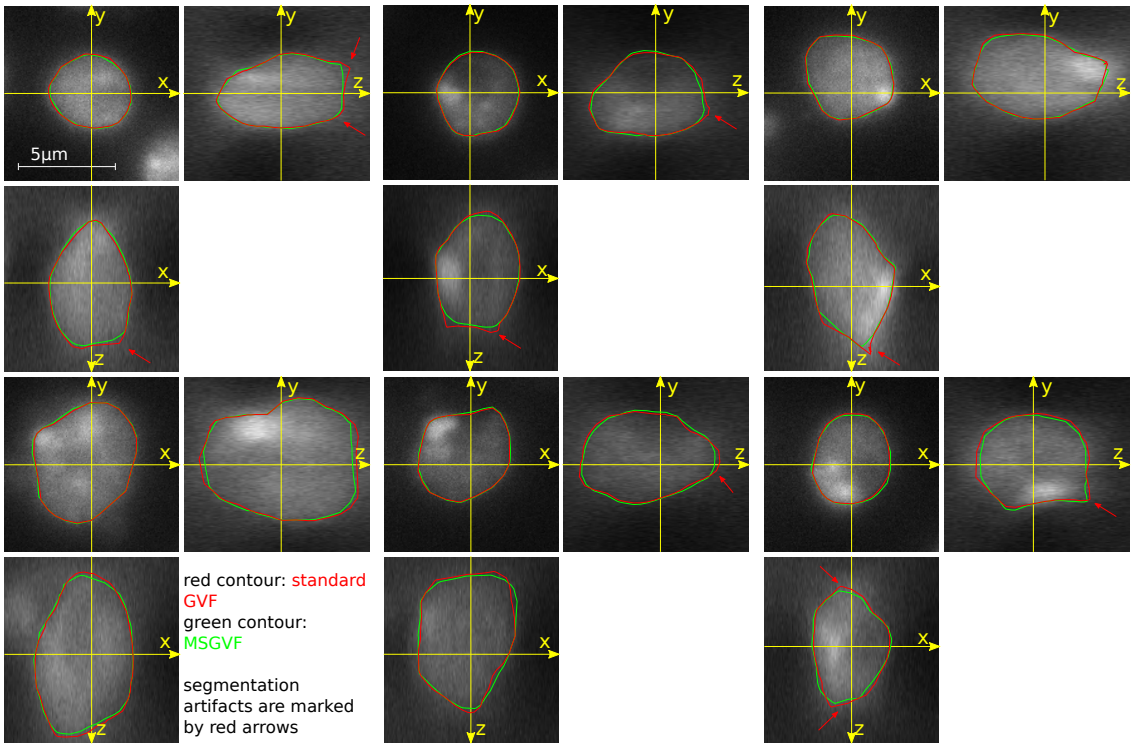


Figure 5.21.: Comparison between the segmentation using Mean Shift GVF and standard GVF on the radially projected gradients for six of the nuclei. The arrows mark positions where the contour with standard GVF becomes too pointed. The results are displayed in three orthogonal views

5.3.3.4. Discussion

The proposed MSGVF method improved the segmentation results for the *Drosophila* S2 cell nuclei by reducing PSF artifacts. The remaining slight elongation of the segmentation is not undesired since we want to be sure to correctly locate the also blurred protein pattern relative to the segmentation mask. A remaining challenge is the segmentation of cells with specially strong inhomogeneities where the segmentation is still not perfect. The MSGVF method, however, is not limited to the segmentation scenario with active surfaces. For example in [Chlap, 2010], MSGVF was used to improve the stability of detected landmark candidates in recordings of Zebrafish embryos.

5.3.4. Semi-Automatic Edge Filtering

In the previous sections, we have presented data force fields that allow for the fully automatic segmentation of fluorescently stained cell nuclei from microscopic recordings. However, depending on the specific application as well as on the imaging technique, the desired object boundary can have very different appearances. In the same dataset, e.g. of a plant cell, the user could be looking for either outer (the cell wall) or inner borders (plasmalemma), or intracellular compartments (as the nucleus or chloroplasts).

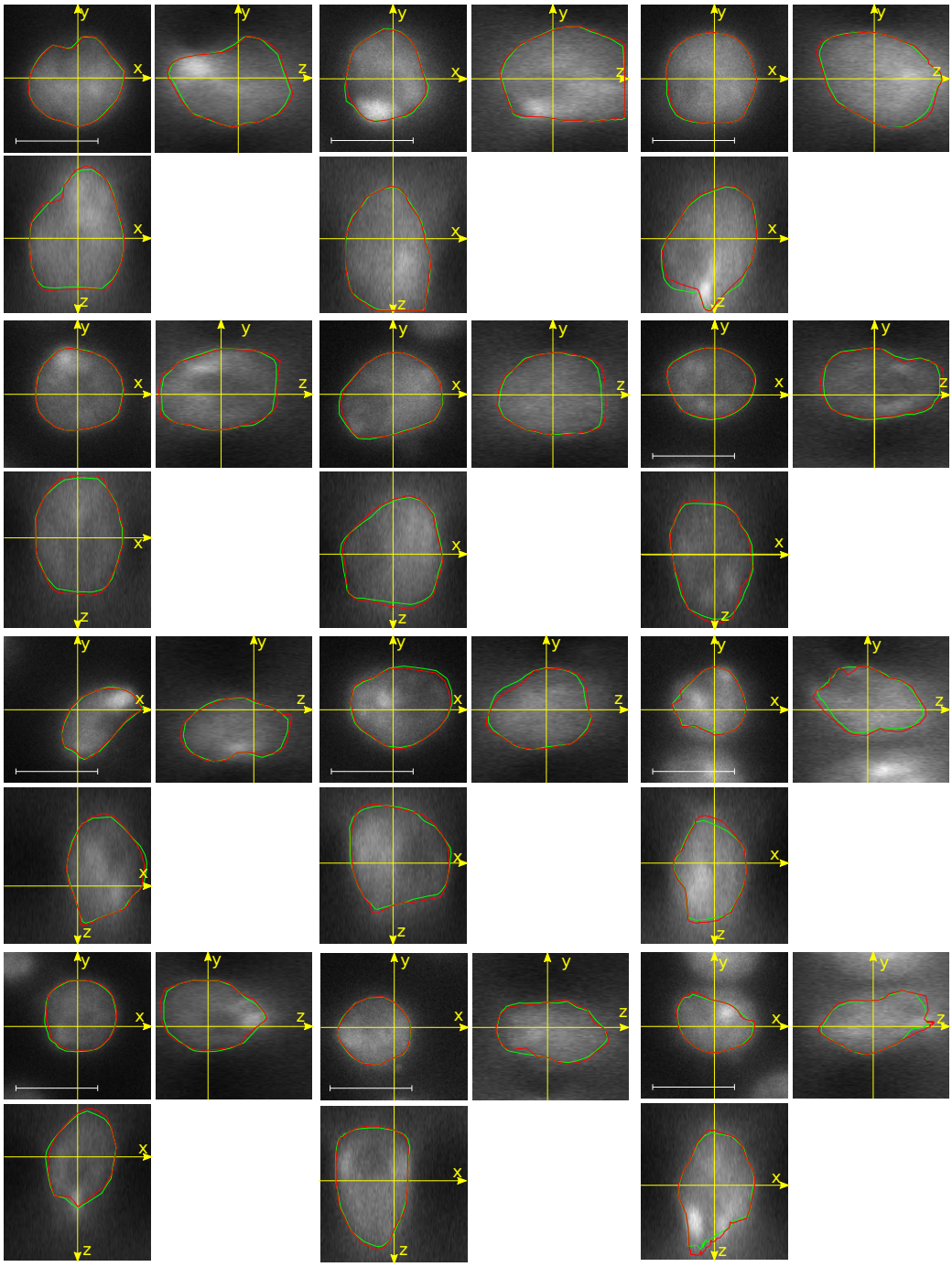


Figure 5.22.: More segmentation results with Mean Shift GVF (green) and RGP GVF (red) for comparison. The scale bars all indicate a length of 5μm. The results are displayed in three orthogonal views as in figure 5.21.

For every new problem setting, the used method needs to be adapted and special prior knowledge about the application has to be included. This knowledge usually can comprehend the texture of the object and the appearance of the object's boundary. This information can be learned from ground

truth data as for example done in [Cootes et al., 2001], if a sufficient amount of labeled data is available. The acquisition of ground truth segmentations is a very tedious step in 3D image analysis settings if it is feasible at all. One or more experts have to draw correct object boundaries in every single slice of every training volume. In some cases, however, the correct boundary might not be visible in one or several consequent slices.
In [Keuper et al., 2010a], we proposed a semi-supervised segmentation method. In this approach, expert knowledge for only a small number of training edge pixels is used to generate full segmentation masks. Thus it combines the advantages of expert guidance with convenient accomplishment.
A user guided tool for the segmentation of medical data has for example also been presented in [Vehkomäki et al., 1997]. There, the authors propose a twofold strategy: they create a graph description of contour fragments with a tessellation of the image plane. The actual segmentation is formulated as a path optimization, where the user has to manually select control points on the contour. The user guided level set implementation in ITK-SNAP [Yushkevich et al., 2006] allows the user, similarly to our method, to define the edge map before starting a level set segmentation. The edge map is defined by a threshold either on the data itself or on the gradients. Thus, the method works satisfactorily only if the edge information is homogeneous over the whole dataset.
In contrast, the presented segmentation method allows for changing appearances of the boundary. The user input is used to design an edge filter for the entire database based on a previous appearance based clustering of boundary candidates. Thanks to this clustering step, the needed user interaction is limited to very simple, quick and intuitive operations.
Since we are working on biological cell data, we have adapted the presented implementation to the segmentation of star-shaped objects. The evaluation was performed on 3D confocal recordings of developing plant cells.

5.3.4.1. Workflow

The general workflow of the presented method is displayed in figure 5.23. We assume that in the given database of recorded objects, the object detection step is already solved and for every object, the estimated position of the center $\mathbf{c}$ is given. For spherical objects, this detection step can for example be performed as described in section 5.2. Given this set of objects, the first step is to randomly choose a sample dataset. In this dataset, we find candidate positions where edge profiles are extracted. These edge profiles are used as features and are grouped into different clusters. The processing up to this step will be presented in detail in section 5.3.4.2. The result of the clustering is mapped into the original dataset: edges belonging to the same cluster are displayed in the same color. The next and most important step is the user interaction. From the displayed edge distributions presented in a 3D slice viewer, the user can decide which edges lie on the desired object contour. This information is used to design a filter for the specified edges, which is applied to the entire database. The generation of the filter is described in section 5.3.4.3. The filter response is used to perform a parametric 3D active surface segmentation using the frequency space representation described in section 5.3.4.4. The user can now verify the resulting segmentations and, if it is not sufficiently good for all datasets, choose the next training sample. The edge appearances in this new sample are used to refine the filter and accordingly, to refine the segmentation.

5.3.4.2. Profile Extraction and Grouping

In most applications, the user is interested in laying a boundary on certain positions with high gradient magnitude, i.e. image edges. In positions where the information is lacking, the user

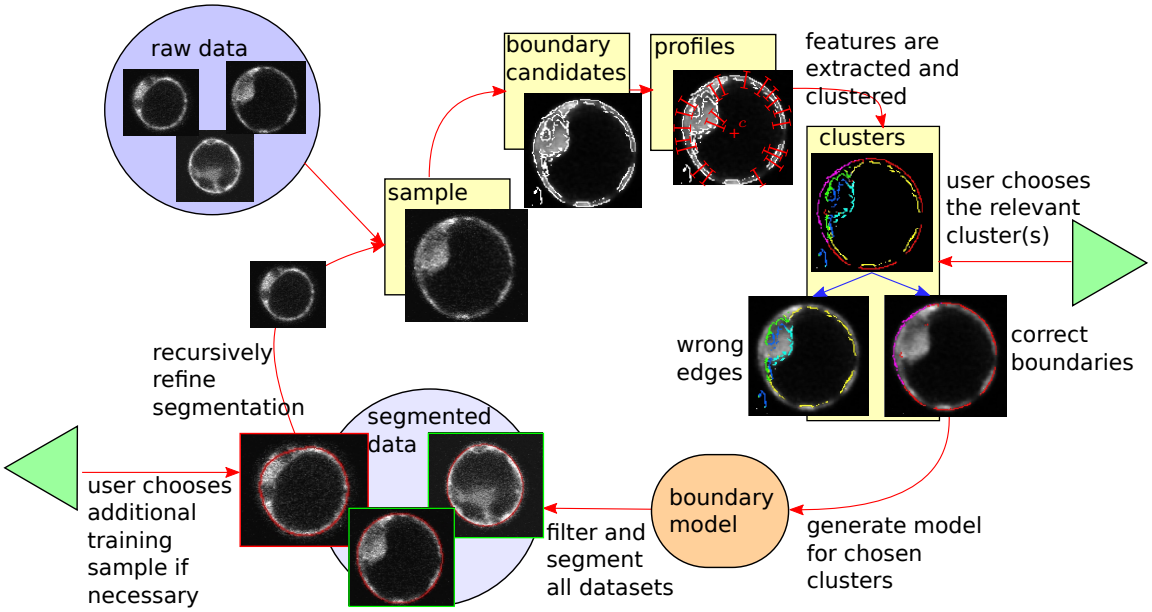


Figure 5.23.: Schematic overview over the whole workflow.

usually wants a smooth interpolation of the boundary. As candidate positions at which the profiles will be extracted, we thus choose points with high gradient magnitude by a threshold. To avoid finding too many candidates and to ensure that we are looking at the most important positions in the sample, we perform a non maximum suppression in gradient direction and take all the remaining maxima in gradient magnitude as candidates. At these positions, gray value profiles are extracted in radial direction from the center. This makes sense because we are assuming that the objects are star-shaped (i.e. there exists a point c such that each line segment connecting c to the object's boundary lies completely within the shape). When looking at more general shapes with spherical topology, one should extract profiles that are normal to the surface at this position, instead. For the extraction of the profiles, two parameters can be adjusted: the step size of the profile and its length. These have to be chosen such that the desired edge appearance can be captured and resolved.
We want to use these profiles as features to describe the appearance at the respective position. For microscopic data we are expecting strong variations in the absolute gray values even within the same recording due to absorption. To make our features robust against these variations, we use the derivative of the profile gray values, which is invariant against a gray level offset. The continuous profile derivative $dprof$ is a function of the radial length $l \in \mathbb{R}$ and the position $\mathbf{x}$. With respect to the center $\mathbf{c}$, $dprof(\mathbf{x}, l)$ is given by

$$dprof(\mathbf{x}, l) = \frac{\mathrm{d}}{\mathrm{d}l} I \left(\mathbf{x} + l \cdot \frac{\mathbf{x} - \mathbf{c}}{\|\mathbf{x} - \mathbf{c}\|} \right), \tag{5.31}$$

where I is the sample dataset. Compare figure 5.24 for a visualization.
The discrete profile vector $\mathbf{p_x}$ is given by

$$\mathbf{p_x}(i) = dprof \left(\mathbf{x}, \lambda \cdot \left(i - \frac{\mathrm{L}}{2} \right) \right), \tag{5.32}$$

where $0 \leq i < \mathrm{L} \in \mathbb{N}$ is the position on the profile, L is the profile length and λ is the step-size. In

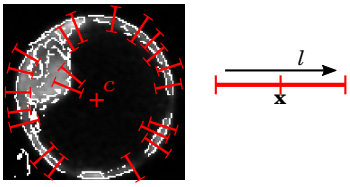


Figure 5.24.: The profile derivative *dprof* is a function of the radial length $l \in \mathbb{R}$ and the position $\mathbf{x}$ where it is extracted. Profile derivatives are extracted in radial direction from the object center.

order to be also invariant against multiplicative gray value changes, we normalize the features by the maximum absolute value.

$$\bar{\mathbf{p}}_{\mathbf{x}}(i) = \frac{\mathbf{p}_{\mathbf{x}}(i)}{\max_j(|\mathbf{p}_{\mathbf{x}}(j)|)} \tag{5.33}$$

The $\bar{\mathbf{p}}_{\mathbf{x}}$ are clustered using K-means clustering (see e.g. [Xu and Wunsch, 2008]). This basic clustering method is suitable for our purpose, because it directly measures the similarity between the profiles and discriminates as soon as the euclidean distance is too large. The number of clusters scales with the number of extracted profiles. In our implementation, there are on average 4000 profiles in one cluster.

5.3.4.3. User Specified Filtering

Once the profiles are grouped into different clusters C_j, the user has to decide which of the clusters are relevant for the actual application. Therefore, the data must be presented such that it is easy to distinguish between relevant and irrelevant clusters. For every cluster, we generate a 3D overlay plot in which all the points belonging to this cluster are marked in the original 3D sample data. The user can view this overlay and decide, whether the marked points lie on the desired boundary or not. This information is used to generate a Gaussian probability density function (PDF) for every chosen cluster, with

$$f_{\mathrm{C}_j}(\bar{\mathbf{p}}) = \frac{1}{(2\pi)^{l/2}|\Sigma_{\mathrm{C}_j}|^{1/2}} \cdot \mathrm{e}^{-\frac{1}{2}(\bar{\mathbf{p}}-\boldsymbol{\mu}_{\mathrm{C}_j})^T \Sigma_{\mathrm{C}_j}^{-1}(\bar{\mathbf{p}}-\boldsymbol{\mu}_{\mathrm{C}_j})}, \tag{5.34}$$

where Σ_{C_j} is the covariance matrix of all profiles $\bar{\mathbf{p}}_{\mathrm{C}_j}$ in the chosen cluster C_j, $|\Sigma_{\mathrm{C}_j}|$ denotes its determinant, and $\boldsymbol{\mu}_{\mathrm{C}_j}$ the vectorial expected value of cluster C_j. With these PDFs, the edge filter is already defined and can be applied to all objects in the database. In the filtering step, we have to extract the profile $\bar{\mathbf{p}}_{\mathbf{x}}$ at every position $\mathbf{x}$ in the dataset. For these profiles, we compute the Mahalanobis distance

$$D_{\mathrm{M}}(\bar{\mathbf{p}}_{\mathbf{x}}, \mathrm{C}_j) = \sqrt{\left(\bar{\mathbf{p}}_{\mathbf{x}} - \boldsymbol{\mu}_{\mathrm{C}_j}\right)^T \Sigma_{\mathrm{C}_j}^{-1} \left(\bar{\mathbf{p}}_{\mathbf{x}} - \boldsymbol{\mu}_{\mathrm{C}_j}\right)} \tag{5.35}$$

to every chosen cluster. The filter response is then given by

$$A(\mathbf{x}) = \min_{\mathrm{C}_j} \left(D_{\mathrm{M}}(\bar{\mathbf{p}}_{\mathbf{x}}, \mathrm{C}_j) \right). \tag{5.36}$$

A has low values, where the distance to the closest cluster center is small, i.e., where the profiles are similar to those belonging to the selected clusters. In order to have high responses at these positions, we compute $\bar{A} = 1 - A/\max(A)$. $\bar{A}$ is used to generate the data force field for the parametric active surfaces.
The filter response $\bar{A}$ is usually well representing all positions where the edges are similar to those chosen by the user in the sample data. Due to the profile normalization, we can even handle linear gray scale variations. On the other hand, the normalization also causes some spot-like filter responses in the background (compare figure 5.25(d)). Accounting for the fact that we are searching for object surfaces, i.e. locally planar structures, we can get rid of these wrong filter responses simply by applying the steerable filter for plane detection described in [Aguet et al., 2005] [5]. The filter response is used as edge map (EM). On the gradients of the edge map, we compute the standard GVF to get a smooth data force field. With this data force field, an active surface can be adapted to the data.
When the segmentation process with the active surface is finished on the whole database, the user can interact again. If the segmentation is not sufficient for all datasets, the user chooses a new sample and runs through steps 5.3.4.2 and 5.3.4.3. The new appearance clusters are added to the model and all wrongly segmented datasets of the previous iteration are segmented using this new model.

5.3.4.4. Active Surface Representation

For the cell segmentation, we have chosen the surface parameterization in spherical coordinates. The surfaces of our single, developing plant cells are star-shaped and the cell centers can easily be found such that this parameterization appears to be adequate. Functions on the sphere S^2, i.e. functions parameterized by longitudinal and latitudinal coordinates, can be represented in the frequency domain using (truncated) spherical harmonic expansions

$$f(\theta,\phi) = \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{f}_l^m Y_l^m(\theta,\phi), \tag{5.37}$$

where

$$Y_l^m(\theta,\phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{\imath m\phi} \tag{5.38}$$

with $\imath$ being the imaginary unit, and the $\hat{f}_l^m$ are the spherical harmonic coefficients with degree l and order m, B is the maximal bandwidth, where $|m| \leq l \leq B$. The P_l^m are the associated Legendre polynomials defined as

$$P_l^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} \left(P_l(x)\right), \tag{5.39}$$

where the $P_l(x)$ are Legendre polynomials. The recursive computation of the associated Legendre polynolmials is described in appendix D.1. For example, frequency domain parameterizations of parametric active surfaces have been used in [Miller et al., 1994; Khairy and Howard, 2008; Kelemen et al., 1999]. They have the advantage that they do not only yield smooth surfaces due to the representation by smooth basis functions, but also provide a parametric shape description that can be useful for further evaluation.
Star-shaped surfaces described by the Euclidean distance r from its center $\mathbf{c}$ to the surface points

[5] We used the C++ implementation of the algorithm from the ArrayToolbox library by Thorsten Schmidt.

that are parameterized as functions on the sphere with longitudinal and latitudinal coordinates can directly be represented in the spherical harmonic domain

$$\mathbf{\Gamma}(\theta,\phi) = r(\theta,\phi) = \sum_{l=0}^{B}\sum_{m=-l}^{l} \hat{r}_l^m Y_l^m(\theta,\phi), \tag{5.40}$$

where the $\hat{r}_l^m$ are the spherical harmonic coefficients with degree l and order m, yielding a shape representation by a vector of expansion coefficients

$$\hat{\mathbf{\Gamma}} = (\hat{r}_0^0, \hat{r}_1^{-1}, \hat{r}_1^0, \hat{r}_1^1, \ldots, \hat{r}_l^l)^T, \tag{5.41}$$

with

$$\hat{r}_l^m = \int_0^{2\pi}\int_0^{\pi} r(\theta,\phi) Y_l^{m*}(\theta,\phi)\sin\theta\, d\theta d\phi, \tag{5.42}$$

where Y_l^{m*} is the complex conjugate of Y_l^m. For a perfectly spherical shape with $r(\theta,\phi) = const.$ it follows $\hat{r}_l^m = 0$ for all $l, m \neq 0$.
If the surface sampling is equidistant, the vector of the L_2 norms of the spherical harmonic coefficients $\hat{r}_l^m$ is a rotation invariant shape descriptor [Kazhdan et al., 2003; Kazhdan, 2004]. Since the phase information is entirely lost, this shape descriptor is not complete, which means that the original shape can not be entirely recovered from the descriptor.
While the above representation is limited to star-shaped objects, spherical harmonics representation can also be used for the representation of arbitrary surfaces with spherical topology. This representation was used for example in [Khairy and Howard, 2008] to perform a parametric deconvolution of 3D images and in [Kelemen et al., 1999] to learn statistics for Hippocampus segmentation. While initialization and regularization are quite easy for star-shaped surfaces, these are critical issues for arbitrary shapes with spherical topology because this parameterization tends to artificially introduce sharp edges even at low bandwidth values (compare [Khairy and Howard, 2008]). This is why in [Kelemen et al., 1999], an icosahedron subdivision representation of the surface in the spatial domain is used additionally to the frequency domain representation in order to perform the surface refinement. More details on the representation of shapes with arbitrary topology with spherical harmonic basis functions are given in appendix D.2.

Regularization The radial parameterization with spherical coordinates has an advantage concerning the regularization of the surface. In most cases, no explicit smoothness constraint is needed, because the regularization can be done implicitly by limiting the bandwidth of the spherical harmonic expansion.
However, in some cases it might be useful to use a smoothness prior together with a high expansion bandwidth even with the radial parameterization. It might for example occur that an object has strongly curved regions on the surface that we want to capture. Still there might be other regions with no edge information at all, where we want to complete the surface with the smallest possible surface area. The smoothness of the surface is established by mean curvature flow. Let us denote the local mean curvature computed on the radially parameterized surface by $\tilde{H}$. The mean curvature flow then reads

$$\frac{\partial \mathbf{\Gamma}}{\partial t} = \tilde{H}\mathbf{n}, \tag{5.43}$$

with $\mathbf{n}$ being the surface normal. Because of the radial parameterization, this can be simplified to

$$\frac{\partial r}{\partial t} = \tilde{H}. \tag{5.44}$$

With the local mean curvature computed on the radial surface parameterization, mean curvature flow does not have the undesired shrinking effect, since $\tilde{H} = 0$ on a sphere of arbitrary size.
The spherical harmonic domain representation of the surface allows for an exact analytical computation of the surface normals and accordingly the surface curvature that is to be minimized. This way, we do not have to approximate forces acting on the surface but can compute them directly. In the following paragraph, we give formulas for the computation of the surface normals and curvature that are needed for the computation of the mean curvature flow.

Surface Curvature Computation The derivatives of the radially parameterized surface can be computed analytically as

$$\frac{\partial r(\theta,\phi)}{\partial \theta} = \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{r}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \theta} \quad \text{and} \quad \frac{\partial r(\theta,\phi)}{\partial \phi} = \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{r}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \phi}. \tag{5.45}$$

The resulting derivative vectors of the radially parameterized 3D surface are

$$\frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta} = \begin{pmatrix} 1 \\ 0 \\ \dfrac{\partial r(\theta,\phi)}{\partial \theta} \end{pmatrix} \quad \text{and} \quad \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi} = \begin{pmatrix} 0 \\ 1 \\ \dfrac{\partial r(\theta,\phi)}{\partial \phi} \end{pmatrix}. \tag{5.46}$$

The computation of the derivatives of the Spherical Harmonic basis functions is given in appendix D.1. The surface normal $\mathbf{n}(\theta,\phi)$ is computed as

$$\mathbf{n}(\theta,\phi) = \frac{\dfrac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta} \times \dfrac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi}}{\left\| \dfrac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta} \times \dfrac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi} \right\|}, \tag{5.47}$$

where $\times$ denotes the cross-product of two vectors. With the surface derivatives, the normal and derivatives of the normal, the first (E, F, G) and second (L, M, N) fundamental forms can be computed. The first fundamental forms are given by

$$\begin{aligned} E &= 1 + \left| \frac{\partial r(\theta,\phi)}{\partial \theta} \right|^2, \qquad F = \left\langle \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta}, \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi} \right\rangle \quad \text{and} \\ G &= 1 + \left| \frac{\partial r(\theta,\phi)}{\partial \phi} \right|^2. \end{aligned} \tag{5.48}$$

The second fundamental forms can be computed as

$$\begin{aligned} L &= -\left\langle \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta}, \frac{\partial \mathbf{n}(\theta,\phi)}{\partial \theta} \right\rangle \\ M &= \frac{1}{2}\left(\left\langle \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta}, \frac{\partial \mathbf{n}(\theta,\phi)}{\partial \phi} \right\rangle + \left\langle \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi}, \frac{\partial \mathbf{n}(\theta,\phi)}{\partial \theta} \right\rangle \right) \\ N &= -\left\langle \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi}, \frac{\partial \mathbf{n}(\theta,\phi)}{\partial \phi} \right\rangle. \end{aligned} \tag{5.49}$$

The local mean curvature is then given by

$$H = \frac{EN + GL + 2FM}{2(EG - F^2)}, \tag{5.50}$$

and the Gaussian curvature is computed as

$$K = \frac{LN - M^2}{EG - F^2}. \tag{5.51}$$

Active Surface Evolution As we are initializing the active surface as a sphere, $\mathbf{\Gamma}(\theta, \phi) =$ const. at the beginning. For every iteration, we project the forces that act on the surface onto their radial components, i.e.

$$\bar{\mathbf{F}}_{\text{data}}(\theta, \phi) = \left\langle \mathbf{F}_{\text{data}}(\mathbf{c} + \mathbf{S}(\theta, \phi) \cdot \mathbf{r}(\theta, \phi)), \mathbf{r}(\theta, \phi) \right\rangle \cdot \mathbf{r}(\theta, \phi), \tag{5.52}$$

where $\mathbf{r}(\theta, \phi)$ is the radial unit vector in direction (θ, ϕ) and $\mathbf{c}$ is the object's center. Then we compute their spherical harmonic expansion. The actual surface update can be performed in spherical harmonic domain. To get the new surface positions in the spatial domain, we then need to perform an inverse Spherical Harmonic transform. The spherical harmonic transform can be computed very efficiently using *S2kit* [6].

5.3.4.5. Evaluation

The evaluation was performed using the three sets of 3D CLSM recordings of living tobacco leaf protoplasts described in section 3.2.3. The whole segmentation process was performed on slightly smoothed data, we applied a Gaussian smoothing with $\sigma = 0.28\mu$m. For the user specified boundary filtering step, we had to specify certain parameters. The chosen profile length is 8 at a step-size of 0.56μm which is the double voxel size in xy-direction. The steerable filter, which was used for filtering out spot-like filter responses in the background, also has a parameter σ_s that specifies the thickness of the planes it searches for. We have set σ_s to 0.56μm. The active surfaces were initialized with the estimated radius r from the detection step. The bandwidth was limited to 16 bands. For the cells with stained Cyt, we have displayed the first training sample and the results of the different steps of the presented framework in figure 5.25 in two orthogonal views. The resulting surface is shown in figure 5.26 for one of the cells in a volume rendering.

5.3.4.6. Results

To evaluate our method, we applied the segmentation to all 187 cells in our database. For each of the three patterns, one cell was randomly chosen as first training cell. The segmentation results for all cells were visually inspected in a 3D slice viewer by two experts, who gave label 1 if the segmentation was correct, and label 0 if not. For those cells that were not correctly segmented after the first iteration, a second training step was performed: one of the cells with label 0 was chosen for each pattern as training sample. Altogether, we performed three iterations. The results can be

[6] *S2kit* is a freely available collection of C routines which compute the discrete Spherical Harmonic transforms of functions defined on the sphere. It is a light version of the SpharmonicKit and is developed at Dartmouth College by S. Moore, D. Healy, D. Rockmore and P. Kostelec. It is available at `www.cs.dartmouth.edu/~geelong/sphere/`.

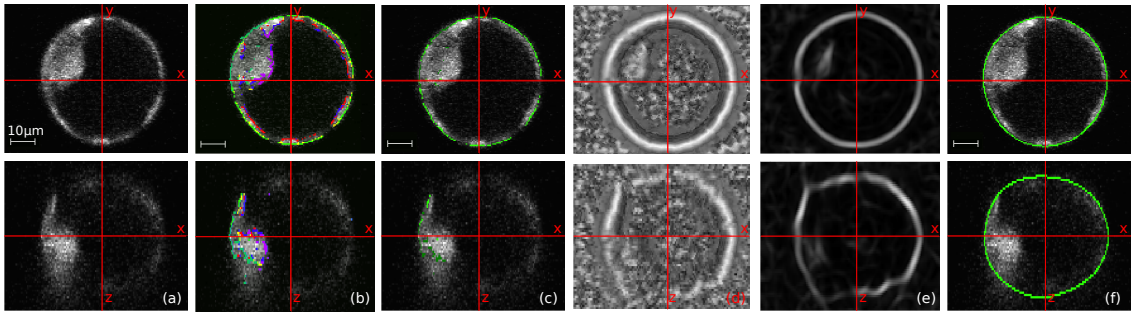


Figure 5.25.: (a) Sample with Cyt staining. (b) The clustering result from the K-means clustering with six clusters. Inner and outer boundaries lie, as expected, in different clusters. One can clearly see, that the absorption in z-direction leads to less candidates for edges and thus less training examples for the edge appearance in the lower z-regions. (c) The two clusters colored in green were chosen as relevant. (d) The filter response $\bar{A}$. (e) The used edge map (after the application of the steerable filter). (f) The segmentation result.

Table 5.5.: Results of our method after 1, 2 and 3 iterations.

Experiment	# of cells	Iteration 1	Iteration 2	Iteration 3
Cytoplasm	55	85.5%	**94.6%**	**94.6%**
Golgi	86	86.1%	88.4%	**96.5%**
ER	46	91.3%	95.7%	**97.8%**

seen in table 5.5. Most of the cells were already correctly segmented after the first iteration. Some results can be seen in figure 5.27.
For further evaluation, we compared the segmentation carried out with our method to the results that could be achieved with the ITK segmentation tool ITK-SNAP [Yushkevich et al., 2006] based on 3D geodesic active contours. The internal forces are based on the gradient magnitude in the dataset. The preprocessing as well as the active contour parameters have to be manually adjusted for each dataset, which is why we performed this segmentation only for three datasets. The ITK-SNAP preprocessing parameters are: the scale of the Gaussian blurring σ, which we set to 0.56μm, the edge contrast κ, which we set to 0.1 and the edge mapping exponent, which we set to 1.6. For the geodesic active surfaces, some parameters have to be adjusted as well. Here, we could not use the same parameters for all three cells. We manually initialized the contours from outside, because the internal structures of the cells made an initialization from inside impossible. The balloon force was set to a value between -0.7 and -0.9, the curvature force, that ranges from 0 (detailed) to 1 (spherical) was set to 0.8 and the advection force, that pushes the boundary back as it tries to cross edges, was set to 4.0. Additionally, the iteration has to be stopped manually. We needed between 461 and 791 steps to get the results displayed in figure 5.28. As can be seen, this segmentation tool can, despite all manual interaction, not handle the heterogeneous boundary and the absorption in z-direction.

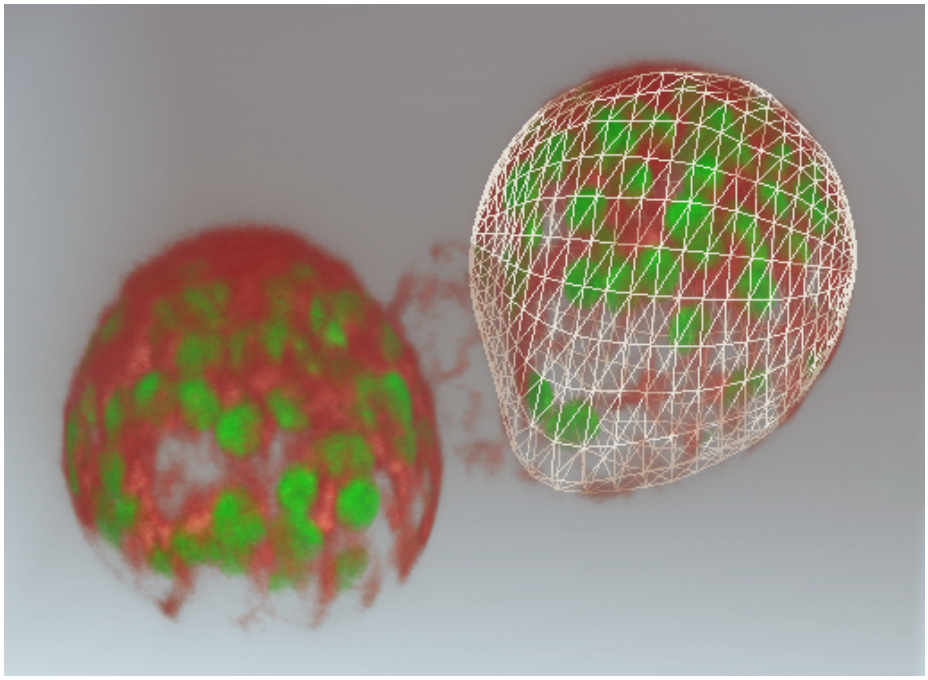


Figure 5.26.: Volume rendering of the two data channels of a recording showing three protoplast cells. The smallest cell between the two large cells does not have any chloroplasts yet, which is why only the protein pattern (red) is visible. The protein pattern has been used to produce the segmentation on one of the cells, here plotted as a surface mesh.

5.3.4.7. Discussion

Our new semi-supervised segmentation method for volumetric datasets can handle heterogeneous edge appearances. In our framework, edge models are learned from user input, while the user interaction is limited to very simple and intuitive operations and no further low-level parameters have to be adjusted. Although current 3D confocal recordings of single plant cells pose numerous challenges, such as inhomogeneous object boundaries, strong gray-value attenuations and noise, the segmentation of the datasets using the proposed method resulted in a highly reliable identification of cell boundaries.

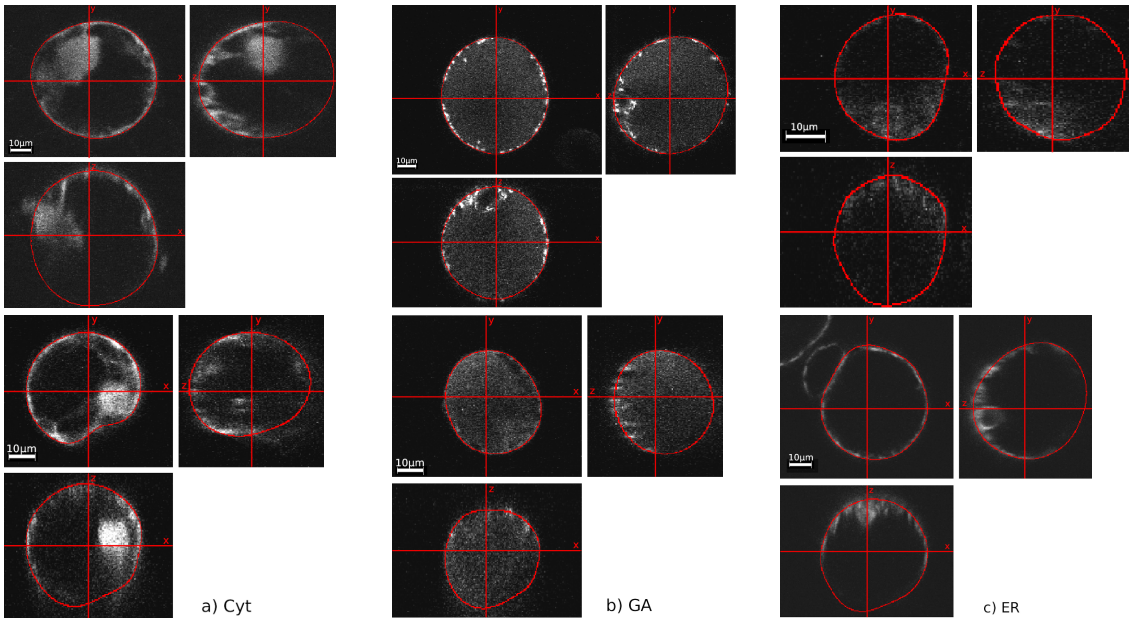


Figure 5.27.: Segmentation results in three orthogonal views. We have displayed two examples for every stained pattern.

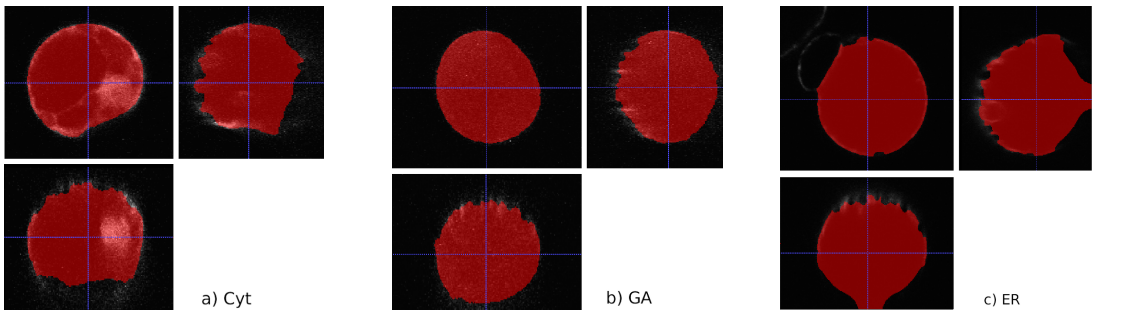


Figure 5.28.: Segmentation results with ITK-SNAP for the same cells as in figure 5.27, second row.

5.4. Weighting Parameter Adjustment

In the active surface energy functional, we always have weighting factors for the single energy terms. These parameters weight the forces pulling the surface during the surface evolution:

$$\mathbf{\Gamma}^{k+1} = \mathbf{\Gamma}^{k} + \nu \mathbf{F}_{\text{int}} + \gamma \mathbf{F}_{\text{data}}, \tag{5.53}$$

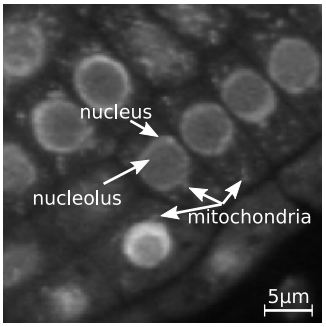
where ν and γ are the weighting parameters. In our implementation, $\mathbf{F}_{\text{int}}$ can usually be decomposed to elasticity and rigidity forces $\mathbf{F}_{\text{int}} = \alpha \mathbf{F}_{\text{ela}} + \beta \mathbf{F}_{\text{rig}}$ that are themselves weighted by α and β. In the remainder of this section, we will assume that the relation between α and β can be kept constant such that we can find an appropriate ν and define $\mathbf{F}_{\text{int}} = \nu \left(\frac{\alpha}{\beta} \mathbf{F}_{\text{ela}} + \mathbf{F}_{\text{rig}} \right)$. In our own implementation, $\frac{\alpha}{\beta} = 1$.

The choice of the weighting parameters that steer the relation between the internal active surface energy and the energy from the data term are crucial for the resulting segmentation quality, and in most cases the same weights do not work for different data sets. When for example looking at the

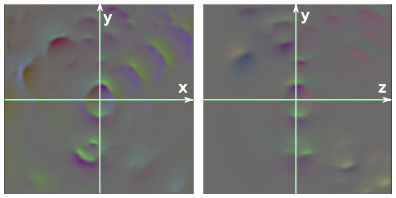
example slice from a DAPI stained *Arabidopsis Thaliana* root tip dataset shown in figure 5.29(a) the importance of the right choice of the weighting becomes obvious. Even though we are using the RGP GVF force field with large capture range, the surface can easily be "irritated" for example by the mitochondria visible in the data. The influence of the parameter choice on the segmentation result is shown in figure 5.29(b)-(e).
An automatic adjustment of these weighting parameters is thus desirable. In microscopic recordings of biological specimen, it is also often the case that the contour of the object we want to segment is not fully visible because of bad contrast. absorption, scattering, or staining artifacts. For example when imaging cell nuclei with DAPI staining, the nucleoli inside the nuclei are not stained and cause holes in the visible nucleus boundaries, whereas regions of dense chromatin result in very bright image regions and thus hamper a good segmentation. This is why we want our weight adjustment method to locally adapt the weighting parameters for every surface point separately - thus promoting smooth surfaces where the data is deficient.
In most cases when segmentation is done in biological data, we are searching for certain boundary properties, e.g. edge profiles, texture, etc., which are in most cases homogeneous over the whole object surface. This assumption can therefore build the basis for the parameter adjustment. In [Keuper et al., 2010c], we have proposed a method for the automatic and dynamic adaption of weighting parameters using exactly this fact. According to the underlying boundary homogeneity assumption, the external data driven active surface forces should be strongly weighted if a boundary estimate looks similar to the rest of the boundary. If a boundary estimate looks much different from the rest of the boundary, the data is considered deficient and high weights should be assigned to the internal active surface forces. One high level parameter still needs to be adjusted in advance. In order to be able to model the object boundary, we need to know approximately how much of the boundary information in the dataset is deficient. Thus we are expressing the classical low level weighting parameters ν and γ from the surface evolution forces in equation (5.53) by one high level parameter, which is the ratio of the object boundary that is certainly visible in the recording. Based on this knowledge, the weights are adjusted individually and dynamically for every surface node during the evolution of the active surface.
The main difference to the active appearance models as e.g. presented in [Cootes et al., 2001], where the relationship between model parameter displacements and residual errors is learned in a training step, is that for our method no prior knowledge about the concrete boundary appearance is included and no training has to be done. In [Allili and Ziou, 2008], a dynamic combination of boundary and region information for 2D segmentation problems is presented, which is, like our approach, based on probability maximization. Unlike in our method, the dynamic weighting is there included into a level set framework and used to combine region and edge information in 2D color images. The internal forces are not dynamically regulated. In [Sarti et al., 2000], a level set segmentation algorithm handling missing edges is presented, that is based on gradient flow. Unlike our algorithm, this method relies on a user-defined initial guess.

5.4.1. Dynamic Weight Adjustment

Since our method will be evaluated on two datasets of cell nucleus recordings, we assume in the following that the objects are star-shaped. Therefore, we are considering the gray value profiles $\mathbf{r}_{\mathbf{x}_i} \in \mathbb{R}^l$ of the surface vertices v_i in radial direction, where l is the number of sampling points on the profile. As for the profile gradients in the previous section, the boundary profiles along the model surface normals can be taken instead, for the segmentation of arbitrary shapes with spherical



(a) 3D CLSM recording of DAPI stained *Arabidopsis thaliana* root tip cell nuclei in xy-view.



(b) Force field RGP GVF

(c) $\nu = 0.9, \gamma = 0.1$

(d) $\nu = 0.5, \gamma = 0.5$

(e) $\nu = 0.1, \gamma = 0.9$

Figure 5.29.: Raw data of the *Arabidopsis thaliana* root tip and the data force field and segmentation results with fixed parameters in two orthogonal views. The influence of the choice of the weighting parameters ν and γ can be seen in (b)-(e).

topology. In every iteration, we compare the possibly new profiles given external forces only to the profiles of the previous step. This is done to encourage the surface to move according to external forces whenever the resulting position fits into the average of the profiles. If a new vertex profile is similar to the other profiles, the underlying data contains valuable information and thus the external forces shall have high weights. If a new vertex profile is not similar to the other profiles, the underlying data is considered to be deficient. In these cases, we want to promote smooth, sphere-like model surfaces and thus assign high weights to the internal forces. This main idea is formalized as described in the following.

To model the vertex similarities we have chosen a unimodal multivariate normal distribution over the vertex profiles. Note that this is different from the multimodal profile appearances we have learned in section 5.3.4.3 from user input. While we allowed for multimodal boundary appearances for the edge filtering method, we have to exclude this case here, where the profiles are evaluated fully automatically. The probability density function (PDF) is given by

$$f(\mathbf{r}_{\mathbf{x}_i}) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} \cdot \mathrm{e}^{-\frac{1}{2}(\mathbf{r}_{\mathbf{x}_i}-\boldsymbol{\mu}_\mathbf{r})^T\Sigma^{-1}(\mathbf{r}_{\mathbf{x}_i}-\boldsymbol{\mu}_\mathbf{r})}, \tag{5.54}$$

where Σ is the covariance matrix of all $|\mathcal{V}|$ profiles $\mathbf{r}_{\mathbf{x}_i}$ and $\boldsymbol{\mu}_\mathbf{r}$ is the vector-valued expected value. Considering the joint distribution is necessary because of the high correlation of the gray values on the profiles. The PDF f allows to compute the probability of a profile $P(\mathbf{r}_{\mathbf{x}_i})$. The vertices lie on valid data with the probability $P(\mathrm{B})$, which is the only high level parameter we use. Since we assume that the majority of vertices, e.g. 80%, lie on valid data, we compute a second PDF $g_f(\mathbf{r}_{\mathbf{x}_i})$ to describe the appearance of the 80% of the profiles $\bar{\mathbf{r}}_{\mathbf{x}_i}$ that are most probable according to f, i.e. that describe valid boundary information:

$$g_f(\mathbf{r}_{\mathbf{x}_i}) = \frac{1}{(2\pi)^{l/2}|\bar{\Sigma}|^{1/2}} \cdot \mathrm{e}^{-\frac{1}{2}(\mathbf{r}_{\mathbf{x}_i}-\boldsymbol{\mu}_{\bar{\mathbf{r}}})^T\bar{\Sigma}^{-1}(\mathbf{r}_{\mathbf{x}_i}-\boldsymbol{\mu}_{\bar{\mathbf{r}}})}, \tag{5.55}$$

where $\bar{\Sigma}$ is the covariance matrix of the $P(\mathrm{B}) \cdot |\mathcal{V}|$ most probable profiles $\bar{\mathbf{r}}_{\mathbf{x}_i}$ according to f and $\boldsymbol{\mu}_{\bar{\mathbf{r}}}$ is the respective expected value. With g_f, we can compute the conditional probability of a profile $\mathbf{r}_{\mathbf{x}_i}$

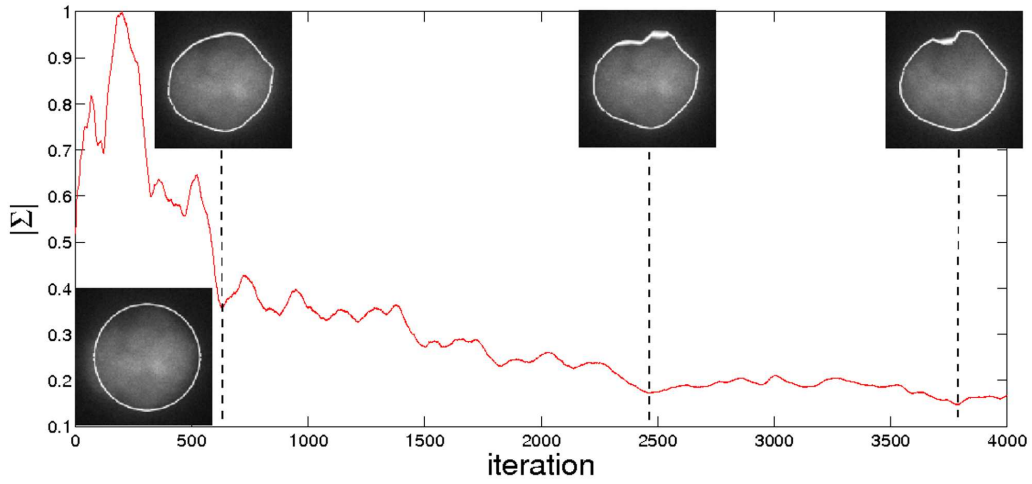


Figure 5.30.: $|\Sigma|$ during active surface evolution for one of the *Drosophila* S2 cell nuclei.

given the fact that it contains valid boundary information $p(\mathbf{r}_{\mathbf{x}_i}|\mathrm{B})$, and thus with Bayes Theorem the conditional probability of a profile $\mathbf{r}_{\mathbf{x}_i}$ to contain boundary information given its appearance:

$$p(\mathrm{B}|\mathbf{r}_{\mathbf{x}_i}) = \frac{p(\mathbf{r}_{\mathbf{x}_i}|\mathrm{B}) \cdot P(\mathrm{B})}{P(\mathbf{r}_{\mathbf{x}_i})}. \tag{5.56}$$

For every vertex v_i, we compute its new position $\mathbf{x}_i^{\text{ext}}$ given external forces only and the according profile $\mathbf{r}_{\mathbf{x}_i^{\text{ext}}}$. Then, we compute $p(\mathrm{B}|\mathbf{r}_{\mathbf{x}_i^{\text{ext}}})$.
The weights ν_i and γ_i that are assigned to vertex v_i in one iteration step are finally computed as:

$$\nu_i = 1 - p(\mathrm{B}|\mathbf{r}_{\mathbf{x}_i^{\text{ext}}}), \qquad \gamma_i = p(\mathrm{B}|\mathbf{r}_{\mathbf{x}_i^{\text{ext}}}), \tag{5.57}$$

to ensure high weights for the external forces if $\mathbf{r}_{\mathbf{x}_i^{\text{ext}}}$ fits well to the majority of the profiles, and to assign low weights for the external forces if the profile does not. In most cases, we do not really want ν_i to be 0, i.e. we want at least "some" smoothness. Thus we are adding a small ϵ to the ν_i.

During the evolution of the model, the determinant of the covariance matrix $|\Sigma|$ contains the information about the profile similarities. Once the model is attracted to the object boundaries, $|\Sigma|$ is small, whereas if the surface lies on heterogeneous regions, $|\Sigma|$ is large. This is why we are taking $|\Sigma|$ normalized by the maximum of its current value and its previous values as an additional convergence criterion, i.e. the evolution stops at step t, if the surface does not change or if $\frac{|\Sigma|_t}{\max(|\Sigma|_{1,\ldots,t})} \leq 0.3$. The normalization has to be done because $|\Sigma|$ is maximal if the surface lies on heterogeneous data, e.g. is partially converged. If the surface is initialized in the background, one might start with a very low $|\Sigma|$ (see figure 5.30). The value 0.3 is chosen manually.
In our implementation, we compute the profile for each vertex over ten sampling points in radial direction from the object center. The resulting covariance matrices Σ and $\bar{\Sigma}$ are thus 10×10 matrices.

5.4.2. Evaluation and Results

The method was first tested on toy examples (see fig. 5.31) and then evaluated on our synthetic cell nucleus data (compare section 5.3.2.1) and two different types of real cell nucleus recordings: on

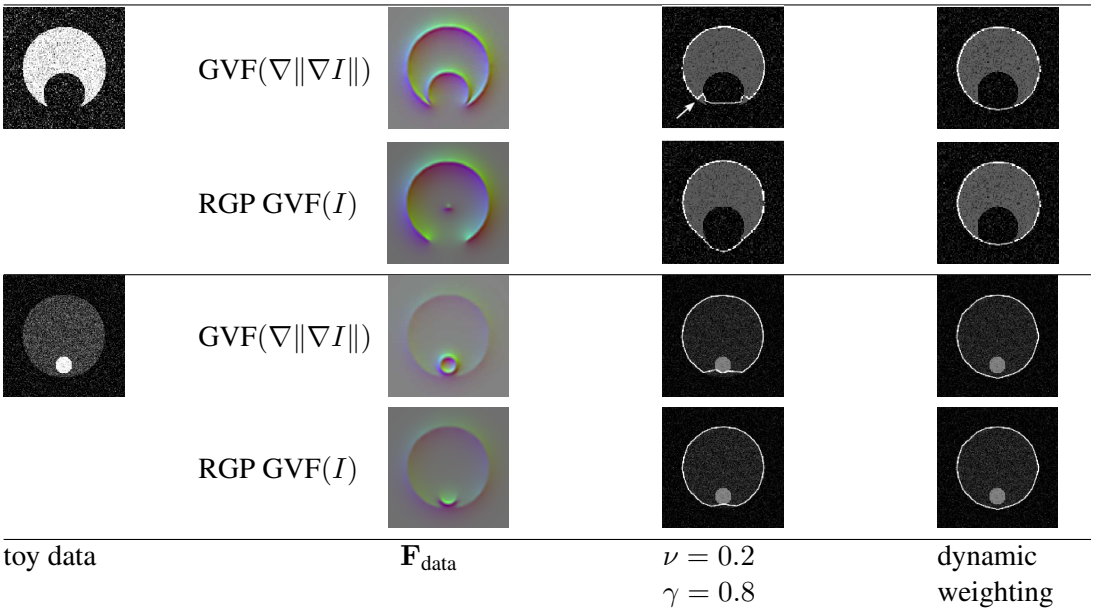


Figure 5.31.: Results for the toy examples with the standard GVF force field and RGP GVF.

the *Drosophila* S2 cell nucleus dataset described in section 3.2.1 and on a dataset of a DAPI stained *Arabidopsis thaliana* root tip recorded with a CLSM (see section 3.2.2). The toy examples are designed to represent the problems that are present in the biological datasets. The first toy example is a sphere with a spherical hole, which should be ignored whenever encountered in our biological data, because it would probably correspond to a nucleolus. The second example is a sphere with a three times brighter sphere inside, corresponding to a spot of dense chromatin, which should not influence the segmentation neither. We have added Gaussian noise to the two examples. With the dynamic weighting of the active surface forces, these toy examples have been well segmented, which, with fixed parameters, could not be done. For comparison, we have segmented the toy data with a standard force field GVF($\nabla\|\nabla I\|$) and the improved RGP GVF force field from section 5.3.2. With the dynamic weighting parameters, the active surface finds the correct contour.

Our synthetic dataset of cell nuclei is quite homogeneous in the chromatin distribution. The assumption that there is deficient boundary information does therefore not hold. However, when segmenting these data with standard GVF data force fields computed directly on the image edges, the surfaces are attracted to boundaries of neighboring cell nuclei. With the dynamic weighting parameters, the influence of these neighboring nucleus boundaries should diminish. The quantitative results are shown in table 5.6 and 5.7. As expected, the results with GVF showed the largest improvement. In the evaluation on the whole volume, the accuracy even grows beyond the accuracy of the MSGVF segmentation. However, the recall is still quite low which means that we still have an undersegmentation in z-direction. The results on the central xy-section show that the resulting accuracy is still below the accuracy that can be achieved with RGP GVF and MSGVF. However, the additional information on the boundary appearance yields a strong improvement. While the other methods had already previously some information on the sought surface appearance, the information gain for the GVF computed on the standard force field is potentially very high with the new, proposed weighting algorithm.

On our real data, the dynamic weighting parameters as well yield improvements. Due to the strong

Table 5.6.: Segmentation results on the synthetic dataset evaluated in the whole volume.

	weights	precision	recall	accuracy
GVF	$\nu = 0.2, \gamma = 0.5$	97.47%	60.01%	58.65%
GVF	dynamic weights $P(\mathrm{B}) = 0.6$	92.95%	81.82%	76.51%
GVF	dynamic weights $P(\mathrm{B}) = 0.7$	93.90%	78.17%	73.87%
RGP GVF	$\nu = 0.2, \gamma = 0.7$	73.63%	96.02%	71.44%
RGP GVF	dynamic weights $P(\mathrm{B}) = 0.6$	73.81%	95.57%	71.49%
RGP GVF	dynamic weights $P(\mathrm{B}) = 0.7$	73.92%	95.55%	71.44%
MSGVF	$\nu = 0.2, \gamma = 0.7$	75.43%	96.03%	73.13%
MSGVF	dynamic weights $P(\mathrm{B}) = 0.6$	75.77%	95.60%	73.21%
MSGVF	dynamic weights $P(\mathrm{B}) = 0.7$	75.70%	95.60%	73.13%

Table 5.7.: Segmentation results on the synthetic dataset evaluated in the central xy-section.

	weights	precision	recall	accuracy
GVF	$\nu = 0.2, \gamma = 0.5$	99.22%	84.73%	84.15%
GVF	dynamic weights $P(\mathrm{B}) = 0.6$	98.80%	90.83%	89.81%
GVF	dynamic weights $P(\mathrm{B}) = 0.7$	98.89%	89.70%	88.78%
RGP GVF	$\nu = 0.2, \gamma = 0.7$	98.14%	94.83%	93.13%
RGP GVF	dynamic weights $P(\mathrm{B}) = 0.6$	98.47%	94.13%	92.75%
RGP GVF	dynamic weights $P(\mathrm{B}) = 0.7$	98.46%	94.0%	92.61%
MSGVF	$\nu = 0.2, \gamma = 0.7$	98.16%	94.89%	93.22%
MSGVF	dynamic weights $P(\mathrm{B}) = 0.6$	98.47%	94.12%	92.75%
MSGVF	dynamic weights $P(\mathrm{B}) = 0.7$	98.48%	94.02%	92.66%

blurring in z-direction it is hard to judge the segmentation of the *Drosophila* S2 cell nuclei in the lower regions, but the overall result seems reasonable for most of the nuclei. We assumed a prior probability of an intact boundary if $P(\mathrm{B}) = 0.8$. See figure 5.32 for some examples. On the left, you can see an overview of a dataset, on the right, the segmentation results for four example cells in orthogonal views.

In figure 5.33, the segmentation results of the proposed dynamic weighting are displayed in direct comparison to the segmentation with fixed parameters. For the fixed parameters, we heuristically choose $\nu = 0.3$ and $\gamma = 0.8$ which shows good results as long as there are not too many bright chromatin spots in the data. The new contours fit the nucleus shape much better whereas the contours with fixed parameters are easily influenced by bright chromatin spots.

For the *Arabidopsis thaliana* nuclei we initialized the surfaces from outside, with spheres with 1.5 times the radius [7]. The segmentation of the *Arabidopsis thaliana* nuclei is more difficult because of the dense tissue. The nuclei often touch one another and cell organelles touch the boundaries. Also, the more central the nuclei lie inside the root, the more difficult is the segmentation with our homogeneity based parameter weighting: as can be seen in figure 5.34, the nucleus boundaries become less and less homogeneous on the inner layers. Furthermore, the nuclei are much more

[7] The nucleolus centers and radii have been manually annotated by Thorsten Schmidt.

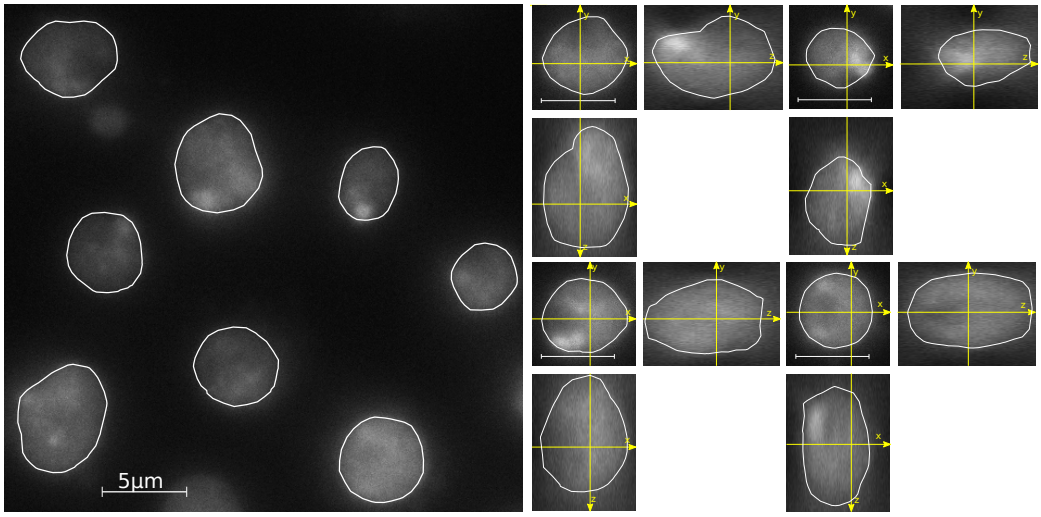


Figure 5.32.: Segmentation results for the *Drosophila* S2 cell nuclei.

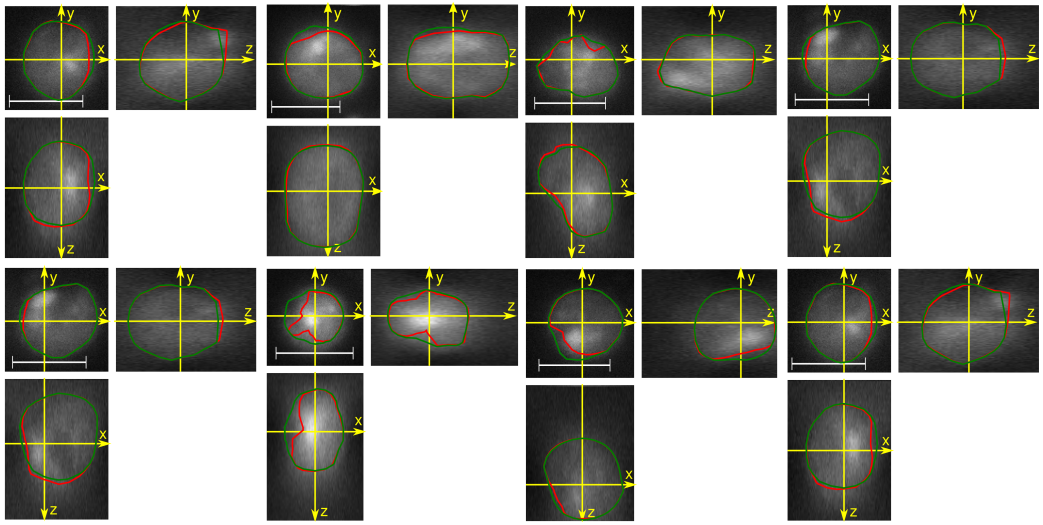


Figure 5.33.: Segmentation results of eight cells in three orthogonal views. The white bar indicates as above the length of 5μm. The contours found with fixed parameters are drawn in red, contours found with the proposed dynamic parameter estimation are drawn in green. The red contours are attracted to the bright chromatin spots if the spots are located near the boundary.

elongated and the nucleoli potentially "split the chromatin in two halves". Despite these facts, most of the nuclei could be properly segmented, see figure 5.34 for an example slice, plotted in three orthogonal views. The colors correspond to the cell layer in the root.

5.4.3. Discussion

We have presented a probability based method to automatically adjust active surface parameters during the surface adaption process. The weighting parameters are regulated dynamically for each

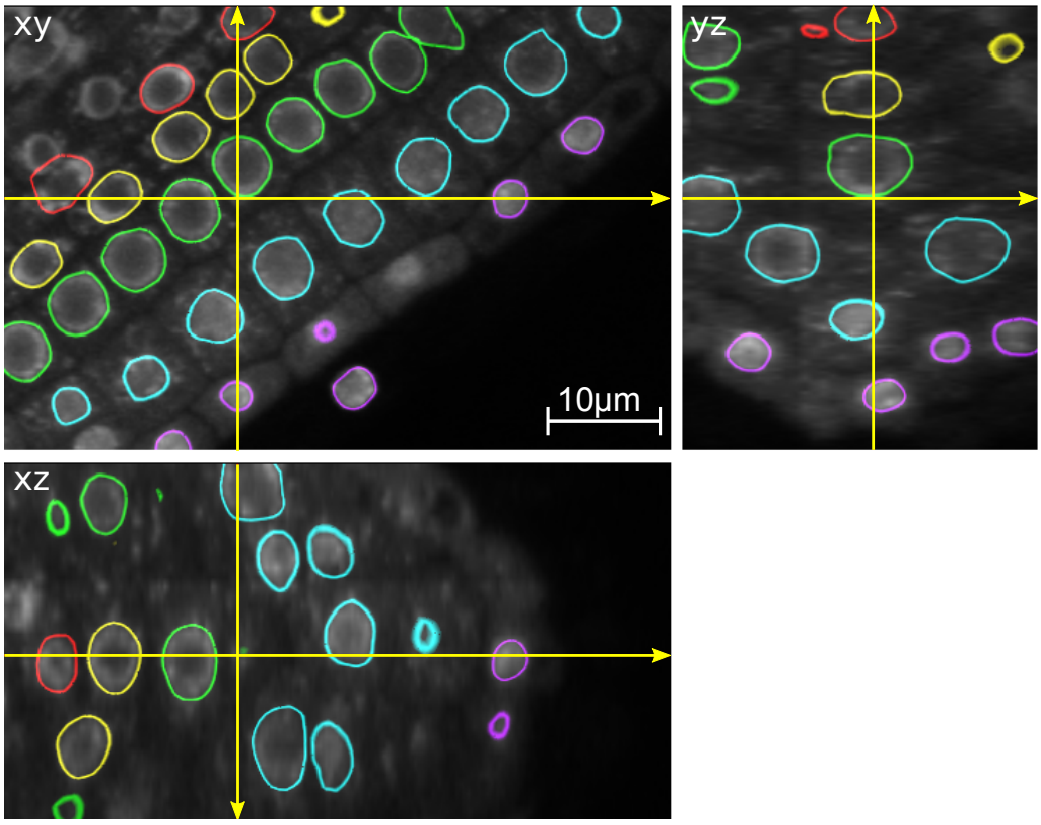


Figure 5.34.: Segmentation results for *Arabidopsis thaliana* root tip nuclei. The colors indicate the different layers of the root from outside to inside: root cap (violet), epidermis(cyan), cortex (green), endodermis (yellow), pericycle (red).

individual vertex based on its appearance - without the input of any prior knowledge. This way, we can avoid tedious parameter adjustment and allow for good segmentation results even in deficient data. For our noisy, biological datasets, the method has some major advantages compared to standard active surfaces with constant weighting parameters all over the surface: first, regions where the boundary information is missing can be closed smoothly, and second, gradients that originate from bright inner regions of the nuclei can be ignored, such that the correct boundary can be found. The presented method however relies on the assumption that the contour is homogeneous in all positions where it is not deficient. Generally, an extension of the method to bi-, or multi-modal edge appearance distributions does not seem to be impossible. However, in this case it is not obvious how the desired edge appearances should be selected without user interaction.

5.5. Conclusion

Active surface models are a means to accurately segment cell nuclei and single cells from volumetric datasets. We have presented methods for the generation of data force fields that incorporate to a certain extent knowledge about the underlying objects. However, the included information is of quite general nature: we use the fact that the objects have closed surfaces with spherical topology and are star-shaped. Furthermore, we use information from the imaging techique which is that,

caused by the blurring, not only strong gradients but also weak gradients with a high local consistency contain valuable boundary information. The proposed data force fields lead to accurate segmentations that work robustly in our blurred and noisy datasets and outperform state-of-the-art methods like standard GVF active surfaces, level sets, and the region competition method from ITK-SNAP.

In order to learn appropriate data force fields for the segmentation of new microscopic data, we have presented a semi-supervised method that allows to generate specialized data force fields for cell segmentation with only few user interactions. The filters learned for the generation of these force fields can be iteratively refined until all object boundary appearances in a certain dataset are covered.

Although we have evaluated these data force field with parametric active surfaces, their application is per se not limited to parametric active surfaces but they can be equally useful in a geodesic level set framework.

Thanks to our proposed method for the active surface weighting parameter adjustment, these factors can be dynamically and automatically determined such that only one high level parameter needs to be set instead of two low level parameters as in the traditional active surface setting. The method automatically learns the boundary appearance of the object boundary if it can be described by a single Gaussian normal distribution. This additional information greatly improves the segmentation result if the image of the object boundary is deficient whereas the results remain comparable to fixed parameter settings if the boundary information is not lossy.

6. Hierarchical Models for Image Segmentation

In this chapter, we present a multi-label segmentation algorithm based on a hierarchical Markov Random Field that we have proposed in [Keuper et al., 2011]. In the proposed method, the segmentation and classification task are handled simultaneously as an image-based multi-label problem. The hierarchical Markov Random Field (HMRF) is built upon a hierarchy of image regions. Thus topological knowledge is introduced into the segmentation and classification algorithm. This topological knowledge is important for many segmentation applications since objects often can not be recognized by their local features. The general idea of using small image regions also leads to a speed-up compared to a pixel-based segmentation, while texture and intensity information inside the regions is preserved. Furthermore, this bears the advantage that a merging of segments by semantic cues becomes much easier. The regions are generated with the method of [Arbeláez et al., 2011, 2009] that we have described in section 2.4.1. The local evidences inside the HMRF are learned using Support Vector Machines (SVM) [Vapnik, 1998]. Thus, we only need a relatively small set of training samples. As in [Mičušík and Pajdla, 2007], we are formulating the segmentation as a MAX-SUM problem for which there exist efficient solvers based on linear programming [Werner, 2007] (see section 2.2.6). An exact MAP solution cannot be found as the problem is NP-hard.
We apply our method to the automatic segmentation of mast cells and the segmentation and classification of their cell organelles from 2D electron microscopic recordings. Our data set has been presented in section 3.4. Cell region segmentation from EM recordings is an ongoing research topic (e.g. [Chang et al., 2009]). The different labels we want to assign are *background*, *cytoplasm*, *nucleus*, *mitochondria*, and *vesicles*. The data set has been manually annotated by experts with these labels. The expert labeling for the example dataset is given in figure 6.1(d). As can be seen, a simple thresholding is not sufficient for the segmentation, because e.g. the vesicles (blue) can have the same gray value as the background (black), while parts of the nucleus (red) have the same intensity as the cytoplasm (yellow). Mitochondria (cyan) can also easily be confused with the nucleus or vesicles. Given these challenges, it is also evident that we need topological knowledge about the possible label constellations (e.g. *The nucleus is enclosed by cytoplasm*).

6.0.1. Related Work

MRFs and Conditional Random Fields (CRF) have been widely used for image segmentation in natural images [Mičušík and Pajdla, 2007; D'Elia et al., 2003; Plath et al., 2009]. In [Mičušík and Pajdla, 2007], the MAX-SUM solver is used to solve classical multi-label MRFs on image regions while seeds are automatically generated using texture and color cues. In [Plath et al., 2009] and [Reynolds and Murphy, 2007], tree-structured multi-class CRFs are presented that, like our method, couple local and global information. [Plath et al., 2009] also use SVMs to learn local ev-

idences. Unlike our method the local segments in [Plath et al., 2009] and [Reynolds and Murphy, 2007] are only dependent on the higher hierarchies whereas no mutual dependency is modeled. This is also the case for [D'Elia et al., 2003], where a tree-structured MRF is presented for binary segmentation tasks. In [Shotton et al., 2008], a semantic image categorization and segmentation algorithm is presented that is based on ensembles of decision trees. The idea of semantic texton forests might be interesting for our task as well, as e.g. the *vesicles* rather form a semantic class.

6.0.2. Method Outline

In our proposed method, we tackle the problem of segmenting multiple semantic classes by learning the different local and global appearances of the structures with the SVM. The information is then combined in a hierarchical MRF. Our segmentation consists of the following steps:

1. With an unsupervised edge based segmentation method, we hierarchically subdivide the image into regions.
2. For all regions, we compute Gabor texture features using a Gabor Filter bank.
3. Two-class SVM classifications are performed for all pairs of labels.
4. From the SVM decision values, we compute multi-class probabilities for all regions.
5. A HMRF is built depending on the region hierarchy and solved using the MAX-SUM-solver.

6.1. Region Hierarchy Generation

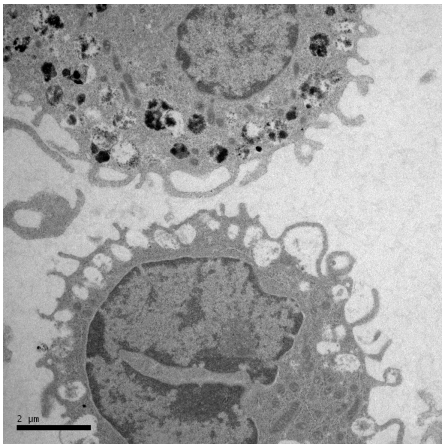
The regions are generated using *gPb-OWT-UCM*, the method proposed in Arbeláez et al. [2011, 2009] (compare section 2.4.1 for a detailed description). The result of this method are closed, non-selfintersecting regions that are hierarchically ordered according to the underlying gradients in the image. The finest regions are in the 0-level of the hierarchy, the coarsest in the top-most level L. For our data, small organelles like the mitochondria are only segmented at the lowest level of hierarchy. We therefore use all of the segmented regions as superpixels. In some cases, the local information of the superpixels and their direct neighbors is not sufficient to even manually predict their labels. Thus, larger regions at a higher level in the hierarchy have to be used in order to introduce global knowledge into the superpixel classification. This higher level is chosen with a fixed threshold at the center of the hierarchy, i.e. at $L/2$. In the remainder of this section, the regions segmented at finest level will be referred to as *superpixels* or *low level regions*, those segmented at the coarser level will be called *high level regions*. The result of the hierarchical region generation on our data is shown in figure 6.1(b).

6.2. Region Learning

6.2.1. Feature Computation

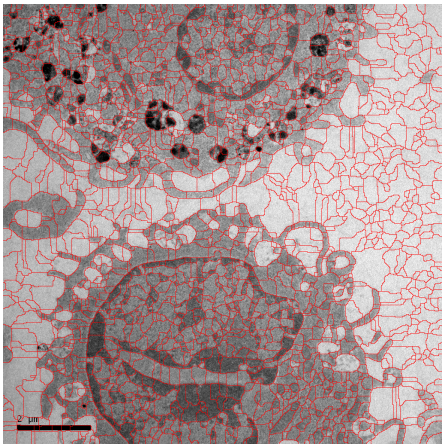
On all high level and low level regions, we compute a set of features based on a Gabor filter bank with complex filters

$$g_{\lambda,\theta}(x,y) = \exp\left(-\frac{x'^2+y'^2}{2\sigma^2}\right)\cdot\exp\left(\imath 2\pi\frac{x'}{\lambda}\right) \tag{6.1}$$

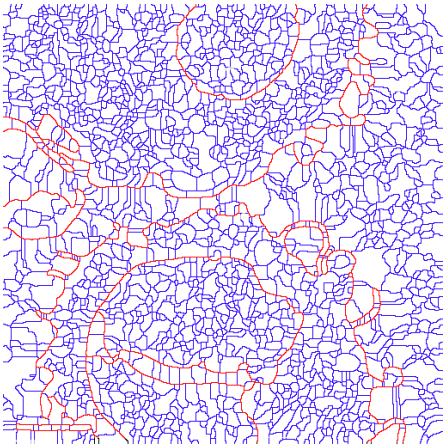


(a) Raw Data

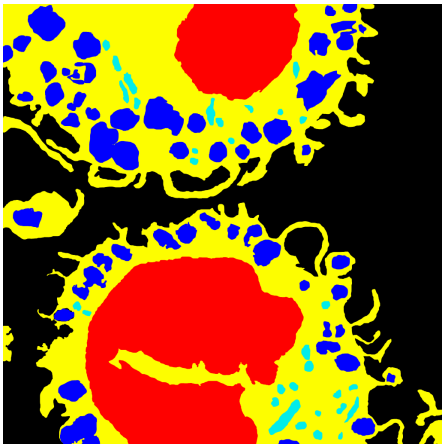
(b) *UCM*



(c) Overlay with all superpixels



(d) The chosen hierarchy levels. The superpixel level is colored in blue, high level boundaries are marked in red.



(e) Ground truth annotation

Figure 6.1.: An example dataset and the region hierarchy built upon it.

Table 6.1.: Pixel class distribution in the regions.

high level region label	ratio of pixels from class				
	1	2	3	4	5
$1'$	97.6%	2.3%	0%	0%	0%
$2'$	8.3%	74.8%	0.8%	1%	15%
$3'$	0.3%	2.5%	97.2%	0%	0%

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\cos\theta + y\sin\theta. \end{aligned}$$

Altogether, we use 20 different filters with five different frequencies λ (from 23.2nm to 69.6nm) and an angular resolution of $45°$ i.e. four different angles θ. The mean energy of the filter responses inside the regions is finally used as feature vectors. Additionally, we append the mean intensity and the mean gray value variance inside the regions to the feature vectors. These feature vectors allow us to learn region classifiers that would permit to directly predict the class membership of each region independently. However, the spatial dependencies between the superpixels contain important information, which is why we base our segmentation on a MRF.

6.2.2. SVM Classification with Probabilistic Output

In order to learn the local evidences, we use a Support Vector Machine (SVM)[1]. In fact, we are training two-class SVMs with RBF kernels for all K^2 pairs of labels.
The training data is generated from the manual annotations in the training set. At the finest scale, each training superpixel is assigned the label (from $1-5$) of the majority of its pixels. This is valid because we assume that each superpixel belongs to exactly one class. At the coarser scale, this is more difficult. Here, we also identify the majority vote of all pixels inside a region, but the regions are mostly composed of pixels from several different classes. Most notably, the regions mainly consisting of cytoplasm also contain parts of the nucleus, mitochondria and vesicles. On the other hand, none of the training regions mainly consisted of mitochondria or vesicles because these organelles are small and can only be discriminated at a very fine level of hierarchy. In order to handle these facts, we learn the distribution of classes inside the high level regions and assign new labels $1'$, $2'$ and $3'$. The pixel label distribution inside these high level region labels is shown in table 6.1. For all pairs of these new labels, we also train two-class SVMs. The decision values of the SVMs give us information about each two-class problem. For the recombination of these two-class decision values into multi-class probabilities, the second method described in [Wu et al., 2004] is used yielding a probability $p_{v_i}^{\ell_i}$ for every region v_i and label ℓ_i [2]. The SVM solution for our segmentation task would be to assign to every superpixel v_i the most probable label ℓ_i according to this probability. We are instead using these probabilities as data term in our HMRF (see section 6.3.1) and compare our results to the pure SVM classification.

[1] We used the C++ SVM implementation from `http://lmb/resources/opensource/libsvmtl.en.html`.
[2] C++ code for the computation of these probabilities has been provided by Olaf Ronneberger.

6.3. Hierarchical Graph Construction

In our implementation, the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed such that there exists a node $v_i \in \mathcal{V}$ for every image region at the two different region levels (*high level regions* and *superpixels*). The edges $\mathcal{E}$ represent the mutual dependencies between the nodes and therefore connect each two neighboring nodes at one level. By the nature of our region generation, each superpixel, belongs to exactly one high level region. In our graph, they are therefore connected to exactly this node in the higher level. Figure 6.2 shows a scheme of the hierarchical graphical structure used. As all five classes are

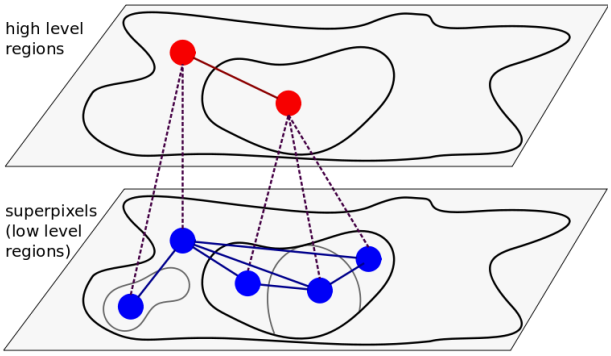


Figure 6.2.: Hierarchical graph structure.

present in all the images, the number of labels K is constant at the superpixel level and set to five. Every node can be assigned one of the five different labels.
For this graph, we want to find the labeling $\boldsymbol{\ell}$ maximizing the energy

$$U(\boldsymbol{\ell}|\mathbf{w}) = \sum_{v_i \in \mathcal{V}} w_i(\ell_i) + \sum_{v_i, v_j \in \mathcal{V}} w_{ij}(\ell_i, \ell_j), \tag{6.2}$$

where the unary and binary potentials in $\mathbf{w}$ are probabilistic terms depending on the region features.

6.3.1. Probabilistic Data Term

The data terms are the unary graph potentials in our hierarchical MRF. These data terms are built upon the probabilistic output of the SVM classification, i.e., we use the SVM to learn the local evidences (see section 6.2.2).
When performing the classification as stated in section 6.2.2, we get a probability $p_{v_i}^{\ell_i}$ for every node v_i and label ℓ_i. The data term $w_i(\ell_i)$ of the graph $\mathcal{G}$ is encoding the quality of assigning the label ℓ_i to node v_i. We thus set

$$w_i(\ell_i) = p_{v_i}^{\ell_i}. \tag{6.3}$$

6.3.2. Edge Term

In literature (e.g. [Mičušík and Pajdla, 2007]), the edge term is mainly used in order to introduce smoothness into the segmentation result. In our case, this is different, because specially the small cell organelles are spread throughout the cell. A strong smoothness constraint would hamper a good classification of these organelles. Furthermore, the cell surfaces are not even and thus the cytoplasm segmentation would neither profit from a smoothness term. On the other hand, we can

see that some classes are always grouped together while they never get into contact with others. Cell organelles for example hardly ever touch the background, while the nucleus has most common surfaces with the cytoplasm. We define the average adjacency ratio matrix AR as the symmetric matrix containing the average number of boundary pixels between class ℓ_i and class ℓ_j normalized by the total number of boundary pixels for all classes. The AR on the superpixel level is given in equation (6.4) for one example training dataset

$$\mathrm{AR} = \begin{pmatrix} 0.241 & 0.118 & 0 & 0 & 0.008 \\ 0.118 & 0.278 & 0.010 & 0.006 & 0.037 \\ 0 & 0.010 & 0.087 & 0 & 0 \\ 0 & 0.006 & 0 & 0.001 & 0 \\ 0.008 & 0.037 & 0 & 0 & 0.049 \end{pmatrix}. \tag{6.4}$$

We base our edge term $w_{ij}(\ell_i, \ell_j)$ on this adjacency ratio in our training data and set

$$w_{ij}(\ell_i, \ell_j) = \frac{2 \cdot \mathrm{AR}(\ell_i, \ell_j)}{\sum_{\ell_k} \mathrm{AR}(\ell_i, \ell_k) + \sum_{\ell_k} \mathrm{AR}(\ell_k, \ell_j)} \tag{6.5}$$

for all v_i and v_j on the superpixel level. Between the high level regions, the AR and $w_{ij}(\ell_i, \ell_j)$ are computed accordingly.
Between the two levels of hierarchy, we also have to set the edge qualities according to figure 6.2. Here, we base our choice on the learned distributions in the classes $1'$, $2'$, and $3'$ (see table 6.1). The edge between the high level node $\bar{v_i}$ and the low level node v_j is set to the corresponding class distribution entry $d_{\bar{\ell_i},\ell_j}$,

$$w_{\bar{\ell_i},\ell_j}(\bar{\ell_i}, \ell_j) = d_{\bar{\ell_i},\ell_j}. \tag{6.6}$$

6.4. Experiments

Our data consists of 27 Transmission Electron Microscopic recordings from Mast Cells, more specifically BMMC (bone marrow-derived mast cells) (compare section 3.4). The data has been manually annotated by experts, who distinguished between five classes: *background*, *cytoplasm*, *nucleus*, *mitochondria* and *other vesicles*. This last class is the least homogeneous as it contains many different cell organelles as lysosomes, the Golgi apparatus, the endoplasmic reticulum, etc. and was therefore hard to classify for the SVM. On average, the UCM yielded 1642 superpixels per image. The average number of high level regions was 43.
We trained RBF-Kernel SVMs for each two-class one versus one classification problem. The cost values were adapted with a gridsearch on one of the training images. The code of the MAX-SUM solver was downloaded from [3].
We have divided our data into three independent sets of images (each containing 9 images). The algorithm was evaluated with a three-fold cross-validation. The results were compared with those we could achieve by directly using the SVM classification (as described in section 6.2.2), as well as those we could achieve by using a simple MRF without hierarchy. For this, we built a MRF only containing the lower level of our region hierarchy. All further parameters of the MRF were chosen identically to our HMRF.

[3] `http://cmp.felk.cvut.cz/cmp/software/maxsum/`.

Table 6.2.: Segmentation accuracy.

	accuracy in class					overall acc.
	background	*cytoplasm*	*nucleus*	*mitochondria*	*other vesicles*	
HMRF	87.3%	61.7%	31.8%	8.1%	10.9%	65.4%
MRF	83.5%	58.6%	21.9%	10.8%	14.9%	61.5%
SVM	85.2%	57.3%	22.2%	9.5%	17.6%	60.5%

Table 6.3.: Precision and Recall of the HMRF method.

	background	*cytoplasm*	*nucleus*	*mitochondria*	*other vesicles*
precision	93.3%	67.4%	55.5%	22.7%	69.3%
recall	93.3%	89.1%	40.4%	18.2%	11.9%

6.5. Results

As a performance measure, we use the segmentation accuracy of all classes (including background). The accuracy is computed by

$$\text{accuracy} = \frac{\text{tp}}{\text{tp} + \text{fp} + \text{fn}}, \tag{6.7}$$

where tp, fp, and fn are the true positives, false positives and false negatives respectively. The segmentation accuracies per class are given in table 6.2. Additionally, we compute the overall segmentation accuracy for all classes. This might be interesting, as the classes are very different in size. Thus, at visual inspection, a segmentation result with low accuracy in the cytoplasm class appears much less reasonable than a low accuracy in the mitochondria class. In the overall segmentation accuracy, we observe an improvement from 61.5% with a simple MRF to 65.42% with our HMRF segmentation. The strongest improvement can be observed in the segmentation of *background*, *cytoplsm* and *nucleus* regions. For these larger regions, the higher level in the hierarchy contains important information.

Actually, the accuracy is reasonably high for the classes *background* and *cytoplasm*. This is important because those are the largest classes and from the biological point of view, the delineation between cell and background is crucial. In the smaller classes *mitochondria* and *other vesicles*, the accuracy is quite low. In order to quantify what this means, we are looking at the precision and recall

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \qquad \text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

of the superpixel classification (see table 6.3). Here, we can see that for the *vesicles* the precision is quite high while the recall is low. This means that whenever our algorithm classifies a region as *vesicle*, it is probably right, whereas many true vesicles are not recognized. On the other hand, the precision for the *cytoplasm* class is lower than its recall. This indicates that regions belonging to different classes (in our case to cell organelles), have been wrongly classified as *cytoplasm*.

When looking at the class-wise accuracy, the SVM segmentation result seems to be already pretty good. However, when looking at the actual segmentation masks (see figure 6.3 and 6.4), one can see that the HMRF approach leads to less cluttered regions than both the other methods (SVM and MRF), which is favorable for the organelle segmentation task. Especially in the SVM classification, cell organelles as vesicles or mitochondria are detected outside the cytoplasm or on the microvilli (the tail-like extensions of the cells) which, from a biological point of view, makes no sense. With

our hierarchical method this was not the case. Thanks to the global knowledge introduced by the hierarchy, specially the cell nuclei have been much better segmented, which, when looking at a small region only, is indeed really difficult.

6.6. Conclusion

The presented multi-label segmentation algorithm is based on a hierarchical graphical model. We could show that our method outperforms the local region classification by SVMs as well as classical MRFs regarding the overall classification accuracy. Compared to classical MRFs, the region consistency reached with our method is much higher thanks to the topological knowledge introduced by the region hierarchy. In the presented method, we only use a fixed two-level hierarchy. While high level information on the nucleus can be included in the hierarchical MRF in the current setting, higher-level information on small structures like mitochondria can so far not be used. In future work, we are therefore hoping to be able to extend this method to deeper hierarchies. A further improvement of the segmentation quality could probably be achieved by choosing more discriminative features to describe the superpixels and high-level regions, such that the initial SVM classification is improved.

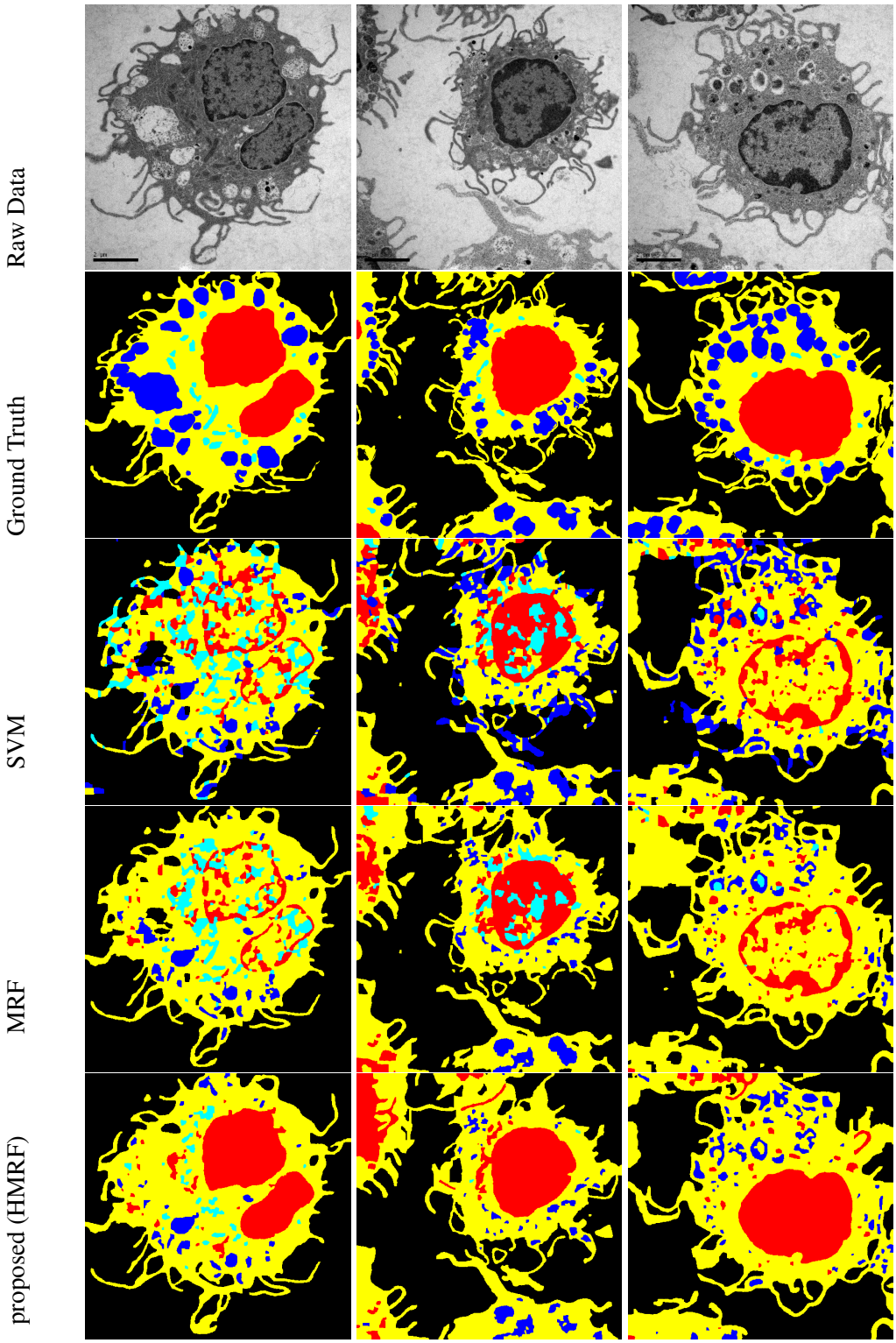


Figure 6.3.: Results of the three different methods on our data. Black corresponds to the label *background*, yellow to *cytoplasm*, red to *nucleus*, cyan to *mitochondria* and blue to *vesicles*. The results of our method, displayed in the last row, are the most consistent.

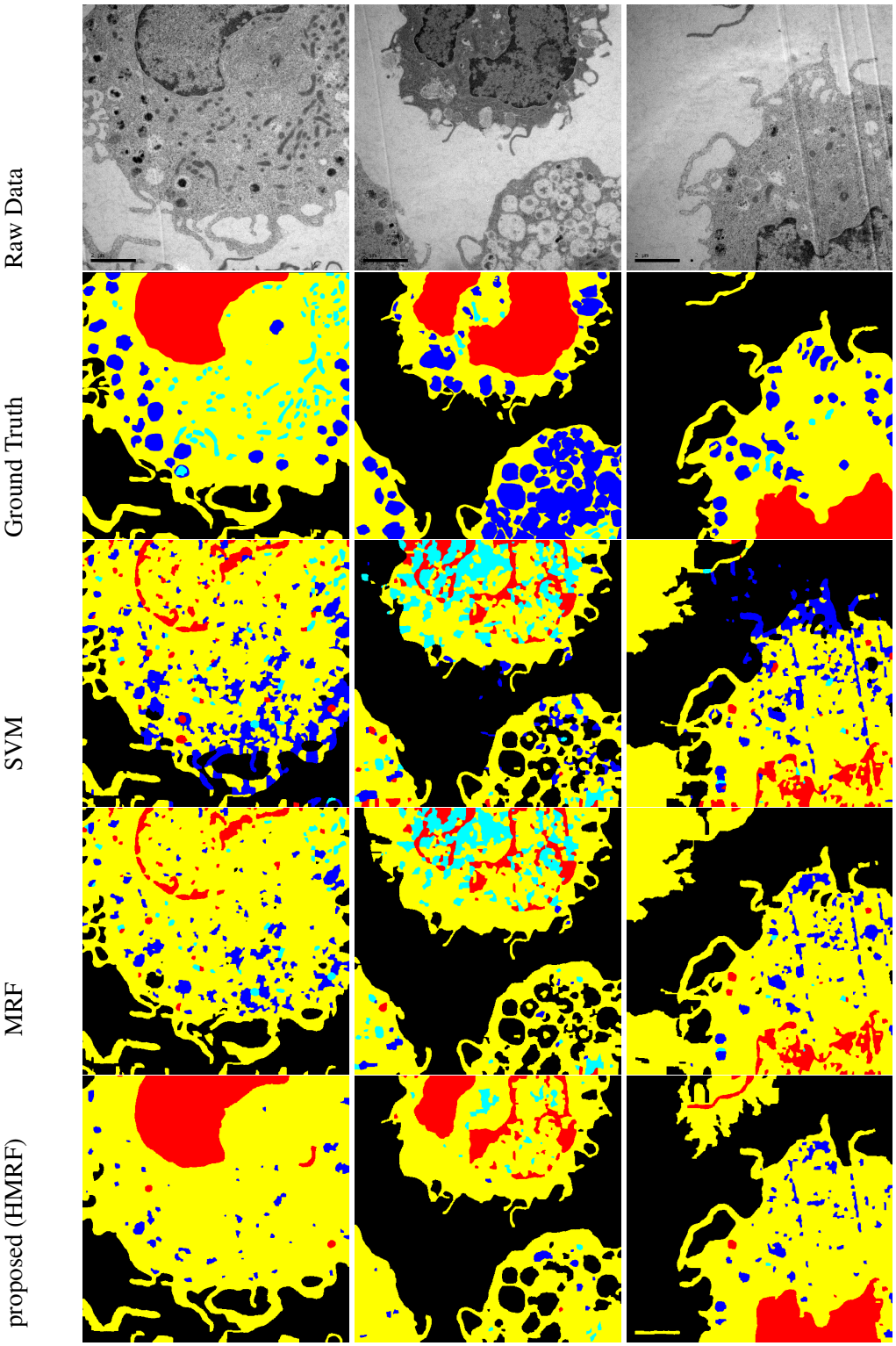


Figure 6.4.: Results of the three different methods on our data. Black corresponds to the label *background*, yellow to *cytoplasm*, red to *nucleus*, cyan to *mitochondria* and blue to *vesicles*. The results of our method, displayed in the last row, are the most consistent.

7. Conclusion

The segmentation of the recorded specimen from microscopic recordings is a difficult task that bears many challenges due to recording noise, blurring, bleaching and signal attenuation effects and inhomogeneities in the fluorescence staining. Due to these issues, segmentation methods that are to be considered state-of-the-art in natural image segmentation can mostly not be applied. Modified or new methods are needed that can cope with the challenges imposed by the imaging techniques.

Blurring effects present in the data due to the Point Spread Function of the recording system can be attenuated by deconvolution techniques. In chapter 4, we have presented two new methods for the deconvolution of conventional fluorescence microscopic data. The first presented method is a regularization of the deblurring kernel for the blind MLEM deconvolution scheme that does not need any recording system specific prior knowledge and nevertheless stabilizes the deconvolution scheme, such that not only the energy but also the SNR converges. The second presented approach, the frequency domain TV regularization of the deconvolution kernel, uses image-frequency-based prior knowledge for the regularization of the deconvolution kernel. With this prior, we achieve a qualitative enhancement of the deconvolution results as well as an improvement compared to state-of-the-art deconvolution methods in terms of RMSE. These results confirm our assumption that the direct manipulation of the frequency domain properties of the deconvolution kernel is beneficial for the deconvolution of widefield microscopic data.
With this maxim, several interesting aspects remain to be investigated. For example, how do spherical aberrations influence the phase of the Object Transfer Function? And how can the phase be regularized without the detour via the spatial domain? In [Holmes, 1992] for example, the rotational symmetry of the PSF around the optical axis is established by averaging over the rotation of the deconvolution kernel in the spatial domain whereas a description of the rotational symmetry by the phase information in the frequency domain could have favorable effects.

Due to the spatial frequencies lost during the recording process, information along the optical axis is not reliable even after deconvolution of the datasets. Therefore, our segmentation algorithms work on undeconvolved data. The manual evaluation of volumetric fluorescence microscopic data as well as the generation of ground truth annotations is very tedious. The methods we have presented for the segmentation of volumetric data are therefore either fully automatic or they rely on only very sparse labeling of selected data points. In contrast to other methods for the segmentation in volumetric, microscopic recordings [Sommer et al., 2011; Fehr et al., 2005; Ronneberger et al., 2005; Kaster et al., 2011], that compute segmentations by point-wise classification the recorded volume, we aim for segmenting the whole object by estimating its boundary position. A combination of the two approaches would be conceivable, where object regions are first identified by a voxel-wise classification. The classification result could serve as additional, region-based data term in our methods, that, so far, only rely on boundary information.
In order to handle the challenges given in volumetric fluorescence microscopic recordings and to

produce robust segmentations, we have presented several methods based on three-dimensional active surface models. The Channel Differential Structure allows for the combination of two recorded channels to build a data term appropriate for the nucleolus segmentation. For star-shaped objects, the object detection result can be used to produce improved data force fields for the segmentation. Artifacts in the segmentation that are due to the strong blurring along the optical axis can be alleviated by the proposed Mean Shift Gradient Vector Flow method. In order to learn new object appearances from minimal user input, we have presented a method to generate specific edge filters from pre-clustered edge appearance candidates.
In standard active surfaces, the relation between regularization and data confidence has to be fixed in advance. In microscopic recordings, however, we commonly observe a strongly varying image quality within one volumetric dataset. Object boundaries might be imaged with a strong contrast in one part of the image, and might be strongly blurred in another image region. Therefore, we have presented a method for the automatic and dynamic adjustment of the weighting of regularization terms during the segmentation process. With this method, we can cope with data deficiencies and enforce smooth object boundaries where the recorded information is lossy. The surface is driven towards the correct boundary wherever the recorded boundary information is sound.
With these methods, we can accurately and robustly segment protoplast cells and cell nuclei from different cell types. The automatic segmentations of *Drosophila* S2 cell nuclei and nucleoli can be used to investigate protein localizations. Statistical evaluations of these data can now be computed in a reasonable amount of time, whereas a manual analysis of the data would be very cumbersome and time-consuming.

For the simultaneous segmentation and classification of electron microscopic recordings into regions belonging to different object classes, we have presented a method based on hierarchical Markov Random Fields. The proposed hierarchical Markov Random Fields allow for the learning of topological knowledge from ground truth annotations. With this approach, segmentation results could be greatly improved compared to a conventional Markov Random Field approach. The topological knowledge introduces a higher consistency especially for the larger objects as *cytoplasm* and *nucleus*.
In future work, we will try to investigate the usage of deeper hierarchical structures such that not only the classes of large objects benefit from the learned topological knowledge. Furthermore, the inclusion of shape information is an important issue that will be a topic of future research.

A. Deriving MLEM Deconvolution

In order to minimize the energy functional

$$J_{\text{MLEM}}(s,h) = \int_{\mathbf{x}\in\Omega} (s*h)(\mathbf{x}) - o(\mathbf{x}) \cdot \log(s*h)(\mathbf{x}) d\mathbf{x} \tag{A.1}$$

with respect to the specimen function s, we have to find critical points of the partial derivative $\frac{\partial}{\partial\epsilon} J_{\text{MLEM}}(s+\epsilon\bar{s},h)|_{\epsilon\to 0} = 0$.

$$\begin{aligned}
\frac{\partial}{\partial\epsilon} J_{\text{MLEM}}(s+\epsilon\bar{s},h) &= \int_{\mathbf{x}\in\Omega} \frac{\partial}{\partial\epsilon} \left(((s+\epsilon\bar{s})*h)(\mathbf{x}) - o(\mathbf{x}) \cdot \log((s+\epsilon\bar{s})*h)(\mathbf{x})\right) d\mathbf{x} \\
&= \int_{\mathbf{x}\in\Omega} (\bar{s}*h)(\mathbf{x}) - o(\mathbf{x}) \cdot \frac{1}{\left((s+\epsilon\bar{s})*h\right)(\mathbf{x})} (\bar{s}*h)(\mathbf{x}) d\mathbf{x} \\
&\overset{\epsilon\to 0}{=} \int_{\mathbf{x}\in\Omega} (\bar{s}*h)(\mathbf{x}) - \frac{o(\mathbf{x})}{\left(s*h\right)(\mathbf{x})} (\bar{s}*h)(\mathbf{x}) d\mathbf{x} \\
&= \int_{\mathbf{x}\in\Omega} \bar{s}(\mathbf{x})(\mathbf{1}*h^m)(\mathbf{x}) - \bar{s}(\mathbf{x}) \left(\frac{o}{s*h} * h^m\right)(\mathbf{x}) d\mathbf{x},
\end{aligned} \tag{A.2}$$

because for any function $a:\Omega\to\mathbb{R}$, $\int_{\mathbf{x}\in\Omega} a(\mathbf{x}) \cdot (h*\bar{s})(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{x}\in\Omega} (a*h^*)(\mathbf{x}) \cdot \bar{s}(\mathbf{x}) d\mathbf{x}$, where h^* is the adjoint of h and equals the mirrored function h^m. $\mathbf{1}$ denotes the constant one-function. The resulting Euler-Lagrange equation is

$$\int_{\mathbf{y}\in\Omega} h^m(\mathbf{y}) d\mathbf{y} - \left(\frac{o}{s*h} * h^m\right)(\mathbf{x}) = 0. \tag{A.3}$$

The partial derivative $\frac{\partial}{\partial\epsilon} J_{\text{MLEM}}(s,h+\epsilon\bar{h})|_{\epsilon\to 0}$ for the minimization of $J_{\text{MLEM}}(s,h)$ with respect to h can be found accordingly.

$$\begin{aligned}
\frac{\partial}{\partial\epsilon} J_{\text{MLEM}}(s,h+\epsilon\bar{h}) &= \int_{\mathbf{x}\in\Omega} \frac{\partial}{\partial\epsilon} \left((s*(h+\epsilon\bar{h}))(\mathbf{x}) - o(\mathbf{x}) \cdot \log(s*(h+\epsilon\bar{h}))(\mathbf{x})\right) d\mathbf{x} \\
&= \int_{\mathbf{x}\in\Omega} (s*\bar{h})(\mathbf{x}) - o(\mathbf{x}) \cdot \frac{1}{\left(s*(h+\epsilon\bar{h})\right)(\mathbf{x})} (s*\bar{h})(\mathbf{x}) d\mathbf{x} \\
&\overset{\epsilon\to 0}{=} \int_{\mathbf{x}\in\Omega} (s*\bar{h})(\mathbf{x}) - \frac{o(\mathbf{x})}{\left(s*h\right)(\mathbf{x})} (s*\bar{h})(\mathbf{x}) d\mathbf{x} \\
&= \int_{\mathbf{x}\in\Omega} \bar{h}(\mathbf{x})(\mathbf{1}*s^m)(\mathbf{x}) - \bar{h}(\mathbf{x}) \left(\frac{o}{s*h} * s^m\right)(\mathbf{x}) d\mathbf{x},
\end{aligned} \tag{A.4}$$

where s^m is the mirrored specimen function s. The resulting Euler-Lagrange equation is

$$\int_{\mathbf{y}\in\Omega} s^m(\mathbf{y}) d\mathbf{y} - \left(\frac{o}{s*h} * s^m\right)(\mathbf{x}) = 0. \tag{A.5}$$

B. Discrete 3D Tikhonov-Miller and TV Regularization

B.1. Tikhonov-Miller Regularizer

In order to minimize the Tikhoniv-Miller regularization term

$$J_{\text{TM}} = \lambda_{\text{TM}} \int_{\mathbf{x}\in\Omega} |\nabla s(\mathbf{x})|^2 d\mathbf{x} \tag{B.1}$$

we need to find a discrete implementation for the derivative

$$\frac{\partial}{\partial s} J_{\text{TM}} = -2\lambda_{\text{TM}} \Delta s(\mathbf{x}), \tag{B.2}$$

where $\Delta s = \frac{\partial^2 s}{\partial x_1^2} + \frac{\partial^2 s}{\partial x_2^2} + \frac{\partial^2 s}{\partial x_3^2}$. We implemented B.2 by using forward and backward differences as

$$\Delta s(u,v,w) = \Delta_-^{x_1}\Delta_+^{x_1} s(u,v,w) + \Delta_-^{x_2}\Delta_+^{x_2} s(u,v,w) + \Delta_-^{x_3}\Delta_+^{x_3} s(u,v,w), \tag{B.3}$$

where the forward and backward differences are defined as

$$\begin{aligned}
\Delta_+^{x_1} s(u,v,w) &= \tfrac{1}{h_{x_1}} \big(s(u+1,v,w) - s(u,v,w)\big) \\
\Delta_-^{x_1} s(u,v,w) &= \tfrac{1}{h_{x_1}} \big(s(u,v,w) - s(u-1,v,w)\big) \\
\Delta_+^{x_2} s(u,v,w) &= \tfrac{1}{h_{x_2}} \big(s(u,v+1,w) - s(u,v,w)\big) \\
\Delta_-^{x_2} s(u,v,w) &= \tfrac{1}{h_{x_2}} \big(s(u,v,w) - s(u,v-1,w)\big) \\
\Delta_+^{x_3} s(u,v,w) &= \tfrac{1}{h_{x_3}} \big(s(u,v,w+1) - s(u,v,w)\big) \\
\Delta_-^{x_3} s(u,v,w) &= \tfrac{1}{h_{x_3}} \big(s(u,v,w) - s(u,v,w-1)\big) .
\end{aligned} \tag{B.4}$$

We assume mirrored boundary pixel values which means that the derivatives evaluate to zero at the image boundaries.

B.2. Total Variation Regularizer

For the minimization of the TV regularization term

$$J_{\text{TV}} = \lambda_{\text{TV}} \int_{\mathbf{x}\in\Omega} |\nabla s(\mathbf{x})| d\mathbf{x} \tag{B.5}$$

we need to find a discrete implementation for the derivative

$$\frac{\partial}{\partial s} J_{\text{TV}} = -\lambda_{\text{TV}} \text{div}\left(\frac{\nabla s(\mathbf{x})}{|\nabla s(\mathbf{x})|}\right), \tag{B.6}$$

where $\mathrm{div}(s) = \frac{\partial s}{\partial x_1} + \frac{\partial s}{\partial x_2} + \frac{\partial s}{\partial x_3}$. For the computation of the divergence term $\mathrm{div}\left(\frac{\nabla s(\mathbf{x})}{|\nabla s(\mathbf{x})|}\right)$, we use the numerically stable scheme presented in [Rudin et al., 1992]:

$$\begin{aligned}
\mathrm{div}\left(\frac{\nabla s(\mathbf{x})}{|\nabla s(\mathbf{x})|}\right) &= \frac{1}{h_{x_1}}\Delta_-^{x_1}\frac{\Delta_+^{x_1}s(\mathbf{x})}{\sqrt{\epsilon+\left(\Delta_+^{x_1}s(\mathbf{x})\right)^2+m\left(\Delta_+^{x_2}s(\mathbf{x}),\Delta_-^{x_2}s(\mathbf{x})\right)^2+m\left(\Delta_+^{x_3}s(\mathbf{x}),\Delta_-^{x_3}s(\mathbf{x})\right)^2}} \\
&+ \frac{1}{h_{x_2}}\Delta_-^{x_2}\frac{\Delta_+^{x_2}s(\mathbf{x})}{\sqrt{\epsilon+\left(\Delta_+^{x_2}s(\mathbf{x})\right)^2+m\left(\Delta_+^{x_3}s(\mathbf{x}),\Delta_-^{x_3}s(\mathbf{x})\right)^2+m\left(\Delta_+^{x_1}s(\mathbf{x}),\Delta_-^{x_1}s(\mathbf{x})\right)^2}} \\
&+ \frac{1}{h_{x_3}}\Delta_-^{x_3}\frac{\Delta_+^{x_3}s(\mathbf{x})}{\sqrt{\epsilon+\left(\Delta_+^{x_3}s(\mathbf{x})\right)^2+m\left(\Delta_+^{x_1}s(\mathbf{x}),\Delta_-^{x_1}s(\mathbf{x})\right)^2+m\left(\Delta_+^{x_2}s(\mathbf{x}),\Delta_-^{x_2}s(\mathbf{x})\right)^2}},
\end{aligned} \tag{B.7}$$

with the function $m(a,b)$ defined as

$$m(a,b) = \frac{\mathrm{sign}a + \mathrm{sign}b}{2}\min(|a|,|b|). \tag{B.8}$$

The added ϵ in the denominator of equation B.7 renders the function robust as the gradient magnitude approaches zero. In our implementation, we have chosen $\epsilon = 0.1$. Again, we assume mirrored boundary pixel values such that the derivatives evaluate to zero at the image boundaries.

C. Deriving the TV regularization of the frequency space magnitudes

The energy formulated in section 4.7.2 for the frequency regularized blind deconvolution is

$$J_{\text{KFTV}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_{\text{KFTV}} \int \Big\| \nabla |\mathcal{F}(h)|(\xi) \Big\| d\xi. \tag{C.1}$$

The minimization of this functional is done using the calculus of variations. Since the Fourier transform is linear, a variation in the spatial domain is equivalent to a variation in the Fourier domain.

$$\int \Big\| \nabla |\mathcal{F}(h+\epsilon g)|(\xi) \Big\| d\xi = \int \Big\| \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)|(\xi) \Big\| d\xi \tag{C.2}$$

The gradient of the kernel regularization term $\int \Big\| \nabla |\mathcal{F}(h)|(\xi) \Big\| d\xi$ can thus be computed as

$$\left. \frac{\partial}{\partial \epsilon} \right|_{\epsilon \to 0} \int \Big\| \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)|(\xi) \Big\| d\xi. \tag{C.3}$$

For the computation of the gradient of the TV regularization of the OTF magnitudes, we use the following fact. Given two square-integrable functions $A : \mathbb{R} \to \mathbb{C} : \xi \mapsto A(\xi)$ and $B : \mathbb{R} \to \mathbb{C}$. Then

$$\int_{\mathbb{R}} A \frac{d}{d\xi} B \, d\xi = - \int_{\mathbb{R}} \frac{d}{d\xi} AB \, d\xi. \tag{C.4}$$

We rewrite the functions as Fourier integrals:

$$\int_{\mathbb{R}} A \frac{d}{d\xi} B \, d\xi = \int_{\mathbb{R}} \int_{\mathbb{R}} a(x_1) e^{-i\xi x_1} dx_1 \cdot \frac{d}{d\xi} \int_{\mathbb{R}} b(x_2) e^{-i\xi x_2} dx_2 \, d\xi \tag{C.5}$$

$$= \underbrace{\int_{\mathbb{R}} \int_{\mathbb{R}} a(x_1) e^{-i\xi x_1} dx_1 \cdot \int_{\mathbb{R}} (-ix_2) \cdot b(x_2) e^{-i\xi x_2} dx_2 \, d\xi}_{\neq 0 \text{ iff } x_2 = -x_1} \tag{C.6}$$

$$= - \int_{\mathbb{R}} \int_{\mathbb{R}} (-ix_1) a(x_1) e^{-i\xi x_1} dx_1 \cdot \int_{\mathbb{R}} b(x_2) e^{-i\xi x_2} dx_2 \, d\xi \tag{C.7}$$

$$= - \int_{\mathbb{R}} \frac{d}{d\xi} \int_{\mathbb{R}} a(x_1) e^{-i\xi x_1} dx_1 \cdot \int_{\mathbb{R}} b(x_2) e^{-i\xi x_2} dx_2 \, d\xi \tag{C.8}$$

$$= - \int_{\mathbb{R}} \frac{d}{d\xi} AB \, d\xi \, . \tag{C.9}$$

This generalizes to n-dimensional functions, as

$$\int_{\mathbb{R}^n} A \nabla B \, d\xi = - \int_{\mathbb{R}^n} \nabla AB \, d\xi \, . \tag{C.10}$$

C. Deriving the TV regularization of the frequency space magnitudes

The energy formulated in section 4.7.2 for the frequency regularized blind deconvolution is

$$J_{\text{KFTV}}(s,h) = J_{\text{MLEM}}(s,h) + \lambda_h \int_{\mathbb{R}^3} \left\| \nabla |\mathcal{F}(h)|(\boldsymbol{\xi}) \right\| d\boldsymbol{\xi}. \tag{C.11}$$

The minimization of this functional is done using the calculus of variations. Since the Fourier transform is linear, a variation in the spatial domain is equivalent to a variation in the Fourier domain.

$$\int_{\mathbb{R}^3} \left\| \nabla |\mathcal{F}(h+\epsilon g)|(\boldsymbol{\xi}) \right\| d\boldsymbol{\xi} = \int_{\mathbb{R}^3} \left\| \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)|(\boldsymbol{\xi}) \right\| d\boldsymbol{\xi} \tag{C.12}$$

The gradient of the kernel regularization term $\int_{\mathbb{R}^3} \left\| \nabla |\mathcal{F}(h)|(\boldsymbol{\xi}) \right\| d\boldsymbol{\xi}$ can thus be computed as

$$\left. \frac{\partial}{\partial \epsilon} \right|_{\epsilon \to 0} \int_{\mathbb{R}^3} \left\| \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)|(\boldsymbol{\xi}) \right\| d\boldsymbol{\xi}. \tag{C.13}$$

$$\begin{aligned}
&\frac{\partial}{\partial \epsilon} \int_{\mathbb{R}^3} \left\| \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)| \right\| d\boldsymbol{\xi} \\
&\quad = \frac{\partial}{\partial \epsilon} \int_{\mathbb{R}^3} \sqrt{\left(\nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)| \right)^2} d\boldsymbol{\xi} \\
&\quad = \int_{\mathbb{R}^3} \frac{1}{2} \frac{1}{\sqrt{\left(\nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)| \right)^2}} \cdot \nabla |\mathcal{F}(h) + \epsilon \mathcal{F}(g)| \\
&\qquad \cdot \nabla \left(\frac{\mathcal{F}(h)\mathcal{F}(g)^* + \mathcal{F}(h)^* \mathcal{F}(g) + \epsilon \mathcal{F}(g)\mathcal{F}(g)^*}{|\mathcal{F}(h) + \epsilon \mathcal{F}(g)|} \right) d\boldsymbol{\xi}
\end{aligned} \tag{C.14}$$

which is, by setting $\epsilon = 0$ and applying the product rule

$$\begin{aligned}
&\overset{\epsilon=0}{=} \int_{\mathbb{R}^3} \frac{1}{2} \frac{\nabla |\mathcal{F}(h)|}{\left\| \nabla |\mathcal{F}(h)| \right\|} \cdot \\
&\quad \left(\nabla \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g) + \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \nabla \mathcal{F}(g) + \nabla \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g)^* + \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \nabla \mathcal{F}(g)^* \right) d\boldsymbol{\xi} \\
&= \frac{1}{2} \left(\underbrace{\int_{\mathbb{R}^3} \frac{\nabla |\mathcal{F}(h)|}{\left\| \nabla |\mathcal{F}(h)| \right\|} \cdot \nabla \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g) d\boldsymbol{\xi} + \int_{\mathbb{R}^3} \frac{\nabla |\mathcal{F}(h)|}{\left\| \nabla |\mathcal{F}(h)| \right\|} \cdot \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \nabla \mathcal{F}(g) d\boldsymbol{\xi}}_{(A)} \right. \\
&\quad \left. + \underbrace{\int_{\mathbb{R}^3} \frac{\nabla |\mathcal{F}(h)|}{\left\| \nabla |\mathcal{F}(h)| \right\|} \cdot \nabla \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g)^* d\boldsymbol{\xi} + \int_{\mathbb{R}^3} \frac{\nabla |\mathcal{F}(h)|}{\left\| \nabla |\mathcal{F}(h)| \right\|} \cdot \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \nabla \mathcal{F}(g)^* d\boldsymbol{\xi}}_{(B)} \right).
\end{aligned} \tag{C.15}$$

Using equation (C.4), (A) equals

$$\begin{aligned}
&\int_{\mathbb{R}^3} \frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \nabla \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g) d\boldsymbol{\xi} - \int_{\mathbb{R}^3} \operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \right) \cdot \mathcal{F}(g) d\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^3} \frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \nabla \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g) d\boldsymbol{\xi} \\
&\quad - \int_{\mathbb{R}^3} \left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} + \frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \nabla \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \right) \cdot \mathcal{F}(g) d\boldsymbol{\xi} \\
&= - \int_{\mathbb{R}^3} \operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g) d\boldsymbol{\xi}, \qquad \text{(C.16)}
\end{aligned}$$

where $\operatorname{div}(\mathbf{x})$ denotes the divergence of vector $\mathbf{x}$. Now, we can use Parseval's theorem and get

$$- \int_{\mathbb{R}^3} \operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)^*}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g) d\boldsymbol{\xi} = - \int_{\mathbb{R}^3} \mathcal{F}^{-1}\left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \right)^* \cdot g \, d\mathbf{x}, \qquad \text{(C.17)}$$

which is, because $\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\|\nabla|\mathcal{F}(h)|\|} \right)$ is symmetric and $\mathcal{F}^{-1}\left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\|\nabla|\mathcal{F}(h)|\|} \right) \right)$ is real valued, equal to

$$- \int_{\mathbb{R}^3} \mathcal{F}^{-1}\left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \right) \cdot g \, d\mathbf{x}. \qquad \text{(C.18)}$$

Now, we compute the same for (B). Because of (C.4),

$$\begin{aligned}
(B) &= \int_{\mathbb{R}^3} \frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \nabla \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g)^* d\boldsymbol{\xi} - \int_{\mathbb{R}^3} \operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \right) \cdot \mathcal{F}(g)^* d\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^3} \frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \nabla \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g)^* d\boldsymbol{\xi} \\
&\quad - \int_{\mathbb{R}^3} \left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} + \frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \cdot \nabla \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \right) \cdot \mathcal{F}(g)^* d\boldsymbol{\xi} \\
&= - \int_{\mathbb{R}^3} \operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|} \right) \cdot \frac{\mathcal{F}(h)}{|\mathcal{F}(h)|} \cdot \mathcal{F}(g)^* d\boldsymbol{\xi}.
\end{aligned} \qquad \text{(C.19)}$$

We can again use Parseval's theorem and get

$$-\int_{\mathbb{R}^3} \operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|}\right)\cdot\frac{\mathcal{F}(h)}{|\mathcal{F}(h)|}\cdot\mathcal{F}(g)^* d\boldsymbol{\xi} = -\int_{\mathbb{R}^3}\mathcal{F}^{-1}\left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|}\right)\cdot\frac{\mathcal{F}(h)}{|\mathcal{F}(h)|}\right)\cdot g\, d\mathbf{x}. \tag{C.20}$$

Now, we can put (A) and (B) together to

$$\text{(C.15)} = -\int_{\mathbb{R}^3}\mathcal{F}^{-1}\left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|}\right)\cdot\frac{\mathcal{F}(h)}{|\mathcal{F}(h)|}\right)\cdot g\, d\mathbf{x}. \tag{C.21}$$

According to the fundamental lemma of calculus of variations, the functional $J_{\text{KFTV}}(s,h)$ is minimized by the Euler Lagrange equation:

$$\int_\Omega s(\mathbf{y})d\mathbf{y} - \left(s^m * \left(\frac{o}{(h*s)}\right)\right) - \lambda_{\text{KFTV}}\mathcal{F}^{-1}\left(\operatorname{div}\frac{\nabla|\mathcal{F}(h)|}{\left\|\nabla|\mathcal{F}(h)|\right\|}\cdot\underbrace{\frac{\mathcal{F}(h)}{|\mathcal{F}(h)|}}_{e^{\imath\cdot\arg(\mathcal{F}(h))}}\right) = 0. \tag{C.22}$$

The resulting multiplicative update scheme for the deconvolution kernel is:

$$\hat{h}_{k+1} = \frac{\hat{h}_k\cdot\left(s^m * \frac{o}{(\hat{h}_k * s)}\right)}{\int_\Omega s(\mathbf{y})d\mathbf{y} - \lambda_{\text{KFTV}}\mathcal{F}^{-1}\left(\operatorname{div}\left(\frac{\nabla|\mathcal{F}(\hat{h}_k)|}{\left\|\nabla|\mathcal{F}(\hat{h}_k)|\right\|}\right)\cdot e^{\imath\cdot\arg(\mathcal{F}(\hat{h}_k))}\right)}. \tag{C.23}$$

The numerical scheme is formulated like in spatial domain (compare appendix B.2).

D. Spherical Harmonics Representation for Active Surfaces

D.1. Spherical Harmonic Basis Functions and their Derivatives

The derivative of the spherical harmonic basis functions

$$Y_l^m(\theta,\phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) e^{\imath m\phi} \tag{D.1}$$

are given by

$$\begin{aligned} \frac{\partial Y_l^m(\theta,\phi)}{\partial\theta} &= \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} \left(-\frac{1}{-1+\cos^2\theta}\right) \cdot \\ &\quad \big((-1-l)\cdot\cos\theta\cdot P_l^m(\cos\theta) + \\ &\quad (1-m+l)\cdot P_{l+1}^m(\cos\theta)\cdot\sin\theta\big) e^{\imath m\phi} \end{aligned} \tag{D.2}$$

and

$$\frac{\partial Y_l^m(\theta,\phi)}{\partial\phi} = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta)\cdot j\cdot m\cdot e^{\imath m\phi}. \tag{D.3}$$

The associated Legendre polynomials can be computed recursively as for example given in [Khairy and Howard, 2008] using the relations:

$$P_l^l(\cos\theta) = \frac{(2l)!}{l!}\left(\frac{1}{2}\sin\theta\right)^l \tag{D.4}$$

for $l \in \mathbb{N}$ and

$$(l+m)(l-m+1)P_l^{m-1}(\cos\theta) = 2m\cot\theta P_l^m(\cos\theta) - P_l^{m+1}(\cos\theta), \tag{D.5}$$

with $P_l^m(\cos\theta) = 0$ for $m > l$.

D.2. Representing Shapes with Spherical Topology

Arbitrary shapes with spherical topology can be represented in the spherical harmonics domain. A method for a bijective mapping of a tesselated surface onto a sphere was proposed by [Brechbühler et al., 1995]. This way, every three-tuple of coordinates can be assigned a pair of angles such that:

$$\mathbf{\Gamma}(\theta,\phi) = \begin{pmatrix} x(\theta,\phi) \\ y(\theta,\phi) \\ z(\theta,\phi) \end{pmatrix}, \tag{D.6}$$

thus laying a latitude-longitude grid over the shape. The so parameterized shape can be represented by three vectors of spherical harmonic expansion coefficients $\hat{X}$, $\hat{Y}$ and $\hat{Z}$, where

$$\hat{X} = \{\hat{x}_0^0, \hat{x}_1^{-1}, \hat{x}_1^0, \hat{x}_1^1, \ldots, \hat{x}_l^l\}, \tag{D.7}$$

with

$$\hat{x}_l^m = \int_0^{2\pi} \int_0^{\pi} x(\theta, \phi) Y_l^{m*}(\theta, \phi) \sin\theta \, d\theta d\phi, \tag{D.8}$$

where Y_l^{m*} is the complex conjugate of Y_l^m. The vectors of coefficients $\hat{Y}$ and $\hat{Z}$ are defined accordingly. As for the spherical harmonic representation of star-shaped surfaces, a rotation invariant shape descriptor can be generated by the L_2 norms of the coefficient vectors for each band [Kazhdan, 2004]:

$$D = \left\{ \sqrt{\left(\hat{x}_0^0\right)^2 + \left(\hat{y}_0^0\right)^2 + \left(\hat{z}_0^0\right)^2}, \ldots, \sqrt{\left(\hat{x}_l^l\right)^2 + \left(\hat{y}_l^l\right)^2 + \left(\hat{z}_l^l\right)^2} \right\}. \tag{D.9}$$

D.2.1. Regularization

In the following, we give formulas for the analytical computation of the surface normals and curvature that are needed for the computation of the mean curvature flow $\frac{\partial \mathbf{\Gamma}}{\partial t} = H\mathbf{n}$.
For the computation of the surface normals, we need the derivatives of the surface which can be computed as [Khairy and Howard, 2008]

$$\frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta} = \begin{pmatrix} \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{x}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \theta} \\ \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{y}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \theta} \\ \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{z}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \theta} \end{pmatrix} \quad \text{and} \quad \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi} = \begin{pmatrix} \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{x}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \phi} \\ \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{y}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \phi} \\ \sum_{l=0}^{B} \sum_{m=-l}^{l} \hat{z}_l^m \frac{\partial Y_l^m(\theta,\phi)}{\partial \phi} \end{pmatrix}. \tag{D.10}$$

From the surface derivatives, the surface normal $\mathbf{n}(\theta, \phi)$ is computed according to equation (5.47). With the surface derivatives, the normal and derivatives of the normal, the first (E, F, G) and second (L, M, N) fundamental forms can be computed. The first fundamental forms are given by

$$\begin{aligned} E &= \left\| \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta} \right\|^2 \qquad F = \left\langle \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \theta}, \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi} \right\rangle \\ G &= \left\| \frac{\partial \mathbf{\Gamma}(\theta,\phi)}{\partial \phi} \right\|^2, \end{aligned} \tag{D.11}$$

the second fundamental can be computed according to equation (5.49).

E. Fast Gradient Vector Flow Computation

The two dimensional gradient vector flow (see section 2.3.3) can easily be extended to three or more dimensions. In 3D we search for a vector field $\mathbf{v} : \Omega \subseteq \mathbb{R}^3 \rightarrow \mathbb{R}^3, \mathbf{v}(\mathbf{x}) = \big(u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x})\big)^T$ that minimizes the energy functional

$$E^{\text{GVF}}(\mathbf{v}) = \int_\Omega \mu(\|\nabla \mathbf{v}\|^2) + \|\nabla M\|^2 \|\mathbf{v} - \nabla M\|^2 d\mathbf{x}, \tag{E.1}$$

for a three dimensional edge map M. The energy is minimized by the solution of the following Euler-Lagrange equations:

$$\begin{aligned}
\mu\nabla^2 u - \left(u - \frac{\partial M}{\partial x_1}\right)\left(\left(\frac{\partial M}{\partial x_1}\right)^2 + \left(\frac{\partial M}{\partial x_2}\right)^2 + \left(\frac{\partial M}{\partial x_3}\right)^2\right) &= 0 \\
\mu\nabla^2 v - \left(v - \frac{\partial M}{\partial x_2}\right)\left(\left(\frac{\partial M}{\partial x_1}\right)^2 + \left(\frac{\partial M}{\partial x_2}\right)^2 + \left(\frac{\partial M}{\partial x_3}\right)^2\right) &= 0 \\
\mu\nabla^2 w - \left(w - \frac{\partial M}{\partial x_3}\right)\left(\left(\frac{\partial M}{\partial x_1}\right)^2 + \left(\frac{\partial M}{\partial x_2}\right)^2 + \left(\frac{\partial M}{\partial x_3}\right)^2\right) &= 0.
\end{aligned} \tag{E.2}$$

The solution can be found by gradient descent.In order to accelerate the computation of the GVF, it can be implemented with a Gauss-Seidel method or the even faster successive over-relaxation (SOR) [Press et al., 1992]:

$$u^{n+1}(\mathbf{x}) = (1-\omega)u^n(\mathbf{x}) + \omega \frac{\sum\limits_{\substack{\mathbf{y}\in\mathcal{N}(\mathbf{x}),\\ y_1\le x_1 \wedge y_2 \le x_2 \wedge y_3 \le x_3}} u^{n+1}(\mathbf{y}) + \sum\limits_{\substack{\mathbf{y}\in\mathcal{N}(\mathbf{x}),\\ y_1> x_1 \vee y_2 > x_2 \vee y_3 > x_3}} u^n(\mathbf{y}) + b}{\frac{1}{\mu}\left(\left(\frac{\partial M}{\partial x_1}(\mathbf{x})\right)^2 + \left(\frac{\partial M}{\partial x_2}(\mathbf{x})\right)^2 + \left(\frac{\partial M}{\partial x_3}(\mathbf{x})\right)^2\right) + \sum\limits_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} 1}, \tag{E.3}$$

where

$$b = \frac{1}{\mu}\frac{\partial M}{\partial x_1}(\mathbf{x})\left(\left(\frac{\partial M}{\partial x_1}(\mathbf{x})\right)^2 + \left(\frac{\partial M}{\partial x_2}(\mathbf{x})\right)^2 + \left(\frac{\partial M}{\partial x_3}(\mathbf{x})\right)^2\right). \tag{E.4}$$

$\omega \in (0, 2)$ is the relaxation parameter. The update for v^{n+1} and w^{n+1} is computed accordingly. For $\omega = 1$, this equals the Gauss-Seidel method, whereas faster convergence can be expected for $\omega > 1$. In [Han et al., 2007], the authors propose an even faster multigrid GVF method. However, in our own implementation we used the SOR implementation as it was fast enough and produced reliable results.

Bibliography

M. Abramowitz. *Fluorescence Microscopy: The Essentials*, volume 4. Olympus Corporation Publishing, 1993.

D. Agard. Optical sectioning microscopy: cellular architecture in three dimensions. *Annu. Rev. Biophys. Bio.*, 13:191–219, 1984.

F. Aguet. Super-resolution fluorescence microscopy based on physical models, 2009. Thesis no. 4418.

F. Aguet, M. Jacob, and M. Unser. Three-dimensional feature detection using optimal steerable filters. In *Int. Conf. on Image Processing (ICIP)*, pages 1158–1161, 2005.

J. Ahlberg. Active contours in three dimensions, 1996. Report No. LiTH-ISY-EX-1708, Linköping University, Sweden.

J. Ahlberg, K. Nordberg, and J. Wiklund. 3-d active contours attracted to planar structure, 2000. Report No. LiTH-ISY-R-2300, Linköping University, Sweden.

M. Allili and D. Ziou. An approach for dynamic combination of region and boundary information in segmentation. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 1–4, 2008.

L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Comm. Pure Appl. Math.*, 43:999–1036, 1990.

B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. Probabilistic image segmentation with closedness constraints. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2611 – 2618, 2011.

P. Arbeláez. Boundary extraction in natural images using ultrametric contour maps. In *Conf. on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW*, page 182, 2006.

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2294–2301, 2009.

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

J.-F. Aujol and A. Chambolle. Dual norms and image decomposition models. *Int. J. Comput. Vision (IJCV)*, 63(1):85–104, 2005.

D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, NJ, 1981.

B. J. Baumgart. Winged-edge polyhedron representation, 1972. Master thesis, Technical Report CS-320, Stanford University.

J. Besag. On the statistical analysis of dirty pictures,o (with discussion). *J. R. Stat. Soc. Series B*, 48(3):259–302, 1986.

S. Beucher. Segmentation d'images et morphologie mathématique, 1990. Thèse, École National Supérieure des Mines de Paris.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2009.

M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th Edition)*. Cambridge University Press, 1999.

R. W. Boyd. *Radiometry and the detection of optical radiation*. New York, John Wiley and Sons, 1983.

Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

Y. Boykov, V. O., and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

J. Boyle and R. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. *Lecture Notes in Statistics*, 37:28–47, 1986.

C. Brechbühler, G. Gerig, and O. Kübler. Parametrization of closed surfaces for 3-d shape description. *Comput. Vis. Image Underst.*, 61(2):154–170, 1995.

X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Global minimizers of the active contour/snake model. In *UCLA CAM Report 05-04*, 2005.

X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.*, 28:151–167, 2007.

T. Brox and J. Weickert. Level set segmentation with multiple regions. *IEEE Trans. Image Process.*, 15(10):3213–3218, 2006.

T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2225–2232, 2011.

Z. Bu and D. Callaway. Chapter 5 - proteins move! protein dynamics and long-range allostery in cell signaling. In R. D., editor, *Protein Structure and Diseases*, volume 83 of *Advances in Protein Chemistry and Structural Biology*, pages 163–221. Academic Press, 2011.

T. Buehler and M. Hein. Spectral clustering based on the graph p-laplacian via the inverse power method. In *Int. Conf. on Machine Learning (ICML)*, pages 81–88. Omnipress, 2009.

A. Carasso. Linear and nonlinear image deblurring: A documented study. *SIAM J. Numer. Anal.*, 36:1659–1689, 1999.

C. S. Casas Delucchi, A. Brero, M. Keuper, O. Ronneberger, A. Wutz, H. Burkhardt, B. Joffe, I. Solovei, and T. Cremer. Chromosome territories and nuclear architecture: A study using native and transgenic Xist inactivation. *Cellular Oncology*, 30(3):251, 2008.

V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer. Math.*, 66:1–31, 1993.

V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vision (IJCV)*, 22 (1):61–79, 1997.

M. Chalfie, Y. Tu, G. Euskirchen, W. Ward, and D. Prasher. Green fluorescent protein as a marker for gene expression. *Science*, 263:802–805, 1994.

A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *preprint*, 2012.

T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266 –277, 2001.

T. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of denoising and segmentation models. *SIAM J. Sci. Comput.*, 66:1632–1648, 2006.

H. Chang, M. Auer, and B. Parvin. Structural annotation of em images by graph cut. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 1103–1106, 2009.

Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17 (8):790–799, 1995.

P. Chlap. Detection and description of landmark candidates for the registration of 3D zebrafish embryos, 2010. Master Thesis, Albert-Ludwigs-Universität Freiburg, Supervisor: Margret Keuper.

S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28:145:1–145:8, 2009.

F.-R.-K. Chung. *Spectral graph theory*. AMS, 1997.

G. Chung and L. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. In *5th Int. Workshop, EMMCVPR*, pages 439–455, 2005.

L. Cohen. On active contour models and balloons. *CVGIP: Image Underst.*, 53(2):211–218, 1991.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24/5:603–619, 2002.

J.-A. Conchello and J. McNally. Fast regularization technique for expectation maximization algorithm for optical sectioning microscopy. In C. J. Cogswell, G. S. Kino, and T. Wilson, editors, *Three-Dimensional Microscopy: Image Acquisition and Processing III*, volume 2655, pages 199–208. Proc. SPIE, 1996.

J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of the complex fourier series. *Math. Comput.*, 19:297–301, 1965.

S. Coons. Surface patches and b-spline curves. *Comput. Aided Geom. D.*, 1974.

T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.

D. Cremers. Dynamical statistical shape priors for level set based tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1262–1273, 2006.

D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *Int. J. Comput. Vision (IJCV)*, 50(3):295–313, 2002.

D. Cremers, T. Pock, K. Kolev, and A. Chambolle. Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

H. Cundy and A. Rollett. *Mathematical Models*. Oxford Univ. Press, 2nd ed., 1961.

C. de Boor. *A Practical Guide to Splines*. Springer, 1978.

B. de Monvel, S. L. Calvez, and M. Ulfendahl. Image restoration for confocal microscopy: Improving the limits of deconvolution, with application to the visualization of the mammalian hearing organ. *Biophys. J.*, 80:2455–2470, 2001.

C. D'Elia, G. Poggi, and G. Scarpa. A tree-structured Markov random field model for Bayesian image segmentation. *IEEE Trans. Image Process.*, 12(10):1259–1273, 2003.

H. Delingette. General object reconstruction based on simplex meshes. *Int. J. Comput. Vision (IJCV)*, 32(2):111–146, 1999.

A. Delong and Y. Boykov. A scalable graph-cut algorithm for n-d grids. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

N. Dey, L. Blanc-Feraud, C. Zimmer, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia. A deconvolution method for confocal microscopy with total variation regularization. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, volume 2, pages 1223–1226, 2004a.

N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivio-Marin, and J. Zerubia. 3D microscopy deconvolution using Richardson-Lucy algorithm with total variation regularization. Technical report, INRIA, 2004b.

N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J. Olivo-Marin, and J. Zerubia. Richardson-Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution. *Microsc. Res. Techniq.*, 69:260–266, 2006.

C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *IEEE Int. Conf. on Data Mining (ICDM)*, pages 107–114, 2001.

E. A. Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Math. Dokl.*, 11:1277–1280, 1970.

A. Dovzhenko, U. Bergen, and H.-U. Koop. Thin alginate layer (tal)-technique for protoplast culture of tobacco leaf protoplasts: Shoot formation in less than two weeks. *Protoplasma*, 204:114–118, 1998.

A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Trans. Image Process.*, 14(9):1396–1410, 2005.

M. Emmenlauer, O. Ronneberger, A. Ponti, P. Schwarb, A. Griffa, A. Filippi, R. Nitschke, W. Driever, and H. Burkhardt. Xuvtools: free, fast and reliable stitching of large 3D datasets. *J. Microsc.*, 233(1):42–60, 2009.

J. Fehr, O. Ronneberger, H. Kurz, and H. Burkhardt. Self-learning segmentation and classification of cell-nuclei in 3D volumetric data using voxel-wise gray scale invariants. In W. Kropatsch and R. Sablating, editors, *Proc. of the 27th DAGM Symp.*, LNCS, pages 377–384. Springer, 2005.

P. Felzenszwalb and D. Huttenlocher. Image segmentation using local variations. In *IEEE Int. Conf. on Computer Vision and Pattern Recogntition (CVPR)*, pages 98–104, 1998.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision (IJCV)*, 59(2):167–181, 2004.

L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton: Princeton University Press, 1962.

B. Frieden. Optical transfer of the three-dimensional object. *J. Opt. Soc. Am. A*, 57:56–66, 1967.

S. Frisken-Gibson and F. Lanni. Diffraction by a circular aperture as a model for three-dimensional optical microscopy. *J. Opt. Soc. Am. A*, 6:1357–1367, 1989.

S. Frisken-Gibson and F. Lanni. Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *J. Opt. Soc. Am. A*, 9: 154–166, 1992.

J.-M. Fritschy and W. Härtig. *Immunofluorescence*. John Wiley & Sons, Ltd, 2001.

B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 670–677, 2009.

B. Fultz and J. Howe. *Transmission Electron Microscopy and Diffractometry of Materials*. Advanced Texts in Physics. Springer, 2007.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6/6:712–741, 1984.

S. Geman, D. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12/7:609–628, 1990.

J.-M. Geusebroek, B. T. H. Romeny, J. Koenderink, R. van den Boomgaard, and P. van Osta. *Color differential structure, Front-End Vision and Multi-Scale Image Analysis*, volume 27 of *Computational Imaging and Vision*, pages 311–327. Springer Netherlands, 2003.

A. Goldberg and R. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35:921–940, 1988.

R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Trans. Image Process.*, 10(10):1467–1475, 2001.

J. W. Goodman. *Introduction to Fourier Optics (2nd edition)*. Electrical and Computer Engeneering. McGraw-Hill Companies, Inc, USA, 1996.

R. Graef, J. Rietdorf, and T. Zimmermann. Live cell spinning disk microscopy. *Advances in Biochemical Engineering/Biotechnology*, 95:1311–1315, 2005.

P. Green. Bayesian reconstructions from emission tomography data using a modified em algorithm. *IEEE Trans. Med. Imag.*, pages 84–93, 1990a.

P. Green. On use of the em for penalized likelihood estimation. *J. R. Stat. Soc. Series B*, 52(3): 443–452, 1990b.

D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Stat. Soc. Series B*, 51:271–279, 1989.

U. Grenander and M. Miller. *Pattern Theory - From Representation to Inference*. Oxford University Press, 2007.

M. Gustafsson, D. A. Agard, and J. W. Sedat. Sevenfold improvement of axial resolution in 3D widefield microscopy using two objective lenses. In *Three-Dimensional Microscopy: Image Acquisition and Processing II*, volume 2412, pages 147–156. Proc. SPIE, 1995.

L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Comput.-aided Des. Integr. Circuits Syst.*, 11(9):1074–1085, 1992.

X. Han, C. Xu, and J. Prince. Fast numerical scheme for gradient vector flow computation using a multigrid method. *Image Processing, IET*, 1(1):48–55, 2007.

M. Hein and T. Buehler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca. In *Advances in Neural Information Processing Systems (NIPS)*, pages 847–855, 2010.

S. Heinzer, S. Wörz, C. Kalla, K. Rohr, and M. Weiss. A model for the self-organization of exit sites in the endoplasmic reticulum. *J. Cell Sci.*, 121(1):55–64, 2008.

T. Holmes. Blind deconvolution of quantum-limited incoherent imagery: maximumlikelihood. *J. Opt. Soc. Am. A*, 9:1052–1061, 1992.

E. Hom, F. Marchis, T. Lee, S. Haase, D. Agard, and J. Sedat. Aida: an adaptive image deconvolution algorithm with application to multi-frame and three-dimensional data. *J. Opt. Soc. Am. A*, 24:1580–1600, 2007.

H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Conf. on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 19–26. ACM, 1993.

L. Hsu, R. Kusner, and J. Sullivan. Minimizing the squared mean curvature integral for surfaces in space forms. *Exp. Math.*, 1:191–207, 1992.

D. Huhn. Texture analysis for the segmentation of t-cells in recordings from electron microscopy, 2009. Student Research Project, Albert-Ludwigs-Universität Freiburg, Supervisor: Margret Keuper.

H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(10):1333–1336, 2003.

O. Juan and Y. Boykov. Active graph cuts. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1023–1029, 2006.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vision (IJCV)*, 1(4):321–331, 1988.

F. Kaster, B. H. Menze, M.-A. Weber, and F. A. Hamprecht. Comparative validation of graphical models for learning tumor segmentations from noisy manual annotations. In *MICCAI 2010 Workshop MCV*, pages 74–85. Springer, 2011.

M. Kazhdan. Shape representations and algorithms for 3D model retrieval, 2004. PhD thesis, Princeton University.

M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symp. on Geometry Processing*, pages 156–164, 2003.

A. Kelemen, G. Szekely, and G. Gerig. Elastic model-based segmentation of 3-d neuroradiological data sets. *IEEE Trans. Med. Imag.*, 18(10):828–839, 1999.

T. Kenig. Blind deconvolution in wide-field fluorescence microscopy, 2009. Master Thesis, Technion - Israel Institute of Technology, Haifa.

T. Kenig, Z. Kam, and A. Feuer. Blind image deconvolution using machine learning for three-dimensional microscopy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2191–2204, 2010.

M. Keuper. Robust invariant features for the analysis and comparison of chromosome territories in cell nuclei, 2007. Diploma Thesis, Albert-Ludwigs-Universität Freiburg, Supervisor: Olaf Ronneberger.

M. Keuper. Mean shifting gradient vector flow, an improved force field for active surfaces in widefield microscopy. Technical report, IIF-LMB, University of Freiburg, 2010.

M. Keuper, J. Padeken, P. Heun, H. Burkhardt, and O. Ronneberger. A 3D active surface model for the accurate segmentation of *Drosophila* schneider cell nuclei and nucleoli. In *Int. Symp. on Visual Computing (ISVC), Part I*, volume 5875, pages 865–874. LNCS, Springer, 2009.

M. Keuper, R. Bensch, K. Voigt, A. Dovzhenko, K. Palme, H. Burkhardt, and O. Ronneberger. Semi-supervised learning of edge filters for volumetric image segmentation. In *Proc. of the 32nd DAGM Symp.*, LNCS, pages 462–471. Springer, 2010a.

M. Keuper, J. Padeken, P. Heun, H. Burkhardt, and O. Ronneberger. Mean shift gradient vector flow: A robust external force field for 3D active surfaces. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 2784–2787, 2010b.

M. Keuper, T. Schmidt, J. Padeken, P. Heun, K. Palme, H. Burkhardt, and O. Ronneberger. 3D deformable surfaces with locally self-adjusting parameters - a robust method to determine cell nucleus shapes. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, pages 2254–2257, 2010c.

M. Keuper, T. Schmidt, M. Rodriguez-Franco, W. Schamel, T. Brox, H. Burkhardt, and O. Ronneberger. Hierarchical Markov random fields for mast cell segmentation in electron microscopic recordings. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 973–978, 2011.

M. Keuper, M. Temerinac-Ott, J. Padeken, P. Heun, T. Brox, H. Burkhardt, and O. Ronneberger. Blind deconvolution with PSF regularization for widefield microscopy. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 1292–1295, 2012.

K. Khairy and J. Howard. Spherical harmonics-based parametric deconvolution of 3D surface images using bending energy minimization. *Medical Image Analysis*, 12:217–227, 2008.

H. Kirshner, D. Sage, and M. Unser. 3D PSF models for fluorescence microscopy in ImageJ. In *MAF'12*, page 154, 2011.

H. Kitano. Computational systems biology. *Nature*, 420:206–210, 2002.

M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. An experimental comparison of discrete and continuous shape optimization methods. In *European Conf. on Computer Vision (ECCV)*, pages 332 – 345, 2008.

P. Kohli and P. Torr. Dynamic graph cuts for efficient inference in Markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2079–2088, 2007.

V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.

V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, 2004.

N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2985 –2992, 2009.

N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

E. Koziolek. Evaluation of deconvolution methods in volumetric microscopic data, 2011. Student Research Project, Albert-Ludwigs-Universität Freiburg, Supervisor: Margret Keuper and Maja Temerinac-Ott.

J. Lellmann and C. Schnörr. Continuous multiclass labeling approaches and algorithms. *SIAM J. Imag. Sci.*, 4(4):1049–1096, 2011.

J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *Int. Conf. on Computer Vision (ICCV)*, pages 646–653, 2009.

M. Leventon, E. Grimson, and O. Faugeras. Level set based segmentation with intensity and curvature priors. In *IEEE Mathematical Methods in Biomedical Image Analysis*, pages 1–4, 2000.

J. Lichtman and J.-A. Conchello. Fluorescence microscopy. *Nature Methods*, 2:910–919, 2005.

J. Lim, P. Arbeláez, C. Gu, and J. Malik. Context by region ancestry. In *IEEE Int. Conf. on Computer Vision*, pages 1978–1985, 2009.

A. Lipson, S. G. Lipson, and H. Lipson. *Optical Physics*. Cambridge Universtiy Press, 2011.

M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *IEEE Int. Conf. on Computer Vision and Pattern Recogntition (CVPR)*, pages 2097 – 2104, 2011.

X. Lou, U. Köthe, J. Willbrodt, and F. Hamprecht. Learning to segment dense cell nuclei with shape prior. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1012 – 1018, 2012.

L. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79/6:745–754, 1974.

M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(2):158–175, 1995.

M. Mantyla. *An Introduction to Solid Modeling*. Computer Science Press, Rockville, Ma., 1988.

J. Markham and J.-A. Conchello. Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur. *J. Opt. Soc. Am. A*, 16(10):2377–2391, 1999.

J. Markham and J.-A. Conchello. Fast maximum-likelihood image-restoration algorithms for three-dimensional fluorescence microscopy. *J. Opt. Soc. Am. A*, 18(5):1062–1071, 2001.

D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

J. Menter. Temperature dependence of collagen fluorescence. *Photochem. Photobiol. Sci.*, 5:403–410, 2006.

J. Mille, R. Boné, P. Makris, and H. Cardot. Greedy algorithm and physics-based method for active contours and surfaces: a comparative study. In *IEEE Int. Conf. on Image Processing (ICIP)*, pages 1645–1648, 2006.

K. Miller. Least squares methods for ill-posed problems with a prescribed bound. *SIAM J. Math. Anal.*, 1:52–74, 1970.

M. Miller, S. Joshi, D. Maffiitt, J. McNally, and U. Grenander. Membranes, mitochondria and amoebae: shape models. *J. Appl. Stat.*, 21/1:141–163, 1994.

A. Mishra, D. A. Clausi, and P. Fieguth. From active contours to active surfaces. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2121–2128, 2011.

B. Mičušík and T. Pajdla. Multi-label image segmentation via max-sum solver. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

V. Morath, M. Keuper, M. Rodriguez-Franco, S. Deswal, G. Fiala, B. Blumenthal, D. Kaschek, J. Timmer, G. Neuhaus, O. Ronneberger, and W. Schamel. Semi-automatic determination of cell surface areas used in systems biology. *Frontiers in Bioscience, Elite*, 5:533–545, 2013.

D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pur. Appl. Math.*, pages 577–685, 1989.

L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(12):1163–1173, 1996.

O. Nakamura and S. Kawata. Three-dimensional transfer-function analysis of the tomographic capability of a confocal fluorescence microscope. *J. Opt. Soc. Am. A*, 7:522–526, 1990.

J. Nuñez and J. Llacer. A fast bayesian reconstruction algorithm for emission tomography with entropy prior converging to feasible images. *IEEE Trans. Med. Imag.*, 9/2:159–171, 1990.

P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1583–1590, 2011.

S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton–jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

N. Otsu. A threshold selection method from grey level histograms. *IEEE Trans. Syst. Man Cyb.*, 9 (1):62–66, 1979.

P. Pankajakshan, L. Blanc-Feraud, Z. Kam, and J. Zerubia. Point-spread function retrieval for fluorescence microscopy. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 1095–1098, 2009.

N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1034–1040, 1999a.

N. Paragios and R. Deriche. Geodesic active regions for motion estimation and tracking. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 688–694, 1999b.

N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: A level set approach. In *European Conf. on Computer Vision (ECCV)*, pages 224–240, 2000.

N. Paragios and R. Deriche. Geodesic active regions: a new framework to deal with frame partition problems in computer vision. *J. Vis. Commun. Image R.*, 13(1):249–268, 2002.

N. Paragios, O. Mellina-Gottardo, and V. Ramesh. Gradient vector flow fast geodesic active contours. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 67–73, 2001.

J. Philip. Optical transfer function in three dimensions for a large numerical aperture. *J. Mod. Optic.*, 46:1031–1042, 2009.

N. Plath, M. Toussaint, and S. Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *Int. Conf. on Machine Learning (ICML)*, pages 817–824. ACM, 2009.

T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 810–817, 2009.

R. B. Potts. Some generalized order-disorder transformations. *Math. Proc. Cambridge*, 48:106–109, 1952.

W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes in C*. Cambridge University Press, New York, USA, 2nd edition, 1992.

C. Preza and J.-A. Conchello. Depth-variant maximum-likelihood restoration for three-dimensional fluorescence microscopy. *J. Opt. Soc. Am. A*, 21(9):1593–1601, 2004.

K. Renz. Comparison of different tracking methods in 3D datasets, 2010. Master Thesis, Albert-Ludwigs-Universität Freiburg, Supervisor: Thorsten Schmidt and Margret Keuper.

J. Reynolds and K. Murphy. Figure-ground segmentation using a hierarchical conditional random field. In *Canadian Conf. on Computer and Robot Vision*, pages 175–182, 2007.

B. Richards and E. Wolf. Electromagnetic diffraction in optical systems. ii. structure of the image field in an aplanatic system. *Proc. of R. Soc. Lond. A*, 253:358–379, 1959.

W. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am. A*, 62/1: 55–59, 1972.

Roerdink and Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *FUNDINF: Fundamenta Informatica*, 41(1-2):187–228, 2000.

O. Ronneberger, J. Fehr, and H. Burkhardt. Voxel-wise gray scale invariants for simultaneous segmentation and classification. In *Proc. of the 27th DAGM Symp.*, LNCS, pages 85–92. Springer, 2005.

O. Ronneberger, Q. Wang, and H. Burkhardt. Fast and robust segmentation of spherical particles in volumetric data sets from brightfield microscopy. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 372–375, 2008.

L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, 1992.

P. Sarder and A. Nehorai. Deconvolution methods for 3-d fluorescence microscopy images. *Signal Processing Magazine*, 23(3):32–45, 2006.

A. Sarti, R. Malladi, and J. A. Sethian. Subjective surfaces: A method for completing missing boundaries. *PNAS*, 97/12:6258–6263, 2000.

T. Schmidt, M. Keuper, T. Pasternak, K. Palme, and O. Ronneberger. Modeling of sparsely sampled tubular surfaces using coupled curves. In *Proc. of the 34nd DAGM Symp.*, volume 7476 of *LNCS*, pages 83–92. Springer, 2012. (to appear).

A. Schönle and S. Hell. Calculation of vectorial three-dimensional transfer functions in large-angle focusing systems. *J. Opt. Soc. Am. A*, 19(10):2121–2126, 2002.

R. B. Schudy and D. H. Ballard. Model detection of cardiac chambers in ultrasound images, 1978. Technical Report 12, Computer Science Department, University of Rochester.

R. B. Schudy and D. H. Ballard. Towards an anatomical model of heart motion as seen in 4-d cardiac ultrasound data. In *Conf. on Computer Applications in Radiology and Computer-Aided Analysis of Radiological Images*, 1979.

J. Schulz, T. Schmidt, O. Ronneberger, H. Burkhardt, T. Pasternak, A. Dovzhenko, and K. Palme. Fast scalar and vectorial grayscale based invariant features for 3D cell nuclei localization and classification. In *Proc. DAGM 2006, Berlin, Germany*, volume 4174 of *LNCS*, pages 182–191. Springer, 2006.

P. Shaw. Comparison of widefield/deconvolution and confocal microscopy in three-dimensional imaging. In J. B. Pawley, editor, *Handbook of Biological Confocal Microscopy*. Springer, 1996.

L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imag.*, 1(2):113–122, 1982.

C. Sheppard and M. Gu. Three-dimensional transfer functions for high- aperture systems. *J. Opt. Soc. Am. A*, 11:593–598, 1994.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

H. Skibbe, M. Reisert, T. Schmidt, T. Brox, O. Ronneberger, and H. Burkhardt. Fast rotation invariant 3D feature computation utilizing efficient local neighborhood operators. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34:1563–1575, 2012.

G. Slabaugh and G. Unal. Active polyhedron: Surface evolution theory applied to deformable meshes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 84–91, 2005.

C. Sommer, C. Strähle, U. Köthe, and F. A. Hamprecht. ilastik: Interactive learning and segmentation toolkit. In *IEEE Int. Symp. on Biomedical Imaging (ISBI)*, pages 230–233, 2011.

K. Spring. Fluorescence microscopy. *Encyclopedia of Optical Engineering*, pages 549–555, 2003.

J.-L. Starck, E. Pantin, and F. Murtagh. Deconvolution in astronomy: a review. *Publ. Astron. Soc. Pac.*, 114:1051–1069, 2002.

P. A. Stokseth. Properties of a defocused optical system. *J. Opt. Soc. Am. A*, 59:1314–1321, 1969.

G. Strang. Maximal flow through a domain. *Math. Program.*, 26:123–143, 1983.

N. Streibl. Three-dimensional imaging by a microscope. *J. Opt. Soc. Am. A*, 2:121–127, 1985.

D. Svoboda, M. Kašík, M. Maška, J. Hubený, S. Stejskal, and M. Zimmermann. On simulating 3D fluorescent microscope images. In *Int. Conf. on Computer Analysis of Images and Patterns*, CAIP 07, pages 309–316. Springer, 2007.

D. Svoboda, M. Kozubek, and S. Stejskal. Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry. *Cytometry Part A*, 75A(6):494–509, 2009.

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, 2008.

A. Szlam and X. Bresson. Total variation and cheeger cuts. In *Int. Conf. on Machine Learning (ICML)*, pages 1039–1046. Omnipress, 2010.

D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(4):413–224, 1986.

D. Terzopoulos, J. Platt, A. Barr, and K. Fleischert. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4):205–214, 1987.

D. Thomann, D. Rines, P. Sorger, and G. Danuser. Automatic fluorescent tag detection in 3D with superresolution: Application to the analysis of chromosome movement. *J. Microsc.*, 208:49–64, 2002.

A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston, 1987.

D. Toomre and J. Pawley. Disk-scanning confocal microscopy. In J. B. Pawley, editor, *Handbook of Biological Confocal Microscopy*. Springer, 1996.

G. vanKempen, L. van Vliet, P. Verveer, and H. vanderVoort. A quantitative comparison of image restoration methods for confocal microscopy. *J. Microsc.*, 185:354–365, 1997.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

T. Vehkomäki, G. Gerig, and G. Székely. A user-guided tool for efficient segmentation of medical image data. In J. Troccaz, E. Grimson, and R. Mösges, editors, *CVRMed-MRCAS'97*, volume 1205 of *LNCS*, pages 685–694. Springer, 1997.

P. Verveer and T. Jovin. Efficient superresolution restoration algorithms using maximum a posteriori estimations with applicationto fluorescence microscopy. *J. Opt. Soc. Am. A*, 14:1696–1706, 1997.

L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vision (IJCV)*, 50:271–293, 2002.

L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598, 1991.

C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput*, 17:227–238, 1996.

U. von Luxburg. A tutorial on spectral clustering. *Stat. Comp.*, 17(4):395–416, 2007.

T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1165–1178, 2007.

N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York: Wiley, 1949.

S. Wilhelm, B. Groebler, M. Gluch, and H. Heinz. Confocal laser scanning microscopy principles, 2012. brochure.

G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods - A mathematical introduction*. Springer, 2nd edition, 2003.

S. Wörz, C. Kappel, R. Eils, and K. Rohr. Model-based segmentation and quantification of fluorescent bacteria in 3D microscopy live cell images. In J. Pluim and J. Reinhardt, editors, *Medical Imaging 2007: Image Processing*, volume 6512 of *Proc. SPIE.*, 2007.

S. Wörz, P. Sander, M. Pfannmoller, R. Rieker, S. Joos, G. Mechtersheimer, P. Boukamp, P. Lichter, and K. Rohr. 3D geometry-based quantification of colocalizations in multichannel 3D microscopy images of human soft tissue tumors. *IEEE Trans. Med. Imag.*, 29(8):1474–1484, 2010.

T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *J. Machine Learning Research*, 5:975–1005, 2004.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1101–1113, 1993.

C. Xu and J. Prince. Gradient vector flow: A new external force for snakes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 66–71, 1997.

C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Process.*, 7(3): 321–345, 1998.

C. Xu, A. Yezzi Jr., and J. L. Prince. On the relationship between parametric and geometric active contours. In *Proc. of 34th Asilomar Conference on Signals, Systems, and Computers*, pages 483 –489, 2000.

G. Xu and Q. Zhang. Minimal mean-curvature-variation surfaces and their applications in surface modeling. In *Geometric Modeling and Processing - GMP 2006*, volume 4077 of *LNCS*, pages 357–370. Springer, 2006.

R. Xu and D. Wunsch. *Clustering*. Wiley, 2008.

Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object detection for multi-class segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3113–3120, 2010.

P. A. Yushkevich, J. Piven, C. Hazlett, S. H., G. Smith, R. Ho, S. Ho, J.-C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision, Modeling and Visualization Workshop (VMV)*, pages 243–252, 2008.

B. Zhang, J. Zerubia, and J.-C. Olivo-Marin. A study of Gaussian approximations of fluorescence microscopy PSF models. In J.-A. Conchello, C. J. Cogswell, & T. Wilson, editor, *Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XIII*, volume 6090, pages 104–114. Proc. SPIE, 2006.

Z. Zhang and M. Braun. Fully 3D active surface models with self-inflation and self-deflation forces. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 85–90, 1997.

H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1):179–195, 1996.

S. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9):884–900, 1996.