Diss. ETH No. 21026

Unsupervised segmentation and shape posterior estimation under Bayesian image models

A dissertation submitted to
ETH Zurich

for the degree of
Doctor of Sciences

presented by
Janick Oliver Cardinale
MSc ETH in Computer Science
born on April 21th, 1983
citizen of Berne, Switzerland

accepted on the recommendation of
Prof. Dr. Ivo F. Sbalzarini
Prof. Dr. Gábor Székely
Prof. Dr. Badrinath Roysam

2013

Declaration of Authorship

This thesis is a presentation of my own original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature. The work was done under the guidance of Professor Ivo F. Sbalzarini at ETH Zurich, Switzerland. I hereby declare that this thesis has not been submitted before to any institution for assessment purposes.

Zurich, January 31, 2013 Janick Cardinale

Acknowledgments

First and foremost, I would like to thank Ivo, my thesis supervisor. Working with Ivo has been very motivating, as he naturally shares his enthusiasm about our work and was always amazingly open, positive, and supportive. Thanks to him I truly enjoyed the last four-and-a-half years. We shared many great moments in the office, getting sushi, on top of Uetliberg, and on the rocks at the shore of the Mediterranean sea. Thank you!

I would also like to thank the MOSAICiens who have made the last four-and-a-half years fantastic. I always, really always, enjoyed entering the CAB building. Special thanks goes to Greg, for sharing his broad knowledge. Greg was always a guide during my PhD and without him my thesis would not be the same. I am also happy to have worked with Jo when starting as a master student. I profited a lot from his mentoring during PhD. In addition to his scientific knowledge, I also really enjoyed his profound skills in analyzing football games. R.R. deserves special mentioning. Rajesh always contributed a great story, irrespective of the context. But most of all, I remember great chats we had in front of CAB or in Hive. I profited a lot Sylvains scientific knowledge and the way he could share it. I enjoyed working with Gong. Gong also has been a great colleague and also his enthusiasm has been inspiring. It was very pleasant to share the office with Birte. I very much appreciated her honesty, calmness, and her sense of humor. I'd like to thank Herr Kollege Ömer, who was always contributing to the good mood in MOSAIC. Christian is the guy who invited

me to share a flat with him, to play football with him, to watch a lot of football... Thank you a lot. All our discussions around the flat were fun and very interesting. Last but not least, Omar, thanks a lot for always having repaired my computer. I shared an office with Omar for over 2 years and it was always enjoyable. Omar is just a great guy - what else can I say?

Tina, thank you so much for all your patience over the last few days. And thank you for the time we had so far. I'm very much looking forward to a great time with you!

Above all, I'd like to thank my family, Marlise, Franco, and Mirko for their unconditional support throughout the last almost 30 years. Without them, I would not have been able to study or make it through my thesis. I am deeply grateful.

Janick

Abstract

Automated image analysis is concerned with extracting quantitative information from images. In this process, image segmentation is a key step. Unsupervised image segmentation partitions an image into a priorly unknown number of "meaningful" regions that are usually intended to correspond to individual objects represented in the image. Because information is lost during image acquisition, multiple solutions are usually possible. Image segmentation hence constitutes an ill-posed problem that requires regularization. Bayesian image models regularize the segmentation problem by including prior knowledge and assigning posterior probabilities to segmentations. In addition, the probability density function of the posterior provides information about the uncertainty or robustness of a particular segmentation.

In this thesis we consider Bayesian image segmentation and uncertainty quantification and provide novel optimization and sampling algorithms for it. We first present a discrete multi-region algorithm to locally optimize the posterior with respect to possible segmentations. In order to jointly estimate the number of regions, we constrain foreground regions to be connected components, amounting to an intuitive topological prior. We present a multi-region competition optimizer over discrete contours for which the posterior is a black-box function. We apply the algorithm to various image models with a focus on images acquired by fluorescence microscopy. We present local and global shape priors to regularize the geometric estimation problem. We consider piecewise smooth and piecewise constant image models with Gaussian and Poisson noise. Furthermore, we jointly segment and deconvolve images by including a blurring kernel in the image-formation model. Due to the local character of the topological prior, the algorithm is competitive, especially for large numbers of regions and in 3D images.

In order to assess and improve the segmentation robustness, we also introduce a novel algorithm to sample from the posterior probability in time and space. This provides a measure of segmentation uncertainty or, more specifically, solution robustness. We present two Markov chain Monte Carlo methods for both discrete and continuous contour representations. The resulting methods approximate the posterior probability density over the high-dimensional space of segmentations. This allows extracting sub-pixel segmentations at specified confidence levels. The present sampling approaches can

also improve solution quality, as they may overcome local optima in the posterior.

Zusammenfassung

Automatisierte Bildanalyse befasst sich mit der Extraktion von quantitativen Informationen aus Bildern. Dabei spielt Bildsegmentierung eine Schlüsselrolle. Bildsegmentierung partitioniert ein Bild in eine vorher unbekannte Anzahl "sinvolle" Regionen, welche mit Objekten im Blid übereinstimmen sollen. Weil während der Bildgebung Information verloren geht, sind oft mehrere Partitionierungen möglich. Bildsegmentierung ist desshalb ein schlecht gestelltes Problem und bedarf Regularisierung. Bayessche Bildmodelle regularisieren das Problem mit einer A-priori-Wahrscheinlichkeit. Zudem ordnen sie Segmentierungen eine A-Posteriori-Wahrscheinlichkeit zu. Diese kann zur Abschätzung der Unsicherheit oder Robustheit einer Lösung verwendet werden.

Diese Arbeit befasst sich mit Segmentierungen und derer Unsicherheitsschätzungen unter Verwendung Bayesscher Bildmodelle. Dazu werden neue Wahrscheinlichkeitsoptimierungs- und Abtastverfahren vorgestellt. Ein neuartiger Optimierungsalgorithmus findet Segmentierungen so, dass die A-Posteriori-Wahrscheinlichkeit lokal maximiert wird. Um gleichzeitig die Anzahl der Regionen der Partitinierung zu ermitteln, werden Regionen an die Bedingung gebunden, dass sie zusammenhängend sind. Dies entspricht einer intuitiven topologischen Annahme. Das Optimierungsverfahren ist gradientenfrei. So können beliebige A-Priori-Wahrscheinlichkeiten, wie zum Beispiel Annahmen über die lokale und globale Gestalt der zu segmentierenden Objekte, als Regularisierung miteinbezogen werden.

Wir betrachten Bildmodelle für stückweise konstante und stückweise glatte Bildsignale sowie Gaussches- und Poissonrauschen. Desweiteren werden Bildfaltungsprozesse in der Bildgebung im Bildmodell berücksichtigt. Der vorgestellte Algorithmus ist insbesondere leistungsstark, wenn das Bild eine grosse Anzahl Regionen enthält. Wir benutzen diese Multiregionssegmentierung vorallem für zwei- und dreidimensionale Bilder aus der Fluoreszenzmikroskopie.

Um die Robustheit einer Segmentierung abzuschätzen und zu verbessern, werden zwei Algorithmen vorgestellt, welche die A-Posteriori-Wahrscheinichkeit mit Segmentierungsvorschlägen in Raum und Zeit derart abtastet, dass zumindest lokal die Wahrscheinlichkeitsfunktion beschreiben und charakterisiert werden kann. Wir stellen zwei *Markov-chain Monte Carlo* Algorithmen vor, um einerseits kontinuierlich und andereseits diskret representierte Objekte im hochdimensionalen Segmentierungsraum abzutasten. Weil mit solchen

stochastischen Methoden, im Vergleich zu lokalen Optimierungen, lokale Maxima überwunden werden können, kann zusätzlich die Segmentierungsgüte im Sinne von höheren A-Posteriori-Wahrscheinlichkeiten verbessert werden.

Contents

Nomenclature

List of Acronyms

BG	background
AR	acceptance rate
CMA-ES	covariance matrix adaptation evolution strategy
CPU	central processing unit
DRS	discrete region sampling
DT	digital topology
EMCCD	electron multiplying charge-coupled device
ER	Endoplasmatic reticulum
FBR	forward-backward ratio
FFT	fast Fourier transform
FG	foreground
GC	graph-cut based PEaRL-like algorithm
GIMH	Gibbs-inspired metropolis Hastings algorithm
GPGPU	general-purpose graphics processing unit
HMC	homogeneous Markov chain
iff	if and only if
ITK	insight toolkit software library
KL	Kullback-Leibler
MAP	maximum-a-posteriori

MCMC	Markov-chain Monte Carlo
MH	Metropolis Hastings
MRF	Markov Random Field
MS	Mumford-Shah
PC	piecewise constant image model
pdf	probability density function
pmf	probability mass function
PS	piecewise smooth image model
SDF	signed-distance function
SNR	signal-to-noise ratio

Greek Characters

α	MH ratio
β	balloon-force coefficient
δ_x	Kronecker delta
ϵ	smoothness hyper-parameter in inner product definition
σ	standard deviation of a Gaussian
Γ	overall contour
$\tilde{\Gamma}$	set of discrete contour points
Γ_i	contour of region i
κ	curvature
λ	wave length
$\tilde{\lambda}$	Poisson distribution parameter
λ	length penalizing hyper parameter
Λ	subdomain of an image
μ_i	intensity mean of region i
ν	intensity gradient penalizing hyper parameter in the MS functional
Ω	image domain
ω	realization of random field
Ω_i	region domain
ϕ	level-set function
$\pi(\cdot)$	target distribution
ϕ	level-set function

ρ	PSF support radius
R_κ	sphere radius for curvature approximation
Σ	covariance matrix
$\hat{\Sigma}$	estimated covariance matrix
σ_i	standard deviation of intensities within region i
θ	region merging threshold
$\boldsymbol{\theta}$	photometric parameter vector
Ξ	finite state space

Latin Characters

A	particle
B	background intensity
$\mathcal{C}$	particle container
$\boldsymbol{c}$	region intensity vector
C_n	set of connected components
d	image dimension
$\mathcal{E}$	energy function
E	scale parameter of Sobolev inner product
F	excess noise factor
f	forbidden label
G	EMCCD gain
G	graph of parent-child relations
H	Heaviside step function
H^n	Hilbert space
I	image
J	model image
K	point-spread function
$\tilde{K}$	convolution kernel for Sobolev kernel approx.
L	label image
L^2	Hilbert space H^0
l'_A	candidate label of particle A
L'	perturbed label image
M	number of regions
m	binary mask (GIMH)

m	maximal connectivity type (8 in 2D, 26 in 3D)
$\boldsymbol{M}$	moment vector
$\mathcal{M}$	particles container
$\bar{n}$	background connectivity type
N	number of particles
n	foreground connectivity type
N_n^k	geodesic neighborhood
N_i	number of discrete points within region i
$\mathcal{N}$	Gaussian pdf
O_i	indicator function for region i
$\mathcal{P}$	set of regular particles
$\mathcal{P}_{\boldsymbol{x}}$	set of regular particles at a particular state $\boldsymbol{x}$
p	probability density function
$\mathcal{P}_f$	set of floating particles
$\mathcal{P}_l$	set of regular particle belonging to region l
q	proposal density or mass function
q_f	off-boundary sampling probability
R	local statistics support radius
r	reference count
$\mathbb{R}$	real numbers
S_x^R	binary spherical mask with radius R with center x
$s_{\boldsymbol{\theta}}$	image generating function
t	time, iteration
T_n	topological number
V	set of vertices
V_C	potential of clique C
$w(\cdot)$	proposal weight function for particles
$W^{n,p}$	Sobolev space
w_A	weight of a particle A
w_i	intensity correction factor
$\bar{X}$	complement of X
$\boldsymbol{x}'$	proposal state
$\boldsymbol{x}_t$	state at time t

X	a binary object, set of binary points
X_s	random variable at time or site s
x_A	position of a particle A
$\mathbb{X}$	Markov random field
Z_i	series of images
$\boldsymbol{z}_{1:t}$	measurement vector

Special symbols

$*$	convolution operator
$\circ$	operator to apply a particle to a state
∇	Nabla operator
O	complexity function, "big O-notation"

CHAPTER

ONE

Introduction

Automated image analysis is a key information source in many disciplines, including biology, medicine, robotics, material sciences, and astronomy. Image analysis is concerned with extracting information from an image. Depending on the nature of this information, image analysis relies on methods covering low-level image-processing tasks, such as denoising, deblurring, inpainting, or zooming, and more high-level tasks, such as image segmentation. Often, image segmentation is a key step in image analysis.

Image segmentation aims at partitioning an image I into "meaningful" regions. Regions in an image are usually defined through their photometric or texture features. Multi-region segmentation then amounts to grouping pixels according to their features. Regions may hence comprise several disconnected sets of pixels, and the number of regions (i.e., the number of feature groups) frequently needs to be imposed, penalized, or learned *a priori*.

Multi-region *image segmentation* partitions a digital image domain $\Omega \subset \mathbb{R}^d$ (here the dimension $d = 2$ or 3) into a background (BG) region Ω_0 and

$(M-1) > 0$ disjoint foreground (FG) regions Ω_i, $i = 1, \ldots, M-1$, bounded by contours or surfaces[1] Γ_i, $i = 1, \ldots, M-1$. In *unsupervised* image segmentation, the number of regions M is not priorly known[2].

General-purpose image segmentation remains challenging. The human visual system outperforms computer vision in most image-segmentation tasks to this day. There are meany reasons for this. A main reason is that the human brain associates any incoming visual signal with previous visual experiences (Danuser, 2011). Such priors include object's typical or expected shape, size, color, texture, irradiance, shading, elasticity, etc. All of these expectations are processed together with the perceived signal and its context. Mimicking this, segmentation algorithms are equipped with prior information about how an image has been recorded and about the expected objects in the image.

Methodology Model-based Bayesian image segmentation is well suited for incorporating prior knowledge. For example, one can provide a simulation model of the image-acquisition process. This model describes the mapping from a real-world scene to a digital image. We call this the *image-formation model*. Image segmentation then amounts to undoing this mapping and recovering the geometry of the objects in the imaged scene. Image segmentation therefore constitutes an *inverse* problem. This inverse problem is often numerically unstable and usually does not have a unique solution. The problem is then *ill-posed*.

Using a Bayesian formulation allows including prior knowledge into the segmentation process. This regularizes the problem in a principled way and possibly renders it well-posed. The Bayesian approach assigns each possible segmentation a *posterior* probability of being correct. The segmentation task then amounts to maximizing this probability, yielding a maximum-a-posteriori estimate of the imaged scene.

[1] The term "contour" is used throughout this thesis to mean either "outline" (2D) or "surface" (3D). Similarly, we use "pixel" to mean either "pixel" (2D) or "voxel" (3D).

[2] This is not to be confused with image segmentation using unsupervised machine-learning methods.

Focus In this thesis, we focus on images acquired with imaging systems for which we can model the image-formation process. The present algorithms are motivated by and specialized for images acquired by fluorescence microscopy. Fluorescence microscopy is a key information source in biological research. Large amounts of 2D images, 3D images, and time-lapse image sequences are acquired every day. Clearly, automated analysis is needed for processing speed and reproducibility. The algorithms presented in this thesis can be applied to a wide range of 2D and 3D images, also beyond fluorescence microscopy.

We consider the posterior probability to be a black-box function. Nevertheless, we introduce in chapter 3 some formulas for commonly used Bayesian models. This involves likelihoods considering piecewise constant and piecewise smooth images models. We also consider the transfer function (impulse-response function) of the image-formation process and discuss energies tailored to images corrupted by Gaussian and Poisson noise. We further show how to include local shape priors, such as the usual contour-length penalizing term, and moment-based global shape priors.

Contributions In unsupervised image segmentation, a fundamental problem is to determine the number of regions M. Many algorithms require M to be fixed or upper-bounded. Strategies for region-number regularization involve region penalization and statistical tests based on region features. The number of regions in the image model hence enters the prior. In order to jointly estimate the number of regions, their photometry, and their geometry, we propose to exploit the intuitive physical property that objects are connected. This amounts to a topological prior. We evolve deformable models such as to locally optimize the posterior density. In order to fulfill topological priors, the topology of the evolving objects needs to be tracked during this optimization procedure. We use concepts from *digital topology* to characterize topological events in discrete object representations (chapter 2). In chapter 4, we present a discrete multi-region competition segmentation algorithm that optimizes posteriors under the topological constraint that regions are connected components. The development of this algorithm, called Region Competition (RC), is the main contribution of chapter 4. RC optimizes a wide range of energies introduced in chapter 3 under the above stated topological prior. For example,

we present several multi-region segmentations using a piecewise constant deconvolving model. This constitutes an extension to previous work on deconvolving active contours (Helmuth and Sbalzarini, 2009). We also present how to efficiently compute Sobolev-gradient approximations using particle–particle interactions, constituting the second contribution of this chapter. The resulting algorithm is particularly competitive for computationally expensive posteriors and for large numbers of regions.

The user of an image-segmentation algorithm remains largely unaware of the quality and robustness of the found solution. While some algorithms provide guarantees or bounds on the final energy of a solution, these are not directly interpretable in terms segmentation quality. Other algorithms use convex relaxations to report globally optimal solutions w.r.t. the objective function. However, such algorithms are only available for a restricted set of image-formation models. We therefore transform the proposed posterior optimizer into a posterior sampler. We evolve the discrete deformable model such that the sampled segmentations approximate the posterior distribution. This allows assessing segmentation robustness. In chapter 5, we present a novel, efficient, particle-based Metropolis-Hastings algorithm, called discrete region sampler (DRS). We prove that the algorithm converges to the correct posterior distribution. Moreover, the algorithm is tunable via biasing the discrete proposal distribution. We demonstrate that the algorithm is unbiased and outperforms a state-of-the-art algorithm by Chang and Fisher (2012) in terms of solution quality, while being competitive w.r.t. computation time.

A discrete explicit representation is not able to represent contour variations smaller than the pixel size. In chapter 6 we therefore use a continuous explicit representation to sample the spatial coordinates and photometries of subcellular structures in time-lapse movies. We use a Bayesian recursive filter, i.e. a particle filter, to track the posterior in this sequential setting. The likelihood of small objects strongly depends on the noise realization. This frequently leads to bad particle-based representations of the posterior. We address these particle-weight degeneracy and sample impoverishment issues with an adaptive Metropolis-Hastings method. The novel adaptation scheme is inspired by the rank-μ update used in the covariance matrix adaptation evolution strategy introduced by Hansen and Ostermeier (1996). We validate the resulting framework using synthetic

data and apply it to microtubule-length tracking in mitotic yeast cells.

CHAPTER

TWO

Preliminaries

The first part of this chapter briefly discusses different forms of shape and contour representations. In the second part we introduce the concept of digital topology (DT), which was originally designed for binary labels. We introduce a specialized form of DT that allows multiple foreground regions.

2.1 Geometry representations

Geometry (shape, object, contour) representation is crucial in automated image segmentation. Segmentation methods and object representations are closely linked through various aspects: First, segmentation algorithms can make use of quantities that are also provided by the respective representation, for example contour curvature. Second, if the representation is limited to shapes of a certain topology, the solution space of the a segmentation algorithm is constrained as well. Third, segmentation algorithms can only generalize to higher dimensions if the geometry representation allows.

An overview of geometry representations can be found in Montagnat et al. (2001). Here, we focus on deformable models. Deformable models have extensively been used for image segmentation since the introduction of *active contours* (Kass et al., 1988).

2.1.1 Deformable models

Deformable models are characterized by a geometry representation and an evolution law (Montagnat et al., 2001). The evolution law acts on the degrees of freedom of the geometric description. This is frequently done such as to optimize an objective function or energy function, see Sec. 3. Deformable models can also be used to represent shape evolution, for example in tracking applications. Thorough reviews on deformable models can be found in (Zhang and Lu, 2004; Montagnat et al., 2001).

Deformable models can be *continuous* or *discrete*. In either case they can be *implicit* (also called geometric models) or *explicit* (also called parametric models) representations. Energetic relations between explicit and implicit representations have been studied by Xu et al. (2000).

2.1.1.1 Continuous deformable models

Continuous deformable models provide direct access to differential quantities such as contour normals, curvature, or higher-order derivatives.

Explicit continuous models Explicit continuous representations are usually parametrized as $\Gamma = s(t), t \in [0, 1[$. Γ is hence the range of a function $s : \mathbb{R} \to \mathbb{R}^d$. The function s is often not known analytically and is approximated using interpolation, frequently with splines. This allows representing open and closed contours. Explicit continuous representations are usually not used for unknown object topologies, as they cannot naturally change the their topology during evolution. A related issue is that contour self-intersections need to be explicitly prevented. Also, generalization to higher dimensions is non trivial.

In order to evolve the contour, spline nodes are moved in direction of the negative energy gradient w.r.t. to the node positions.

Implicit continuous models Implicit continuous representations avoid these issues. They are usually implemented as level sets. In level-set methods (Osher and Sethian, 1988), the *level-set function* $\phi : \mathbb{R}^d \to \mathbb{R}$ represents the contour Γ as its zero-level set $\Gamma = \{x|\phi(x) = 0\}$. Γ is then referred to as *interface*. The FG region is defined as the set $\{\phi(x) < 0\}$, whereas the set $\{\phi(x) > 0\}$ is the BG region. Level-set methods allow efficiently querying spatial coordinates for region information. The contour naturally changes topology as it evolves, and the method generalizes to higher dimensions. However, level-set methods are limited to binary segmentations of closed curves. Multi-region extensions have nevertheless been proposed and are discussed in Sec. 4.4. Also, since Γ is embedded in a higher-dimensional space, the method is computationally involved. In order to relax this computational overhead, narrow-banded versions have been proposed (Adalsteinsson, 1994).

In order to evolve the interface, a partial differential equation (PDE) is numerically integrated. This PDE is usually obtained from the Euler-Lagrange equation w.r.t. ϕ of the energy $\mathcal{E}(H(\phi))$. The interface hence evolves so as to follow the direction of the energy's steepest descent. Such minimization using variational calculus is often used for explicit continuous representations as well. The implicit formulation, however, benefits from more stable numerical integration.

Soft membership functions are related implicit continuous representations. They are used in convex formulations of the binary partitioning problem. It has been shown that for certain energy functionals, thresholding of the membership function yields a global minimum of the original, binary minimization problem (Chan and Esedoglu, 2005). Γ is then implicitly given as $\Gamma = \{x|M(x) = t\}$, with M and t being the soft-membership function the threshold, respectively. Since Γ is represented as an iso-surface of M, M is a level-set function. However, the term "level-set method" is usually not used in the context of soft membership functions.

2.1.1.2 Discrete deformable models

Discrete deformable models are mostly implemented using grids. On the one hand, geometric information is then only available at grid nodes. Therefore, representation accuracy is usually limited to pixel resolution. Furthermore, methods using such representations usually suffer from a lack of differentiability. On the other hand, the discrete representation enables using efficient combinatorial optimization methods, such as graph-cuts (Greig et al., 1989; Boykov et al., 2001). Moreover, the representation allows efficient spatial region querying. The contour, however, is not directly accessible from the grid. Discrete representations thus usually use additional data structures, such as linked lists, to store Γ (Shi and Karl, 2008; Malcolm et al., 2008; Cardinale et al., 2012). A striking feature of discrete deformable models is the possibility for efficient topological control during contour evolution. This is achieved using DT as discussed in the following section.

Explicit discrete models Explicit discrete representations directly store region labels at grid nodes. Grid nodes usually coincide with the pixels. Such representations allow encoding multiple regions effortlessly. Furthermore they allow efficiently querying spatial coordinates for region information. However, this representation does not provide direct access to the contour.

In order to evolve the contour, grid nodes change their label such as to decrease the energy.

Implicit discrete models Discretized versions of the implicit level-set method have been proposed by Shi and Karl (2008) in order to accelerate the method. They proposed to quantize the level-set function to a finite set of values and use linked lists for fast access to the interface. Using these data structures they performed real-time tracking using implicit shape representations.

Before evolving the contour, the energy gradient is sampled at pixels on both sides of the interface. The level-set function then changes sign at

pixels with negative energy gradient.

2.2 Digital topology

"Topology is a major area of mathematics concerned with the most basic properties of space, such as *connectedness*"[1]. Digital topology (or *discrete topology*) is concerned with topological properties of discrete spaces, in particular of grids. For binary images, i.e., images with one FG and one BG region, the concept of digital topology allows characterizing topological properties of discrete points. These characteristics enable detecting region genus changes when changing the label of a discrete point (Bertrand, 1994; Lamy, 2007; Han et al., 2003; Ségonne, 2005).

We first briefly introduce the notions of *connectivity*, *geodesic neighborhoods*, and *topological numbers*. For more details on these topics we refer to Bertrand (1994); Lamy (2007); Han et al. (2003). We adopt the notation and definitions from Bertrand (1994); Bertrand et al. (1997); Ségonne (2005). We then extend these concepts to multiple FG regions.

2.2.1 Connectivity

Jordan's curve theorem states a fundamental property of closed curves in an embedding space. We define regions in a discrete space such as to satisfy this property. The theorem states that a simple closed curve (without self-intersections) partitions the space into two connected components. The theorem has been proven by Jordan (1887) in 2D and has been generalized to higher dimensions by Lebesque and Brouwer in 1911. Their work is spread over different papers, an overview can be found in (Dieudonné, 2009), from which we adapt the theorem:

Theorem 1 (Jordan-Brouwer Theorem)**.** *Given a subset* Γ *of* $\mathbb{R}^d$ *homeomorphic to the boundary of a topological* n*-ball (viz. homeomorphic to an* $(n-1)$*-sphere), then*

[1] from http://en.wikipedia.org/wiki/Topology, 18.01.2013

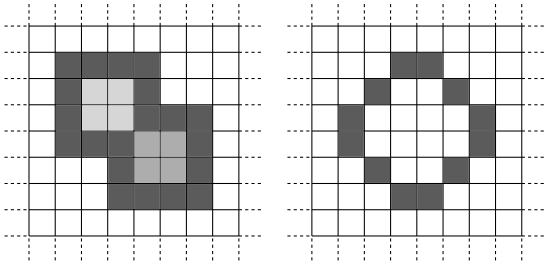


Figure 2.1: Examples of topological paradoxes in 2D. (a) If the FG and BG regions both have connectivity $n = 4$, a simple closed curve (dark gray) partitions the space into 3 connected components. (b) If both regions have a connectivity of $n = 8$, the closed curve does not partition the space at all.

- *the complement* $\mathbb{R}^d - \Gamma$ *has exactly two connected components.*
- Γ *is the boundary of every connected component of* $\mathbb{R}^d - \Gamma$.

We consider discrete spaces that respect the above theorem. Let the FG X be a set of discrete points x and the BG its complement $\bar{X}$, such that $X \cap \bar{X} = \emptyset$ and $X \cup \bar{X} = \Omega$. Both FG and BG have a certain connectivity type. In 2D, two points are 4-connected if they share an edge and 8-connected if they share a corner. In 3D, two points are 6-connected if they share a face, 18-connected if they share an edge, and 26-connected if they share a corner. In order to avoid topological paradoxes, only the following combinations of FG (n) and BG ($\bar{n}$) connectivities are admissible according to Jordan's theorem: $(n, \bar{n}) \in \{(4, 8), (8, 4), (6, 26), (26, 6), (6, 18), (18, 6)\}$. Figure 2.1 shows two examples of a topological paradox when using incompatible connectivity pairs in 2D.

2.2.2 Topological numbers

Based on the chosen connectivity we can define discrete geodesic neighborhoods. The n-neighborhood $N_n(x)$ is the set of n-connected points

adjacent to point x.

Definition 1. *Let $X \subset \Omega$. The geodesic neighborhood of order k of a point $x \in X$ is the set $N_n^k(x, X)$ defined recursively by:*

$$\begin{cases} N_n^1(x, X) = \{N_n(x) \backslash x\} \cap X \\ N_n^k(x, X) = N_m^1(x, X) \cap \bigcup \{N_n(y),\, y \in N_n^{k-1}(x, X)\} \end{cases}$$

with $m = 8$ in 2D and $m = 26$ in 3D.

Intuitively, the geodesic neighborhood $N_n^k(x, X)$ comprises all points $y \in N_m^1(x, X) \backslash x$ that are n-connected to x along a path that is not longer than k (Bertrand, 1994).

From this, a topological number can be defined as the number of n-connected components $\#C_n(\cdot)$ within a geodesic neighborhood:

Definition 2. *The topological numbers $T_n(x, X)$ relative to the point x and the set X are:*

$$\begin{aligned} T_4(x, X) &= \#C_4(N_4^2(x, X)) \\ T_8(x, X) &= \#C_8(N_8^1(x, X)) \\ T_6(x, X) &= \#C_6(N_6^2(x, X)) \\ T_{6^+}(x, X) &= \#C_6(N_6^3(x, X)) \\ T_{18}(x, X) &= \#C_{18}(N_{18}^2(x, X)) \\ T_{26}(x, X) &= \#C_{26}(N_{26}^1(x, X)) \,. \end{aligned}$$

The notation $n = 6^+$ indicates that the dual connectivity $\bar{n}$ is 18, whereas the dual connectivity for $n = 6$ is 26.

Topological numbers are an efficient tool to characterize points in binary images. They can be computed from purely local information. For example, if $T_n(x, X) = T_{\bar{n}}(x, \bar{X}) = 1$, we know that changing the region label of point x does not change the genus of neither the FG, nor the BG. All points for which this is true are called *simple points*. Figure 2.2 and Tab. 2.1 visualize and summarize the topological characteristics of points that can distinguished using these topological numbers.

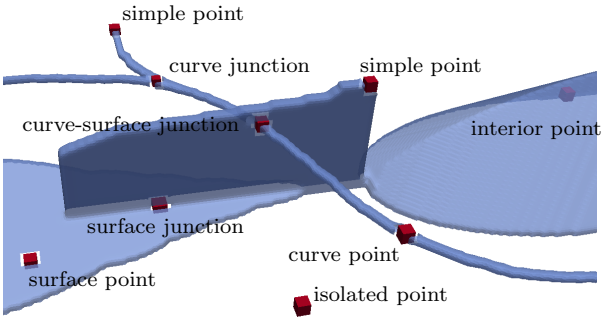


Figure 2.2: A 3D space partitioned into FG (blue) and BG (white). The figure illustrates topological characteristics of different discrete points (red).

$T_n(x, X)$	$T_{\bar{n}}(x, \bar{X})$	Topological type of x
0	1	isolated point
1	0	interior point
1	1	simple point
2	1	curve point
> 2	1	curve junction
1	2	surface point
1	> 2	surface junction
> 1	> 1	curve surface junction

Table 2.1: Topological characterization based on topological numbers. Figure 2.2 illustrates the different point types.

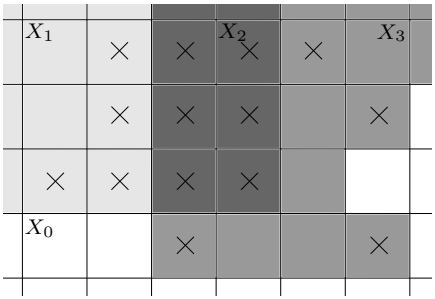


Figure 2.3: Illustration of FG-simple points in 2D. X_1, X_2, and X_3 are FG regions (gray) with connectivity type $n = 4$. X_0 is the BG region (white). Marked points are FG simple.

2.2.3 Multi-region extension

The concept of digital topology is limited to binary images. This follows from Jordan's theorem. If three regions share the embedded space, two of them need to be of the same connectivity type. Consequently, topological paradoxes as illustrated in Fig. 2.1 might arise.

Nevertheless, topological numbers can be used to classify points also in a multi-region framework by splitting the FG $X = \bigcup_{i=1}^{M-1} X_i$ into multiple disjoint sub-regions X_i. The BG region remains a single set $\bar{X} = X_0$. We therefore introduce the multi-region extension of a simple point in a straightforward manner:

Definition 3. *A point x is* foreground simple *(FG-simple) iff $T_n(x, X_i) = T_{\bar{n}}(x, \bar{X}_i) = 1$ for all $i > 0$.*

Intuitively, $T_n(x, X_i)$ is the topological number when considering all other regions $X_j, j \neq i$ to be part of the BG. Changing the region label of a FG-simple point does not change the genus of *any* FG region. Figure 2.3 illustrates FG-simple points in 2D.

This extended definition of FG simplicity allows distinguishing different topological events on the FG regions. We use it to explicitly avoid topological paradoxes. The methods presented in Secs. 4 and 5 therefore use

of this notion of foreground-simple points.

CHAPTER

THREE

Bayesian image models

In image segmentation, a Bayesian model is determined by a likelihood $p(I|\Gamma, \boldsymbol{\theta})$ and a prior $p(\Gamma)$, where I is the image data and Γ the segmentation contour. $\boldsymbol{\theta}$ contains photometric parameters such as region intensities. The posterior pdf is obtained using Bayes' formula

$$p(\Gamma, \boldsymbol{\theta}|I) = \frac{p(I|\Gamma, \boldsymbol{\theta}) \cdot p(\Gamma, \boldsymbol{\theta})}{p(I)}. \tag{3.1}$$

The likelihood expresses how likely it is to observe the measured image I given a certain segmentation Γ and parameters $\boldsymbol{\theta}$. The likelihood therefore formalizes the image formation model. The image-formation model describes the mapping between a ground-truth object state $(\Gamma_{\mathrm{GT}}, \boldsymbol{\theta})$ and an expected image J conditioned on that state. For many image-formation models, also for all the models discussed in this work, $\boldsymbol{\theta}$ is determined by the model and I. For example, in many popular models the image intensity of a region i is equal to the mean intensity of the area enclosed by Γ_i in I. We therefore mostly omit $\boldsymbol{\theta}$ throughout this work.

Generating J from Γ is called the *forward problem*. Solving the forward problem is usually straightforward. In microscopy, for example, the forward model is often linear (see Sec. 3.1.2). The image-formation models considered in this work are of the form

$$I = \underbrace{(s_{\boldsymbol{\theta}}(\Gamma) * K)}_{J} \odot W. \tag{3.2}$$

K is the impulse-response function of the imaging equipment and $*$ is the convolution operator. The function $s_{\boldsymbol{\theta}}$ is called the image generating function. It assigns an intensity to each pixel according to the image-formation model. $W \in \Omega$ is the noise image of independently distributed random variables, and $\odot$ denotes either an element-wise addition or an element-wise multiplication for additive or multiplicative noise, respectively. Finding the contour Γ and the associated photometric values $\boldsymbol{\theta}$ from given I,K, and $s_{\boldsymbol{\theta}}$ constitutes an *inverse problem*.

Prior terms measure how likely a certain segmentation Γ is. The most popular prior term, proposed by Mumford and Shah (1989), penalizes the Euclidean contour length $|\Gamma|$, leading to smooth region boundaries. Other priors may include global shape characteristics and penalize deviations of the segmented shape from a template shape.

Energy differences

In this work, two particular applications of Bayesian models are investigated: image segmentation (chapter 4) and shape sampling (chapters 5, 6).

In model-based image segmentation, the optimization problem is often formulated as a maximum-a-posteriori (MAP) problem

$$\max_{\Gamma} p(\Gamma|I) \tag{3.3}$$

for Γ given an image I. In order to find Γ, the anti-logarithm of the posterior, called the *energy function* $\mathcal{E}$, is minimized. The logarithm changes the product of the likelihood and the prior to a sum of energy terms. The energies corresponding to the likelihood and the prior are called *external*

and *internal* energy, respectively. Plenty of minimization methods for the total energy $\mathcal{E}$ have been presented in the literature. The method of choice depends on the characteristics of $\mathcal{E}$. Gradient-based local optimization of $\mathcal{E}$ is a popular approach for non-convex $\mathcal{E}$. In a discrete setting, the energy gradient becomes an energy difference $\Delta\mathcal{E}$.

In shape-sampling applications, we are interested in characteristics of the posterior probability distribution $p(\Gamma|I)$ from which we might, for example, assess the robustness of a particular segmentation. From Markov chain theory (see chapter 5.1.2) we know that, in order to reconstruct this probability density, we (only) need to be able to evaluate posterior ratios $\frac{p(\Gamma'|I)}{p(\Gamma|I)} = \exp(-\Delta\mathcal{E})$ for two given contours Γ and Γ'.

In both applications, the quantity of interest is the energy difference $\Delta\mathcal{E}$ when deforming Γ to Γ'. In our case of discrete geometry models, the smallest perturbation to Γ is a flip of a discrete point from one region to another. This chapter introduces various external and internal energy functions with a focus on efficient discrete energy-difference evaluations. We assume the noise to be realized independently for each pixel. Using Bayes' formula (3.1) we decompose $\Delta\mathcal{E}$ as

$$\begin{aligned}\Delta\mathcal{E} = -\log\left(\frac{p(\Gamma'|I)}{p(\Gamma|I)}\right) = -\sum_{i=0}^{M-1}\Bigg(&\log\prod_{x\in\Omega_i'} p\left(I(x)|J'(x)\right)\cdot p(\Gamma') \\ &-\log\prod_{x\in\Omega_i} p\left(I(x)|J(x)\right)\cdot p(\Gamma)\Bigg),\end{aligned} \tag{3.4}$$

where the images J and J' are computed images using Eq. (3.2). Ω_i and Ω_i' are the regions enclosed by Γ_i and Γ_i', respectively. M is the total number of regions.

This chapter is organized in two sections. The first one considers various likelihood terms and the associated external energies. In the second section we discuss prior terms and the associated internal energies. We also present how to compute discrete energy differences for local and global shape features.

3.1 Likelihood terms

The likelihood $p(I|J)$ evaluates how likely it is to observe I given a model image J. The corresponding external energy hence measures data fidelity. Data fidelity is often measured using a certain norm. This norm depends on the underlying statistical noise assumption. A well-known example is that a Gaussian noise model leads to a L^2-norm-based energy (Brox and Cremers, 2007). However, energy functions without an explicit link to an underlying statistical model have also been introduced in the literature (Kass et al., 1988; Caselles et al., 1997).

In the following, we discuss the main ingredients of the likelihood terms w.r.t. the image formation model in Eq. (3.2).

Photometric region model Mumford and Shah (1989) argued that, following some basic observations in 2D photography, *"the segmentation problem in computer vision consists in computing a decomposition [...] of the image domain such that (a) the image varies smoothly and/or slowly within each region, and (b) the image varies discontinuously and/or rapidly across most of the boundary* Γ *between different regions."*. The segmentation task is thus formalized as finding piecewise smooth function J, differentiable within regions, that optimally approximates the image I. This formulation leads to the well-studied Mumford-Shah (MS) energy functional

$$\mathcal{E}(J,\Gamma) = \int_\Omega (J-I)^2 \mathrm{d}\Omega + \nu \int_{\Omega-\Gamma} \|\nabla J\|^2 \mathrm{d}\Omega + \lambda|\Gamma| \tag{3.5}$$

The first term is the external energy which is responsible for data fidelity. The third term is the prior and will be discussed in Sec. 3.2. The second term penalizes variations in J within regions and determines the smoothness of J. With $\nu \to \infty$ the model degenerates to the piecewise constant limit (often referred to as the *cartoon limit*). This functional is notoriously difficult to globally minimize (Ambrosio and Tortorelli, 1990; Pock et al., 2009). Simplified functionals are therefore often more appropriate in practice. We distinguish between *piecewise smooth* (PS) and *piecewise constant* (PC) models. Although PC models are a limit case of PS models, it is useful to treat them separately because the PC assumptions greatly

simplifies and accelerates energy computation.

Noise We focus on likelihood functions for which the underlying statistical noise models is from the exponential family. A systematic overview of likelihoods of this type, together with an information theoretic interpretation based on Bregman distances, is given by Paul et al. (2013). In this work, we focus on Gaussian and Poisson noise. Gaussian noise is a generic and popular model. The MS functional for example assumes Gaussian noise with fixed variance. Gaussian noise sources are for example electron-multiplying cameras used in low-light microscopy (see chapter 6). Poisson noise is often found in low-intensity applications when photon-shot noise is the dominant noise source. Important applications include light microscopy, emission tomography, or single-photon emission computed tomography.

Blur The process of image acquisition maps the light irradiance of a real-world scene to a scalar field in Ω. This mapping is often modeled by its impulse-response function, the *point-spread function* (PSF). For microscopes and telescopes, this mapping is almost perfectly linear. This is because light corresponding to high spatial frequencies of the aperture misses the objective lens. The acquisition system is hence a low-pass filter. A detailed description of image formation in confocal microscopes can be found in Helmuth (2010). Image formation in these cases can be modeled as a convolution of the real-world scene with the PSF. The convolution operation in Eq. (3.2) with point-spread function K hence models blurring processes during image acquisition.

Frequently, one is interested in reconstructing the shapes of the imaged real-world objects from the observed image, attempting to undo the PSF blur. In the presence of noise, this inverse problem cannot be solved directly. The process of solving a regularized version of this inverse problem is often referred to as *deconvolution*, and multiple regularization methods are available (Vogel, 2002; Sibarita, 2005; Hansen et al., 2006). In deconvolving active contours (Helmuth and Sbalzarini, 2009) the actual inverse problem never needs to be computed. Forward simulations of image formation are used to evaluate the model. This has enabled highly accurate and

robust reconstructions of small, diffraction-limited objects in biological cells using fluorescence microscopy (Helmuth et al., 2009). Here we extend the concept of deconvolving active contours to higher-dimensional images and to multiple regions.

3.1.1 PIECEWISE CONSTANT MODELS

We discuss mathematical and implementation details of the calculation of $\Delta\mathcal{E}$ for various combinations noise models and the region-intensity models with and without a convolution operation.

3.1.1.1 piecewise constant Gaussian-noise model

The piecewise constant model with Gaussian noise leads to a model image J with intensities equal to the region intensity means (Mumford and Shah, 1989). Using the Gaussian probability density in Eq. (3.4). The energy difference simplifies to

$$\Delta\mathcal{E}^{\mathrm{PC}}_{\mathrm{Gauss}} = -\sum_{i=0}^{M-1}\left(N'_i \log\frac{1}{\sqrt{2\pi}\sigma'_i} - N_i \log\frac{1}{\sqrt{2\pi}\sigma_i} - \frac{N'_i}{2} + \frac{N_i}{2}\right). \quad (3.6)$$

N_i is the number of pixels in region i and σ_i the standard deviation of the intensities in region i. The PC assumption leads to an expression that only depends on region variances and sizes.

Fixed Variance The popular energy functional of Chan and Vese (2001) is a special case the above model when fixing the standard deviations to $\sigma_i = \sqrt{\frac{1}{2}}$. Doing so, the first and second terms in Eq. (3.6) vanish, as they sum to 0. For a piecewise constant fixed-variance Gaussian noise model, Eq. (3.4) simplifies to

$$\Delta\mathcal{E}^{\mathrm{PC}}_{\mathrm{LS}} = -\sum_{i=0}^{M-1}\left(\sum_{x\in\Omega_i}(I(x)-\mu_i)^2 - \sum_{x\in\Omega'_i}(I(x)-\mu'_i)^2\right), \quad (3.7)$$

where μ_i is the mean intensity in region i. When only slightly perturbing Γ, most of the terms vanish. If only one pixel at position x is changing from region i to region j, Eq. (3.7) further simplifies to $(I(x) - \mu_i')^2 - (I(x) - \mu_i)^2 + (I(x) - \mu_j')^2 - (I(x) - \mu_j)^2$.

The energy differences in Eqs. (3.6) and (3.7) can be computed in constant time when the sum of intensity squares $\sum_{x\in\Omega_i} I(x)^2$, the sum of intensities $\sum_{x\in\Omega_i} I(x)$, and the number of points per region are stored.

3.1.1.2 piecewise constant Poisson-noise model

The Poisson distribution has one parameter $\tilde{\lambda}$ equal to the distribution's mean. We therefore set the parameter for each region equal to its intensity mean, i.e., $\tilde{\lambda} = \mu_i$. The pixel-wise likelihood for location x to belong to region i is then the Poisson pdf value for $I(x)$ with $\tilde{\lambda} = J(x)$. If only one discrete point changes its region label, only two regions are affected. The external-energy difference for a one-point change from region i to j using a PC model with Poisson noise is:

$$\Delta\mathcal{E}^{\mathrm{PC}}_{\mathrm{Poisson}}(x) = -\sum_{k=\{i,j\}} \left(\log \mu_k' \sum_{y\in\Omega_k'} I(y) - N_k'\mu_k' - \log \mu_k \sum_{y\in\Omega_k} I(y) + N_k\mu_k \right). \tag{3.8}$$

As in the Gaussian case, energy differences can efficiently be computed if the sum of intensities $\sum_{x\in\Omega_i} I(x)$ and the number of points are stored for each region.

3.1.2 piecewise constant deconvolution model

Regardless of the noise model, region-wise constant photometric parameters are not equal to the region intensity means anymore when including a convolution (compare sections 3.1.1.2, 3.1.1.1). This is due to the inten-

sity flux across region boundaries caused by the convolution. The region intensity values $\boldsymbol{c} = \{c_i\}, i = 0 \cdots (M-1)$, hence need to be estimated separately. In the following we assume $\boldsymbol{c}$ to be known and we show the intensity estimation in Sec. 3.1.2.3.

If the PSF has finite support and only one discrete point changes region, a naïve evaluation of the energy difference requires two local convolutions around x. This can be avoided by explicitly storing the model image

$$J = c_0 + \left(\sum_{i=1}^{M-1} c_i O_i \right) * K. \tag{3.9}$$

J is re-computed using a fast Fourier transform (FFT) whenever the photometric parameters are updated. When a particle at position x changes from region i to region j, the binary indicator O_i is updated to $O_i - \delta_x$ and O_j becomes $O_j + \delta_x$, where δ_x is the Kronecker delta (discrete unit impulse) at x. Due to the linearity of the convolution operator, we have:

$$\begin{aligned} &\left(\sum_{k=1}^{M-1} O_k c_k + \delta_x c_j - \delta_x c_i \right) * K \\ &= \underbrace{\left(\sum_{k=1}^{M-1} O_k c_k \right) * K}_{J - c_0} + (\delta_x c_j - \delta_x c_i) * K. \end{aligned} \tag{3.10}$$

The first term on the right-hand side corresponds to the pre-computed model image J without the background intensity. The second term is a scaled, discretized PSF mask. The model image J is then updated as $J' = J + \delta_x * K \cdot \frac{c_j}{c_i}$. Hence, Eq. 3.10 allows computing $\Delta\mathcal{E}^{\text{dec}}$ as a *local* operation: We iterate through a spherical local window (centered at x) with a radius ρ equal to the radius of PSF support $S^\rho(x)$. At each pixel y in the window we calculate $J'(y)$. Also, we fix the photometric parameters $\boldsymbol{c}$ when computing the energy difference for one discrete point; see Sec. 3.1.2.4 for further discussion.

3.1.2.1 Gaussian noise

With the convolution in Eq. (3.2) two complications arise: First, the convolution changes the likelihood for all pixels within the PSF support. This alters the region statistics for all regions in the support of the PSF. Second, the noise acts on $J(x)$, rather than on the region intensity means. This renders the photometric problem more difficult. The region-wise noise variance, however, remains constant within each region:

$$\sigma_i^2 = \frac{1}{N_i - 1} \sum_{x \in \Omega_i} (I(x) - J(x))^2. \tag{3.11}$$

From Eq. (3.4) we derive the energy difference for a region change of point x:

$$\begin{aligned} \Delta\mathcal{E}_{\text{Gauss}}^{\text{PC,dec}}(x) = - \sum_{i=0}^{M-1} \Bigg[& N_i' \log \frac{1}{\sqrt{2\pi}\sigma_i'} - N_i \log \frac{1}{\sqrt{2\pi}\sigma_i} \\ & + \sum_{y \in S_x^\rho \cap \Omega_i} \left(\frac{(I(y) - J'(y))^2}{2\sigma_i'^2} - \frac{(I(y) - J(y))^2}{2\sigma_i^2} \right) \Bigg]. \end{aligned} \tag{3.12}$$

We first iterate I, J, and L within the PSF support and compute the region-wise squared residuals $r_i = (I(y) - J(y))^2|_{L=i}$ and $r_i' = (I(y) - J'(y))^2|_{L=i}$. In order to efficiently evaluate the energy differences using local operations, we also store σ_i^2.

The σ_i need to be recomputed from scratch whenever the model image J is regenerated, which is usually the case after intensity re-estimating $\boldsymbol{c}$ as discussed in Sec. 3.1.2.3.

Fixed variance When fixing the variance to $\sigma^2 = 0.5$, we obtain a multi-region deconvolving Chan-Vese energy. The corresponding energy difference reads:

$$\Delta\mathcal{E}_{\text{LS}}^{\text{PC,dec}}(x) = - \sum_{y \in S_x^\rho} \left((I(y) - J(y))^2 - (I(y) - J'(y))^2 \right) \tag{3.13}$$

Compared to the version without deconvolution, the computational domain extends to the support of the PSF.

3.1.2.2 Poisson noise

As was the case for Gaussian case, the likelihood has to be re-evaluated within the PSF support. For each pixel, we set the distribution parameter $\tilde{\lambda}$ to the expectation value which depends on space, i.e. $\tilde{\lambda}(x) = J(x)$. Using the Poisson pdf with $\tilde{\lambda}(x) = J(x)$ in Eq. (3.4) leads to

$$\Delta\mathcal{E}^{\mathrm{PC,dec}}_{\mathrm{Poisson}}(x) = -\sum_{y\in S^{\rho}_{x}} \left(I(y)\log J'(y) - J'(y) - I(y)\log J(y) + J(y)\right). \tag{3.14}$$

Since the Poisson distribution has only one free parameter, energy-difference computation is significantly easier than in the Gaussian case.

3.1.2.3 Intensity estimation

Estimating the region intensities $\boldsymbol{c}$ requires special attention in deconvolving image model, especially for objects that are small compared to the width of the PSF. Intensity estimation with asymptotic confidence intervals for two piecewise constant regions has been done using Fisher scoring (Paul et al., 2013). Here, we estimate $\boldsymbol{c}$ by fitting the model image J to the data image I. This has the advantage that it can be done for any number of region and potentially also for piecewise smooth models. We do this by formulating the problem as a 2D linear regression for each FG region. Considering the pixels within a region i as the data points, we find an affine transform of the set $A_i = \{I(x) : |L(x)| = i\}$ to the set $B_i = \{J(x) : |L(x)| = i\}$. Figure 3.1 illustrates region-wise intensity correction in a one-dimensional example. We minimize $\sum_{i=0}^{M-1}(\boldsymbol{w}\cdot[1,\, A_i]^T - B_i)^2$ with respect to $\boldsymbol{w}$. The regression coefficient w_0 then serves as an estimate for the BG intensity, while w_1 is used as a correction factor for the FG intensity of the corresponding region, hence $c_i \leftarrow w_1 c_i$.

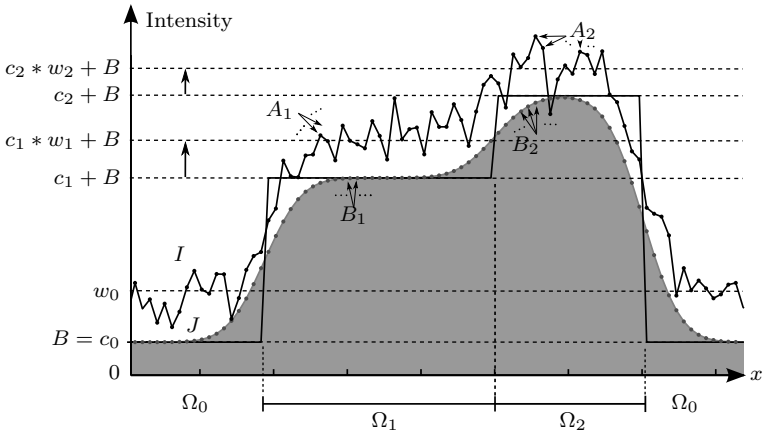


Figure 3.1: 1D example illustrating the region intensity estimation. The geometry is fixed, and, the domain contains 3 regions, 2 of which are foreground regions. In this example, the inital estimates c_0, c_1, and c_2 are too low. Based on the region-wise linear correlation between the model image J and the data image I, the individual correction factors w_i are estimated using regression.

In confocal microscopy images the background region is often very large and its intensity I_0 can therefore be estimated separately. The stability of the intensity estimates of small regions can be increased by exploiting robust estimate of the background intensity. We find that a more robust heuristic alternative to linear regression is to estimate the correction factors w_i by the mean or median of the set $\{(A_i(x) - I_0)/(B_i(x) - I_0)\}$. Figures 3.2 and 3.3 show the relative intensity estimation errors for multivariate linear regression and for this heuristic. Experiments are performed for SNRs between 3 and 20. For each SNR we assess the ℓ^1 estimation error for an increasing number of wrongly segmented pixels. For Fig. 3.2 we convolve an object of size 10 pixel with a Gaussian PSF of width $\sigma_{\mathrm{PSF}} = 2$. The background intensity is known exactly. In Fig. 3.3 we use the same PSF, but the foreground region only contains 4 pixels. Moreover, we provide an initial background intensity that is 20% too high. In both cases the heuristic using the median gives the best estimates for any Γ-SNR combination. The estimation breaks down when more than half of the foreground region are assigned to the background region. Moreover, the heuristic shows to be robust w.r.t. inaccurately estimated background intensties.

3.1.2.4 Joint photometric and geometric problem in PC deconvolving energies

Efficient evaluation $\Delta\mathcal{E}^{\mathrm{PC,dec}}$ relies on using fixed intensities $\boldsymbol{c}$. The energy differences in Eqs. (3.12) and (3.14) therefore depend on the frequency of photometric updates.This amounts to performing alternating minimization: For all discrete points we evaluate $\Delta\mathcal{E}$ after updating of $\boldsymbol{c}$ and J. Then the active contour moves. In worst case this may lead contour oscillations. It is not clear if this process is guaranteed to converge. We have the following options:

- We interpret $\boldsymbol{c}$ as a function of Γ, i.e., the photometric parameters are determined by Γ and I. Hence, we update J and $\boldsymbol{c}$ in every iteration. This has the advantage that energy-difference calculations respect the intensity changes of regions when changing the label of pixel. This solution, however, is computationally expensive. Delayed

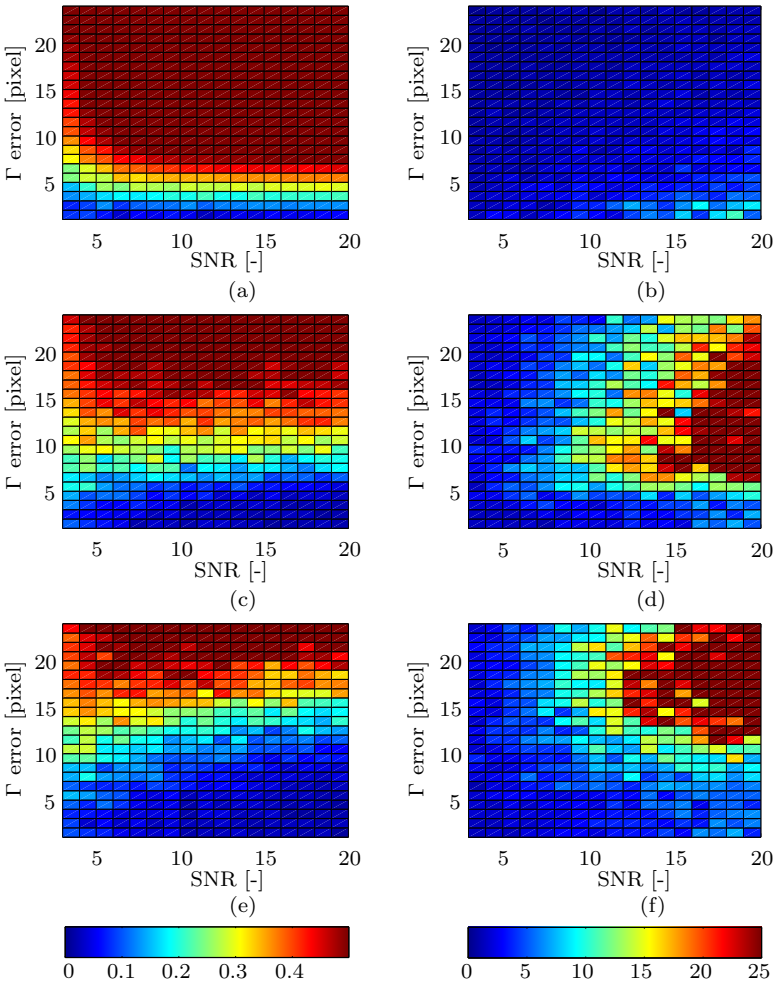


Figure 3.2: Relative ℓ^1 error (left row) and standard deviation (right row) of intensity estimates for perfectly known background intensity and 10 pixel object size. (a),(b) Linear regression results. (c),(d) Mean-of-$\{(A_i(x) - I_0)/(B_i(x) - I_0)\}$ heuristic. (e),(f) Median-of-$\{(A_i(x) - I_0)/(B_i(x) - I_0)\}$ heuristic.

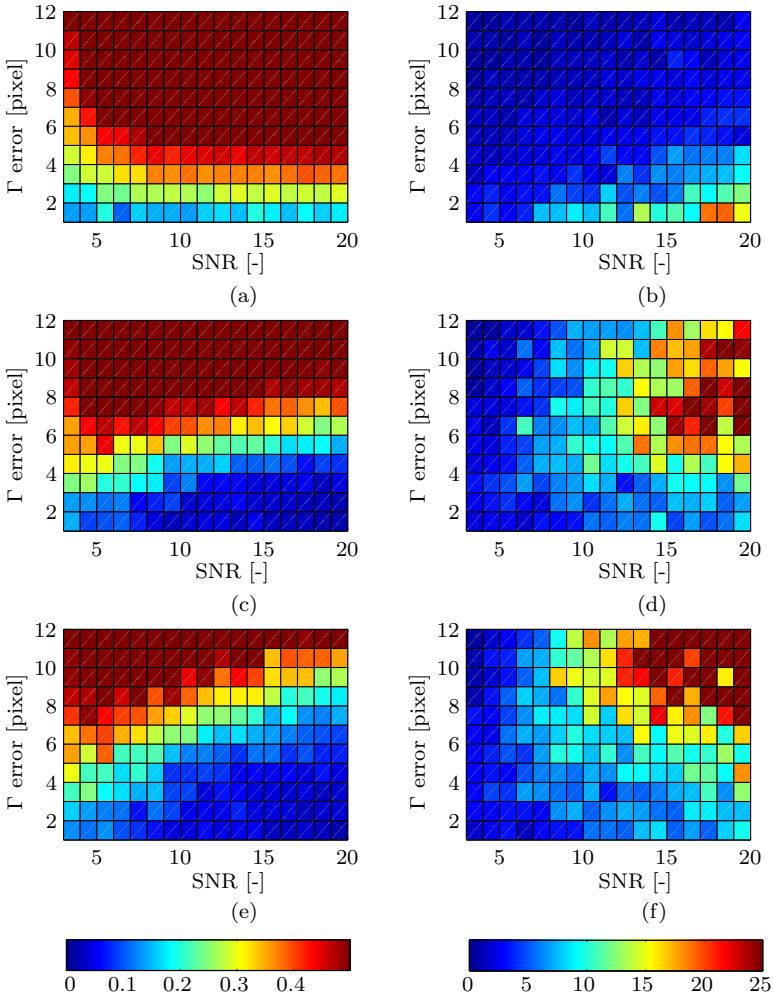


Figure 3.3: Relative ℓ^1 error (left row) and standard deviation (right row) of intensity estimates for 20% overestimated background intensity and 4 pixel object size. (a),(b) Linear Regression results. (c),(d) Mean-of-$\{(A_i(x) - I_0)/(B_i(x) - I_0)\}$ heuristic. (e),(f) Median-of-$\{(A_i(x) - I_0)/(B_i(x) - I_0)\}$ heuristic.

update of J and $\boldsymbol{c}$ after an empirical number of iterations works well in practice. The approximation error depends on the sensitivity of $\boldsymbol{c}$ w.r.t. Γ.

- We interpret the PC deconvolution problem as a joint problem of intensity and geometry estimates. Hence, we extend the state space with $\boldsymbol{c}$ and consider the joint space.

- We fix $\boldsymbol{c}$.

Options one and two involve the photometric estimates in the posterior and are therefore favorable. Here we use option one with delayed updates.

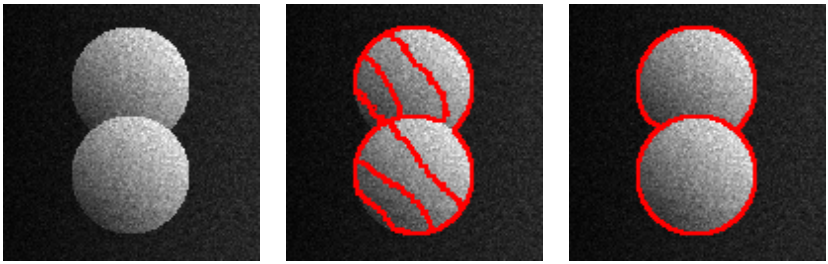
3.1.2.5 Complexity

After updating $\boldsymbol{c}$, the entire model image J is re-computed from its definition. This involves a FFT, which has computational complexity in $O(|\Omega| \log |\Omega|)$, where $|\Omega|$ is the total number of pixels in the image. Unlike for non-deconvolving energy functionals, the computational cost here depends on the size of the image.

3.1.3 PIECEWISE SMOOTH MODELS

PC models may lead to undesired partitioning of regions that smoothly vary in intensity. Figure 3.4 illustrates segmentations using a PC and a PS model on an image with a regions with non-zero intensity gradient. Varying intensities within objects are common in practice.

Equation (3.5) has a global character. Whenever the segmentation locally changes, the approximation J changes globally. The related optimization problem is highly non-convex (Mumford and Shah, 1989; Pock et al., 2009). We argue that local piecewise smooth models are better adapted to optimization algorithms. We use piecewise smooth models based on local statistics. The local statistics approach has previously been used (Zhu and Yuille, 1996; Li et al., 2008; Lankton and Tannenbaum, 2008). Brox



(a) Data *(b) PC model* *(c) PS model*

Figure 3.4: Comparison of a segmentation result using a piecewise constant and a piecewise smooth image model. (a) The data image containing regions with smooth intensity variations. (b) PC model solution. The regions found are oriented perpendicularly to the intensity gradient. The result in that case is sensitive to the number of initial regions. (c) Result using the PS model with energy $\mathcal{E}_{LS}^{PS}$. Both results have been generated using the algorithm presented in chapter 4.

and Cremers (2007) have shown that the MS functional is a first-order approximation a Bayesian model where region statistics are computed almost locally. In the following, we outline this relationship; for details we refer to Brox and Cremers (2007). Their argumentation is based on a result by Nielsen et al. (1997), who have shown that filtering I with filter h of frequencies $\hat{h} = 1/(1 + \sum_k^\infty \alpha_k \omega^{2k})$ yields a minimizer of

$$\mathcal{E}(J) = \int_{\Omega_i} \left((J - I)^2 + \sum_{k=1}^{\infty} \alpha_k \left(\frac{\mathrm{d}^k J}{\mathrm{d}x^k} \right)^2 \right) \mathrm{d}x. \tag{3.15}$$

We focus on a particular case of h. When parametrizing h with $\alpha_k = \frac{\lambda^k}{k!}$ we retrieve a Gaussian filter with standard deviation $\sigma = \sqrt{2\lambda}$. Exploiting Eq. (3.15), we minimize $\mathcal{E}$ by setting $J(x)$ to a region-wise Gaussian-weighted mean, i.e.,

$$J(x) = \mu_i(x) = \frac{\int_{\Omega_i} \mathcal{N}_\sigma (y - x)\, I(y) \mathrm{d}y}{\int_{\Omega_i} \mathcal{N}_\sigma (y - x)\, \mathrm{d}y}. \tag{3.16}$$

$\mathcal{N}_\sigma$ is the Gaussian pdf with standard-deviation σ. The integral in

Eq. (3.15) is over regions Ω_i with natural boundary conditions. Neglecting terms of order $k > 1$ and fixing the variance of the noise to 0.5 exactly recovers the MS functional (3.5). If we do not fix the variance, we obtain a local version of $\mathcal{E}^{\mathrm{PC}}_{\mathrm{Gauss}}$ with local standard deviation

$$\sigma_i(x) = \frac{\int_{\Omega_i} \mathcal{N}_\sigma\,(y-x)\,(I(y)-\mu_i(x))^2 \mathrm{d}y}{\int_{\Omega_i} \mathcal{N}_\sigma\,(y-x)\,\mathrm{d}y}. \tag{3.17}$$

We cut the support of the Gaussian kernel in Eqs. (3.16) and (3.17) in order to obtain approximations to the weighted means and variances, respectively. This enables evaluating $\Delta\mathcal{E}$ using local windows. The global character of the MS functional is lost when cutting the Gaussian kernel support. Also, for $\sigma \to \infty$ we recover to the cartoon limit, i.e., $\nu \to \infty$ in (3.5). The larger σ, the more expensive an energy-difference evaluation becomes. In the limit of $\sigma \to \infty$, we can nevertheless efficiently compute energy differences as described for PC models.

The energy differences for a fixed-variance PS Gaussian noise model and a Gaussian noise model are computed similarly to the PC versions presented in sections 3.1.1.1. The only difference is that J is computed using statistics in local windows.

3.1.3.1 Gaussian noise

From Brox and Cremers (2007) we know that Eq. (3.7) with the space-dependent statistics in Eqs. (3.16) and (3.17) corresponds to the Mumford-Shah functional. Cutting the support of the Gaussian yields a very good approximation. Let R be the local-statistics-support radius. The computational complexity for calculating a one-pixel MS approximation energy difference $\Delta\mathcal{E}^{\mathrm{MS}}$ is $O(R^d)$. Large radii hence become computationally very expensive.

However, solving the MS functional is not the primary goal. We are rather interested in finding an efficient energy model to segment and sample piecewise smooth regions. We therefore further approximate Eqs. (3.16) and (3.17) by discarding the Gaussian weights. This yields approximate statis-

tics for smaller masks, because low-Gaussian-weight contributions are neglected. The piecewise smoothness is then approximated with piecewise constant local patches. Let S_x^R be the hyper-sphere of radius R centered at x. With the adapted local statistics

$$\mu_i(x) = \sum_{y \in \Omega_i \cap S_x^R} \frac{I(y)}{|\Omega_i \cap S_x^R|}, \ \sigma_i(x) = \sum_{y \in \Omega_i \cap S_x^R} \frac{(I(y) - \mu_i(x))^2}{|\Omega_i \cap S_x^R|}, \tag{3.18}$$

and $N_i(x) = |\Omega_i \cap S_x^R|$ we can compute the energy difference for a one-pixel label switch as:

$$\begin{aligned} \Delta\mathcal{E}_{\text{Gauss}}^{\text{PS}}(x) = -\sum_{i=0}^{M-1} \Bigg(& N_i'(x) \log \frac{1}{\sqrt{2\pi}\sigma_i'(x)} - N_i(x) \log \frac{1}{\sqrt{2\pi}\sigma_i(x)} \\ & - \frac{N_i'(x)}{2} + \frac{N_i(x)}{2} \Bigg). \end{aligned} \tag{3.19}$$

Fixed variance Using the local-window statistics in Eq. (3.7), the energy difference for changing the region label of pixel x from i to j is

$$\Delta\mathcal{E}_{\text{LS}}^{\text{PS}}(x) = (I(x) - \mu_i'(x))^2 - (I(x) - \mu_i(x))^2 + (I(x) - \mu_j'(x))^2 - (I(x) - \mu_j(x))^2. \tag{3.20}$$

3.1.3.2 Poisson noise

Even though the relation between the MS energy and space dependent statistics has only been derived for a Gaussian noise model, we also use local PC patches to approximate a piecewise smooth Poisson model. The Poisson distribution parameter $\tilde{\lambda}$ then becomes local to a patch, and also we only consider samples from within that patch. We use Eqs. (3.18) and (3.8) to obtain the energy difference when pixel x switches from region i

to j for a piecewise smooth model with Poisson noise:

$$\Delta\mathcal{E}^{\mathrm{PS}}_{\mathrm{Poisson}}(x) = -\sum_{k=\{i,j\}} \left(\log \mu'_k(x) \sum_{y\in\Omega'_k\cap S^R_x} I(x) - N'_k(x)\mu'_k(x) \right.$$
$$\left. - \log \mu_k(x) \sum_{y\in\Omega_k\cap S^R_x} I(x) + N_k(x)\mu_k(x) \right). \tag{3.21}$$

3.1.4 Data-dependent balloon potential

Assume we need to decide whether a pixel x with $\nabla I(x) \approx 0$ belongs to a foreground or the background region. The MS functional does not provide an answer, as it does not distinguish between foreground and background objects. It is rather formulated to find the edge set of an optimal piecewise smooth approximation J to an image I. Pixel x will not be classified as belonging to the edge set Γ, since the length term penalizes unjustified contour. For closed regions, the same argument causes regions to shrink. For local energy-minimization algorithms, such as active contours, the decision depends on the initialization. Cohen (1991) hence introduced balloon forces that create pressure within regions and inflate the closed regions like a balloon. Xu and Prince (1998) introduced wide-range gradient vector flows that attract contours to edges with large $\|\nabla I\|$. Using these flows, local optimization becomes more robust when the contour is far from an edge. However, in order to be more robust w.r.t. the initializations, the energy should decide if x belongs to a closed foreground region or to the background region. We hence suggest using a data-dependent balloon force that favors high-intensity regions to be in the foreground. Changing the sign of the coefficient favors the foreground to segment low-intensity regions. Let $L \in \mathbb{N}^\Omega$ be a label image that maps pixels to region labels. The background region has 0. Then, the data-dependent balloon potential is

$$\mathcal{E}_{\mathrm{balloon}} = -I \cdot H(-|L| + 1), \tag{3.22}$$

where $H(\cdot)$ is the Heaviside distribution. This generates an outward flow with a strength that depends on the image intensity. This flow counteracts the curvature flow induced by the length prior in a data-dependent manner.

3.2 Prior terms

We focus on local contour-length priors and global shape priors based on 1D moments. We implement these priors (see Appendix A) for the optimization and sampling algorithms introduced in chapters 4 and 5.

$|\Gamma|$ is one of the most widely used internal energies for active contour models. Penalizing $|\Gamma|$ seems intuitive because of its geometric meaning. The corresponding Gibbs prior probability pdf $p = \exp(-|\Gamma|)$ may seem rather arbitrary. However, for a discrete contour representation this is the maximum-entropy distribution. Before we discuss discrete approximations for $|\Gamma|$ and global shape priors, we eleaborate this theoretical result.

3.2.1 MRF - Gibbs equivalence

(Hammersley and Clifford, 1968) relates the Gibbs distribution to Markov Random Fields (MRF). This relation holds for all energies for which the calculation of $\Delta\mathcal{E}$ has a finite support. From Geman and Geman (1984) we reproduce the following two definitions and the following theorem. We then interpret the result by example.

Let S denote a set of N sites (for example pixels) and G be a neighborhood system on S (for example the neighborhood relation to compute the local contour length). In other words, (S, G) is an undirected graph. Furthermore $\mathbb{X} = \{X_s, s \in S\}$ is any family of random variables X_s indexed by S with possible realization $\omega = \{X_{s_1} = x_{s_1}, \ldots, X_{s_N} = x_{s_N}\}$.

Definition 4 (MRF)**.** *$\mathbb{X}$ is a MRF with respect to the neighborhood system G if*

- $P(\mathbb{X} = \omega) > 0$ *and*

- $P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r, r \in G_s)$

for all sites $s \in S$ *and all* ω.

Intuitively, a random variable X_s in a MRF is conditionally independent from all other random variables in that field, given the values of the random variables in X_s's neighborhood.

Definition 5 (Gibbs distribution)**.** *A Gibbs distribution relative to* (S, G) *is a probability measure* π *on the possible realizations* $\{\omega\}$ *of the form*

$$\pi(\omega) = \frac{1}{Z} \exp\left(-\frac{1}{T} \sum_{c \in C} V_c(\omega)\right).$$

T is the temperature. Z is a normalizing partition function. The exponent is the energy. And C is the set of cliques of the graph (S, G), and the family V_C is called the potential. Every clique contributes to the total energy through V_C.

The following theorem by Hammersley and Clifford (1968) specifies the equivalence between the two objects defined above:

Theorem 2 (Hammerley-Clifford)**.** $\mathbb{X}$ *is a MRF with respect to* G *if and only if* $\pi = P(\mathbb{X} = \omega)$ *is a Gibbs distribution with respect to* G.

In order to exploit this equivalence, we interpret an image as being a realization of a random field of labels. We therefore identify the sites s with pixels and define G as all pixels involved in computing internal energy differences.

The label probabilities for pixel x depend on the potentials of all cliques of the graph (S, G). For a neighborhood as illustrated in Fig. 3.5, only singletons (cliques of size 1) and doubletons (cliques of size 2) exist. The former are used to include external potentials, the latter for the prior as shown in the following example. For a simple Manhattan distance metric of the contour, we assume G to be edges to the 4 face-connected neighbors in 2D. Every interior pixel $x = (i, j)$ we have 5 cliques, one singleton and

4 doubletons. Then, $V_{\{i,j\}}$ is equal to the external energy. $V_{\{(i,j),(i-1,j)\}}$, $V_{\{(i,j),(i+1,j)\}}$, $V_{\{(i,j),(i,j-1)\}}$, and $V_{\{(i,j),(i,j+1)\}}$ are equal to 1 if the labels of the different sites are unequal, 0 else. For binary L, this is the Ising model.

3.2.2 Length regularization

In continuous active contour representations, such as in level-set methods, the contour length can easily be computed. In discrete representations, however, it needs to be approximated from the discrete contour pixels using concepts from digital geometry.

Various discrete length approximations for deformable models have been presented in the literature. Zhu and Yuille (1996) argued that blurring an image with a Gaussian filter has similar effects as including a length-regularization term in the energy functional. One problem with this approach, however, is that edges get smoothed. Also, spurious intensity fluxes across close regions can be a problem since they change the mean intensities of these regions. Another approximation used in Song and Chan (2002); Yu et al. (2006), and in techniques based on the Ising model, counts the number of region changes on the pixel grid. While this approach is computationally efficient, it causes the regions to tend to polygonal shapes instead of developing smooth contours (Yu et al., 2006). Also, the contour generally does not evolve smoothly, due to the discrete objective function. Shi and Karl (2008) hence smoothed the contour of a discretized level function using a Gaussian kernel, followed by a re-discretization step. A drawback of this approach is that the smoothing is not represented in the energy functional. The resulting trade-off between regularity and data fidelity is hence difficult to control (Kybic and Kratky, 2009).

Here we use *cut metrics* and *regularizing curvature flows*. The former has so far mainly been used in graph-cut segmentations, but perfectly suits the algorithms presented in chapters 4 and 5.

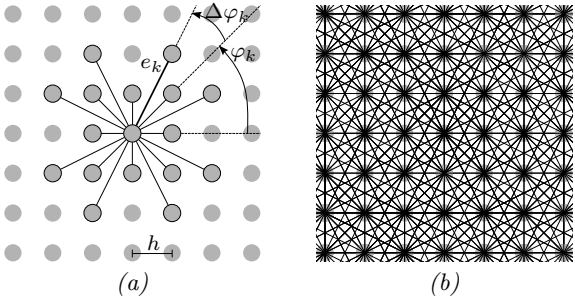


Figure 3.5: (a) The 16 neighborhood of a pixel. (b) Associated grid. See main text for variables and details.

3.2.2.1 Cut metrics

The Ising model only roughly approximates the contour length. Boykov and Kolmogorov (2003) hence introduced more sophisticated contour length approximations on a lattice. We briefly restate their result for the Euclidean case (Riemannian metrics are also considered in the original work) and connect them to the results above. The length approximation is based on the Cauchy-Crofton formula

$$\int n_\Gamma \, \mathrm{d}\mathcal{L} = 2|\Gamma|, \tag{3.23}$$

with n_Γ the number of intersections of a straight line with the contour Γ. $\mathcal{L}$ is the set of all lines. For the discrete version, consider the grid spanned by all directions in G, as depicted in Fig. 3.5b. For each grid line or angle φ (see Fig. 3.5a), Boykov and Kolmogorov (2003) derived the optimal edge weight w_k in order to approximate the Euclidean metric: This results in the following edge-weights calculation:

$$w_k = \frac{h^2 \Delta\varphi}{2|e_k|}. \tag{3.24}$$

Here, $|e_k|$ is the distance along the grid line $k, k = 1 \ldots K$, h is the pixel spacing, and $\Delta\varphi_k$ the angle difference to the $(k-1)$-st grid line ($\varphi_K = \pi$). The variables are in Fig. 3.5.

In order to define a discrete length prior, we set the MRF potentials for all doubletons to the edge weights, i.e., $V_{C,|C|=2} = w_k$.

For vanishing h and $\Delta\varphi$ the approximation converges to the Euclidean contour length. However, using Euclidean contour length approximations causes the final contour to favor certain directions (Kybic and Kratky, 2009).

3.2.2.2 Contour length regularization using curvature flow

Smoother approximations have been studied by Kybic and Kratky (2009), who proposed a regularizing flow for discrete level-set methods that approximates the local curvature κ as

$$\kappa(x) = C\left(\frac{|S_x^{R_\kappa} \cap X_{|L(x)|}|}{|S^{R_\kappa}|} - \frac{1}{2}\right), \tag{3.25}$$

with $S_x^{R_\kappa}$ a hyper-sphere of radius R_κ centered at x and $|S^{R_\kappa}|$ its volume. C is a constant that depends on the image dimension d and on R_κ. We adopt this approach, exploiting the fact that curvature regularization is equivalent to contour-length minimization.[1]

We therefore directly add the curvature-regularizing flow to $\Delta\mathcal{E}$. The direction of the flow is given by the outward normal on the contour. We adapt the sign of κ to account for the direction of the flow: for expanding regions, κ is subtracted from the energy difference, for shrinking regions it is added to it. $R_\kappa = 4$ is found to provide a good trade-off between regularity and resolution.

3.2.3 Moment-based shape priors

In contrast to local shape priors, global shape priors regularize the overall shape of a region. Many approaches have been proposed. Osada et al. (2002) used distances between distributions of angles and distances be-

[1] This is seen by applying variational calculus to $\sum_{i>0} \lambda|X_i| = \lambda \sum_{i>0} \int_{\Gamma_i} \mathrm{d}s$.

tween randomly sampled points on the contour. This approach is usually referred to as *shape distributions* or *shape contexts* (Belongie et al., 2002). Other shape-matching approaches are based on Hausdorff distances, Fourier descriptors, shape-moment distances, or skeleton distances. Zhang and Lu (2004) and Veltkamp and Hagedoorn (2001) give an overview. Cremers et al. (2002) presented statistical shape priors for continuous deformable models.

In moment-based shape priors, one computes k moments of the distribution of the points in a region. This results in a k-dimensional moment vector $\boldsymbol{M}$. Some vector norm between the moment vector $\boldsymbol{M}_{\Omega_j}$ of region Ω_j and the moment vector $\boldsymbol{M}_T$ of a *template* shape is then used as shape-distance measure. Such moment vectors are called *geometric moments*. Geometric moments can be transformed to other bases in order to render them rotation invariant. Examples of moment-based shape priors are found in Kim and Kim (2000) and Rose et al. (2009). However, it is challenging and computationally expensive to compute rotation-invariant measures based on moments, especially in 3D.

We therefore consider a simple, but efficient 1D metric, which is naturally invariant to all rigid-body transformations. We consider the moments of all line-length segments from every discrete point within a region to a fixed reference point. We use the *centroid* of the region as reference fixed point. Conceptually, this is the $D1$-shape measure proposed by Osada et al. (2002). But instead of sampling points on the contour, we consider all discrete points within a region.

Such a 1D moment-based measure is not as accurate as 2D moment-based measures. Nevertheless, Mertzios and Tsirikolias (1991) observed that when deforming a shape only locally, this metric behaves well in the sense that it increases monotonically with the deformation. In a local segmentation context, the shape is usually quite close to its target shape, such that 1D moments are accurate enough.

Let d_i be the Euclidean distance between a discrete point $x_i \in \Omega_j, j > 0$, and the centroid $\bar{x}$ of Ω_j, i.e., $d_i = \|x_i - \bar{x_j}\|$. Furthermore, let $\langle d_i \rangle$ be the mean of these distances over all points in that region. Then, the *normalized*

uni-variate central moment M_k of order k is

$$M_k(\mathcal{D}_{\Omega_j}) = \frac{1}{M_2(\mathcal{D}_{\Omega_j})} \left(\sum_{d_i \in \mathcal{D}_{\Omega_j}} (d_i - \langle d_i \rangle)^k \right)^{\frac{1}{k}}. \tag{3.26}$$

$\mathcal{D}_{\Omega_j}$ is the set of distances $\{d_i\}$ of all points in Ω_j. Normalizing with the second moment M_2 renders the measure scale-invariant. Using the resulting moment vector we can define the 1D-moment-based energy difference

$$\Delta\mathcal{E}_{\text{shape}} = (\boldsymbol{M}_T - \boldsymbol{M}_{\Omega'_j})^2 - (\boldsymbol{M}_T - \boldsymbol{M}_{\Omega_j})^2, \tag{3.27}$$

where Ω'_j is the perturbed shape.

3.3 Discussion

We provided explicit formulas for the probability ratios of discrete segmentations of noisy images under various image models. We started from Bayes' formula and separately treated likelihood and prior terms. All energies considered multiple regions.

In the likelihood part, we distinguished between piecewise constant and piecewise smooth models. We provided formulas for images corrupted with Gaussian noise, Gaussian noise with fixed variance, and Poisson noise.

For the PC case we further distinguished image-formation models that involve a blurring kernel K and models where K is the Dirac-delta function. Future work on PC deconvolving energies, however, needs to study the effect of delayed intensity updates. Alternatively, schemes that directly incorporate intensity changes into the label-jump probability may be studied.

In order to account for PS models, we based the energy computations on local statistics. We have shown a relationship between local-statistics-based energy computations for PS image models and the MS functional using the work of Brox and Cremers (2007). Accounting for a convolution with a kernel K in a PS image model is ongoing work. The difficulty is

that intensities then depend on the geometric state within the local patch.

We have further reviewed the explicit link due to Geman and Geman (1984) between a digital image, interpreted as a realization of a field of random variables, and the Gibbs distribution. This allowed computing prior probabilities of segmentations from a modeled energy. Using this result, we presented prior ratios for two different discrete length approximations.

We also presented an efficient, translation, scale, and rotation invariant global shape prior based on 1D moments. The resulting shape descriptions are not accurate enough to reconstruct a shape, but the prior can be used to correct local errors of a segmentation.

CHAPTER

FOUR

Unsupervised multi-region segmentation with discrete models

4.1 Introduction

In unsupervised multi-region segmentation the number of regions, their photometric features, and their contours are to be jointly estimated from the image. This requires additional regularization on top of the usual smoothness priors (see Sec. 3.2.2).

Related work Most multi-region methods use region-number priors (e.g., (Brox and Weickert, 2004; Delong et al., 2011)). Alternatively, a length-/area balancing term is used (Sandberg et al., 2010). Brox and Weickert (2004) proposed recursive splitting of regions into pairs of sub-regions such as to minimize an energy that includes a region-number penalty . A separate level set is evolved for each region. In order to prevent regions from overlapping, an additional penalization term is introduced into the energy functional. The number of level functions that need to be evolved

is reduced to one in unsupervised region-competition methods (Kadir and Brady, 2003). A representation with only one level function, however, cannot capture multiple contours touching in one point. Other multi-region segmentation methods impose a fixed number of regions (or an upper bound on it) that is often learned prior to contour evolution using, e.g., pixel-feature clustering or model selection. This is for example the case in multi-phase level sets, which evolve $\log_2 M$ level functions in order to segment a fixed number of M regions (Vese and Chan, 2002). Besides the increased computational cost of evolving multiple level functions, undefined statistics from empty regions may hamper the evolution (Brox and Weickert, 2004). Mansouri et al. (2006) presented a multi-region-competition (Zhu and Yuille, 1996) implementation where the contours are implicitly represented by multiple level functions. Lie et al. (2006) represented multiple regions using a single level function that converges to a piecewise constant function indicating the different regions. Homeomorphic level sets prevent topological changes during energy minimization (Fan et al., 2008). Song and Chan (2002) introduced a fast discrete level-set method for the two-region piecewise constant CV model. He and Osher (2007) generalized this method to an arbitrary, but priorly known number of piecewise constant regions and related the approach to topological derivatives (Larrabide et al., 2008). Yu et al. (2006) optimized a two-region piecewise smooth image energy using a discrete level function on a lattice. Fast discrete level-set methods have been used for real-time tracking of a known, fixed number of regions (Shi and Karl, 2008) and for fast approximate surface evolution (Malcolm et al., 2008). Graph min-cut algorithms (Boykov et al., 2001) are efficient combinatorial optimizers for discrete problems with theoretical performance guarantees, both for priorly known numbers of regions (Boykov et al., 2001) and for unknown numbers of regions using a region-number penalty (Delong et al., 2011).

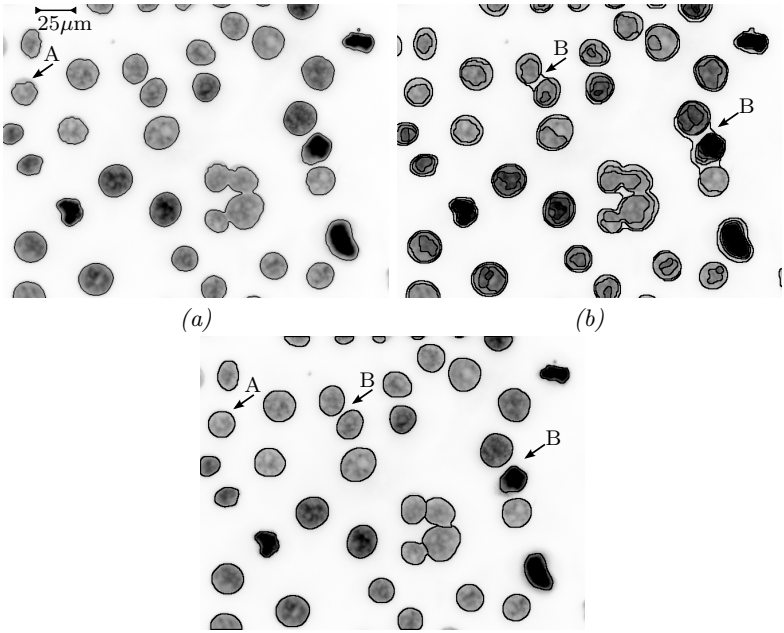
Connected-component prior Here we replace the prior or penalization on the region number (or its upper bound) by the topological constraint that foreground regions have to be connected components. We therefore define a foreground (FG) region as a connected set of pixels in a certain digital geometry representation, amounting to a topological constraint. This definition is motivated threefold: (1) We wish that regions determined by a segmentation algorithm delineate different physical ob-

jects represented in an image (see Fig. 4.1). This frequently causes the problem of choosing an appropriate number of regions so as to avoid over- and under-segmentation (see arrows A and B in Fig. 4.1b). (2) It resolves the dependence between the number of regions and the regularization constant in the energy (see Fig. 4.2). (3) It can be evaluated using only local information, whereas region-number penalties require global information (see Sec. 2.2).

Another advantage of such a prior is that topological constraints can be evaluated using local information only, whereas region-number priors require global information. The present definition of a region regularizes the problem of estimating the number of regions jointly with their photometric features and contours. We use extended concepts from digital topology (see Sec. 2.2) to enforce the topological region definition, and we present an efficient discrete energy minimization algorithm that can locally minimize a range of well-known energy functionals, including the energies presented in Sec. 3, under this hard constraint.

We present an implementation of a versatile discrete-contour multi-region-competition algorithm in 2D and 3D, inspired by discrete level sets (Shi and Karl, 2008). The algorithm is based on the idea of using computational particles to represent the evolving contour and is able to segment a priorly unknown and arbitrary number of connected regions. Regions are dynamically fused and split during energy minimization. This enables jointly estimating the number of connected regions in an image, their photometric features, and their contours. We use digital topology (see Sec. 2.2) to provide optional control over region splits and merges during contour evolution. The topological constraint for foreground regions to be connected components, however, is always present.

We demonstrate the applicability of the present method to three well-known segmentation energy functionals: The first energy describes images containing an unknown number of regions where each region has a different, but constant (homogeneous) intensity. The energy is regularized using a penalty on the approximated length of the overall contour. The second energy extends this model to account for regions containing piecewise smooth intensity distributions. The third energy extends explicit deconvolving active contours (Helmuth and Sbalzarini, 2009) to handle topological changes



(a) *(b)*

(c)

Figure 4.1: A motivation for defining FG regions as connected components. The image shows a collection of cell nuclei, which are distinct real-world objects (image: Dr. Prisca Liberali, University of Zurich). (a) Segmentation (black outlines) using graph cuts (Boykov et al., 2001) to minimize the two-region CV energy. Due to their different intensities, not all nuclei are correctly delineated (see, e.g., arrow A). (b) Graph-cut segmentation minimizing a ten-region piecewise constant energy. It is not clear what number of regions to choose in order to avoid over-segmentation and fusion of objects (see arrows B). (c) Segmentation using the present algorithm constraining FG regions to be connected components. The algorithm finds 39 connected FG regions, corresponding to the 39 nuclei in the image.

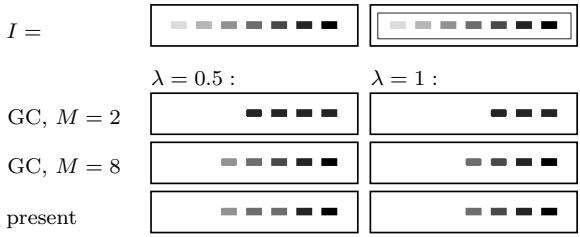


Figure 4.2: Illustration of the dependence between the number of regions and the length-regularization coefficient λ in a piecewise constant image. 1^{st} row: raw image I (left) and initialization for the present algorithm (right). 2^{nd} and 3^{rd} rows: resulting reconstructed images using graph cuts (GC) (Boykov et al., 2001) with $M = 2$ and $M = 8$ regions, respectively. The lowest intensity that is detected depends on both M and λ. 4^{th} row: present reconstruction when defining a FG region as a connected component. The result corresponds to the GC result with the ground-truth number of $M = 8$ regions. The lowest intensity detected only depends on λ.

during energy minimization and to arbitrary dimensions. This renders the method less sensitive to the topology of the initial segmentation.

The remainder of this chapter is organized as follows: In Sec. 4.2 we discuss the digital objects of our discrete explicit deformable model. Sec. 4.3 presents an interpretation of the topological prior in terms of an energy. In Sec. 4.4 we present an efficient discrete algorithm for region-competition energy minimization under hard topological constraints. In Section 4.5 we demonstrate the approximation of Sobolev gradients using particles. Section 4.6 demonstrates the applicability of the present framework to three well-known image models on both synthetic and real-world images in 2D and 3D, and compares its performance with that of a multi-label graph-cut minimizer (Delong et al., 2011). Section 4.7 summarizes and discusses the results.

4.2 Digital geometry representation

4.2.1 Connectivity of regions

We constrain FG regions in the image to be represented by connected sets of pixels. All void space between FG regions is represented by one and the same background (BG) region. Regions that can be captured by this representation must be larger than a single pixel. Consequently, regions cannot be connected via edges or corners of the pixel lattice. The FG regions are hence defined as face-connected neighborhoods, i.e., 4-connected in 2D and 6-connected in 3D. In the following we refer to this type of connectivity as the *FG connectivity*. According to Jordan's theorem, the BG region then needs to be 8-connected in 2D and 18 or 26-connected in 3D (Ségonne, 2005). Here we use the (FG, BG)-connectivity pairs $(4, 8)$ and $(6, 26)$ for 2D and 3D, respectively.

4.2.2 Contour

The discrete contour $\tilde{\Gamma}_i$ around FG region X_i, $i = 1, \ldots, M-1$, is defined by all pixels with at least one FG-connected neighbor belonging to a different region $X_j \neq X_i$, $j = 0, \ldots, M-1$. These *contour points* are part of the corresponding FG region, i.e., $\tilde{\Gamma}_i \subset X_i$, making all FG regions closed, connected subsets of Ω. The BG region is the open complement set $X_0 = \Omega \setminus \bigcup_{i=1}^{M-1} X_i$. Consequently, the continuous $(d-1)$-dimensional boundary $\Gamma \in \Omega$ of the FG region is the pixel edge as illustrated in Fig. 4.3. In-between any two pixels of different regions, there is at least an infinitesimal stripe of the BG region. Since FG regions are closed and have the smallest connectivity type, they can never touch. Topological paradoxes are hence avoided. In the following, however, we still say that FG regions are touching whenever two pixels with different labels are FG neighbors.

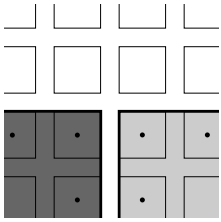


Figure 4.3: Pixel grid with two closed foreground regions (gray) and the open background region (white). Pixels with a dot are discrete contour points. These pixels belong to the respective FG region and contribute to the region statistics. The bold lines illustrate the continuous contour Γ.

4.3 Connected components as region-merging energy

Since we define regions as connected components, they may naturally split during the energy-minimization process, provided these topological changes are permitted by the user. The criterion for regions to merge can be formulated as a hard region-merging penalty in the energy functional:

$$\mathcal{E}_{\text{merge}} = \sum_{(i,j)>0:X_i\sim X_j} H[D_{\text{KL}}(P_{X_i}||P_{X_i\cup X_j}) + D_{\text{KL}}(P_{X_j}||P_{X_i\cup X_j}) - \theta]\,. \quad (4.1)$$

$H(\cdot)$ is the Heaviside distribution and $X_i \sim X_j$ indicates that X_i and X_j are FG-connected competing regions. Two regions merge if this is favorable for the overall energy. In order to reflect the discrete-event character of topological changes, the weight of this contribution to the total energy in is ∞.

4.4 Region Competition (RC) - Energy minimization algorithm

We introduce a versatile region-competition mechanism inspired by discrete level-set methods. In the present framework, minimizing an energy $\mathcal{E}$ done using a rank-based discrete optimizer that does not require information about the gradient of the energy functional. This is beneficial since the hard penalty introduced by the topological constraint on regions is not differentiable. We start by introducing the data structures and then describe the minimization algorithm used to perform topologically consistent contour evolution. The algorithm is designed with data locality and parallelism in mind.

4.4.1 Data structures

The present method relies on three main data structures: First, regions are identified using a *label function* (or *label image*) $L : \Omega \mapsto \mathbb{N}$ that maps a discrete space coordinate x to the region label currently assigned to that pixel. Contour pixels are assigned the negative label of the region they bound. This allows identifying contour points directly from the label image. The label of the BG region is fixed to 0.

Second, all points belonging to a contour are stored as computational particles. Each particle A is defined by its location x_A, i.e., the integer pixel coordinates of the corresponding contour point, and its properties. These properties are used to evolve the contour and are stored in a *particle data structure* containing:

- the currently assigned label $l = L(x_A)$ to avoid expensive lookups in the label image;
- the candidate label l' as the label that minimizes $\Delta\mathcal{E}_A$ among all other candidate labels;
- the change in energy $\Delta\mathcal{E}_A$ when changing the current label l to the candidate label l';

Algorithm 1 Discrete region competition

1: Initialization: Set up L and $\mathcal{C}$.
2: **repeat**
3: $\quad\mathcal{M} = \mathcal{C}$
4: $\quad$Optimization: see Algorithm 2
5: $\quad$(Optional) Compute Sobolev gradients: see Sec. 4.5
6: $\quad$Contour propagation: see Algorithm 3
7: $\quad$Topology processing: see Algorithm 4
8: **until** convergence

- lists with the particle indices of the parent and child points of A. Parents are all FG-connected points that belong to a different FG region. They are responsible for expanding the FG region they belong to. Children are all FG-connected points that belong to a different region, including the BG.
- the count r of parents with label l'.

Third, we use a hash map $\Omega \mapsto \mathcal{C}$ as an efficient data structure to iterate over the particles and to map space coordinates x to particle indices A. The hash map allows index lookups in $O(1)$.

4.4.2 Algorithm

We describe an algorithm that iteratively propagates the contour points (viz., the particles) of multiple regions over the image such as to locally minimize an energy functional under topological constraints on the FG regions. After initialization, the algorithm proceeds in iterations (see Algorithm 1), each of which comprising three steps: optimization, contour propagation, and topology processing. Optionally we compute Sobolev gradient approximations before contour propagation. This optional step is discussed in detail in Sec. 4.5.

Application-specific segmentation methods can be derived from the present algorithm by specifying a particular energy functional and a set of topological constraints. The former allows including prior knowledge about the

image-formation process (e.g., the PSF of a microscope in a deconvolving energy (Helmuth and Sbalzarini, 2009; Paul et al., 2013)) and the morphology of the imaged objects (see Sec. 4). The latter allows including prior knowledge about whether FG regions are allowed to fuse or split (or both or none) during the energy minimization process (Bertrand, 1994; Lamy, 2007; Han et al., 2003). Regardless of the topological constraints on contour evolution, however, a FG region is always defined as a FG-connected component.

The input arguments to the algorithm are an energy functional $\mathcal{E}$, the image data I, and, since it is an iterative process, an initial segmentation L_0. Pixels in L_0 that have a special label f can be used to indicate forbidden regions. These regions are treated as boundaries that are never penetrated by any contour, nor do they have an active contour themselves. In order to avoid boundary checking at the border of the image domain Ω, we initially pad the entire image by a layer of pixels with label f.

4.4.2.1 Initialization

All FG pixels with a neighbor of a different label in L_0 are marked as contour points. For each contour point, a particle is generated and added to the hash map $\mathcal{C}$, where the corresponding space coordinate is the key of the map and the particle index its value (line 1 in Algorithm 1).

4.4.2.2 Optimization

In the main loop (line 3 in Algorithm 1), we first copy the current set of particles $\mathcal{C}$ to $\mathcal{M}$. $\mathcal{M}$ is the candidate list containing all particles we consider moving to another region. We first attempt moving them to the BG by setting all candidate labels l' in $\mathcal{M}$ to 0 (line 2 in Algorithm 2). We then calculate for each particle A the energy difference $\Delta\mathcal{E}_A = \Delta\mathcal{E}(x_A, l \to 0) = \mathcal{E}(x_A, 0) - \mathcal{E}(x_A, l)$.

In the next step, we attempt growing the FG regions. To do so, all particles $A \in \mathcal{M}$ perform the following steps: All neighboring points that belong

Algorithm 2 Optimization

1: **for all** $A \in \mathcal{M}$ **do**
2: $l'_A = 0; \Delta\mathcal{E}_A = \Delta\mathcal{E}(x_A, l \to 0) = \mathcal{E}(x_A, 0) - \mathcal{E}(x_A, l)$
3: **for all** $\{B \mid x_B \in \{N_n^1(x_A, L \neq l_A)\}, l_B \neq f\}$ **do**
4: register A in B's parent list; register B in A's daughter list
5: **if** $B \notin \mathcal{M}$ **then**
6: add B to $\mathcal{M}$; Set $l_B = 0$; $r_B = 1$; $l'_B = l_A$;
$\Delta\mathcal{E}_B = \Delta\mathcal{E}(x_B, 0 \to l'_B)$
7: **else**
8: **if** $l_A = l'_B$ **then**
9: $r_B = r_B + 1$
10: **else if** $\Delta\mathcal{E}(x_B, l_B \to l_A) < \Delta\mathcal{E}(x_B, l_B \to l'_B)$ **then**
11: $l'_B = l_A$
12: construct $\mathcal{G}$ from $\mathcal{M}$
13: $\mathcal{M} = \mathcal{M} \backslash \{A : \Delta\mathcal{E}_A \geq 0\}$

to a different region (including the BG) register A as a parent (line 4). Particles for contour points that do not yet exist (since their current label is 0) are created and added to $\mathcal{M}$ (line 6). All particles now know the set of pixels they could potentially move to, and the set of pixels they are attacked from.

The candidate label l'_B of B is set to the label of A if this is favorable in energy (lines 8–11). This means that if the candidate label of B is different from the label of A (else we increase the parent count r since this candidate label is supported by two or more parents, line 9), we set l'_B to the label l_A of the parent if $\Delta\mathcal{E}(x_B, l_B \to l_A) < \Delta\mathcal{E}(x_B, l_B \to l'_B)$. In addition, we remove particles with $\Delta\mathcal{E} \geq 0$ from the candidate list (line 13).

While each individual move in $\mathcal{M}$ is guaranteed to decrease the overall energy, this may not be true for several moves performed simultaneously. This property is inherent to discrete contour-propagation methods and can cause contour and energy oscillations. We therefore monitor the history of the contours and halve the percentage of accepted moves whenever the contours do not propagate anymore. This amounts to reducing the step size in a rank-based optimization scheme. Unless the algorithm has already converged, the step size eventually reduces to 1, i.e., only a single move

Algorithm 3 Contour propagation

1: find maximal-connected sub-graphs G_k of G.
2: **for all** $G_k = \{V_k, E_k\}$ **do**
3: sort V_k according to $\Delta\mathcal{E}$
4: **for all** $A \in V_k$ with $\mathcal{E}_A < 0$ **do**
5: **if** conditions C1, C2, and C3 are true **then**
6: $\forall$ children B with $l'_B = l_A : \; r_B = r_B - 1$.
7: **else**
8: $\mathcal{M} = \mathcal{M} \backslash A$

from $\mathcal{M}$ is executed in each iteration. This guarantees that the energy can only decrease from then onward, and the algorithm hence converges to a local minimum of $\mathcal{E}$.

4.4.2.3 Contour propagation

The set of moves that will be executed simultaneously needs to be selected according to the topological and causal constraints. Simply executing all minimum-energy moves determined in the optimization step could lead to violations of the topological constraints. Only contour points that are not FG-simple are allowed to cause a topological change in any FG region.

Topological violations can arise from the fact that moves at iteration t may depend on moves in iteration $t + 1$. This is illustrated in Fig. 4.4 for the points $(\mathrm{d}, 2)$ and $(\mathrm{c}, 3)$. Whether region A is allowed to propagate to pixel $(\mathrm{d}, 2)$ without disconnecting depends on the label of pixel $(\mathrm{c}, 2)$ in iteration $t + 1$. The move in iteration t is only valid if pixel $(\mathrm{c}, 2)$ will still belong to region A in iteration $t + 1$. But $(\mathrm{c}, 2)$ has a parent at $(\mathrm{c}, 3)$, proposing it to join region B. This point at $(\mathrm{c}, 3)$ in turn is a candidate for label C through the parent at $(\mathrm{d}, 3)$. Situations like this induce topological dependence chains of arbitrary length. We identify the set of moves that are topologically dependent by constructing an undirected graph $G = \{V, E\}$ (line 12 in Algorithm 2). The vertices V correspond to particles and the undirected edges E to parent–child relationships. Topologically dependent sets are then given by the maximal-connected sub-graphs G_k of G. The

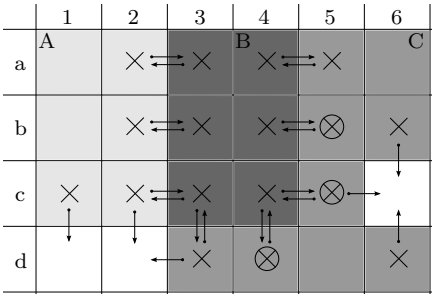


Figure 4.4: Illustration of 3 adjacent FG regions A (light gray), B (dark gray), and C (gray) in 2D. Points in the background region are white. Particles are shown as crosses. Points without a particle are interior points; they are not FG-connected to any other region. The arrows point from parents to the corresponding children. The circles indicate non-foreground-simple points; interior points are not considered. See main text for details about the algorithm.

maximal-connected sub-graph in the example of Fig. 4.4 contains the vertices $\{(\mathrm{c},2),(\mathrm{c},3),(\mathrm{d},2),(\mathrm{d},3)\}$.

The contour is then propagated by selecting all compatible moves in G_k, such as to minimize the sum of their energy differences. This is done independently for each sub-graph G_k. In order to avoid enumerating all compatible moves, we use a sub-optimal heuristic (Algorithm 3). This starts by sorting the vertices V_k of each sub-graph by ascending $\Delta\mathcal{E}$ (line 3 in Algorithm 3) and purging all invalid moves from $\mathcal{M}$ in this order. Moving particle A is valid if it fulfills all of the following conditions (line 5):

C1: if A is a child, its parent count is $r_A \geq 1$;

C2: if A is a parent, all of its children B that have already been accepted as a move have $r_B > 1$;

C3: if A is a parent, at least one of its children is not yet accepted or has a candidate label $l'_B \neq l_A$.

C1 ascertains that the particle is connected to the propagating region. C2 ensures that no child of this particle would lose connection to the

Algorithm 4 Topology processing

1: change = true
2: **while** change **do**
3: change = false
4: **for all** $A \in \mathcal{M}$ **do**
5: **if** x_A is FG-simple **then**
6: Update data structures: Algorithm 5(x_A)
7: change = true
8: **for all** $A \in \mathcal{M}$ **do**
9: **if** holes are disallowed `AND` $(T_n(x_A, L = l'_A) \geq 2$ `OR` $T_{\bar{n}}(x_A, L \neq l_B) \geq 2)$ **then**
10: **next** A
11: **if** $T_n(x_A, L = l_A) \geq 2$ **then**
12: **next** A **if** splits are disallowed
13: store the seed set $\mathcal{S} = \{N_n^1(x_A, L = l)\}$.
14: Update data structures: Algorithm 5(x_A)
15: **for all** $X_i \sim X_j$, $i, j = 1, \ldots, M-1$, **do**
16: **if** fusions are allowed `AND` region merging criterion is true **then**
17: merge regions X_i and X_j and add seed to $\mathcal{S}$
18: Recompute L using flood fill from seeds $\mathcal{S}$

propagating region if this parent changed its label. C3 prohibits moves to interior points. Valid moves for a parent A reduce the parent counts of all its children B with $l'_B = l_A$ (line 6).

In summary, Algorithm 3 identifies topologically dependent particles and selects a topological compatible subset thereof. These particles are ranked in order of decreasing energy differences before the moves are executed. This ranking is independent of the particle processing order. It only depends on the energy differences of the particles, which is a sole property of the image and the contour state.

4.4.2.4 Topology processing

We detect and account for topological changes in the FG regions using

concepts from digital topology (Bertrand, 1994; Lamy, 2007; Han et al., 2003; Ségonne, 2005; Shi and Karl, 2008). The BG region is allowed to change its topology arbitrarily. A genus change in a FG region can be a *split* of the region into several regions, a *fusion* of two or more regions into one, or the introduction of a *hole* into a region.

Splits and the introduction of holes are detected using the FG topological number (see Sec. 2.2.2). If $T_n(x_A, \{y : L(y) = l'_A\}) \geq 2$ or $T_{\bar{n}}(x_A, \{y : L(y) \neq l_B\}) \geq 2$, changing the label of particle A to l'_A introduces a hole in region l_B (line 9 in Algorithm 4). Similarly, if the FG topological number for the label l_B is larger than 1, the corresponding region splits, unless splits are disallowed by the user (lines 11–12).

If region fusions are allowed, all competing pairs of regions (indicated by $\sim$ in line 15) undergo a region-merge check (line 16). In principle, this check depends on the energy functional $\mathcal{E}$. Different energy-independent merging criteria, however, have been introduced based on region statistics (Zhu and Yuille, 1996; Calderero and Marques, 2008; Ayed and Mitiche, 2008). Here we use the symmetric Kullback-Leibler merging criterion (Calderero and Marques, 2008) based on measuring the similarity between the empirical intensity distributions P_{X_i} and P_{X_j} in the two regions X_i and X_j, $i, j > 0$. The regions fuse if

$$D_{\mathrm{KL}}(P_{X_i}||P_{X_i \cup X_j}) + D_{\mathrm{KL}}(P_{X_j}||P_{X_i \cup X_j}) < \theta\,, \tag{4.2}$$

where $D_{\mathrm{KL}}(\cdot||\cdot)$ is the Kullback-Leibler divergence between the two distributions in the argument. The merging threshold θ is a free parameter of the method. For $\theta = 0$, regions are prevented from fusing.

Whenever region labels change due to splits or fusions, a seeded flood fill in L is performed to identify the new connected components. For fusions, the seed point is one of the pixels where the regions touch. For splits, all FG points neighboring points where the regions were last in contact are seeds. The points of last contact are easily found as those that are not FG-simple (line 13). If a seed point moves to a different region, another point in its geodesic neighborhood of order 1 becomes the new seed. The flood fill (line 18) then reconstructs the label image L.

Algorithm 5 Data-structure update

1: $L(x_A) = l'_A$.
2: **if** $l_A \neq 0$ **then**
3: Add $x' \in N_n^1(x_A, l_A)$ to $\mathcal{C}$; $L(\boldsymbol{x}') = -l_{A'}$
4: **if** $l_{A'} \neq 0$ **then**
5: Remove all interior points in $x' \in N_{\bar{n}}^1(x_A, l'_A) \cup x_A$ from $\mathcal{C}$ and set $L(\boldsymbol{x}') = |l'_A|$.
6: **else**
7: $\mathcal{C} = \mathcal{C} \backslash x_A$

4.4.2.5 Data-structure update

During topology processing, moves that do not induce topological violations are executed and the data structures are updated (Algorithm 5 called from lines 6 and 14 of Algorithm 4). The labels of the corresponding pixels are changed to the respective candidate labels, and the label image is updated accordingly (line 1 in Algorithm 5). These changes may causes the creation of new contour points, the particles of which are added to the hash map $\mathcal{C}$ (line 3). Similarly, the particles from pixels that newly became interior points are removed from $\mathcal{C}$ (lines 5 and 7).

4.5 Approximation of Sobolev gradients

The RC algorithm amounts to a gradient descent, where the gradient is approximated at discrete points in order to obtain $\Delta\mathcal{E}$. We used a L^2-type gradient. We therefore considered energies belonging to the L^2 inner-product function space. This *inner product* has certain undesirable properties for deformable models, which have been extensively discussed by Sundaramoorthi et al. (2007). For discrete models, the following two of these properties are of special interest: First, the inner product does not contain any regularity terms. This results in non-smooth contour/time (hyper) surfaces. Hence, in the presence of noise, the contour becomes non-smooth during evolution. Curvature regularization via priors is typically used to prevent this. Second, the L^2-type gradient is ignorant w.r.t. the type of

curve motion, such as global translations or local adaptations. Intuitively, the contour therefore locally optimizes "on a small scale" and frequently steps in local minima.

In a Sobolev function space, functions are associated with a norm that contains L^p-norms and derivatives of the function itself. The metric on that space induced by this inner product hence includes smoothness terms that allow addressing the regularity issues mentioned above. Using such a metric does not affect the global minimum of the function, but it amounts to *preconditioning* the L^2-type gradient.

Sobolev gradients have been introduced by Neuberger (1997). Deformable models using a Sobolev-type inner products have been presented by Sundaramoorthi et al. (2005); Charpiat et al. (2005). Detailed theoretical and practical considerations can be found therein. It has been shown that curve evolution using Sobolev gradients are smoother and more robust w.r.t. contour initialization. This comes at the cost of more expensive gradient computations. Often, however, often less iterations are necessary.

Here, we show how to use Sobolev gradients in RC *using particle interactions*. We first discuss the inner product considered. Then we compute energy differences based on the metric induced by this inner product.

4.5.1 Inner product

We closely follow the argumentation of Sundaramoorthi et al. (2007). They consider a Sobolev space $W^{1,2}$, which is a Hilbert space H^1, and define its inner product as

$$\langle h, k \rangle_{H^1} := \bar{h} \cdot \bar{k} + \epsilon \cdot E^2 \cdot \langle \nabla h, \nabla k \rangle_{L^2}, \tag{4.3}$$

where h and k are elements of the tangent space of Γ_i. The tangent space is the set of all possible deformations of Γ_i. The ∇-operator in Eq. (4.3) is w.r.t. the L^2-norm. The variable $\epsilon \in \mathbb{R}^+$ is a hyper parameter for smoothness and $E > 0$ determines the length scale of the smoothness terms in the inner product. Sundaramoorthi et al. (2007) set $E = |\Gamma_i|$, we

further discuss E in Sec. 4.5.2.1. The average $\bar{h}$ of h over Γ_i is

$$\bar{h} = \frac{1}{|\Gamma_i|} \int_{\Gamma_i} h(s) \mathrm{d}s. \tag{4.4}$$

The average $\bar{k}$ is defined similarly. The L^2 inner product is

$$\langle h, k \rangle_{L^2} = \frac{1}{|\Gamma_i|} \int_{\Gamma_i} h(s) \cdot k(s) \, \mathrm{d}s. \tag{4.5}$$

4.5.2 Sobolev gradient computation

In order to compute the first-order Sobolev gradients using the metric induced by the inner product in Eq. (4.3), Sundaramoorthi et al. (2007) presented the following equality

$$\nabla_{H^1} \mathcal{E}(s) = \int_{\Gamma_i} \tilde{K}(\hat{s} - s) \cdot \nabla \mathcal{E}(\hat{s}) \mathrm{d}\hat{s} = (\tilde{K} * \nabla \mathcal{E})(s) \tag{4.6}$$

with convolution kernel

$$\tilde{K}(r) = \frac{1}{E} \left(1 + \frac{(|r|/E)^2 - (|r|/E) + 1/6}{2\epsilon} \right), \, r \in [-E/2, E/2]. \tag{4.7}$$

Figure 4.5 shows $\tilde{K}$ for different ϵ. Equation (4.6) enables computing the first-order Sobolev gradient from the L^2 gradient. Note that the convolution domain is Γ_i.

4.5.2.1 Kernel selection

Sundaramoorthi et al. (2005) set E equal to the contour length $|\Gamma_i|$ in order to obtain a scale-invariant inner product in Eq. (4.3). Gradient information is then shared along the entire contour. This enables global contour movements such as translations.

Here, we fix E and therefore set a scale with respect to the image coor-

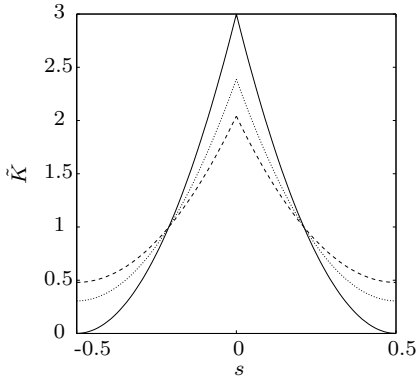


Figure 4.5: Kernel $\tilde{K}$ for $E = 1$ and different ϵ. The solid, dotted, and dashed curves show $\tilde{K}$ for $\epsilon = 1/24$, $\epsilon = 0.06$, and $\epsilon = 0.08$, respectively.

dinate system. The support of $\tilde{K}$ then becomes finite, which is beneficial for computational efficiency. Moreover, it enables us to approximate the arc distance $\hat{s} - s$ between two points, which is used to evaluate the kernel in Eq. 4.6. RC's discrete representation, however, does not allow computing this quantity. Since we consider a relatively small support of $\tilde{K}$, we approximate intrinsic distances by the Euclidean distance $d(\cdot, \cdot)$. We set $\epsilon = \frac{1}{24}$. The kernel then smoothly approaches zero at its tails, as depicted in Fig. 4.5. The length scale E is a user defined parameter.

4.5.2.2 Discrete approximation

Since the particles store a discrete $L^2 - type$ gradient approximation on both sides of the contour Γ_i, we can approximate the Sobolev gradient by convolving over particles. In particle methods, discrete convolution amounts to pairwise particle–particle interactions (Hockney and Eastwood, 1988; Eldredge et al., 2002; Koumoutsakos, 2005; Schrader et al., 2010). We therefore efficiently compute this convolution by summing kernel-weighted energy differences of neighboring particles.

Let $\mathcal{Q}_p$ be the set of particles in $\mathcal{M}$ that are located in the support of $\tilde{K}$ and have the same label as p, i.e., $\mathcal{Q}_p = \{q \in \mathcal{M} |\, d(x_q, x_p) < E/2,\, l_p = l_q\}$.

Similarly, let $\mathcal{Q}'_p$ be the particles within the kernel support that lie on the other side of the contour, i.e., $\mathcal{Q}'_p = \{q \in \mathcal{M} | \, d(x_q, x_p) < E/2, \, l_p = l'_q\}$. For each particle p we then update the energy difference as follows:

$$\Delta\mathcal{E}_p \leftarrow \frac{1}{|\mathcal{Q}_p|} \sum_{q \in \mathcal{Q}_p} \tilde{K}(d(x_q, x_p)) \Delta\mathcal{E}_q - \frac{1}{|\mathcal{Q}'_p|} \sum_{q \in \mathcal{Q}'_p} \tilde{K}(d(x_q, x_p)) \Delta\mathcal{E}_q \quad (4.8)$$

We use cell lists (Hockney and Eastwood, 1988) with a cell edge-length equal to the interaction cutoff radius of $E/2$ in order to efficiently find the neighbors (interaction partners) of each particle. This reduces the average time complexity of the discrete convolution from $O(N^2)$ to $O(N)$ for a total of N contour particles.

If both terms in Eq. (4.8) have the same sign, the approximated Sobolev gradients on both side of the contour have opposite directions. This happens when the optimizer found an extremum of the energy. Consequently, the energy differences on one side of the contour are negative. Since the discrete representation does not allow sub-pixel deformations, the contour then oscillates.

4.6 Benchmarks and applications

We demonstrate the capabilities and limitations of the proposed topological region prior and RC minimizer by applying them to synthetic benchmark images with three different energy models. In each case, we also illustrate the practical applicability of the method to real-world images and provide computational timings. All times reported have been measured on a single 2.67 GHz Intel i7 core with 4 GB RAM using the Intel C++ compiler (v. 12.0.2). As a benchmark, we compare with iterated extended α-expansions with label costs as a region-number penalty (Delong et al., 2011). We use an 8-neighborhood with edge weights following the Cauchy-Crofton formula (Boykov and Kolmogorov, 2003), see Sec. 3.2.2.1. Compared to other multi-label optimization techniques like $\alpha - \beta$ *swaps*, α-expansions guarantee to find a solution with an energy at most twice the energy of the global minimum (Delong et al., 2011). The α-expansions are iterated in a PEARL-like manner in order to solve the joint estimation

problem of region numbers, intensities, and contours (Isack and Boykov, 2011). We choose this graph-cut-based benchmark algorithm, referred to as GC below, since it is also discrete and provides good theoretical performance guarantees. The corresponding source code was obtained from `http://vision.csd.uwo.ca/code/`. For all benchmarks we use the overall energy

$$\mathcal{E} = \mathcal{E}_{\text{data}} + \lambda \mathcal{E}_{\text{length}} + \alpha \mathcal{E}_{\text{merge}}. \tag{4.9}$$

All test cases and results are summarized in Tab. 4.1. For test cases with available ground-truth data we additionally provide two quality measures: the relative error in energy and the percentage of missclassified pixels. The latter is w.r.t. the optimal label pairing between the ground-truth and the found labels.

4.6.1 PIECEWISE CONSTANT MODEL

We first consider images comprising an unknown and arbitrary number of connected FG regions with each region having a potentially different, but constant mean intensity. After experiments using synthetic data, we present and compare results for natural scene images. We conclude this subsection with a 3D real-world application to segmenting fluorescently stained nuclei. All results in this subsection are performed using the fixed-variance Gauss-noise model presented in Sec. 3.1.1.1.

4.6.1.1 Benchmarks on synthetic data

Figure 4.6 illustrates the behavior of the present algorithm and of GC using the above energy functional on a synthetic image. The image contains 6 regions, each of which having a different, but constant mean intensity. The present algorithm is started with an initial segmentation far from the correct result and with a wrong number of initial regions (Fig. 4.6e). This demonstrates the capability of the algorithm to merge regions and to correctly delineate boundaries between touching regions. The total computational time used for this example is 0.39 s, despite the unfavorable

Optimizer	**Initialization**	**Optimizer parameters**	$\mathcal{E}_{\text{data}}$
Icecream PC, 130 × 130, Fig. 4.6			
present	6 × 6 bubbles	$\theta = 0.2, R_\kappa = 4$	PC
GC	$M = 12$	labelcost = 5	PC
GC	$M = 6$	labelcost = 0	PC
Icecream PC, 410 × 410			
present	8 × 8 bubbles	$\theta = 0.2, R_\kappa = 8$	PC
GC	$M = 12$	labelcost = 5	PC
GC	$M = 6$	labelcost = 0	PC
Icecream PC, 100 × 100 × 100			
present	5 × 5 × 5 bubbles	$\theta = 0.2, R_\kappa = 4$	PC
GC	$M = 12$	labelcost = 5	PC
GC	$M = 6$	labelcost = 0	PC
Zebrafish embryo nuclei, 512 × 512 × 39, Fig. 4.9			
present	local maxima	$\theta = 0, R_\kappa = 2$	PC
Bird, 481 × 32, Fig. 4.8a/b			
present	18 × 12 bubbles	$\theta = 0.5, R_\kappa = 8$	PC
GC	$M = 5$	labelcost = 50	PC
Icecream PS, 130 × 130, Fig. 4.10			
present	5 × 5 bubbles	$\theta = 0.2, R_\kappa = 4$	PS
GC	5 × 5 bubbles	labelcost = 20.5	PS
GC	3 × 3 bubbles	labelcost = 40	PS
Icecream PS, 100 × 100 × 100			
present	3 × 3 × 3 bubbles	$\theta = 0.3, R_\kappa = 4$	PS
GC	$M = 3$	labelcost = 20.5	PS
Zebrafish embryo germ cells, 188 × 165 × 30, Fig. 4.13			
present	bounding box	$R_\kappa = 0.04$	PS
Cloud, 481 × 32, Fig. 4.14a/b			
present	18 × 12 bubbles	$\theta = 0.2, R_\kappa = 8$	PS
GC	3 × 5 bubbles	labelcost = 175	PS
Elephants, 130 × 130, Figs. 4.8c/d and 4.14c/d			
present	18 × 12 bubbles	$\theta = 0.5, R_\kappa = 8$	PC
GC	$M = 5$	labelcost = 50	PC
present	18 × 12 bubbles	$\theta = 0.2, R_\kappa = 8$	PS
GC	3 × 5 bubbles	labelcost = 175	PS
Convolved artificial image, 49 × 72, Fig. 4.15			
present	bounding box	$\theta = 0.2, R_\kappa = 4$	DEC
Endosomes, 512 × 386, Fig. 4.17			
present	local maxima	$\theta = 0.1, R_\kappa = 2$	DEC
Endoplasmatic reticulum, 250 × 325 × 16, Fig. 4.18			
present	Otsu threshold	$\theta = 1000, R_\kappa = 4$	DEC

Table 4.1: Test cases benchmarking the present optimizer(present) against multi-label GC.

Energy parameters	**Final $\mathcal{E}$**	$\frac{(\mathcal{E}-\mathcal{E}_{\mathrm{GT}})}{\mathcal{E}_{\mathrm{GT}}}$	**misscl. px [%]**	**Iter.**	**CPU time**
Icecream PC, 130 × 130, Fig. 4.6					
$\lambda = 0.04$	71.42	4e-5	0.24	64	0.39s
$\lambda = 0.04$	71.28	**-1e-3**	**0.11**	3	0.28s
$\lambda = 0.04$	75.85	0.06	4.7	3	**0.09s**
Icecream PC, 410 × 410					
$\lambda = 0.04$	467.2	5.2e-3	0.11	110	7.34s
$\lambda = 0.04$	464.3	**-1.0e-3**	**0.04**	5	8.18s
$\lambda = 0.04$	760.8	0.63	19	3	**1.3s**
Icecream PC, 100 × 100 × 100					
$\lambda = 0.04$	1863	5.5e-3	0.11	62	57s
$\lambda = 0.04$	1844	**-4.4e-3**	**0.04**	5	76.9s
$\lambda = 0.04$	1880	0.014	2.09	5	**38.5s**
Zebrafish embryo nuclei, 512 × 512 × 39, Fig. 4.9					
$\lambda = 0.04$	*	-	-	44	7.3m
Bird, 481 × 32, Fig. 4.8a/b					
$\lambda = 0.2$	*	-	-	83	**4.06s**
$\lambda = 0.2$	*	-	-	9	8.81s
Icecream PS, 130 × 130, Fig. 4.10					
$\lambda = 0.04, \beta = 0.05, R = 8$	87.94	8.5e-4	0.07	71	**0.49s**
$\lambda = 0.04, \beta = 0.05, R = 8$	87.87	**-5.3e-5**	**6e-3**	9	10.2s
$\lambda = 0.04, \beta = 0.05, R = 8$	87.87	**-5.3e-5**	**6e-3**	8	3.47s
Icecream PS, 100 × 100 × 100					
$\lambda = 0.04, \beta = 0.05, R = 8$	4618	6.7e-4	0.01	77	**4m**
$\lambda = 0.04, \beta = 0.05, R = 8$	4615	**-2.7e-6**	**1e-3**	4	12m
Zebrafish embryo germ cells, 188 × 165 × 30, Fig. 4.13					
$\lambda = 0.08, \beta =$5e-3, $R = 4.5\mu$m	*	-	-	207	5.3m
Cloud, 481 × 32, Fig. 4.14a/b					
$\lambda = 0.2, \beta = 0.1, R = 30$	*	-	-	157	**57.77s**
$\lambda = 0.2, \beta = 0.1, R = 30$	*	-	-	16	12.3m
Elephants, 130 × 130, Figs. 4.8c/d and 4.14c/d					
$\lambda = 0.2$	*	-	-	163	**11.26s**
$\lambda = 0.2$	*	-	-	13	42.57s
$\lambda = 0.2, \beta = 0.05, R = 30$	*	-	-	385	**25.57s**
$\lambda = 0.2, \beta = 0.05, R = 30$	*	-	-	17	13.2m
Convolved artificial image, 49 × 72, Fig. 4.15					
$\lambda = 0.04$	27.13	-5e-2	1.52	53	2.3s
Endosomes, 512 × 386, Fig. 4.17					
$\lambda = 0.04$	*	-	-	41	32s
Endoplasmatic reticulum, 250 × 325 × 16, Fig. 4.18					
$\lambda = 0.15$	*	-	-	55	7.11m

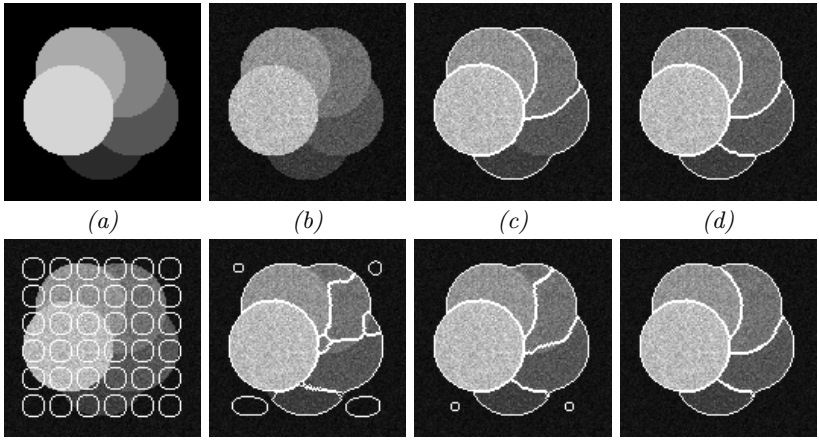
* Final energy not comparable due to different definitions of a region.

choice of initial contours.

The evolution of the total, external, and internal energies for this case is shown in Fig. 4.7. The present algorithm converges after 64 iterations. The dot symbols mark the time points at which fusions between two or more regions occurred. GC rapidly finds a solution with an energy that is slightly lower than that of the ground truth. This can be explained by the noise introducing spurious local minima in the energy landscape. The CPU times until convergence are comparable for the two algorithms. In order to test how the results scale with image size, we also consider the same problem with the image zoomed (not padded) to 410×410 pixels, and with a 3D version of the image (see Tab. 4.1). In all cases GC is sensitive to the initial number of regions (Figs. 4.6c and 4.6d) when using uniformly distributed initial region intensity estimates. With an initial number of $M = 12$ regions GC solves the problem with a CPU time comparable to the present algorithm; for $M = 6$ GC fails to find the correct segmentation. The GC implementation requires $\approx 1.76\,\mathrm{GB}$ of main memory for this 3D case; the present code uses $\approx 125\,\mathrm{MB}$.

4.6.1.2 Application to real data

We assess the real-world applicability of the present algorithm by applying it to 2D natural-scene images from the Berkeley database (Martin et al., 2001) and to a 3D confocal fluorescence microscopy image of stained nuclei in a zebrafish embryo. The results are shown in Figs. 4.8 and 4.9. In Fig. 4.8 we visually compare with GC results; the energies, however, cannot be compared due to the different definitions of what constitutes a region. The nuclei in Fig. 4.9 are small enough to justify the model of constant intensity within each nucleus. Different nuclei, however, have different intensities, e.g., arrows A and B in Fig. 4.9a, benefitting from a multi-region segmentation approach. The final label image after 44 iterations is shown in Fig. 4.9b. For better visualization, the gray-scales are the region labels rather than the estimated intensities. An overlay of the original image and the final contours is shown in Fig. 4.9c for the area highlighted by the yellow rectangle in Fig. 4.9b. Figure 4.9d shows the result when allowing region fusions, illustrating the effect of topological control during



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

(e) *(f)* *(g)* *(h)*

Figure 4.6: Synthetic example using the energy $\mathcal{E}^{\mathrm{PC}}$. (a) Piecewise constant ground-truth image. (b) Ground-truth image corrupted with Poisson noise. The 5 FG regions correspond to peak signal-to-noise ratios (SNR) of 4, 5.25, 6.5, 7.75, and 9, respectively. (c) Final result from GC when initialized with the ground-truth number of $M = 6$ regions. The GC algorithm fails due to inaccurate estimates of the region intensities. (d) Correct GC result with 6 final regions when initializing with $M = 12$ regions. (e–h) Contour evolution at iterations 0, 15, 25, and 64 of the present algorithm with contour points (particles) shown in white. The correct number of 5 connected FG regions is found.

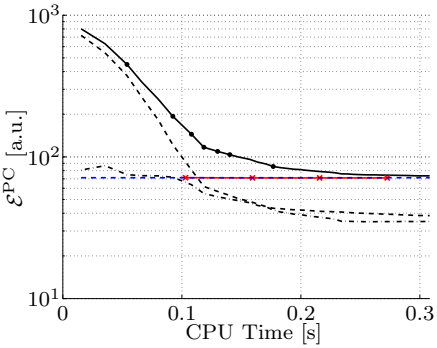


Figure 4.7: Energy evolution for both algorithms. For the present algorithm we show $\mathcal{E}_{\mathrm{length}}$ (dash-dotted), $\mathcal{E}^{\mathrm{PC}}_{\mathrm{data}}$ (dashed), and the total energy (sold). Circles mark region-fusion events. The red line shows the GC energy evolution for an initial $M = 12$. Crosses mark iterations. The residual energy of the ground-truth image is indicated by the horizontal dashed blue line.

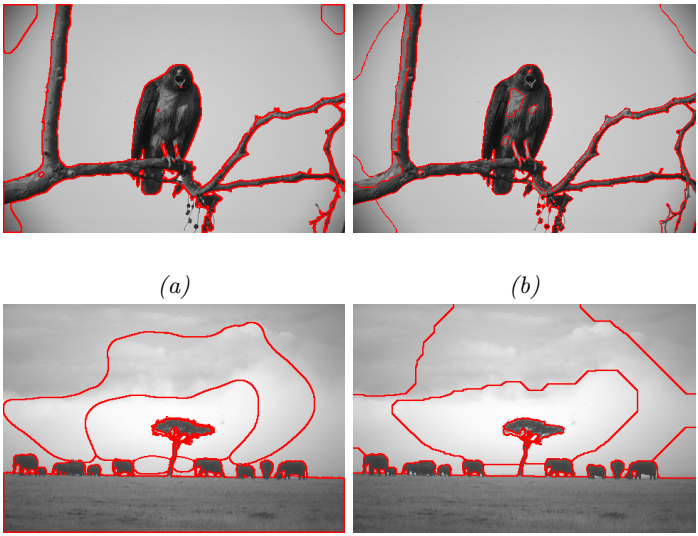
contour evolution.

4.6.2 PIECEWISE SMOOTH MODEL

We benchmark the RC algorithm on an artificial image with shaded regions. We use the piecewise smooth Gaussian-noise model with fixed variance described in chapter 3.1.3. We then present results of a 3D fluorescent microscopy image with similar features. Finally, we compare RC and GC on two 2D natural-scene images. The total energy for these benchmarks is $\mathcal{E} = \mathcal{E}^{\mathrm{PS}}_{\mathrm{LS}} + \beta\mathcal{E}_{\mathrm{balloon}} + \lambda|\Gamma|$. The last term is approximated by adding κ to $\Delta\mathcal{E}$ (see Sec. 3.2.2.2).

4.6.2.1 Benchmarks on synthetic data

Figure 4.10 illustrates the behavior of the present algorithm (Figs. 4.10a to 4.10d) on an image with linearly shaded FG and BG and compares it



(a) *(b)*

(c) *(d)*

Figure 4.8: Visual comparison on natural-scene images using $\mathcal{E}^{PC}$. (a/c) Segmentation result using the present algorithm; (b/d) using GC. GC finds 3 regions in (b) and 4 in (d). The present algorithm finds 3 connected FG regions in (a) and 9 in (c).

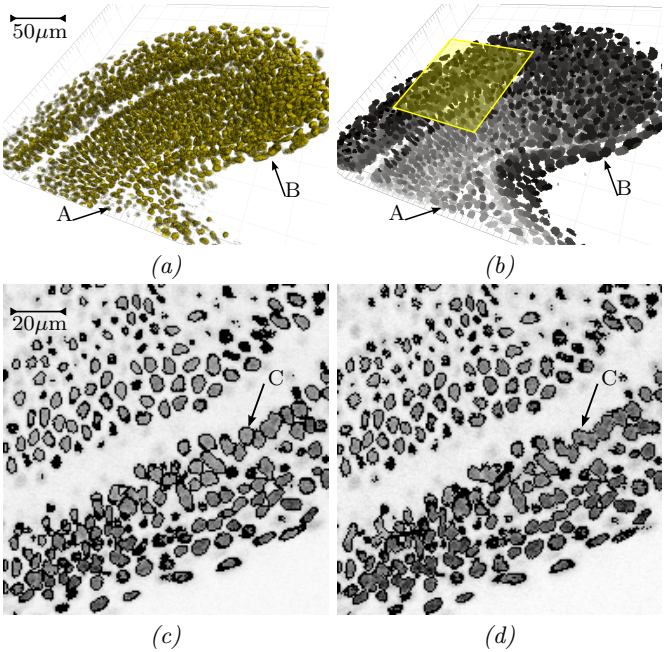
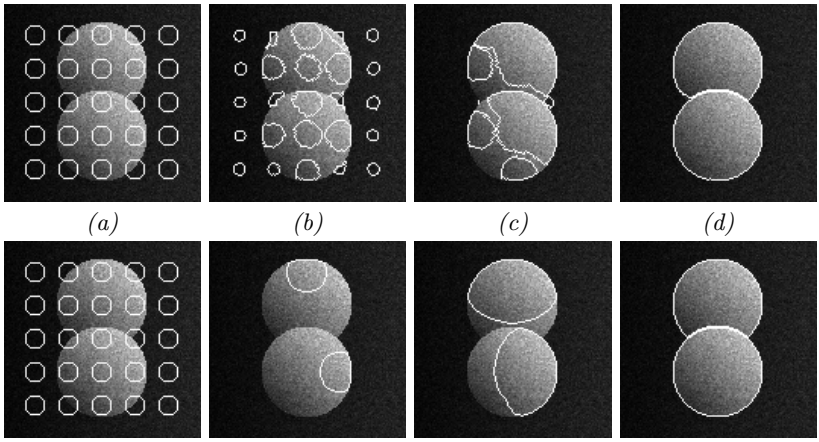


Figure 4.9: Real-world application using $\mathcal{E}^{PC}$ to segment nuclei in a zebrafish embryo imaged by confocal fluorescence microscopy. (a) Visualization of the nuclei in the raw 3D data (image: Dr. Andrew Oates and Bhavna Rajasekaran, MPI-CBG Dresden). (b) Maximum-intensity projection of the final label image L. The algorithm is initialized with small FG regions placed at all local intensity maxima after Gaussian ($\sigma_x = 5\,px$, $\sigma_y = 5\,px$, $\sigma_z = 2\,px$) blurring. The topology is fixed to the initial topology, with the exception that regions are allowed to vanish. On average $1.03 \cdot 10^6$ candidate particles are processed per iteration. 99.99% of the particles stop moving after 25 iterations. The algorithm converges after 44 iterations, finding 3218 connected FG regions. Since every connected component is a separate region with its own intensity estimate, nuclei of different brightnesses (e.g., arrows A and B) are correctly segmented. (c) Magnified z-plane showing an overlay of the original image with the final contours (black) in the region highlighted in yellow in (b) (intensities inverted for display purposes only). Touching nuclei are not fused if region merges are disallowed during contour evolution. (d) Allowing regions to merge, touching nuclei of similar intensities are assigned to the same region (e.g., arrow C) and the final number of connected FG regions is 1452. The visualizations in (a) and (b) were done using Imaris by Bitplane, Inc.



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

(e) *(f)* *(g)* *(h)*

Figure 4.10: Synthetic example using the energy $\mathcal{E}^{\mathrm{PS}}$. Two overlapping linearly shaded disks on a linearly shaded background, corrupted with Poisson noise. The brighter parts of the disks (top right) approximately correspond to a peak SNR of 8.7, while the low-intensity parts (bottom left) have SNR ≈ 3.2. (a–d) Contour evolution at iterations 0, 5, 15, and 70 of the present algorithm. The correct number of 2 connected FG regions is found. (e–h) Evolving contour at iterations 0, 1, 4, and 9 of the GC algorithm, also finding the correct number of regions.

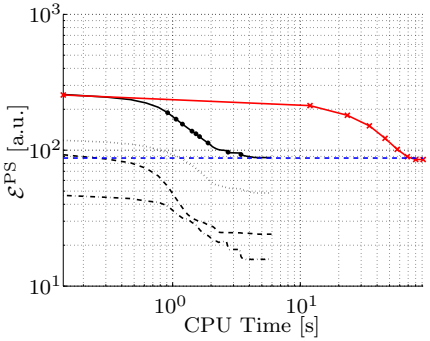


Figure 4.11: (i) Energy evolution for the two algorithms. For the present algorithm we show the evolution of $\mathcal{E}^{PS}_{\mathrm{data}}$ (dashed), $\mathcal{E}_{\mathrm{length}}$ (dash-dotted), $\mathcal{E}_{\mathrm{balloon}}$ (dotted), and of the total energy $\mathcal{E}^{PS}$ (solid). Dots mark region-fusion events. The red line shows the energy for GC. Crosses mark iterations. The residual energy of the ground-truth image is indicated by the horizontal dashed blue line.

with GC (Figs. 4.10e to 4.10h). In the high-SNR areas, the data term of the energy dominates the evolution, and the contours immediately stick to intensity edges. Within the shaded FG circles, the regions expand as driven by the balloon force. After 5 iterations, regions that are not separated by large intensity gradients begin to merge.

The present algorithm is robust with respect to different choices of the patch radius R. However, R should be chosen smaller than the length scale of intensity variations and large enough such that $|S^R|$ constitutes a representative sample to construct the local intensity histograms P.

Figure 4.11 shows the evolution of all energy terms for the present example. When initialized with 25 bubbles as shown, GC is about 20 times slower than the present algorithm, since it evaluates the energy everywhere in the image, whereas the present algorithm evaluates it only at the particles. Both methods find solutions close to the ground truth and correctly estimate the number of regions.

The results for a 3D version of the image in Fig. 4.10 are given in Tab. 4.1. In the 3D case, GC is initialized with the ground-truth number of regions

and an initial contour close to the ground-truth solution in order to keep CPU times reasonable. The present algorithm is again initialized with bubbles.

4.6.2.2 Application to real data

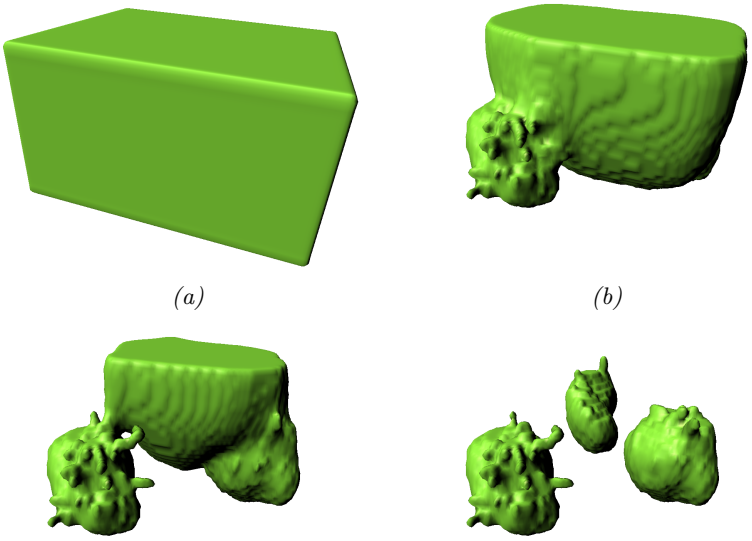
Real-world applications of the present image model are shown in Figs. 4.13 and 4.14. The data consist of a 3D confocal image of primordial germ cells in a zebrafish embryo (Fig. 4.13a) and 2D natural-scene images from the Berkeley database (Fig. 4.14) (Martin et al., 2001). The difficulty in segmenting these images is that the intensity is inhomogeneous within each object, as illustrated in Fig. 4.13b. Also the background is inhomogeneous in all images, requiring a piecewise smooth model. Figure 4.12 shows the surface evolution for the primordial germ cells image. The final segmentations obtained with the present algorithm are shown in Figs. 4.13c, 4.14a, and 4.14c. The segmentations using GC are shown in Figs. 4.14b and 4.14d. Comparing Figs. 4.14c/d with Figs. 4.8c/d illustrates the difference between a piecewise constant and a piecewise smooth image model.

4.6.3 Deconvolution model

We test the algorithm on blurred data using the deconvolving model in Eq. (3.13) with external energy $\mathcal{E}_{\text{data}} = \mathcal{E}_{\text{LS}}^{\text{PC,dec}}$. We first consider synthetic data and then provide a 2D real-world example of segmenting fluorescently labeled endosomes.

4.6.3.1 Benchmark on synthetic data

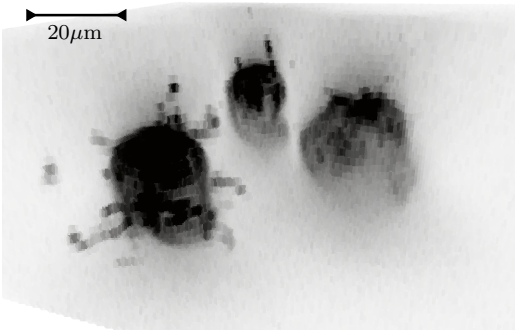
Figure 4.15 illustrates the behavior of the present algorithm using the deconvolving energy functional on a synthetic image. The image simulates a realistic scenario in fluorescence microscopy with a pixel size of 80 nm and a half-width of the PSF of 120 nm. The image blurred by the PSF (Fig. 4.15b) corrupted with Poisson noise (Fig. 4.15c) with a peak SNR



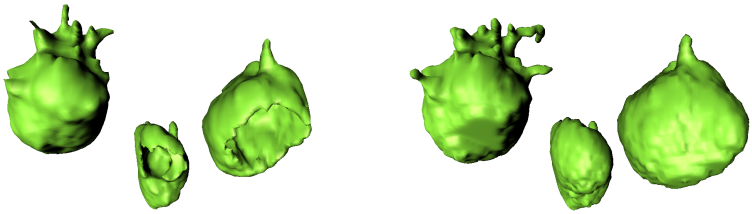
(a) *(b)*

(c) *(d)*

Figure 4.12: Top view of the 3D contours minimizing the energy $\mathcal{E}^{PS}$ after 50 iterations (b), 100 iterations (c), and 257 iterations (d). The algorithm is initialized with a box-shaped region (a). The length-term coefficient λ is set to 0.08, the balloon force coefficient to $\beta = 0.005$, and the local patch radius to $R = 4.5\,\mu m$ (corresponding to $9 \times 9 \times 3$ voxels). After 208 iterations oscillations are detected and after 257 iterations the process converges. The average CPU time per iteration is 1.25 s, resulting in a total processing time of 5.3 minutes. All visualizations in this figure were done using Imaris 7.2 by Bitplane, Inc.

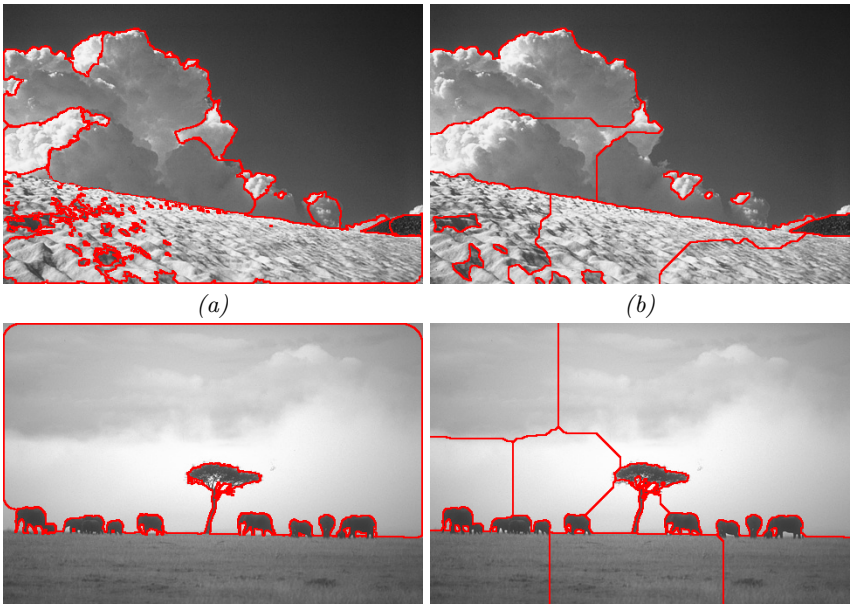


(a)



(b) *(c)*

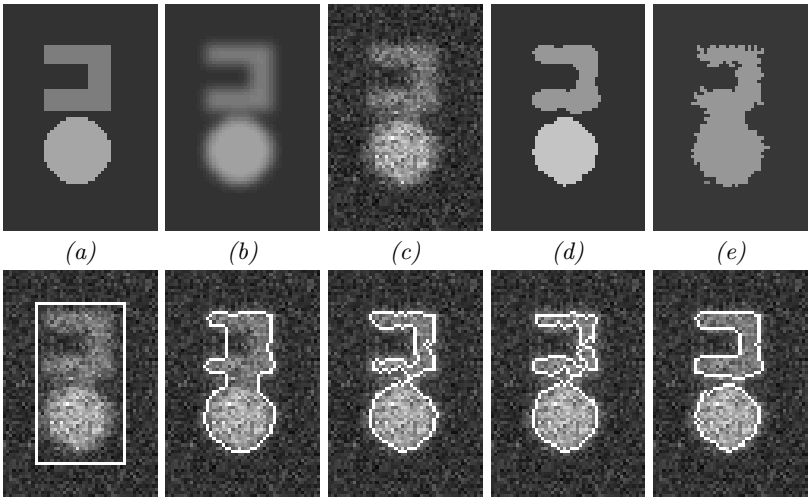
Figure 4.13: Real-world application using $\mathcal{E}^{PS}$ to segment primordial germ cells in a zebrafish embryo. (a) The raw 3D confocal image showing 3 cells with a fluorescent membrane stain (image: Mohammad Goudarzi, University of Münster). Intensities are inverted for display purposes only. (b) Intensity isocontour illustrating the inhomogeneity of the objects (bottom view). (c) Final segmentation using the present algorithm with $\mathcal{E}^{PS}$ (bottom view). The algorithm is initialized with a single box-shaped contour encompassing all objects and ultimately finds 3 connected FG regions. Visualizations were done using Imaris by Bitplane, Inc.



(a) *(b)*

(c) *(d)*

Figure 4.14: Visual comparison on natural-scene images using $\mathcal{E}^{PS}$. (a/c) Segmentation result using the present algorithm; (b/d) using GC. GC finds 6 regions in (b) and 9 in (d). The present algorithm finds 17 connected FG regions in (a) and 14 in (c).



(a) *(b)* *(c)* *(d)* *(e)*

(f) *(g)* *(h)* *(i)* *(j)*

Figure 4.15: Synthetic example using the energy $\mathcal{E}^{dec}$. (a) The ground-truth image. (b) The image convolved with a Gaussian point-spread function with $\sigma = 1.75\,px$, modeling a confocal fluorescence microscope. (c) The blurred image after addition of Poisson noise. The intensity of the u-shaped object corresponds to a peak SNR of 3, that of the circular region to an SNR of 4. (d) The reconstructed image using the deconvolving model. (e) The reconstructed image using a piecewise constant model with $\lambda = 0.1$, $\theta = 0.8$. (f–j) The contour after 1, 10, 20, 35, and 53 iterations, finding the correct number of 2 connected FG regions. The algorithm reduces the step size upon detecting oscillations after 30 and 77 iterations.

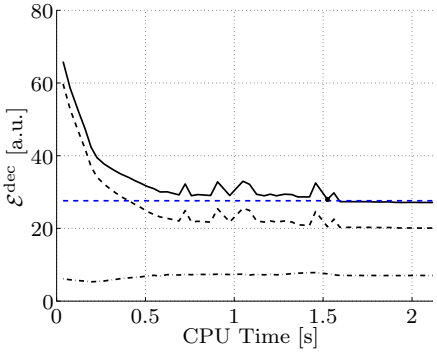


Figure 4.16: Energy evolution for the deconvolving model. The solid line represents the total energy, the dashed line $\mathcal{E}^{dec}_{\text{data}}$, and the dash-dotted line $\mathcal{E}_{\text{length}}$. The dot symbol indicates a region-merging event. The residual energy of the ground-truth image is indicated by the horizontal dashed blue line.

of 3 and 4 for the dimmer and brighter object, respectively. The width of the gap between the objects is equal to the half-width of the PSF.

Without using the information of how many objects are represented in the image, we start the segmentation from a single, rectangular initial contour (Fig. 4.15f). Figures 4.15f to 4.15j show the evolution of the contour. Since the area of the circle is larger than the area of the u-shaped object, the intensity estimate is initially dominated by the circle. This causes initial over-segmentation of the u-shaped object. At iteration 19, the lower region splits into two regions with independent intensity estimates. This causes the regions segmenting the u-shaped object to merge again, resulting in a correct detection in the end. Figure 4.16 shows the evolution of the energies during this segmentation process.

We compare the results with those obtained using the piecewise constant energy without deconvolution. The corresponding final reconstruction is shown in Fig. 4.15e. The PC model is not able to separate the two objects. It is moreover necessary to set λ to be 10 times larger than for the deconvolving energy in order to prevent overfitting the blurry object boundaries with many small regions.

4.6.3.2 Application to real data

A deconvolving energy functional is particularly useful when segmenting near-diffraction-limited objects as they occur, e.g., in intra-cellular imaging. We illustrate this in Fig. 4.17 using a single plane of a 3D confocal image showing endosomes labeled with fluorescent Rab5 protein (Helmuth et al., 2009). Endosomes are small membrane-bound organelles of about 20 to 200 nm size. Accurately reconstructing the outlines of the many blurred, dense objects in this image is challenging when not accounting for the microscope PSF. Here we use a simple Gaussian model PSF whose width is determined by fitting it to point-like structures in the image. A separate measurement of the actual PSF of the microscope was not performed. Initially, we place small circular contours around every local intensity maximum in the image. These contours then rapidly evolve to concentrate around the endosomes. The number of regions in the image does not need to be known when initializing the algorithm. This is an advantage over explicit deconvolving active contours (Helmuth and Sbalzarini, 2009). Explicit deconvolving active contours, however, provide sub-pixel resolution, whereas the present method is limited to pixel-level accuracy. This prevents the correct detection of objects covering less than 2 pixels. After 73 iterations, the algorithm converges to the reconstructed model image shown in Fig. 4.17b. The original image overlaid with the final outlines in the region indicated in Fig. 4.17a is shown in Fig. 4.17c. The two touching objects in the lower-right corner are properly separated based on their different intensities.

We apply RC using a deconvolving energy $\mathcal{E}^{\mathrm{PC,dec}}_{\mathrm{Poisson}}$ to a 3D image of an endoplasmatic reticulum (ER). ERs also are near-diffraction-limited subcellular structures. We compare joint deconvolution and segmentation to a PC segmentation without deconvolution in Fig. 4.18. The segmentation without deconvolution tends to include out-of-focus light into the foreground object. The deconvolving energy therefore causes the contour to resolve fine structures in more detail, as shown in Fig. 4.18d. Also, the objects are usually smaller. Consequently, small isolated structures may be missed since the curvature regularization acts stronger on smaller objects. For the sake of comparison, however, we here used the same prior hyperparameter $\lambda = 0.15$ for both cases. We approximate the PSF by a

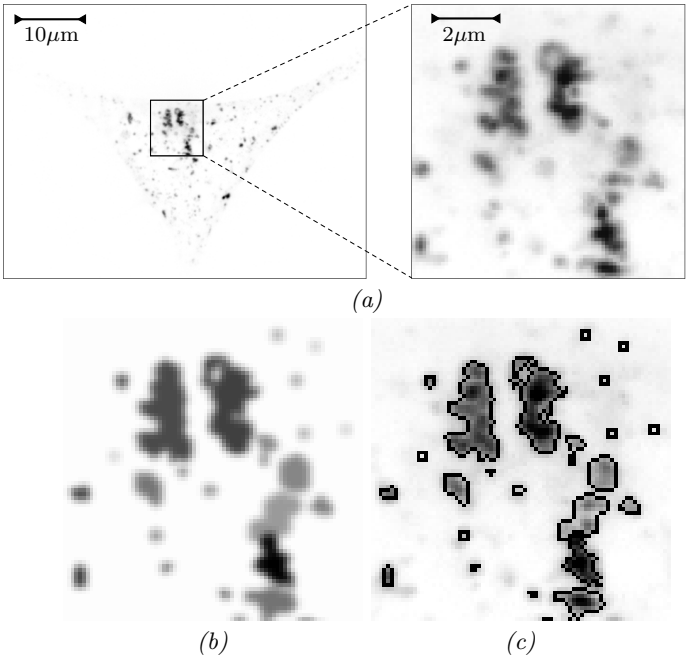


Figure 4.17: Real-world application of the deconvolving model to fluorescently labeled endosomes in live HER911 cells. (a) Confocal fluorescence microscopy image after background subtraction using a rolling-ball algorithm (image: Prof. Urs Greber, University of Zurich, and Dr. Christoph Burckhardt, Harvard University). (b) Final reconstructed image in the inset window shown in (a). (c) The final contours (black pixels) overlaid onto the original image data. Starting from 1541 spherical FG regions centered at local intensity maxima, the algorithm finds 72 connected FG regions. We approximate the PSF by a Gaussian with $\sigma = 1.011\,px$, found by fitting to signals of point-like structures in the image. Intensities are inverted for display purposes only.

Gaussion with $\sigma_{x,y} = 1\,\text{px}$ and $\sigma_z = 1.5\,\text{px}$. We stop the algorithm after 50 iterations, which needs 7.11 minutes for the image of size $250 \times 325 \times 16$ pixels.

4.6.4 Cell segmentation with spherical shape prior

Figure 4.19 shows the effect when using the global shape prior discussed in Sec. 3.2.3. We first segment the images without shape prior. Regions are initialized at local intensity maxima after blurring. We then use the obtained segmentations as initialization for a second segmentation with shape prior. We used a circle as a template shape.

4.6.5 Applications using Sobolev gradient approximations

We compare RC results using approximated L^2 gradients with RC results using approximated Sobolev gradients (Sec. 4.5). We first challenge both approaches using an artificial deconvolution problem. Figure 4.20 shows the artificial data, intermediate results, and final results for both approaches. We always initialize with 25 bubbles on a cartesian grid. During the first 15 iterations, regions evolve almost identically for both approaches. After 59 iterations, the Sobolev-gradient approach is closer to the final solution. After 88 iterations, it starts oscillating and therefore falls back to the L^2-gradient mode. It converges after 282 iterations. The L^2-gradient approach converges after 500 iterations. The results for both algorithms are shown in Figs. 4.20h and 4.20l. The average iteration times are 0.74s with and 0.73s without Sobolev gradients. They are almost identical because of the deconvolving energy calculations , i.e., computing the L^2-gradient approximations is computationally more expensive than computing the Sobolev-gradient approximations.

In Fig. 4.21 and Fig. 4.22 we show real-world data with low SNR. Both examples illustrate that Sobolev gradients maintain smooth contours, resulting in less noise-sensitive segmentations. The L^2-gradient result requires stronger curve-length penalization in order to regularize the contour. However, this fails for the example in Fig. 4.21 as regions start to collapse when

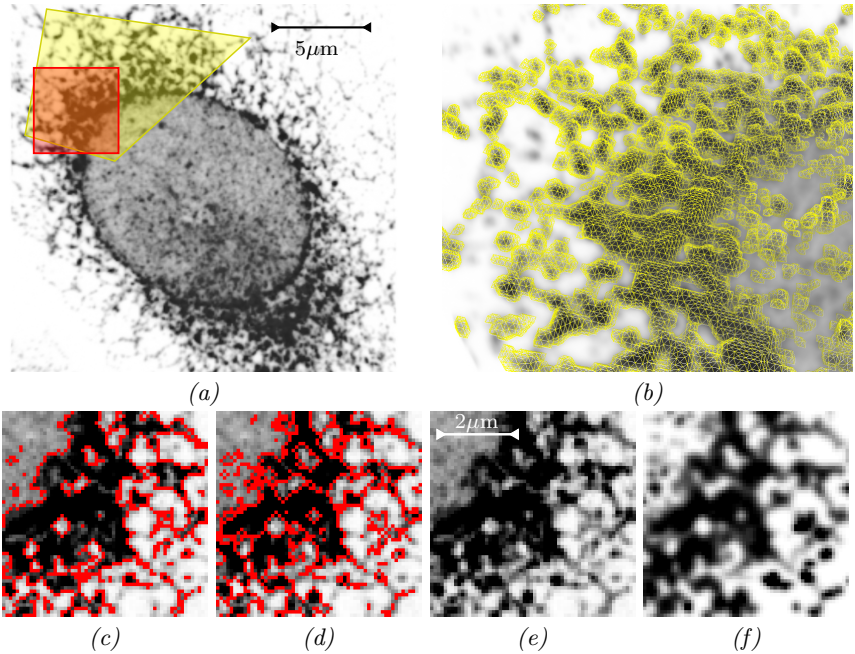


Figure 4.18: Piecewise constant segmentations of ER data with and without deconvolution model. (a) Confocal fluorescence microscopy image (image: Helenius group, ETH Zürich). (b) Data with segmentation contour (yellow mesh) when using $\mathcal{E}^{PC,dec}_{Poisson}$ as the external energy. The illustrated region corresponds to the yellow frustum shown in (a). (c) 2D slice with segmentation contour using the external energy $\mathcal{E}^{PC}_{Poisson}$. The region corresponds to the region indicated by the red window in (a). (d) Same as (c) but segmentation using the deconvolving external energy $\mathcal{E}^{PC,dec}_{Poisson}$. (e) Original data of the region indicated in red. (f) Reconstructed image J of the segmentation shown in (b) and (d). Intensities have been inverted for display purposes only.

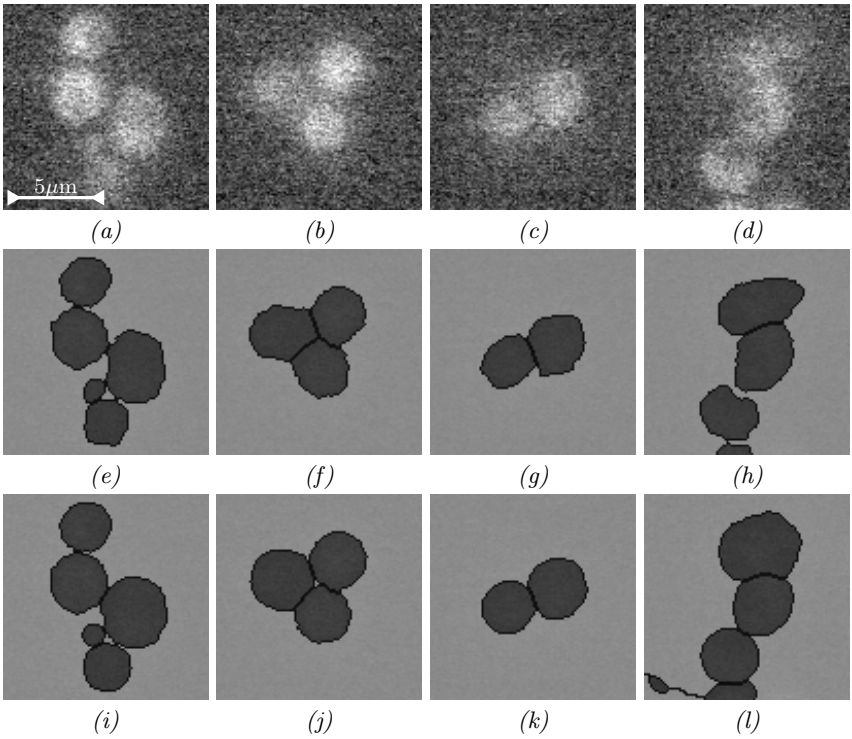
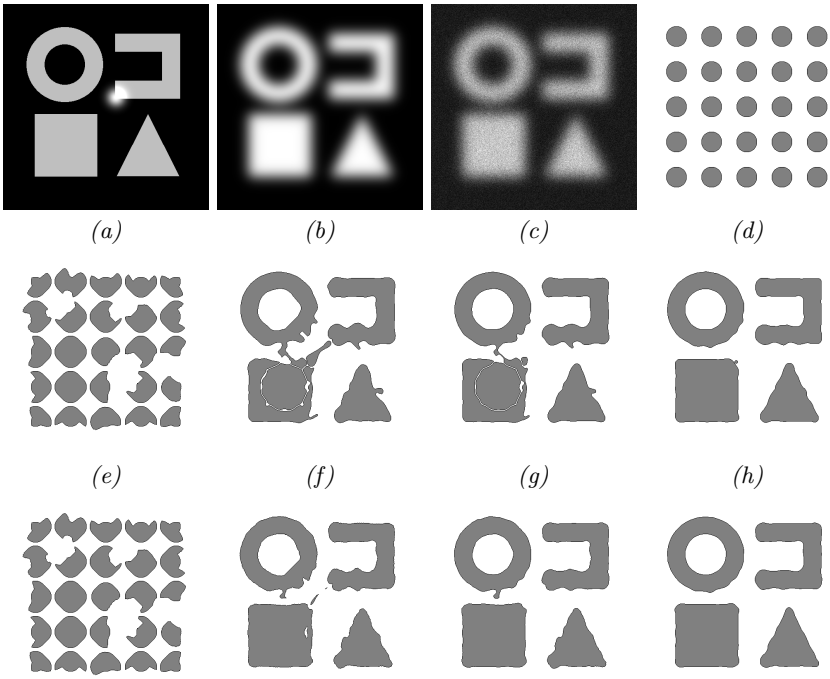


Figure 4.19: Fluorescently labeled cells (image: Michael Unger, BISON Group, ETHZ) segmented with and without global spherical shape prior. The topology has been fixed during contour evolution. (a–d) shows the low-SNR raw data. (e–h) shows the segmentation results without global shape prior. (i–l) shows the segmentation results with global shape prior. The lower-most region in (l) can not satisfy the spherical prior due to the image boundary and tries to compensates in another direction.

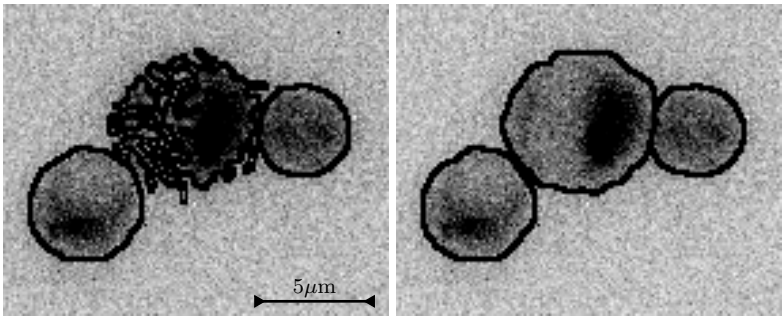


(a) *(b)* *(c)* *(d)*

(e) *(f)* *(g)* *(h)*

(i) *(j)* *(k)* *(l)*

Figure 4.20: Region evolution using L^2 and Sobolev gradient approximations for the deconvolving energy $\mathcal{E}_{LS}^{PC,dec}$. (a) Ground-truth data with overlaid Gaussian point-spread function of width $\sigma = 15$ pixels. (b) Convolved image. (c) Convolved image with Poisson noise of SNR 6. (d) Initialization for both algorithms. (e–h) Regions after 15, 59, 88, and 500 iterations using the L^2-type gradient. (i–l) Regions after 15, 59, 88, and 282 iterations using the Sobolev-type gradient.



(a) (b)

Figure 4.21: Segmentation results of fluorescently labeled cells using the energy $\mathcal{E}_{LS}^{PC}$. Topology is locked and the regions are initialized at local intensity maxima after image blurring. (a) Segmentation when using L^2-gradient approximations. (b) Segmentation when using Sobolev-gradient approximations.

λ is increased. Similarly, a stronger length regularization fragments the segmentation of the membrane, especially in domains where the membrane in Fig. 4.22d has weak signal. The runtime using Sobolev gradients, however, is smaller only for the example in Fig. 4.21, where a more expensive energy is used. In this example the algorithm using approximated L^2 gradients does not converge after 200 iterations (43 s). When using Sobolev gradients, the algorithm converged after 93 iterations (16 s). In the example in Fig. 4.22, however, the algorithm with approximated L^2 and Sobolev gradients needs 106 iterations (254 s) and 200 iterations (549 s), respectively.

An additional segmentation example using Sobolev gradients for images of proton emission patterns is shown in appendix B.2.

4.7 Discussion and conclusions

We have presented a discrete multi-region-competition framework based on the topological constraint that each foreground region has to correspond to a connected set of pixels in some discrete geometry representation. An energy-minimization algorithm that accounts for this topological

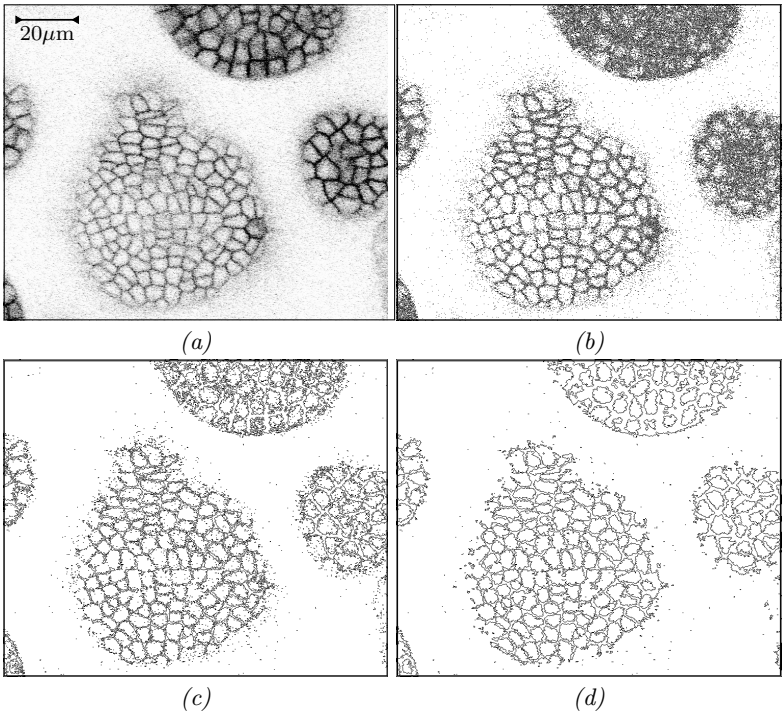


Figure 4.22: Shoot apical meristems of Arabidopsis labeled with plasma membrane localized fluorescent protein. (a) Data image of size 756×622 *pixels (image: Liu et al. (2010)). (b) Initial contour from Otsu thresholding. (c) Segmentation contour using approximated* L^2 *gradients. (d) Segmentation contour using approximated Sobolev gradients.*

constraint has been implemented in both 2D and 3D and tested using three popular energy functionals. The number of regions in an image does not need to be known *a priori*, and the initial segmentation can have a different topology than the final result. We have presented a novel discrete contour-propagation scheme. We enforced the topological region definition and provide optional control over region merging and splitting during contour evolution. The contours are represented by computational particles that evolve as driven by the energy-minimizing flow. Like discrete level-set methods (Shi and Karl, 2008), the present algorithm only requires evaluations of the energy functional, but not of its gradient. This is beneficial given the non-differentiable topological constraint. Contour oscillations are suppressed by adaptive step-size reduction in the rank-based minimization algorithm.

We have shown that the chosen particle-based approach allows to approximate Sobolev gradients using efficient particle–particle interactions. We demonstrated that the presented approximations allow to benefit from three Sobolev-gradient-based optimization properties. First, less length regularization is needed as Sobolev gradients naturally regularize the contour evolution. This allows more detailed segmentations for low-signal images. For the same reason, Sobolev gradients are beneficial for particularly noisy data. Third, the Sobolev-gradient-based optimization path often needs less iterations. For expensive energies this may amortize the Sobolev-gradient approximation computation and therefore may lead to better performance.

We illustrated the algorithm on synthetic images and demonstrated its applicability to real-world data using three different energy functionals. We compared with results obtained using a state-of-the-art discrete energy minimizer based on multi-label graph cuts (GC) (Delong et al., 2011). The first energy represented a piecewise constant image intensity model. The second functional used a piecewise smooth image model to allow for inhomogeneous intensity distributions within regions. Applying the method to a deconvolving energy unites image deconvolution and segmentation and extends explicit deconvolving active contours (Helmuth and Sbalzarini, 2009) to handle topological changes during energy minimization and to higher-dimensional images.

The benchmarks demonstrated that the solution quality and the runtime of the present algorithm are competitive. Compared with GC, the present algorithm is particularly competitive for large region numbers and for costly energy functionals, such as piecewise smooth energies. GC rapidly finds low-energy solutions. For alternating geometric/photometric optimization of multi-region energies, however, the large geometric step sizes of the GC-based PEARL algorithm may lead into wrong energy funnels. The resulting segmentations then wrongly delineate regions, as for example in Fig. 4.6c. If GC correctly estimates the intensities, it typically finds better solutions than RC in terms of energy and missclassified pixel, because RC converges to local minima.

Due to the discrete contour representation, the present method is limited to single-pixel accuracy. Sub-pixel accurate segmentations, such as those achieved by explicit deconvolving active contours (Helmuth and Sbalzarini, 2009) would require continuously varying particle positions, hampering the efficient solution of the energy-minimization problem and the application of digital topology.

A limitation of the present method compared to GC is that contours can only advance at most one pixel per iteration. For initial contours far from the final solution, segmentation may hence be slow. Nevertheless, the timings of the present implementation as reported for each test case are encouraging when compared with GC. The computational cost of the algorithm depends on the energy functional to be minimized. In the example of Fig. 4.13, evaluations of the energy functional accounted for 88% of the computational time (66% for $\mathcal{E}^{\mathrm{PS}}_{\mathrm{data}}$, 22% for the curvature-regularizing flow), whereas topology processing took 1%, contour propagation 4%, and data-structure update 3%. Table 4.2 shows this breakdown of the computational cost for each of the three energy functionals considered. For the PC model, curvature approximation and contour propagation are the most expensive parts. This is due to lookups in L and I. For the same reason, the computational time using the PS model is dominated by evaluating the data energy. For the deconvolving energy functional, pre-computing the model image J dominates the processing time. The time complexity of the algorithm with the PC and PS image models is $O(|\tilde{\Gamma}|)$, i.e., linear in the total number of particles and independent of the image size. For the deconvolving energy functional, however, the convolution renders the

	PC	PS	DEC
evaluating $\mathcal{E}_{\text{data}}$	1%	**66**%	**97**%
evaluating $\mathcal{E}_{\text{length}}$	**31**%	22%	<1%
optimization	21%	4%	2%
contour propagation	**31**%	4%	<1%
topology processing	4.5%	1%	<1%
data-structure update	11.5%	3%	<1%

Table 4.2: Relative computational costs of the different steps of the algorithm for the three energy functionals (PC, PS, dec) considered here. All times were measured using the respective biological example images.

complexity dependent on the image size as $O(|\Omega| \log |\Omega| + |\tilde{\Gamma}|)$. The computational performance of the present implementation could for example be improved by storing the image data along a space-filling curve, which is expected to improve cache efficiency, as points that are close in the image will also be close in memory.

Faster executions can be achieved using general-purpose graphics processing units (GPGPU) for the energy-difference computation. GPGPUs are highly parallel streaming multi-processors. Ebrahim (2011) implemented the energy $\mathcal{E}_{\text{LS}}^{\text{PS}}$ and κ computation (see Sec. 3.2.2.2) using the OpenCL API (Stone et al., 2010), which aims at standardizing programming heterogeneous parallel hardware. Ebrahim (2011) benchmarked the GPGPU implementation running on a nVidia Tesla C2050 against RC running on an Intel Xeon 3.2 GHz CPU for 2D and 3D images. For 3D images the overall speedup gained using the GPU was 20. Interestingly, the speedup was mainly due to the spatial locality caching of the GPU's texture memory rather than processing power. In 2D, this caching effect is weaker. Nevertheless, speedups of 60% were reported. Future work includes supplementing the work by (Ebrahim, 2011) by implementing more energies as OpenCL kernels and testing these on different hardware platforms.

CHAPTER

FIVE

Efficient shape sampling with Markov-chain Monte Carlo

In the previous chapter we jointly estimated the number of regions, their contours, and their intensities. We introduced a heuristic probability density optimizer w.r.t. region contours. We hence found a (local) maximum-a-posteriori (MAP) estimate. The method suffers from various shortcomings. First and most importantly, the solution found may be a local maximum. The method is therefore sensitive to the initialization and regularization parameters. Second, even if the global optimum is found, we do not get any information about the sensitivity or robustness of that solution. Third, in particular when segmenting fine structures, the approach in chapter 4 suffers from the object representation being limited to pixel resolution. This resolution limit might cause oscillations that we need to detect, which is difficult, and handle, which is expensive.

Here, instead of optimizing an energy functional, we *sample* from the posterior distribution $p(L|I)$ induced by the energy. The term sampling refers to drawing multiple samples from the distribution. The goal is to find a representative collection of segmentations $\mathcal{L} = \{L_i\}$ representing the distri-

bution with pdf $p(L|I)$. In order to do so, we use MCMC methods. More precisely, we design a Metropolis-Hastings algorithm.

Due to the stochastic nature of MCMC methods, the deformable contour is able to overcome local probability maxima. Also, since we are not aiming for the maximum only, but for a full description of the posterior, we gain additional insight into the robustness of the MAP solutions. Fan et al. (2007) successfully characterized multi-modal distributions using shape sampling for image segmentation. Investigating the mode associated to a MAP solution allows characterizing the robustness of a particular segmentation. With a representative sample from the posterior in hand we can also assess marginals, such as the *probability of boundary* (Chang and Fisher III, 2011) or confidence-band estimates for the segmentation probability. The image-formation model needs to be exact though, else such confidence bands are lower bounds.

An important difference to the optimization problem in the previous chapter is that we here fix the number of regions. The task of sampling a segmentation space with unknown number of regions is very difficult and is not considered. In chapter 4 we defined a region as a connected component. This, together with an initialization, was sufficient to determine the final number of regions. Since we now fix the number of regions, we do not need to satisfy such a topological constraint. Nevertheless, we also present a method to sample topologically constrained shapes.

The next section gives a brief overview of Markov chain theory to make this chapter self-contained. We then review related works on shape sampling. In Sec. 5.3 we present a novel discrete method, and its implementation, to sample target distributions for probabilities introduced in chapter 3. In Sec. 5.4 we apply the algorithm to synthetic and real-world data for different image models. We benchmark the algorithm against a state-of-the-art method (Chang and Fisher, 2012). We conclude this chapter with a discussion and outlook.

5.1 Introduction to MCMC

We are given $\pi(\cdot)$, a distribution that we do not know explicitly, but we're able to evaluate (query) an unnormalized version of π. For example, π could be the posterior $p(L|I)$ of segmentations given the image I and a Bayesian image model (see chapter 3). Given the image model and I, we can query the (unnormalized) probability of a particular segmentation L. Our goal is to sample from π in order to get a representative collection $\mathcal{L}$. The idea of MCMC methods is to construct a stochastic process, a Markov chain, whose visited states $\{\boldsymbol{x}_t\}$ mimic a sample from the target distribution π. We hence design a Markov chain such that its equilibrium distribution is π. Then, we simulate the Markov chain for a long time and store all visited states. Using this sample we are able to compute the desired statistics.

The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) is a MCMC method. The next section shows that this algorithm, by construction, produces a Markov chain that is *reversible*, *aperiodic*, and *irreducible*. These are sufficient conditions for converging to the correct equilibrium distribution π (Smith and Roberts, 1993). Rigorous mathematical derivations are beyond the scope of this thesis. A good and thorough overview of MCMC can be found in Brooks (1998). Here we reproduce only the necessary theorems and give references to the corresponding proofs. The first subsection summarizes Markov theory. The second one explains why the Metropolis-Hastings algorithm produces a chain with equilibrium distribution equal to the target distribution.

5.1.1 Excerpt of Markov chain theory

The definitions and theorems in this subsection were selected or adapted from Bremaud (1999). Proofs of all theorems stated in this subsection can be found therein.

In the context of this chapter we focus on discrete state spaces. Let Ξ be the finite state space of a Markov chain. This is, the state-transition probabilities can be written in matrix form. The element $p_{ij} = p(\boldsymbol{x}_{t+1} =$

$j|\boldsymbol{x}_t = i)$ is the probability for the chain to move from state i to state j. We also restrict ourselves to homogeneous Markov chains (HMC), i.e., chains for which the transition probabilities (in the discrete case the *transition matrix* or *transition graph*) are independent of time:

Definition 6 (Homogeneous Markov Chain)**.** *Let $\{x_n\}$, $n \leq 0$, be a discrete-time stochastic process with countable state space Ξ. If for all integers $n \leq 0$ and all states $i_0, i_1, \ldots, i_{n-1}, i, j$,*

$$p(\boldsymbol{x}_{n+1} = j|\boldsymbol{x}_n = i, \ldots, \boldsymbol{x}_0 = i_0) = p(\boldsymbol{x}_{n+1} = j|\boldsymbol{x}_n = i) \tag{5.1}$$

whenever both sides are well-defined, this stochastic process is called a Markov chain. It is said to be a homogeneous Markov chain if in addition, the right-hand side of (5.1) is independent of n.

The exclusive dependence on the past through the predecessor state is called the *Markov property*. The distribution of a discrete-time HMC at a certain time t is a probability vector of length $|\Xi|$ with the i-th entry the probability of being in state i after t iterations. Due to the Markov property, the distribution of a Markov chain is completely determined by the initial distribution of the states in Ξ and the transition matrix. When running the stochastic process sufficiently long, this probability vector may converge. If so, the distribution to which the chain converges is called the *stationary-*, the *equilibrium-*, or the *steady-state* distribution:

Definition 7 (Stationary Distribution)**.** *A stationary probability distribution π satisfies for all states i*

$$\pi(i) = \sum_{j \in \Xi} \pi(j) p_{ji}. \tag{5.2}$$

From this definition the following theorem immediately follows:

Theorem 3 (Steady State)**.** *A chain started with a stationary distribution is stationary.*

Three key questions are: 1. Will a certain Markov chain ever converge to a stationary distribution? 2., If so, to which distribution?; there might

be multiple, and 3., In what sense does the chain converge, and are we allowed to compute averages over $\mathcal{L}$?".

In order to formulate necessary conditions for convergence to exactly one equilibrium distribution, we use *irreducibility*, *positive recurrence*, and *aperiodicity*.

Irreducibility is a characteristic of the transition graph. Informally, it guarantees that the entire state space is accessible from every state. Irreducibility implies that the stationary distribution is unique, provided it exists.

Definition 8 (Communication)**.** *A state j is said to be accessible from state i if and only if there is at least one path $i, i_1, \ldots, i_{m-1}, j$ from i to j such that*

$$p_{ii_1} \cdot p_{i_1 i_2} \cdots p_{i_{m-1} j} > 0.$$

States i and j are said to communicate if i is accessible from j and j is accessible from i.

The communication relation is reflexive, symmetric, and transitive and is therefore an equivalence relation. It generates a partition of Ξ into disjoint equivalence classes called *communication classes*. In other words, within a communication class all possible pairs of states communicate. And, if two states communicate they belong to the same communication class.

Definition 9 (Irreducibility)**.** *If there exists only one communication class, then the chain is said to be irreducible.*

In other words, a chain is irreducible if it is possible to access every state from every other state. Irreducibility hence allows accessing the whole state space, irrespective of the initial state.

Irreducibility, however, does not guarantee that the chain visits a certain state twice in finite time. Summary statistics on $\mathcal{L}$ may therefore be invalid. We have to ensure that such statistics exist.

Recurrence For irreducible HMC *recurrence* is a good notion for stability. As we will see later, this allows us to compute the desired statistics. Recurrence bounds the *return time* T_i, i.e., the time needed by a chain starting in state i to return to state i:

Definition 10 (Recurrence)**.** *State $i \in \Xi$ is called recurrent if*

$$p(T_i < \infty) = 1.$$

A recurrent state $i \in \Xi$ is called positive recurrent if $\mathbb{E}[T_i] < \infty$.

Recurrence is easy to prove if Ξ is finite:

Theorem 4. *An irreducible HMC with finite state space is positive recurrent.*

From the following equivalence theorem

Theorem 5 (Stationary Distribution Criterion)**.** *An irreducible HMC is positive recurrent if and only if there exists a unique stationary distribution.*

and theorem 4 we can answer question 2: An irreducible chain in a finite state space is recurrent and hence, if at all, converge to a unique equilibrium distribution.

Aperiodicity and convergence to steady state We have not yet discussed sufficient conditions for a chain to converge to an equilibrium distribution. It can be shown (via coupling of Markov chains) that a HMC of period one will eventually converge. We therefore use the concept of periodicity:

Definition 11 (Arithmetic Definition of Period)**.** *The period d_i of state $i \in \Xi$ is*

$$d_i = \gcd\{n \leq 1; p_{ii}(n) > 0\} \tag{5.3}$$

with $d_i = \infty$ if there is no $n > 0$ with $p_{ii}(n) > 0$. A state i is called aperiodic if $d_i = 1$.

In words, if a state i always returns back to the same state i in a multiple of k steps, the state has a period of k. This implies that two communicating states have the same period. A chain is said to be *aperiodic* if all states in Ξ are aperiodic. In order to prove aperiodicity we need to show that for any state $\boldsymbol{x}$ the greatest common divisor of 2 different path lengths of $\boldsymbol{x} \to \boldsymbol{x}$ is 1. The easiest way to achieve this is to design a kernel with $P(\boldsymbol{x}_t = \boldsymbol{x}_{t-1} | \boldsymbol{x}_{t-1}) > 0$.

For an ergodic chain, i.e., an irreducible, positive recurrent, and aperiodic chain, the following holds:

Theorem 6 (Convergence to steady state)**.** *Let $\boldsymbol{P}$ be an ergodic transition matrix on a countable state space. For all probability distributions μ and $\tilde{\mu}$ on Ξ,*

$$\lim_{n \to \infty} \frac{1}{2} |\mu^T \boldsymbol{P}^n - \tilde{\mu} \boldsymbol{P}^n| = 0 \tag{5.4}$$

This result, together with theorems (4) and (5), shows that a discrete HMC under mild regularity conditions (aperiodicity and irreducibility) will eventually converge to the equilibrium distribution π, independent of the initial distribution μ. This is a sufficient conditions answering questions 1 and 2.

Ergodic theorem Finally, the following theorem answers question 3:

Theorem 7 (Ergodic Theorem)**.** *Let $\{\boldsymbol{x}_n\}$ be an irreducible positive recurrent Markov chain with the stationary distribution π, and let $f : \Xi \to \mathbb{R}$ be such that*

$$\sum_{i \in \Xi} |f(i)| \pi(i) < \infty.$$

Then for any initial distribution μ

$$\lim_{N \to \infty} \frac{1}{N} \sum_{k=1}^{N} f(\boldsymbol{x}_k) = \sum_{i \in \Xi} f(i) \pi(i) \tag{5.5}$$

almost surely.

This theorem allows us to compute empirical averages over the output of

a stochastic process since they converge to probabilistic averages.

For a HMC in a finite state space, irreducibility and aperiodicity guarantee convergence to a unique equilibrium distribution in such a way that we can compute averages in the form of Eq. (5.5). In the following, we ensure convergence by constraining the chain these two conditions.

5.1.2 The Metropolis-Hastings algorithm

Following our plan of constructing a Markov chain with one and only one equilibrium distribution that is equal to the target distribution, the next step is to find the correct transition matrix $\boldsymbol{P}$. In this subsection we assume $\boldsymbol{P}$ fulfills the regularity conditions of being positive recurrent, aperiodic, and irreducible. Finding $\boldsymbol{P}$ becomes easier when constraining it to be a reversible transition matrix and profiting from implications of the reversible property:

Definition 12 (Detailed balance)**.** *A MC is said to be reversible with respect to any initial distribution π if for any i and j*

$$\pi_i \cdot p_{ij} = \pi_j \cdot p_{ji}. \tag{5.6}$$

A chain satisfies *detailed balance* if the above equation holds.

We marginalize Eq. (5.6) over j and get

$$\sum_j \pi_j \cdot p_{ji} = \sum_j \pi_i \cdot p_{ij} = \pi_i \sum_j p_{ij} = \pi_i,$$

which exactly matches Def. 7. Hence,

Theorem 8. *Let $\boldsymbol{P}$ be a transition matrix on the countable state space Ξ, and let π be some probability distribution on Ξ. If for all states $i, j \in \Xi$ the detailed balance equations are satisfied, then π is a stationary distribution of $\boldsymbol{P}$.*

We hence search for $\boldsymbol{P}$ such that Eq. 5.6 holds. The following intuitive argument is adapted from the paper of Chib and Greenberg (1995). As-

sume that we have a transition probability matrix $\boldsymbol{Q}$ that does not fulfill the detailed balance condition, for example

$$\pi_i \cdot q_{ij} > \pi_j \cdot q_{ji}.$$

In this situation moving from state i to state j happens too often. In order to correct for this imbalance we introduce a probability distribution with transition probabilities $\alpha_{kl} \in [0, 1]$ as follows:

$$\pi_i \cdot q_{ij} \cdot \alpha_{ij} \overset{!}{=} \pi_j \cdot q_{ji} \cdot \alpha_{ji}.$$

Since moving from j to i happens too rarely, we set α_{ji} to its maximum value $\alpha_{ji} = 1$ and solve for α_{ij}:

$$\alpha_{ij} = \frac{\pi_j}{\pi_i} \cdot \frac{q_{ji}}{q_{ij}} \tag{5.7}$$

α is called the *Metropolis-Hastings ratio*. From state i we then move to state j only with probability α_{ij} in order to compensate for the imbalance. This yields the desired $\boldsymbol{P}$. We can simulate this chain with an acceptance-rejection scheme as described in algorithm 6.

Algorithm 6 Metropolis-Hastings

1: **repeat**
2: set $i := x_t$ and draw a state j from q_{i_t}
3: **if** $u \sim \mathcal{U}(0, 1) \leq \alpha_{ij} = \min\left(\frac{\pi_j \cdot q_{ji}}{\pi_i \cdot q_{ij}}, 1\right)$ **then**
4: $x_{t+1} \leftarrow j$
5: **else**
6: $x_{t+1} \leftarrow i$
7: **until** convergence

This is the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970). The algorithm is useful and successful because of mainly two properties: First, the proposal distribution q can be freely chosen. We design it to be irreducible and aperiodic. Second, from the MH ratio (line 3) we see that we only need to be able to query an unnormalized version of $\pi(\cdot)$, as the decision to move only depends on a ratio.

The first term on the right-hand side of Eq. (5.7) is called the *posterior ratio*, the second term the *forward-backward ratio* (FBR). We further call the *acceptance rate* (AR) the fraction of accepted moves among all trials. Roughly, acceptable ARs are in the range between 10 and 50 percent. At equilibrium, a chain with an AR above 50% usually propose samples with posterior values similar to each other. According to line 3 such samples are very often accepted. Consequently, the chain is not selective and the samples are biased by the proposal distribution unless the chain is ran for long-enough time. In order to improve one usually alters q such as to propose $\boldsymbol{x}'$ with larger distance to $\boldsymbol{x}$ in parameter space. We say that the *step size* is increased. Note that before the chain reaches equilibrium, the AR may be larger than 50%. Similarly, if the step size is too large, in a reasonably large parameter space, we need to be lucky to find a proposal with comparable or better posterior. In this case the chain sticks around $\boldsymbol{x}$, and this sample is overrepresented. It is hence important to properly tune the step size of a MH algorithm.

5.2 Related work

MCMC methods have extensively been used in image processing since the seminal paper of Geman and Geman (1984). They established an analogy between statistical physics models and images using the equivalence between Markov random field (MRF) and the Gibbs-Boltzman-distribution (Grimmet, 1973). In their work they introduced the Gibbs sampler, which can be interpreted as a variant of the MH algorithm, for MRFs leading to a highly parallel MAP algorithm for image restoration.

Tu and Zhu (2002) presented a MCMC method for models introduced in the seminal paper of Zhu and Yuille (1996). They defined a state space allowing jump-diffusion dynamics for the chain. Diffusion dynamics was used to sample region deformations and competition. Regions were explicitly represented. This enabled the use of jump dynamics for merging and splitting. Jump dynamics was also used to jump between different image models for uniform, textured, cluttered, and shaded regions. Such models focus on natural scene images.

Fan et al. (2007) designed a proposal for shape perturbation on implicitly defined regions using level sets. The level-set perturbation was applied in the vicinity of a point sampled on the contour. The method therefore needed to switch back and forth between implicit and explicit representations, which generates a heavy computational overhead. Their proposal is biased toward smooth shapes, as it incorporates a regularizing flow. This rendered proposal asymmetric and calculating the backward proposal density became non-trivial. The authors used locally linear approximations of the shape to compute this value. Topological changes were not explicitly treated in the paper. An interesting aspect discussed by Fan et al. (2007) is *conditional simulation*. In conditional simulation a certain part of the state space is fixed. This can be very useful for semi-automated segmentation tasks, where experts label parts of image.

Chen and Radke (2009) demonstrated the superiority of sampling approaches over optimization by characterizing multi-model distributions and visualizing the corresponding segmentations. Their approach sampled shapes from a distribution over the space of signed-distance functions (SDF). Shape deformation was achieved by deforming the contour around a *foot point* by pushing it in the direction of the normal. The presented deformation scheme was able to maintain the SDF property. Nevertheless, the proposal scheme did not allow simple computation of the FBR. In fact, the backward proposal could not be calculated exactly. The approximation used depends on the curvature, and the equilibrium distribution was hence biased. The method did not allow topological changes.

Later, Chang and Fisher III (2011) drastically lowered the number of iterations needed to converge to the stationary distribution by sampling with biased proposals. They presented a proposal distribution that was biased along the gradient of the underlying energy functional. Deformation was achieved by adding Gaussians to the level-set function. This proposal was parametrized and contained three random variables of which multiple combinations might lead to the same shape deformation. Thanks to a clever proposal scheme, good approximations could be made in order to compute the FBR. Furthermore, the authors extended the method to represent and sample a multiple, but fixed numbers of regions.

Recently, the same authors presented a sampling scheme similar to a Gibbs

sampler called Gibbs-inspired Metropolis Hastings (Chang and Fisher, 2012). Level sets were used to represent and deform shapes. The level-set values were changed within a mask. Perturbation were chosen such that the MH ratio is 1, like in Gibbs sampling. Using digital topology the method allowed sampling shapes in topologically constrained spaces. In terms of computational performance the method clearly outperformed the previous methods. We hence use this latest method to compare with.

5.3 Discrete region sampling

We present a discrete multi-region method to sample the posterior $p(\Gamma|I)$. The algorithm can be seen as a sampling version of RC (see Sec. 4), which is a discrete particle-based MAP algorithm. We therefore call the present algorithm *discrete region sampling* (DRS) .

Inspired by the algorithms reviewed above, DRS supports biasing the proposal distribution and sampling in a topologically constrained space. The work by Chang and Fisher III (2011) particularly influenced the design of the present approach. A fundamental difference between the present approach and all reviewed approaches is the explicit, discrete object representation. Nevertheless, the present algorithm naturally changes the object's topology. Due to the finite state space, proving ergodicity is trivial. Also, the FBR can be computed exactly.

DRS fundamentally differs from the above algorithms in terms of the sampling strategy. Previous methods focus on increasing the step sizes while maintaining the AR. DRS efficiently performs a large number of small steps with a relatively high acceptance rate. This strategy exploits the fact that many image models are induced by energies whose differences can be efficiently computed for small shape deformations.

This section is structured as follows: We first introduce the state space and the discrete move set. We then discuss the FBR computation for different step sizes. In subsection 5.3.3.2 we show how to bias the proposal distribution toward smooth shapes and how to sample topologically constrained shapes with only minor changes to the algorithm. In subsection 5.3.4 we present a summary of DRS. Before concluding the section

with benchmarks and applications, we show that the present Markov chain fulfills the regularity conditions discussed in the previous section.

5.3.1 The move set and the state space

Objects are represented using a label image L, as in chapter 4. In a discrete multi-region object representation, Γ is restricted to closed objects and it can be bijectivly mapped onto a corresponding label image L. In order to perturb Γ, we perturb L. We do so by either adding or removing discrete points to a region. Or both. In order to calculate the FBR, we need to be able to compute the probability of selecting a certain discrete point. The proposal therefore is a discrete probability distribution. In order to keep track of the discrete candidate points, we mark them with a particle, similar to the particles introduced in chapter 4.

5.3.1.1 Particles

Definition 13. *A particle A is a triplet (x_A, l'_A, w_A) with position $x_A \in \Omega$, candidate label $l'_A \in [0; M]$, and weight $w_A \in \mathbb{R}$.*

Applying a particle causes the label image to inherit the particle's candidate label at position x_A. We define the operator $\circ : (\Xi, \mathcal{P}) \to \Xi$ to indicate a transition from one state to another using particle A, i.e., $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t \circ A$.

We distinguish between two types of particles, *regular* and *floating* particles.

Regular particles The particles mark the positions where L can be perturbed to an intermediate label image L'. Therefore, we need to place them such that L' is a comparably good candidate in terms of the posterior probability. A large perturbation activity close to the contour of an object is likely to fulfill this criterion. We therefore introduce *regular particles*. Their locations are bound to the contour. More formally:

Definition 14 (Regular particle set)**.** *A regular particle A is a particle*

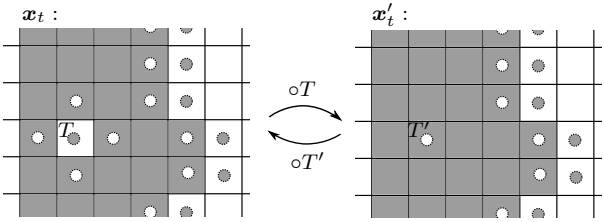


Figure 5.1: Regular particles in a topological trap. The gray shaded region is a foreground region, the background region is white. Circles indicate particles. See main text for details.

such that $L(x_A) \neq l'$ *and* $|N_n^1(x_A, X_{l'})| > 0$ *where* n *denotes the connectivity of region* l'*. The set* $\mathcal{P}$ *contains every possible regular particle exactly once.*

Consequently, $\mathcal{P}$ is entirely defined by L.

Floating particles In order to ensure reversibility, a *reverse particle* A' is needed in $\boldsymbol{x}_{t+1}$ that, when applied, leads back to state $\boldsymbol{x}_t$, i.e., $A' = (x_A, L_{\boldsymbol{x}_t}(x_A), w_{A'})$. When considering regular particles only, we allow situations as illustrated in Fig. 5.1. All particles in $\boldsymbol{x}_t$ correspond to Def. 14. The reverse particle of particle T' at $\boldsymbol{x}'_t$, however, does not. We hence either reject the move because the FBR is 0, or we augment the particle set. We avoid this trap by augmenting the particle set with unbounded *floating particles*. We denote the set of floating particles $\mathcal{P}_f$. The discrete proposal distribution is therefore based on particles in $\mathcal{P} \cup \mathcal{P}_f$.

A transition from a regular to a floating particle occurs when a connected component collapses w.r.t. the region connectivity. At the same time, a hole is closed. Similarly, a transition from a floating particle to a regular particle occurs when a new connected component is created. At the same time, a hole may be created.

5.3.1.2 The state space

In order to allow topology-changing proposals, we introduced floating particles. Unlike regular particles, the label image L does not naturally encode these particles. We therefore augment the set $\mathcal{L}$ by the set of all possible floating particle sets $\{\mathcal{P}_f\}$ with at least one particle belonging to each region (the number of regions is fixed to M). We define hence the state space Ξ as

$$\Xi = \mathcal{L} \times \{\mathcal{P}_f\}. \tag{5.8}$$

For M topologically unconstrained objects, Ξ is of size

$$|\Xi| = M^{|\Omega|} \cdot (|\Omega| - 1)^{M-1}.$$

This state space is huge[1]. However, for all posteriors of the Bayesian image models discussed in Sec. 3, $L(x)$ depends only on a few other pixels in the vicinity of x. Most entries in the state vector are hence uncorrelated.

5.3.1.3 Moving in the state space

The Markov chain moves from state $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$ by either applying particles (geometric moves) or by changing the particle set $\mathcal{P}_f$ (non-geometric moves). We call the latter *off-boundary sampling*.

Geometric moves The particles operate on L and hence on Γ, i.e., they constitute the geometric part of the move set. A particle is selected according to its weight, which represents an unnormalized discrete probability. For single-particle moves

$$q_A = \frac{w_A}{\sum_{B \in \mathcal{P}} w_B} \tag{5.9}$$

[1] Assume a 2-region problem in three dimensions with a reasonable image size, e.g. $|\Omega| = 512 \times 512 \times 16$ pixel. In this example, the state space size is $|\Xi| \approx 10^{1262618}$.For comparison, the Eddington number, the number of protons in the observable universe, is $1.57 \cdot 10^{80}$.

Algorithm 7 Apply a particle

1: **if** $A \in \mathcal{P}_f$ and $L(x_A) = l'_A$ **then** reject.
2: $L(x_A) = l'_A$
3: Remove A
4: **if** A' fulfills Def. 14 **then**
5: insert A' in $\mathcal{P}$.
6: **else**
7: **if** $A' \in \mathcal{P}_f$ **then** insert A' in $\mathcal{P}_f$ **else** reject.
8: Ensure $\mathcal{P}$ satisfies Def. 14 $\forall B : x_B \in N_M(x_A)$.
9: Filter $\mathcal{P}$ to satisfy topological constraints.
10: Update particle weights $\forall B : x_B \in N_M(x_A)$ (depending on q)

is the proposal probability of the move proposed by particle A. The proposal and FBR computation for multi-particle moves is discussed in the next section.

We now detail the procedure of applying a particle. Algorithm 7 shows the individual steps. If A is drawn from $\mathcal{P}_f$ we first check if it is applicable (line 1). A particle is not applicable if the label image at that location is equal to the A's candidate label. In this case the chain performs a self-transition and we therefore reject the move. In all other cases we update the label image (line 2). We remove A from either $\mathcal{P}$ or $\mathcal{P}_f$ in line 3. The reverse particle is then inserted in the appropriate container (lines 4–7). If that location in $\mathcal{P}_f$ is already occupied we reject the move immediately. In lines 8 and 10 we update the regular particles in the neighborhood. The weight update is only required when biased proposals are used. While scanning the neighborhood for updating $\mathcal{P}$, all particles not satisfying any given topological constraints are removed from $\mathcal{P}$. This ensures that $\mathcal{P}$ only contains particles that, when applied, lead to topologically valid shapes.

It is possible that a regular and a floating particle with identical candidate labels share the same location. Also, while waiting to be selected by the proposal process, the pixel of a floating particle F may change its label. When F is selected in such a situation, the chain performs a self-transition, i.e., $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$. This has the same effect as rejecting the move.

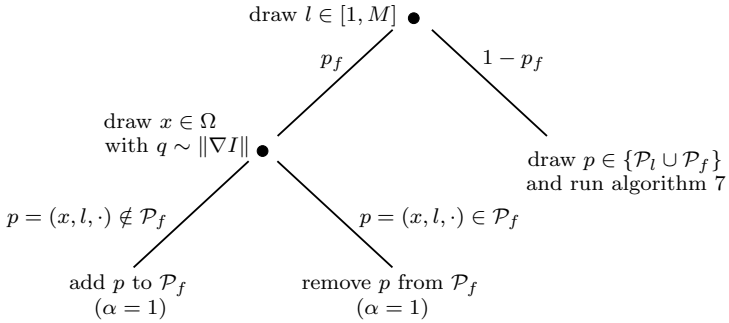


Figure 5.2: Decision tree of the Markov kernel.

Off-boundary sampling The notion of floating particles provides an opportunity to extend the proposal q almost arbitrarily. We may hence place unbounded particles at interesting locations far from the contour. Interesting locations have large image gradients $\|\nabla I\|$. We call this option *off-boundary sampling*. For the sake of reversibility, off-boundary particles can also be removed.

In order to ensure correct FBR calculation, off-boundary sampling only introduces moves that are not possible in any other way. Namely, we insert or remove a particle from $\mathcal{P}_f$. This cannot be achieved by regular particles without also changing the label image, which is part of the state space. Figure 5.2 illustrates how off-boundary sampling is integrated. With probability q_f we operate on the set $\mathcal{P}_f$. Off-boundary sampling can be disabled by setting $q_f = 0$. At any value of $q_f \in [0, 1[$ the equilibrium distribution remains the same.

The posterior remains unaffected by off-boundary moves. Consequently, the MH ratio is equal to the FBR. The equilibrium distribution for floating particle locations is hence uniform. On average, the probability for a floating particle to exist is 0.5 at each location. Since we bias the proposal to insert and remove floating particles more often at positions where $\|\nabla I\|$ is large, convergence may be slow. Off-boundary sampling is, however, useful during the burn-in phase in order to explore interesting parts of the state space. After burn-in phase, we gradually reduce q_f to 0. Since the Markov chain converges to the same equilibrium for any fixed q_f, this annealing is optional. The calculation of the FBR is discussed in the next

section.

5.3.2 FBR computation

5.3.2.1 FBR computation for multi-particle moves

When operating on flat π or sampling with too small step size, the MH algorithm performs a random walk following the proposal distribution. Consequently, the samples drawn do not represent the target distribution, but rather they reflect the proposal. More representative samples can be drawn by increasing the step size (Liu et al., 2000).

A simple way to increase the step size is to draw (and apply) multiple particles per move. For simplicity, we sample *with replacement*. The order in which particles are drawn is irrelevant. We therefore sample from a discrete multinomial distribution. We are in a degenerate case since we have as many different categories (also called classes) as particles. Assume the chain is in state $\boldsymbol{x}_t$. The quantity $q(\boldsymbol{x}'|\boldsymbol{x}_t)$ is the probability of drawing the particles leading to state $\boldsymbol{x}'$. Consider k' trials with replacement. The experiment consists of k different outcomes ($k \leq k'$). The integer n_{p_i} counts how many times category i (or the particle p_i) has been sampled. Then the multinomial proposal pmf reads

$$q(\boldsymbol{x}'|\boldsymbol{x}_t) = q(p_1, \ldots, p_k|\boldsymbol{x}_t) = \frac{k'!}{n_{p_1}! \cdots n_{p_k}!} \cdot q_1^{n_{p_1}} \cdots q_k^{n_{p_k}}. \tag{5.10}$$

For this proposal pmf, the FBR computation simplifies to

$$\frac{q(\boldsymbol{x}'|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}')} = \prod_i \frac{q(A_i')}{q(A_i)}, \tag{5.11}$$

where A_i' is the reverse particle of A_i. This FBR can be computed efficiently. The move set is designed such that the reverse particle always exists.

Note that the step size is automatically reduced if the same particle is

selected twice. Sampling with replacement avoids situations where not enough particles are available. This is especially useful when using any kind of stratified sampling (see Sec. 5.3.3.1). Nevertheless, a sampling strategy without replacement could be chosen as well.

Alternative step-size increase strategies Constraining the topology while increasing the step size with the above strategy leads to an intractable FBR. In order to build a discrete proposal we would have to enumerate all topologically valid particle groups of size smaller or equal to k. In order to reduce the acceptance rate while ensuring topological constraints using digital topology, the following approaches can be considered:

- Liu et al. (2000) introduced the multiple-try Metropolis-Hastings algorithm. The algorithm produces a reversible chain with smaller AR at the cost of evaluating multiple candidates per iteration.
- A more efficient possibility is to assign multiple pixels to one particle[2]. Although this leads to a resolution loss of the final probability density map, qualitatively the result will be as useful as having the full resolution. This is because there is no significant "change" of probability within the particle's scope. If there was a significant change, the MH ratio would be selective without step-size increase.

5.3.2.2 FBR computation for two conditional particles (conditional proposals)

For certain image models, such as the deconvolving model (see Sec. 3.1.2), the posterior probability changes drastically when growing or shrinking a region. A move with a smaller energy difference is to add and remove two neighboring discrete point at the same time. Figure 5.3 compares a single-particle-move energy difference with the energy difference for the

[2]Sometimes such a collection of pixels is called a *super-pixel*. Any space-filling pattern homotopic to a rectilinear grid could be used to define super-pixels. We presume a rectilinear grid in order to be compatible with digital topology.

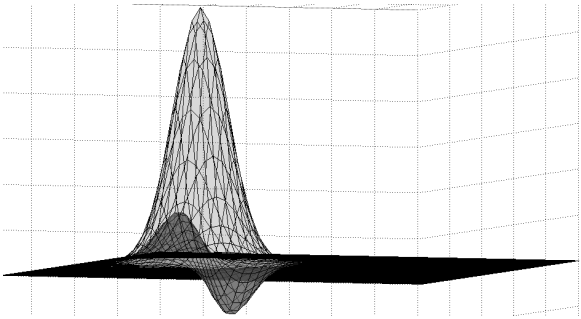


Figure 5.3: The volume below the light gray surface corresponds to the change in energy when adding a discrete point using the energy function $\mathcal{E}^{dec}$ with a Gaussian PSF. The volume below the dark curve shows the change in energy when locally deforming the contour by moving a discrete point.

two-particle proposal when using the deconvolving model from Sec. 3.1.2.

In order to include local area-preserving moves in the proposal we apply two particles A and B in one move. We first sample a particle A. The particle location x_A then serves as foot point, similar to the work of Chen and Radke (2009). We then create a new discrete distribution from which we sample the partner particle B. The distribution contains all particles in the set $Q_A = \{\{\mathcal{P}_{\boldsymbol{x}\circ A} | x \in N_n^+(x_A)\} \cup A\}$. The neighborhood set $N_n^+(x_A)$ contains x_A. In order to still allow growing an shrinking regions, we also include particle A in Q_A. If B is the same particle as A, we perform a one-particle move.

On the one hand, drawing the candidates A and B is straightforward. On the other hand, computing the FBR for a combined move involves 8 probability mass functions:

$$\frac{q(\boldsymbol{x}|\boldsymbol{x}')}{q(\boldsymbol{x}'|\boldsymbol{x})} = \frac{q(A'|\boldsymbol{x}') \cdot q(B'|\boldsymbol{x}' \circ A') + q(B'|\boldsymbol{x}') \cdot q(A'|\boldsymbol{x}' \circ B')}{q(A|\boldsymbol{x}) \cdot q(B|\boldsymbol{x} \circ A) + q(B|\boldsymbol{x}) \cdot q(A|\boldsymbol{x} \circ B)}. \tag{5.12}$$

A' and B' are the reverse particles of A and B, respectively. The probability $q(A)$ follows from Eq. (5.15) in state $\boldsymbol{x}_t$. For the other proposal pmf values we follow the scheme illustrated in Fig. 5.4: We first need to find B. Starting at $\boldsymbol{x}_t$ we apply A in order to find out what particles are contained in $N_n^+(x_A)$ after having applied A. We sample B from the set Q_A. Knowing B we can calculate the conditional probability $q(B|\boldsymbol{x}_t \circ A)$. This intermediate state is equivalent to $\boldsymbol{x}' \circ B'$ (see Fig. 5.4). We therefore also compute $q(A'|\boldsymbol{x}' \circ B')$. In order to calculate $q(B)$ we need to go back to the original state by applying A'. Back in state $\boldsymbol{x}_t$ we calculate $q(B|\boldsymbol{x}_t)$. Then, we apply B. In state $\boldsymbol{x}_t \circ B$ we calculate $q(A|\boldsymbol{x} \circ B)$ and $q(B'|\boldsymbol{x}' \circ A')$. Eventually, we move to state $\boldsymbol{x}' = \boldsymbol{x}_t \circ B \circ A$ in order to calculate $q(A'|\boldsymbol{x}')$ and $q(B'|\boldsymbol{x}')$. These are the two last ingredients for the FBR in Eq. (5.12).

The additional data structure manipulations for this two-particle proposal scheme generate computational overhead. For a deconvolving energy, however, conditional sampling amortizes by increasing the acceptance rate. Experiments are presented in Sec. 5.4.3.

5.3.2.3 FBR computation for floating particles

In Sec. 5.3.1.1 we introduced off-boundary sampling using floating particles. Recall from Fig. 5.2 that we first decide (with probability $1 - q_f$) for a move to deform the contour or for a move to alter $\mathcal{P}_f$. Then, we decide between inserting (with probability q_I) and deleting a floating particle. The proposal probability of inserting a floating particle at posit on x is

$$q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = q_f \cdot q_I \cdot \frac{1}{Z_e}\|\nabla I(x)\|, \; Z_e = \sum_{x \in \Omega} \|\nabla I(x)\| \tag{5.13}$$

If we decide to delete a particle, we sample a particle uniformly. The probability for deleting of a floating particle hence is

$$q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = q_f \cdot (1 - q_I) \cdot \frac{1}{|\mathcal{P}_f|}. \tag{5.14}$$

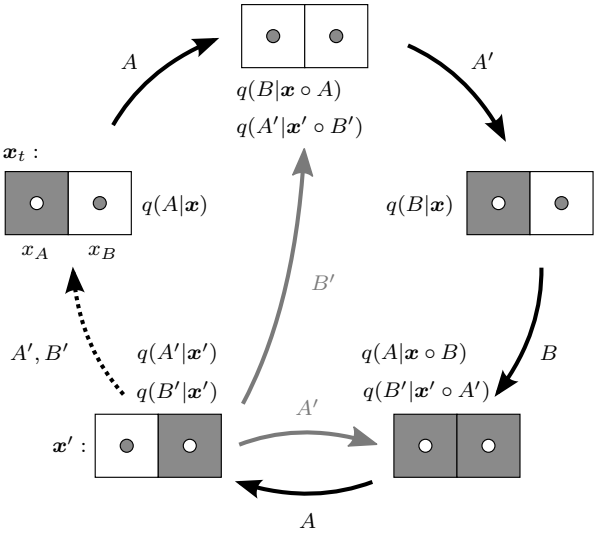


Figure 5.4: Illustration of the move order for the conditional proposal computation in the two-region case. x_A and x_B are the positions of two hypothetical particles A and B, respectively. The color in the particle (small circle) indicates the candidate label. The starting state is $\boldsymbol{x}_t$. The transition labels indicate the particle(s) that are applied in order to reach the next (intermediate) state. State $\boldsymbol{x}'$ is the newly proposed state. If the proposal is accepted, $\boldsymbol{x}_{t+1}$ is set to $\boldsymbol{x}'$. The dashed transition is only used when rejecting $\boldsymbol{x}'$. The transitions in gray illustrate that $\boldsymbol{x} \circ A = \boldsymbol{x}' \circ B'$ and $\boldsymbol{x} \circ B = \boldsymbol{x}' \circ A'$. The conditional probabilities $q(A'|\boldsymbol{x}' \circ B')$ and $q(B'|\boldsymbol{x}' \circ A')$ can therefore be calculated in the same intermediate state as $q(B|\boldsymbol{x} \circ A)$ and $q(A|\boldsymbol{x} \circ B)$, respectively. We denote by $q(\cdot)$ the different proposals to be calculated in the corresponding states.

This probability is undefined if there is no floating particle. In this case we reject the move since the FBR is zero.

From Eqs. (5.13) and (5.14) we can calculate the FBR regarding the floating particles.

5.3.3 Biasing the proposal distribution

The proposal distribution is a crucial part for any MH algorithm. It has to be designed such that the chain is ergodic (see 5.1.1). At the same time it is responsible for the algorithm's performance. In theory, no matter what proposal distribution is used, when running a correct MH algorithm long enough, the Markov chain will eventually converge to its equilibrium distribution π. In practice, the proposal distribution determines how long it takes from the equilibrium distribution and how long one has to sample to obtain a representative sample from π. For good performance we need to focus on the interesting parts of the state space. It is about deciding *"where to look first?"*. If prior knowledge is available about what relevant states look like, one should use this knowledge. It can be shown that the proposal distribution should be close to the target distribution in order to achieve fast convergence (Mengersen and Tweedie, 1996).

The same can be formulated more technically: For the Metropolis-Hastings algorithm, all samples with a MH ratio larger than 1 are accepted. There is no difference between much-better samples and slightly-better samples. A good proposal allows taking larger steps while maintaining acceptable acceptance ratios. With fixed acceptance ratio, a larger step size therefore allows faster exploration of the space. The chain is said to be well *mixing*.

The present method focuses on taking many small steps around interesting locations, namely around the contour. We take advantage of both efficient proposal computation and local target-density ratio computation for small contour perturbations. Instead of taking one big but computationally expensive step, we perform many small computationally cheap steps. While a big step might be rejected, the small-step strategy has a high probability of moving forward in the same computational time. Additionally, floating particles enhance the local perturbation sampling. They explore the state

space at interesting locations. Nevertheless, the particle set $\mathcal{P}$ is usually very large and, for reasons discussed above, it may be beneficial to bias particle selection, i.e., bias the proposal distribution. In the following we discuss how we bias the proposal. Using stratified sampling we focus on interesting objects or locations. Using smooth proposals we favor smooth contours.

5.3.3.1 Stratified sampling

The particle set $\mathcal{P}$ can be split into subsets amounting to discrete *stratified sampling*. For example, we might partition particles according to the regions they belong to. Reconsider the Markov kernel decision tree in Fig. 5.2. We first choose a region l from a uniform distribution with $q_l(l) = \frac{1}{M-1}$. Then, we sample the particles from the subset $\mathcal{P}_l$ that contains only particles belonging to region l. This ensures that small and large regions receive the same amount of processing time. Therefore, the final posterior representation is of better quality for small regions. This stratified sampling approach leads to relative quantification errors that may be useful for certain applications, for example in protein quantification (see Sec. 5.4.4). Note that any other partitioning of $\mathcal{P}$ could be used as well.

5.3.3.2 Smooth proposals

Assuming that the target distribution favors smooth contours, it is favorable to propose smooth samples. In image segmentation, this assumption is reasonable since the prior probability often contains a length penalty causing the contour to be smooth. This has already been exploited by Fan et al. (2007) and Chen and Radke (2009). Chang and Fisher III (2011) biased the proposal with the full energy gradient. Here we show how to incorporate a smooth contour bias into a discrete proposal using particle weights.

For smooth boundary proposals, w_i is related to the contour length caused by the pixel at the particle's position. As weights we use the length approximations for contours and surfaces introduced by Boykov and Kolmogorov

(2003). These approximations are discussed in Sec. 3.2.2.1. They are based on the Cauchy-Crofton formula which relates the Euclidean length of a curve to the number of grid lines intersecting it. Although there exist grids leading to more accurate length estimation, we use a rectilinear grid with an 8-neighborhood in 2D and a 26-neighborhood in 3D. The length approximation need not to be very accurate here since it is only used to bias the proposal. Efficient computation is more important. Let $l = L(x_A)$, then

$$q(A|\boldsymbol{x}) = \frac{1}{Z} \sum_{y \in N_m^1(x_A)} w(|y - x|) \cdot \delta_L \left(L(y) \right), \; Z = \sum_{A \in \mathcal{P}} w_A, \tag{5.15}$$

where δ_L is the Kronecker delta. Recall that $N_M^1(\cdot)$ is the full 3^d neighborhood set. Z is the partition function. For efficient evaluation, the weight function $w(\cdot)$ can be pre-computed for all input values. Whenever a particle is applied, all particles in the fully connected neighborhood need their weights updated.

Note that through particle weights we could incorporate any type of bias into q without altering the algorithm. In Sec. 5.4.3 we show that smooth proposals indeed increase the acceptance rate.

5.3.3.3 Topologically constrained proposals

Recently, Chang and Fisher (2012) proposed sampling in topologically constraint spaces. Digital topology (see Sec. 2.2) is used to detect topological events and prevent them if necessary.

Topologically constrained sampling reduces the state space Ξ to a subspace. Although this reduces the space size, it does not necessarily simplify the sampling task. Many paths towards a good solution may temporarily leave the valid space. Moreover, we are interested in Markov chain summary statistics. Gaps between object boundaries ensure a particular genus, however, eventually vanish in the summery statistics (unless they are favored by the target distribution).

We argue that topologically constrained sampling is useful only if seed

points of connected objects are known. Then, the sampler can delineate the object boundaries starting from the known seed points.

In order to fulfill topological constraints in the present framework, we set $q_f = 0$ and filter both particle sets $\mathcal{P}$ and $\mathcal{P}_f$ such that the topological numbers (see Sec. 2.2.2) at all particle positions fulfill the constraints. The sampler will therefore only consider topologically valid shape perturbations.

Whenever a particle A is applied, particles in the fully-connected neighborhood $y \in N_M(x_A)$ are invalidated or validated. Therefore, $\mathcal{P}$ needs to be updated locally. We do so by collecting particles for all y and computing their topological numbers. We then ensure that all particles with a valid topological number pair are contained in $\mathcal{P}$, and none of the others (line 9 in algorithm 7).

5.3.4 Overall algorithm

We summarize DRS in algorithm 8. Lines starting with an asterisk are optional. Instructions starting with two asterisks are executed only when using conditional proposals. The algorithm is started with an initial label image L_0 which implicitly determines the number of regions. L_0, together with an empty set of floating particles, defines the initial state $\boldsymbol{x}_0$. We initialize the regular particle set $\mathcal{P}$ (line 1) such as to respect topological constraints. If biased proposals are used, we compute the particle weights (line 2), else all weights are set to 1. In the main loop we first sample a region r (line 4). When using stratified sampling (sectoin 5.3.3.1) we only consider the subset of particles belonging to this region. With probability q_f (lines 5 and 6) we then perform off-boundary sampling (lines 7–15) as described in Sec. 5.3.1.3. We linearly decrease q_f during the burn-in phase. We then draw at least one particle A according to the proposal $q(\cdot)$ (line 18). Recall that $\mathcal{P}$ is maintained (by algorithm 7) such as to only contain particles leading to a topologically valid shape. For large step sizes multiple particles are drawn by repeating lines 17–19. The step size must be set to 1 if topological constraints or conditional proposals are used. When using conditional proposals we also need to find the partner particle B (line 21), for which we first simulate A (line 18). We then draw B

Algorithm 8 DRS - Discrete Region Competition Sampling

Initialization: Set up $\boldsymbol{x}_0$, i.e., $\mathcal{P}$ from L_0, $\mathcal{P}_f = \emptyset$
* For all particles, calculate w according to eq. (5.15)
repeat
  Sample a region r [*and filter $\mathcal{P} \cup \mathcal{P}$ for region r]
  Sample $u_1 \sim \mathcal{U}[0;1]$ [* adapt q_f]
  if $q_f < u_1$ **then**
    if Coin toss is head **then**
      Sample $x \sim \|\nabla I\|$
      Compute FBR using Eq. (5.13) and (5.14).
      Insert $F = (x, r, w_p)$ if $u_2 \sim \mathcal{U}[0;1] < \max(\text{FBR}, 1)$.
    else
      Sample a F from $\mathcal{P}_f$, **if** $\mathcal{P}_f \neq \emptyset$, **else next iteration**
      Compute FBR using Eq. (5.13) and (5.14).
      Delete F if $u_3 \sim \mathcal{U}[0;1] < \max(\text{FBR}, 1)$.
    next iteration
  $\Delta\mathcal{E} = 0$
  for i=1...step size **do**
    Sample particle A_i from $\{\mathcal{P} \cup \mathcal{P}_f\}$ according to q_i
    $\Delta\mathcal{E} = \Delta\mathcal{E} + \Delta\mathcal{E}(x_A, L(x_A) \rightarrow l'_A)$ [** $+\Delta\mathcal{E}(x_B, L(x_B) \rightarrow l'_B)$]
  Compute $q(\boldsymbol{x}'|\boldsymbol{x})$ using Eq. (5.10)
  ** Apply A. Sample B from $\mathcal{P}|x_B \in N_n^+(x_A)$. Apply A'. .
  Move to $\boldsymbol{x}'$ by applying A_i [**and B]
  ** Compute conditional proposals in eq. (5.12) using eq. (5.15)
  Calculate $q(\boldsymbol{x}|\boldsymbol{x}')$ using (5.15).
  if $u_4 \sim \mathcal{U}[0;1] < \max\left(\exp(-\Delta\mathcal{E})\frac{q(\boldsymbol{x}|\boldsymbol{x}')}{q(\boldsymbol{x}'|\boldsymbol{x})}, 1\right)$ **then**
    $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}'$ **else** apply A'_i [**and B']; $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$
until max. number of iterations reached

from particles located in the neighborhood $N_n^+(x_A)$. During this simulated move, conditional proposal probabilities can be calculated as illustrated in Fig. 5.4. We finally move to the proposed state $\boldsymbol{x}'$ by applying particle A (and B when using conditional proposals) using algorithm 7. In this new state we compute the backward proposals (line 24). We then compute $\Delta\mathcal{E}$ according to an energy described in chapter 3 (line 19). For multi-particle moves we add all particle contributions to $\Delta\mathcal{E}$. Finally, we apply the MH algorithm in lines 25 and 26.

5.3.5 Implementation

We implemented DRS as image-processing filter in the ITK image-processing software library (Ibanez et al., 2005). All linear combinations of energies introduced in chapter 3 can be sampled with this implementation. DRS is implemented for 2D and 3D images.

The present algorithm uses large iteration numbers. An efficient implementation is hence crucial. In this subsection we discuss implementation details on critical routines and data structures. We start with the implementation for particle containers. We then discuss efficient approximate sampling using particle weights.

5.3.5.1 Data structure for efficient particle sampling

In line 18 of algorithm 8 we sample from $\mathcal{P}$ and $\mathcal{P}_f$, which are both set data structures. We need to be able to quickly insert and erase particles from containers. At the same time, we need to efficiently access particles in random-access fashion as generated by the random number generator. Implementations of sets (hashed or not) are usually unordered. We therefore build enumerated set using one vector and one hash map as follows: Particles are stored as keys in the hash map. The corresponding values are equal to the indices in the vector. Vector element i a pointer to particle i.

Look-up, random access, insertion, and deletion then all have a complexity of $O(1)$:

- We find a particle looking up the particle in the hash-map.

- In order to access the particle at position i (i has been generated by the number generator), we dereference the i-th vector entry.

- For inserting a particle A, we first look up the corresponding entry in the hash map. If it already exists, we are done. Else we insert a key/value pair into the map with A being the key and the value being the vector's length. We then append a pointer to A to the vector.

- For deleting, we move the last element in the vector to the position of the deleted particle and update the hash map.

S a data structure is sometimes referred to as a *constant indexed (hash) set*.

5.3.5.2 Discrete distribution sampling

In Sec. 5.3.3.2 we weighted particles in order to bias the proposal toward smooth contours. Without biasing, i.e., $\forall A \in \mathcal{P} : q_A = \frac{1}{|\mathcal{P}|}$, we sample a uniform number in $[0, |\mathcal{P}|[$ and access the particle at this position. For unequal weights, the optimal algorithm has a computational complexity of $O(|\mathcal{P}|)$ (Bringmann and Panagiotou, 2012). For large numbers of particles this renders the algorithm with smooth proposals intractable. A good approximation can, however, be obtained when assuming that particles are unordered and that their weights are moderately equilibrated within the container. We consider N consecutive elements (with periodic boundary conditions) in the indexed set $\mathcal{P}$ as a representative sample. Here, this is a valid assumption since all particles have one out of three different weights. We set $N = 30$.

We generate all random numbers with a Mersenne Twister random number generator (Matsumoto and Nishimura, 1998).

5.3.6 Irreducibility and aperiodicity

We argue that our move set described above induces an irreducible and aperiodic transition graph in Ξ, provided each particle is assigned a probability larger than 0. For the sake of completeness, these two properties are proven here.

Irreducibility Recall that for a finite-state HMC to be irreducible, the probability of accessing every state i from every other state j needs to be non-zero (see Sec. 5.1.1). We assume the move set with $q_f = 0$, which is a subset of the move set when $q_f > 0$. The following proof shows irreducibility for our move set when Ω is larger than $2 \times 2(\times 1)$.

By definition, any state j has at least $c > 0$ connected component(s) (including isolated floating particles) of any region l. Assume the connected component's centers of mass falling on pixel at positions $x_i, i = 1 \dots c$.

For any region there is always a regular particle with candidate label l until the region is equal to the image domain Ω. This follows from the definition of regular particles. Similarly, for every region of more than c pixel in size there are always at least $c + 1$ regular particles with candidate label 0. Since there is always a particle A with $x_A \neq x, q_p > 0$ there is a non-zero probability for every region to collapse at a center-of-mass position x. This accessible state is illustrated as state (0) for the two-region case in Fig. 5.5.

We partition the image domain into $2 \times 2(\times 1)$ subdomains. There are $|\Xi| = 2^4 \cdot 2^4$ possible states for this patch. Figure 5.5 illustrates the non-zero probability path to 16 states (modulo rotation) covering all possible configurations of the label image. Note that the rotated configuration (3') is accessible from (1). Moreover, there is a non-zero probability path to create a floating particle, e.g., (1) $\rightarrow$ (2) $\rightarrow$ (3) $\rightarrow$ (4′) $\rightarrow$ (5). State (5) modulo the floating particle corresponds to state (1). Transitions through application of regular particles enable generating floating particles in any of the 4 fields in any order. Due to the independence of floating and regular particles all 2^4 label configurations can be combined with all 2^4 floating particles configurations (exept the state with blank label image and 0 floating particles). Therefore, every $j \in \Xi$ can be accessed for this

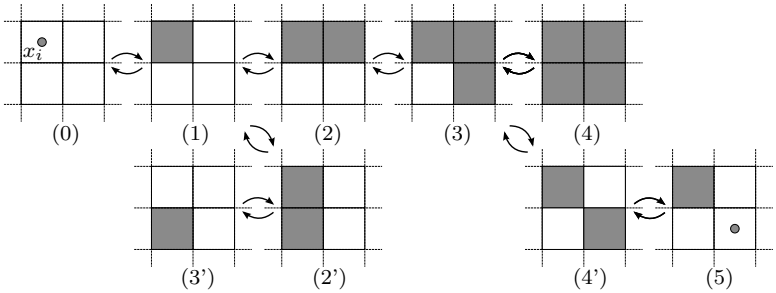


Figure 5.5: A 2×2 patch of a 2D image or 3D slice. Circles indicate floating particles with corresponding candidate label. Regular particles are not shown. For better visibility the diagram illustrates only a subset of the full transition graph. Note that state (3') corresponds to (1) rotated by 90 degrees.

patch. To explore the entire connected component (at which we collapsed to enter state (0)) we move the 2×2 sliding window by one pixel row.

When repeating the above procedure for all regions, any state can be recovered reached, the "empty" state without floating particles or with blank label image, which by definition does not belong to Ξ.

Aperiodicity The easiest way to break a period is to allow moving from state i to the same state i with non-zero probability (see chapter 5.1.1). Such self-transitions have non-zero probability when a floating particle with $L(x_A) = l'_A$ exists.

Note that when considering regular particles only, the chain has a period of 2. This is because applying a reverse particle is the same as recovering the original state, i.e., $\boldsymbol{x} = \boldsymbol{x} \circ A_i \circ A'_i$.

5.4 Applications

We first apply DRS to various artificial images. In order to validate the algorithm, we generate a ground-truth posterior. Moreover, we com-

pare DRS with Gibbs-inspired Metropolis-Hastings (GIMH) (Chang and Fisher, 2012) using two different ground-truth target probabilities. We then present a toy example to illustrate the usefulness of floating particles. In subsection 5.4.3 we use the algorithm to analyze the deconvolution problem using forward simulation in different situations. In the same subsection we discuss the effect of smooth and conditional proposals. The second set of applications considers fluorescence microscopy images. A first real-world application for protein quantification is presented in subsection 5.4.4. The example shows the multi-region sampling capability of DRS and motivates uncertainty quantification. In Sec. 5.4.5.1 we demonstrate the robustness of the present sampling approach on real-world data. We also apply DRS in 3D in Sec. 5.4.6.

5.4.1 Comparison with Gibbs-inspired MH

We compare the present algorithm with GIMH (Chang and Fisher, 2012). GIMH outperforms previous algorithms by an order of magnitude (Chang and Fisher, 2012) in terms of sampling speed. GIMH and DRS are gradient-free, allowing one to easily integrate new energies. Also, both samplers support multiple regions and allow constraining the shape topologies.

GIMH Chang and Fisher (2012) propose to use a level set ϕ for shape representation. They therefore operate on an infinite-dimensional state space. The foreground is defined as $\phi > 0$. The posterior only changes when ϕ changes its sign. In each iteration, a spherical mask m is sampled and the level set is perturbed as $\phi_{t+1} = \phi_t + f \cdot m$ with $f \in \mathbb{R}$. Then, for all f that change sign(ϕ) anywhere, the posterior is computed. Since ϕ imposes an ordering of the pixels, the number of possible posteriors for any perturbation is tractable (given m and conditional on pixels that are not in m). From this discrete conditional posterior, the algorithm samples the new state. Conceptually, this procedure is related to block-Gibbs sampling since the posterior is not conditionally sampled along one, but along many dimensions at a time (namely for all pixels within m).

GIMH also adapts its step size. Large masks m, corresponding to large steps, allow fast convergence. The problem with large step sizes, together

with the pixel ordering according to ϕ, is that pixels are coupled to neighboring pixels through the mask m. Drawing f hence depends on the mask size: Consider the mask falls into a constant image region with probabilities $p_0 = 0.6$ and $p_1 = 0.4$ to belong to the foreground and background, respectively. Assume $\phi > 0$ for half of the pixel in m. Then, the conditional posterior to draw f such that k pixels change to the foreground is $\left(\frac{p_0}{p_1}\right)^k$. Hence, the larger the mask m, the higher the probability for a large step size, i.e., a large $|f|$.

Similarly, only few pixels can already heavily influence the decision of other pixels within the mask. In the conditional posterior space, one orthogonal dimension hence dominates the conditional posterior.

This pixel dependence within the mask m biases the samples from GIMH toward the mode. This effect becomes stronger the larger the masks is. This bias toward the mode is illustrated in the following experiment.

Test cases We use a two-region model for which we know the ground-truth target probability. This probability indicates how probable it is for a region to occupy any pixel. For simplicity, we ignore any prior.

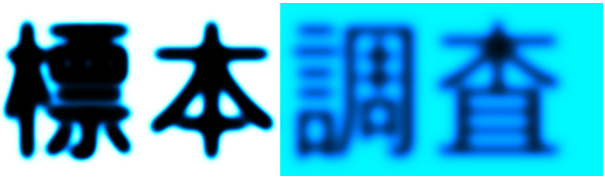
We assume the pixel-wise probabilities for the foreground to be drawn from a fixed Gaussian distribution $p_{\mathrm{FG}} \sim \mathcal{N}(\mu_{\mathrm{FG}}, \sigma^2_{\mathrm{FG}})$. The background region is drawn from $p_{\mathrm{BG}} \sim \mathcal{N}(\mu_{\mathrm{BG}}, \sigma^2_{\mathrm{BG}})$. We are given an image I shown in Fig. 5.6a. We can compute the ground-truth target distribution π. The probability of observing the foreground region at pixel x is

$$\pi(x) = \frac{p_{\mathrm{FG}}(I(x))}{p_{\mathrm{FG}}(I(x)) + p_{\mathrm{BG}}(I(x))}.$$

For both algorithms we calculate the mean shape from the samples generated by the chain[3]. We then report the pixel-wise L^1 and L^2 distance to the ground-truth target distribution.

[3]For GIMH thinning is used for technical reasons. To the best of our knowledge, this does not significantly change the results reported here.

(a)



(b)

Figure 5.6: (a) Data image of size 146×546 *pixels. The background value is 10 (black) while the brightest pixel is of intensity 255. The Kanji characters denote the word "sampling". (b) Two different target probabilities* π_1 *(left) and* π_2 *(right). Each pixel reports the probability of belonging to the foreground region. Black and white correspond to probabilities 1 and 0, respectively.*

We perform this experiment on two target distributions with parameters

- $(\mu_{\mathrm{BG},1} = 15,\ \sigma_{\mathrm{BG},1} = 10,\ \mu_{\mathrm{FG},1} = 50,\ \sigma_{\mathrm{FG},1} = 10)$ and
- $(\mu_{\mathrm{BG},2} = 10,\ \sigma_{\mathrm{BG},2} = 20,\ \mu_{\mathrm{FG},2} = 30,\ \sigma_{\mathrm{FG},2} = 30)$.

Figure 5.6b shows the color-coded π_1 and π_2 side by side. The second case is somewhat more difficult since pixels in the whole image have a reasonable probability of belonging to region 1. The lowest probability in this second case is $\min(p_{\mathrm{FG}}(I(x)) = 0.348$. In the first case $\min(p_{\mathrm{FG}}(I(x)) = 0.00038$.

Fig. 5.7 plots the L^1 and L^2 errors versus CPU time for both algorithms. No burn-in phase is considered. We initialized GIMH with a random level set with values in $[-0.5, 0.5]$. We used bubbles on a grid to initialize DRS.

Due to larger step sizes, GIMH approximates the target distributions faster during the first few seconds. In the first case, we measured a lower L^1 er-

ror for DRS' samples after approximately 8s; the L^2 error is lower after 4s. This is due to the neighbor dependencies in the GIMH masks, which cause the samples to be biased toward the mode. For a broader target distribution such as π_2, this difference becomes more significant. While GIMH improves the approximation only during the first computation second, DRS constantly decreases the error. The resulting probability maps are shown in Fig. 5.8. The results from GIMH are clearly biased toward the mode. For π_2, GIMH fails to represent the $p = 0.348$ confidence level in the vicinity of the Japanese characters. The probability map of DRS approximates the target distributions more accurately in both cases. It is, however, less smooth though due to its discrete character. The map becomes smoother the longer we run the algorithm. Figures 5.8d and 5.8f show that also the DRS approximation slightly overestimates the probability of the foreground region. We believe that this is because the floating particle distribution has not yet converged to the vicinity of its equilibrium. The effect on the summary statistics of the geometric subspace, however, is minor. Different initial floating particle distributions may be tested in future research.

5.4.2 Topology change

The toy example in Fig. 5.9 illustrates how off-boundary sampling enables DRS to change the shape genus.

Figure 5.9a shows a simple shape with a hole. The image is corrupted with Poisson noise. We initialize the chain with a circular shape (indicated by the red circle). After only a few iterations, new connected components emerge. Floating particles are more frequently introduced and removed at the boundary of the object where the image gradient is large. In Fig. 5.9b the traces of several accepted floating particles are visible, not only in the interior of the triangle, but also at the corners. Traces of topological changes are annotated with red arrows. The algorithm require 1.4s for 200000 iterations on this image of size 300×300 pixels.

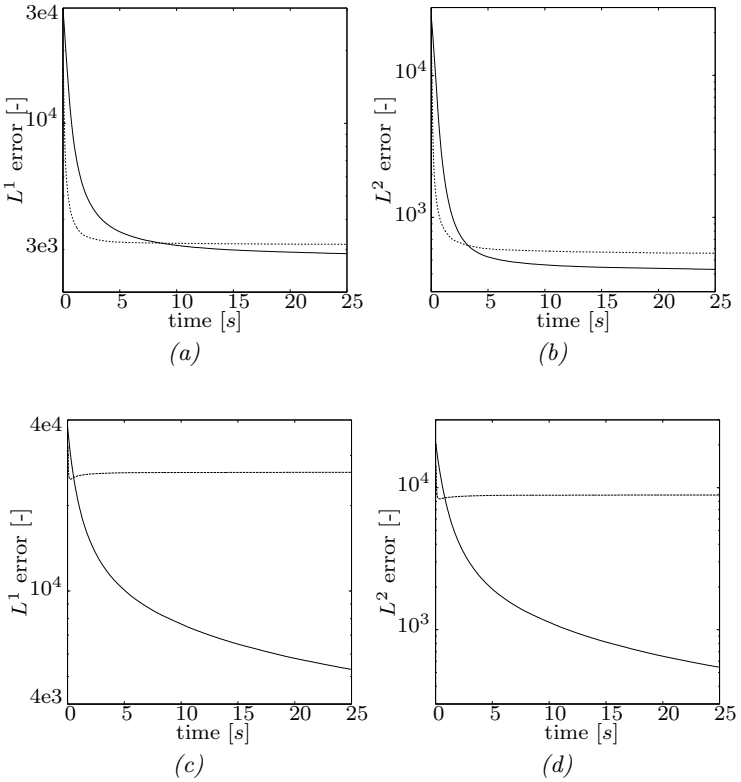


Figure 5.7: Sampling error versus CPU time. The solid and dashed lines show the errors for DRS and GIMH, respectively. (a) and (b) show the L^1 and L^2 distances between the reconstructed distribution and the ground truth π_1. (c) and (d) show the distances to π_2. We consider the first 25 seconds of computational time without any burn-in phase.

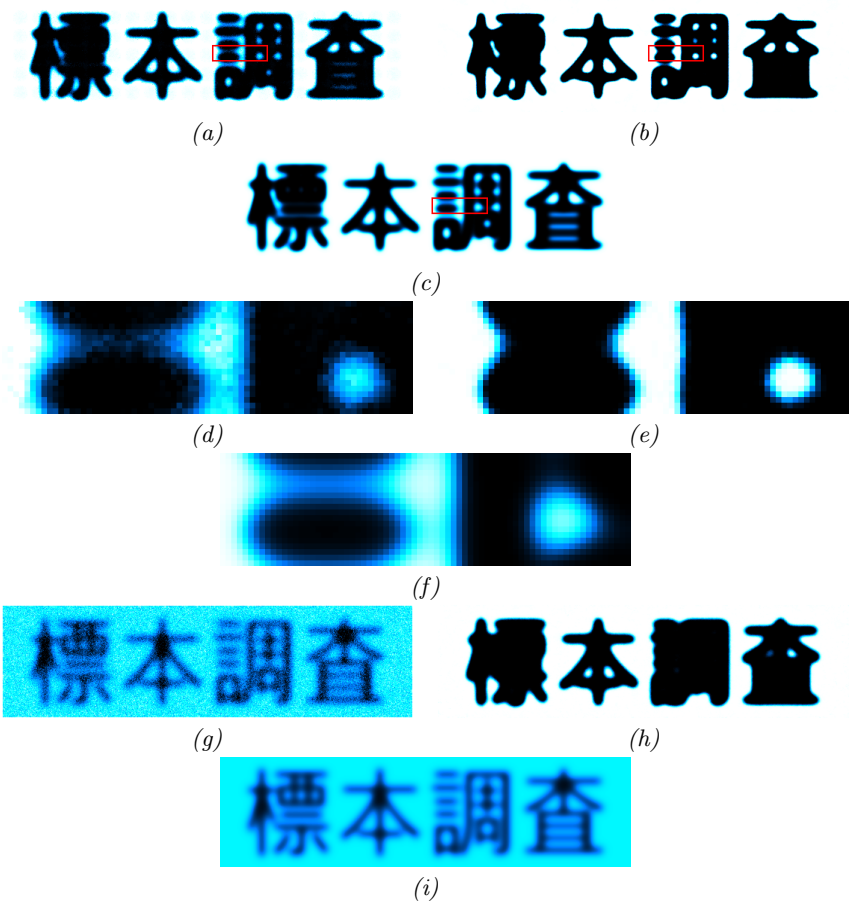
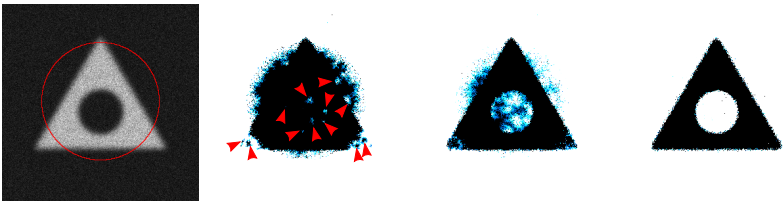


Figure 5.8: Visualization of the resulting probability estimates. (a) DRS estimation of π_1. (b) GIMH estimation of π_1. (c) Ground truth π_1. (d),(e), and (f) show (a),(b), and (c) in the respective zoom window (red). (g) DRS estimation of π_2. (h) GIMH estimation of π_2. (i) Ground truth π_2.



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

Figure 5.9: Illustration of the ability of DRS to change the contour genus using floating particles. (a) Initialization contour overlaid on the original image data. Panels (b),(c), and (d) show the probability maps after 10 000, 50 000, and 150 000 iterations, respectively. Red arrows mark topological changes.

5.4.3 Deconvolution using smooth and conditional proposals

Besides robust identification of the probability mode, sampling the shape space provides information about the shape of the target probability. The first goal of this subsection is to analyze certain aspects forward-simulation-based deconvolution. In Secs. 5.3.2.2 and 5.3.3.2 we introduced conditional and smooth proposals. Both techniques require additional computations. Conditional proposals require building a second pmf from particles in the neighborhood of a previously drawn particle. Biasing the proposal, i.e., assigning state-dependent weights to the particles, is computationally expensive for two reasons: First, the weights need to be calculated and updated. Second, we need to draw a particles from a discrete probability distribution (categorial sampling). If the energy is expensive to evaluate, such efforts may be amortized. The deconvolving energy is expensive, depending on the support of the PSF. The second goal of this subsection is to elaborate on the different alternative proposals.

We consider the artificial ground-truth image in Fig. 5.10a. It contains three connected components. After blurring the image with a PSF the objects are difficult to separate. For this example we use a Gaussian PSF of width $\sigma_{\mathrm{PSF}} = 7$. It is superimposed in the upper-left corner of the ground-truth image in Fig. 5.10a. We simulate the image-formation process of a microscope by a convolution followed by corruption with multiplicative

Poisson noise. We consider a simpler case with SNR ≈ 12 and a more difficult case with SNR ≈ 3. The resulting test images are shown in Figs. 5.10b and 5.10c.

We separate the deconvolution problem into the geometric and the photometric subproblem, which are strongly coupled. We first assume the correct photometric parameters to be known. We further assume the object to have piecewise constant intensity. Without any further prior knowledge we find that many shapes are plausible. This is because of the information loss due to low-pass filtering with the PSF and noise. As illustrated in Fig. 5.10e, the confidence bands are wide, even for a high SNR. We observe that the probability of recovering the corners decreases. At an SNR of 3 the probability map is shown in Fig. 5.10f. We conclude that the geometric subproblem is ill-posed since many original states are plausible.

If the photometric parameters are unknown, the uncertainty further increases. Even when initializing with the ground-truth segmentation, variations of the geometry are still plausible. Since the photometries are estimated conditional on the shape, the intensities will be over-estimated at some point. This causes the shape to retract, which further increases the estimated intensities. The sampler therefore walks along a probability ridge and finds a state that seems intuitive: The intensity estimates are twice the ground-truth intensities, and the geometry covers only half of the area of the ground-truth object. The process then continues reducing the area and increasing the intensities. This does not happen because they are more probable than the ground-truth solution, but because there is a large number of plausible states to explore. We therefore need to introduce a prior in order to "keep the shape together", which causes lower intensity estimates. Penalizing the contour length is a good solution, as long as we are not reconstructing filamentous structures.

Iterative approaches For comparison, we apply three different, commonly used deconvolution algorithms: Tikhonov-Miller inverse filtering (van Kempen et al., 1997), total-variation regularized Richardson-Lucy (Richardson, 1972; Lucy, 1974; Dey et al., 2006), and a wavelet-regularized version of the thresholded Landweber method (Vonesch and Unser, 2008).

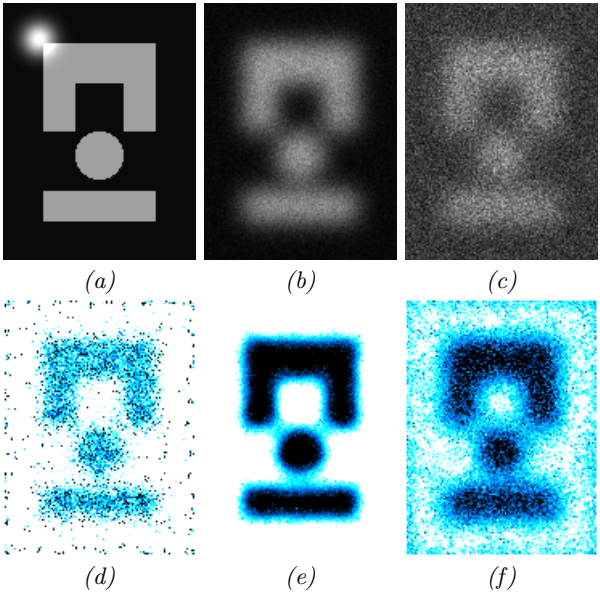


Figure 5.10: (a) Ground-truth data with overlaid point-spread function. (b) Convolved image with Poisson noise of SNR 12. (c) Convolved image with Poisson noise of SNR 3. (d) Intermediate probability map from DRS applied to (b) when using $\mathcal{E}_{\text{Poisson}}^{\text{PC,dec}}$ *without any prior. (e) Same as (d), but with region intensities known to DRS. (f) Same as (e), but applied to (c).*

An overview of popular deconvolution algorithms is discussed by Sarder and Nehorai (2006). We ran all algorithms for 200 iterations. We used the implementations in the software bundle accompanying the work of Vonesch and Unser (2008).

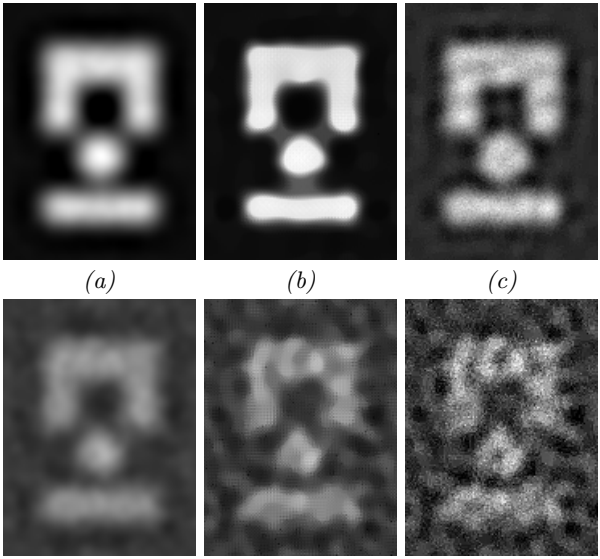
A direct comparison is difficult. The three deconvolution algorithms mentioned above are image reconstruction methods rather than segmentation methods. The results, however, illustrate the difficulty of the task.

Figure 5.11 shows the results from the three algoritmhs for both SNR cases. Tikhonov-Miller inverse filtering is known to have a low deconvolution but high a denoising effect. In our experiment, this algorithm performs well on the low-intensity image. In both cases, however, details are lost. The Richardson-Lucy algorithm is appropriate for Poisson and recovers the shapes quite well in the high-SNR case. We observe that the algorithm roughly creates 3 regions of similar intensity: the background, the foreground, and an intermediate region. The intermediate region is located where the sampler reports uncertainty in Fig. 5.10e. For the low-SNR case, the algorithm seems to amplify the noise. Similar results with less denoising are obtained with the thresholded Landweber algorithm.

Results We regularize the geometric problem using the curvature regularization discribed in Sec. 3.2.2.2. We run the DRS and RC algorithms for both SNR cases. We run DRS for 125 000 iterations with a burn-in phase of 62 500 iterations. Timings are reported in Tab. 5.1.

Fig. 5.12 shows sampling (DRS) and optimization (RC) results using the data image at SNR $\approx$ 12 (Fig. 5.10b) for three different prior weights, $\lambda = \{4, 10, 15\}$. Interestingly, for $\lambda = 10$ the certainty at corners increases compared with $\lambda = 4$. When further increasing λ to 15, the shape is oversmoothed and corners are rounded. Since RC sticks to local minima close to the mode, we assume that the landscape is rugged in the vicinity of the mode.

Fig. 5.13 shows DRS and RC results on the image with SNR $\approx$ 3 with $\lambda = 1, 2$, and 5. We adapted λ to values 1,3, and 5. $\lambda = 1$ is too small and the same effect as described above occurs. $\lambda = 3$ seems to impose a reasonable prior; the objects are almost separated. The sampler re-



(a) *(b)* *(c)*

(d) *(e)* *(f)*

Figure 5.11: Deconvolution results of different algorithms using the artificial images in Figs. 5.10b and 5.10c. The upper row correspond to results for an SNR of 12, the lower row are results for the image with SNR 3. (a),(d) Tikhonov-Miller inverse filtering. (b),(e) Richardson-Lucy algorithm. (c),(f) Wavelet-regularized thresholded Landweber deconvolution.

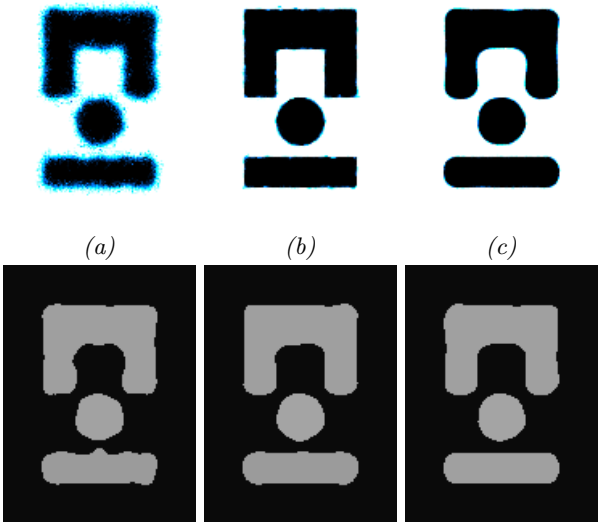


Figure 5.12: Sampling (upper row) and optimization (lower row) results for the deconvolution task on the image in Fig. 5.10b (SNR $\approx$ 12) for different regularization parameters λ: (a),(d) $\lambda = 4$; (b),(e) $\lambda = 10$; (c),(f) $\lambda = 15$.

ports uncertainty between the circle and the bottom bar. With $\lambda = 5$ the confidence-band width gets narrower, as expected when imposing a stronger prior. The sampler, however, takes longer to reach equilibrium. From Fig. 5.13f we can interpret that the Markov chain disconnects the upper from the middle object after the burn-in phase. We observe that the optimizer better separates the objects for low λ. For $\lambda = 10$ and $\lambda = 15$, however, RC gets stuck in local minima induced by the prior.

Proposal scheme The results reported are generated using conditional and smooth proposals. In order to show the effect of other proposals we additionally run all 4 combinations of proposals (i.e., standard, biased, conditional, and biased-conditional) for the SNR $\approx$ 12 case. Acceptance rates for two different PSF sizes, $\sigma_{\text{PSF}} = 7$ and $\sigma_{\text{PSF}} = 2$, are reported

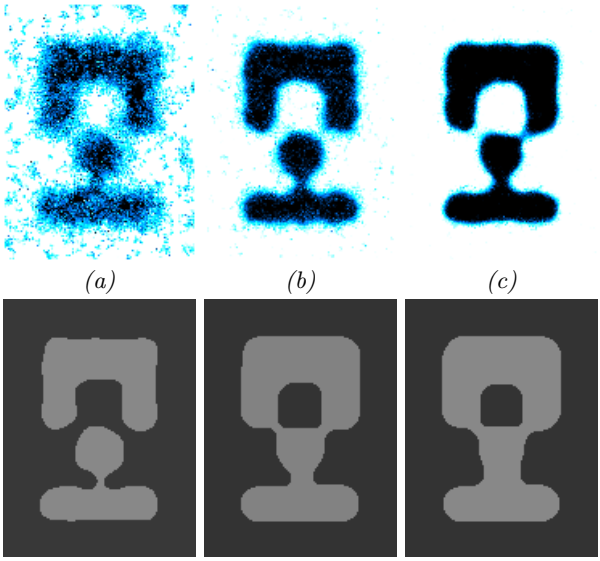


Figure 5.13: Sampling (upper row) and optimization (lower row) results for the deconvolution task on the image in Fig. 5.10c (SNR $\approx$ 3) for different regularization parameters λ: (a),(d) $\lambda = 1$; (b),(e) $\lambda = 3$; (c),(f) $\lambda = 5$.

proposal		$\sigma_{\text{PSF}} = 7$		$\sigma_{\text{PSF}} = 2$	
		avg. AR [%]	time [s]	avg. AR [%]	time [s]
		37.4	41	11.1	**9**
	biased	**55.4**	49.6	16	14.7
cond.		42.6	**36**	18	10
	biased	53.8	39	**29.6**	18.5

Table 5.1: Summary of ARs and average computational times of the plots in Fig. 5.14. The experiments are performed with DRS applied to the image in Fig. 5.10b. Two different PSF sizes for the Poisson-noise-deconvolution image model are tested on 4 different proposal schemes.

in Fig. 5.14. For $\sigma_{\text{PSF}} = 7$, all acceptance rates are above 35%. This is due to the flat PSF. When adding or removing a point the intensity difference at a single pixel within the PSF support is small. The Poisson noise model tolerates such deviations (see Eq. (3.14) in Sec. 3.1.2). When biased proposals are used, the acceptance rate is too high.

With a Gaussian PSF of width $\sigma_{\text{PSF}} = 2$, some pixel intensities change drastically when adding or removing a pixel. Consequently, $|\Delta\mathcal{E}|$ is larger and the decision to accept or reject the move becomes more certain. As motivated in Sec. 5.3.2.2, conditional proposals are well suited in this case. The intensity change per pixel is less if two nearby particles with different candidate labels are applied at the same time (recall Fig. 5.3). Fig. 5.14b shows that using conditional proposals increases the acceptance rate from 10% to 18%, while the computational time only increases slightly from 9s to 10s (see Tab. 5.1). Biased proposals significantly increase the computation time. This is because when the PSF is smaller the energy is less expensive to evaluate, and the proposal-computation overhead becomes relatively more expensive. Combining both specialized proposal schemes increases the AR almost by a factor of 3 while doubling the computational time. For particle moves on images with high SNR, the intensity differences increase. Therefore, the decision to accept or reject a move becomes more certain. Consequently, for images with high SNR, conditional proposals are indispensable.

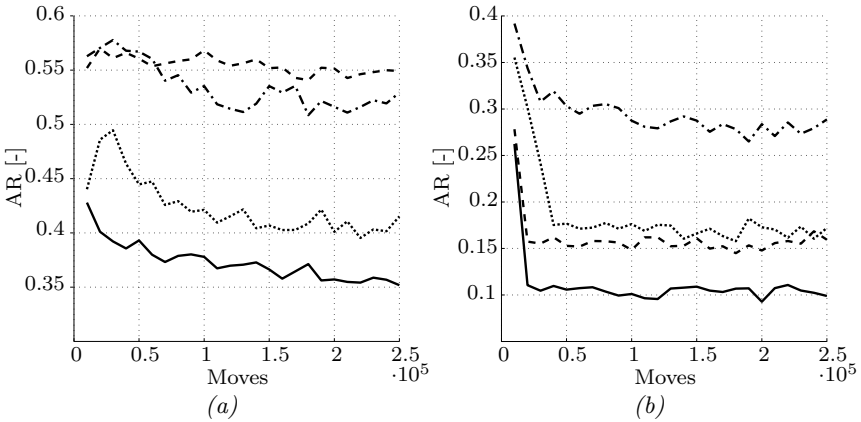


Figure 5.14: Acceptance rates (AR) versus particle moves for deconvolving segmentation of the image in Fig. 5.10b (SNR ≈ 12) with a Poisson noise model. We ran 250000 iterations per run without conditional proposals, and 125 000 iterations for runs with conditional proposals (since they perform 2 particle moves per iteration). (a) AR for $\sigma_{PSF} = 7$. (b) AR for $\sigma_{PSF} = 2$. Solid, dashed, dotted, and dash-dotted lines represent standard, biased, conditional, and biased-conditional proposals, respectively.

5.4.4 Protein quantification in cells - Robust low intensity segmentation

Fluorescence microscopy allows quantifying concentration of fluorescently labeled proteins in cells or cell compartments. After image segmentation, pixel-wise intensity values are used to reconstruct the amount of protein within the region of interest. Such quantification relies on robust segmentation algorithms. Uncertainty estimates of object boundaries further improve the sample-set quality. Fig. 5.15b shows a region of interest of a noisy real-world image. We initialize one region around each local intensity maximum after low-pass filtering. The initial region boundaries are indicated in red in Fig. 5.15a. This initialization process produces 6 false positives at the image boundary.

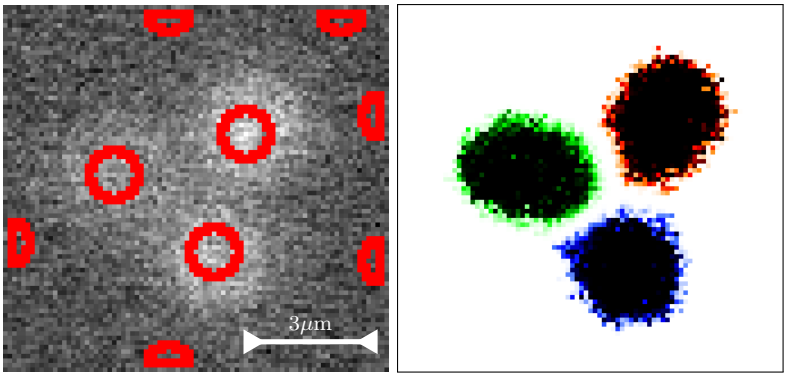
We run DRS for 2 000 000 iterations using a Gaussian-noise piecewise constant image model. For each label the sampler outputs a probability map for the corresponding region. The region maps for the initial false positives are empty, i.e., they report a zero probability. Figure 5.15b shows the probability maps for the three different non-empty regions. For better visibility of the results we slightly separated the probability maps in Fig. 5.15b. We observe smooth probabilities at locations where cells are touching each other since the likelihood term is not decisive in-between the cells. The pixel-wise probabilities are hence dominated by the prior.

In order to improve the quantification quality, we discard all pixels that have a probability below 95% of belonging to a foreground region.

5.4.5 Comparison with optimization

5.4.5.1 Imaginal wing disc in 2D

The following example illustrates DRS' robustness w.r.t. the initialization. We use an image of complex-shaped cellular structures from Zartman et al. (2012). We compare the result from local optimization using RC (see chapter 4) with the DRS approach. For this comparison we consider a 2D

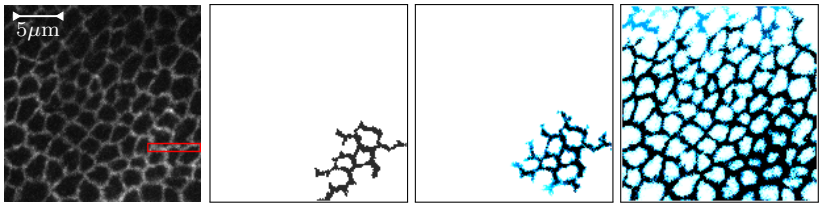


(a) (b)

Figure 5.15: Touching fluorescent cells. (a) Contours (red) of initial regions using local maximum detection after blurring. (b) Probability map of a multi-region segmentation after 2 000 000 iterations. For visibility of the results, the probability maps of the three cells have been artificially separated. White corresponds to probability 0 and black to probability 1.

version of the image; the full image is segmented in the following section. Fig. 5.16 shows the results from both algorithms. The intensity signal is inhomogeneously distributed across the image. We use a piecewise constant image model, which does not match the data image in Fig. 5.16a. In order to challenge the algorithms, we initialize the contour to a small region as marked in Fig. 5.16a. For the sake of the experiment, we set $q_f = 0$ in order to avoid new seeds from which the contour could explore the image. Starting from the small initial patch, RC converges in 0.22s; the result is shown in Fig. 5.16b. It corresponds to a local posterior mode. Within the same time DRS produces the confidence map in Fig. 5.16c. After 10 million iterations, which approximately takes one minute, almost all cell membranes are correctly segmented.

Note that although the DRS result reaches a lower-energy solution, the RC result is nevertheless reasonable. The optimizer is constrained to find a connected region of constant intensity. It would, when initialized with the final state of the sampler, split the region into many smaller regions in order to cluster different region intensities.



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

Figure 5.16: (a) Maximum intensity projection of an image stack of fluorescently labeled cell membranes. The red square indicates the contour of the initial region. (b) The solution found by the RC. (c) The posterior after the same time using DRS. (d) The solution found by DRS, without off-boundary sampling, after 10 000 000 iterations. Image data are from Zartman et al. (2012).

5.4.5.2 Mitochondria

We apply RC and DRS to a real-world application for mitochondria segmentation. We used a piecewise smooth external energy and curvature regularization for the internal energy (see Sec. 3.2.2.2). Both algorithms are initialized using a binary mask obtained by Otsu thresholding. We run RC until convergence and DRS for 5 000 000 iterations. For DRS, we obtained contours by thresholding the probability map at a probability value of 0.5. At image locations with strong signal, the results of both algorithms are nearly identical. Regions with weaker signal that were missed by the Otsu-based initialization are still delineated by DRS.

Figure 5.17 shows an example result. More results are shown in Appendix B.1.

5.4.6 Imaginal wing disc - A 3D example

We apply DRS to a 3D fluorescence microscopy image stack of the imaginal wing disc of a Drosophila larva. We aim at segmenting all cells and reconstructing their neighborhood graph. The image is difficult to segment because of the thin structures to reconstruct, the low SNR, and the inho-

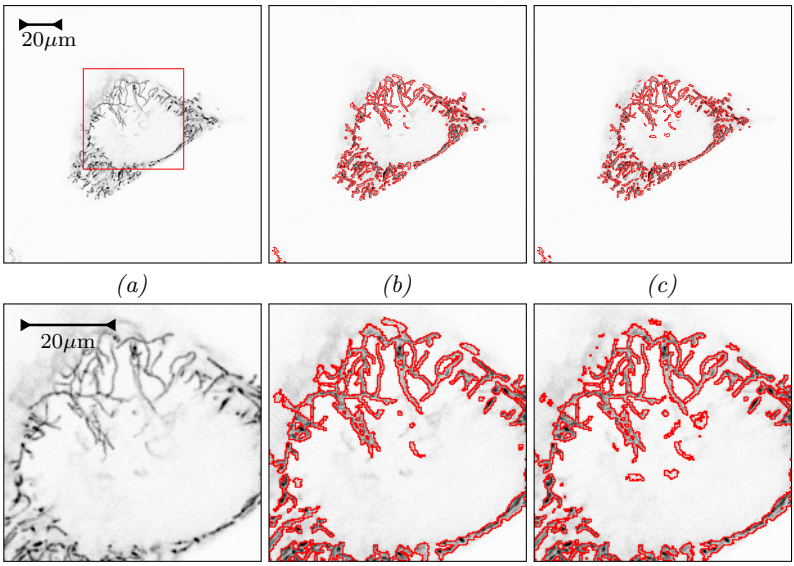


Figure 5.17: Comparisons of RC and DRS segmentations for images of fluorescently labeled mitrochondria in 2D. (a) The image data of size 512×512 *pixels (image: Kathy Ushakov, Dept. of Cell Physiology and Metabolism, University of Geneva Medical School). (b) Contours of the RC segmentation. (c) Contours corresponding to the 0.5 iso-level of the DRS probability map. (d) Close-up data image. (e) Close-up of RC's contours. (f) Close-up of the 0.5 DRS' confidence contour. See Appendix B.1 for additional comparisons.*

mogeneously distributed fluorescent protein. So far, automated methods did not provide satisfactory segmentation quality in 3D.

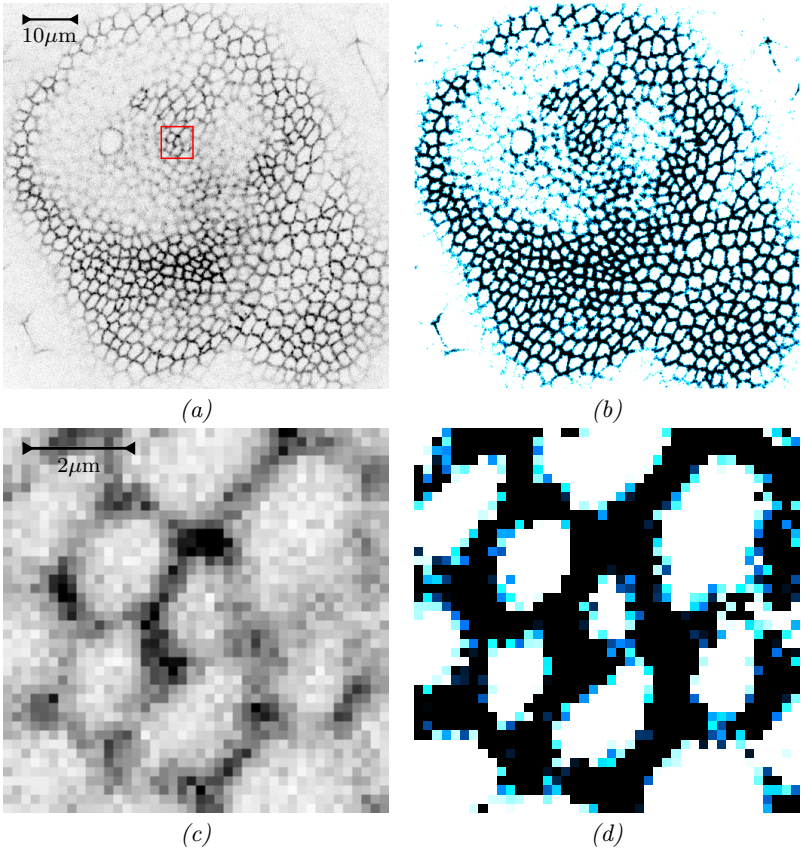
The image is of size $512 \times 512 \times 12$ voxels. A slice of the data is shown in Fig. 5.18a together with a scaled inset in Fig. 5.19a. We use a piecewise smooth Gaussian image model with discrete approximated contour length prior (see Sec. 3.2.2.1). For this energy, we perform 5 million DRS iterations in 13.6 minutes. For comparison, the same number of iterations with the computationally more efficient piecewise constant energy takes 1 minute. The average acceptance rate is 16%. Both experiments are performed on a laptop computer with an Intel i7 processor. The overall memory usage of the algorithm is about 240 MB.

We present 3D shape sampling results, e.g., two confidence surfaces, from DRS in Figs. 5.18b and 5.19. Due to performing energy calculations only in the vicinity of the contour, the present algorithm allows sampling shapes in 3D in reasonable time. We are not aware of any previous implementation producing posteriors for 3D images. GIMH could in principle be extended to 3D, but its sampling time increases linearly with the number of voxels.

5.5 Discussion and outlook

We presented a Markov-chain Monte Carlo method for discrete shape sampling. We tailored the Metropolis-Hastings algorithm to a discrete multi-region sampling (DRS) algorithm. The resulting Markov chain samples shapes of different topology (if not otherwise constrained) despite the explicit shape representation. The algorithm follows the idea of performing many small, computationally cheap moves and certainly accept some of them. In order for this strategy to be efficient, energy differences need to be computationally cheap to evaluate for local shape perturbations.

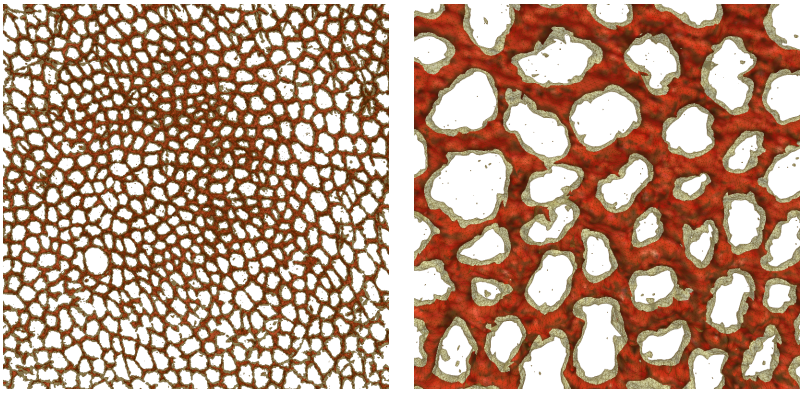
We represented the discrete proposal distribution for local shape perturbation using particles. The proposal was designed such as to locate particles in the neighborhood of the contour. We have shown that assigning weights to particles biases the proposal distribution toward smooth shapes in order to improve the acceptance ratio. For the same reason we introduced conditional moves that combine two particle moves. We have proven that



(a) *(b)*

(c) *(d)*

Figure 5.18: (a) A section from the image stack of a developing Drosophila wing. The image stack was recorded by Zartman et al. (2012). (b) Corresponding probability map of a 3D sampling result using a piecewise smooth Gaussian noise model. (c) Close-up data image in a region where wing disc is curved. (d) Probability map corresponding to (c).



(a) (b)

Figure 5.19: Segmentation confidence surface visualized using Paraview (Henderson, 2007). The red surface interpolates the 0.9 probability iso-level of the 3D probability map. The transparent surface shows the 0.1 confidence level.

the associated Markov chain is ergodic.

We benchmarked the algorithm for two artificially generated target probabilities and compared the results with those from the Gibbs-inspired Metropolis Hastings sampler (Chang and Fisher, 2012). GIMH rapidly reduced the L^1 and L^2 distances to the target distribution due to its larger step size. GIMH converged to a biased distribution. DRS better reconstructed the target probabilities and had smaller L^1 and L^2 errors after a few seconds of compute time.

We applied the algorithm to artificial and real-world images. We first demonstrated the sampler's ability to change the contour topology. We then analyzed the different proposal strategies for the deconvolving segmentation problem. We conclude that for small PSFs and high SNRs conditional proposals are beneficial. Biased proposals amortize their computational overhead for expensive energies, e.g., deconvolving energies with large PSFs and piecewise smooth energies with large local windows. By comparing the algorithm to RC we showed that the sampler is able to overcome local minima and is hence robust to bad initializations. We

demonstrated multi-region sampling using the real-world application for protein quantification in touching cells. We concluded the applications with a 3D real-world segmentation and reconstruction of the corresponding confidence surfaces.

A sampling approach is more robust than local optimization approaches w.r.t. initialization and local optima. The sampler additionally provides confidence bands for the segmentation if the image-formation model is accurate. The sampler, however, has shortcomings though. First, DRS is computationally expensive. For many applications RC is more efficient, as it needs less energy evaluations to find a good segmentation. Second, RC uses the connected-component prior in order to estimate the number of regions in an image. DRS needs the number of regions to be fixed beforehand.

We propose to relax the first shortcoming using approximate initialization of the target distribution and parallel computation.

Good initializations start the chain close to the equilibrium distribution. In practice, this can be achieved using hierarchical resolution approach: A pre-run using a down-sampled image could provide a low-resolution probability map for initialization on next finer level. This approach exploits the fact that neighboring pixel are likely to have similar probability values.

Parallel hardware can be exploited in various ways. The simplest approach is to run multiple Markov chains and then combine their samples. This approach, however, wastes computational time for each chain's burn-in phase. For energies with finite computational support, such as piecewise smooth energies, domain decomposition may be appropriate, as illustrated in Fig. 5.20. Domain decompositions have the advantage of being more cache efficient and also they allow processing images that do not fit the memory of a single computer. An implementation could for example be realized using the Parallel Particle Mesh library (PPM) (Sbalzarini et al., 2006). Within each subdomain Λ_{ij}, a DRS algorithm samples the shape space. Ghost layers are updated on demand. If the whole Markov process is in equilibrium, the data in the ghost layers represent a partial sample from the target distribution. It is not clear though, and remains to be shown, whether such a scheme converges to equilibrium.

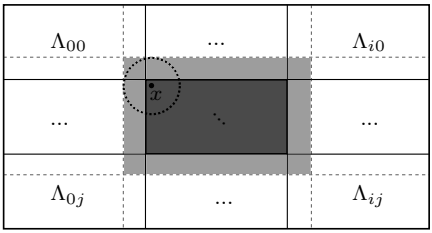


Figure 5.20: Domain decomposition for parallel distributed-memory implementation of DRS using the PPM library (Sbalzarini et al., 2006). Λ_{ij} are the subdomains of the image. Ghost layers (light-blue shaded) are needed to compute the energy differences. The dotted circle indicates the local support of an energy difference computation at particle location x.

The second shortcoming is more severe. In order to extend the method to estimate the number of regions, careful reconsideration on region number regulation is necessary. The prior used in chapter 4 seems unpractical, since too many regions would be created due DRS' stochastic nature. One solution might be to consider a soft region penalty, as is for example used in optimization (Delong et al., 2010; Brox and Weickert, 2004). A jump-diffusion process, as proposed by Tu and Zhu (2002), could be used to switch between different region models.

Another appealing enhancement of DRS seems to be *parallel tempering*. Parallel tempering (Geyer, 1991) robustifies standard sampling methods by running multiple Markov chains in parallel. Each of the chains is assigned a different temperature T_k. In case the underlying model is a Markov random field, this temperature corresponds to the temperature parameter of a Boltzmann distribution. Else, the temperature is an inverse factor on the negative logarithm of the target distribution. One chain runs at temperature $T_0 = 1$, all others run on higher temperatures. From time to time, chains exchange their states according to a Metropolis-Hastings criterion. The method can be seen as running a parallel stochastic process that generates good proposal for the chain at T_0. Parallel tempering only requires slight modifications of a standard MCMC implementation.

CHAPTER

SIX

On-line confidence estimates with application to microtubule tracking

6.1 Introduction

We present on confidence estimation for object detection and segmentation where the object size is below pixel resolution. We hence use a continuous explicit geometry representation. Small objects are difficult to segment in noisy images. This is because few data points are available for intensity estimation. According to the law of large numbers, the estimation variance is hence larger. We therefore impose a shape prior by explicitly representing the object in parametrized form. This prior is additionally with the fixed-region-number constraint from chapter 5.

Using knowledge about the image-formation process allows sub-pixel estimates. Sub-pixel accuracy is achieved by exploiting convolution artifacts in image formation. We hence use a deconvolving energy (Eq. (3.12)) in the likelihood term. Intensities, however, are not estimated separately, but become part of the state, for the reason outlined above. We improve inten-

sity estimation by filtering the posteriors over time-lapse image sequences using Bayesian recursive filters. The underlying assumption is that the intensities vary smoothly across consecutive frames.

We design a Bayesian recursive estimator, i.e., a particle filter. Particle filters represent the posterior distribution using a particle-based, i.e., non-parametric, function approximation. In order to avoid problems such as degeneracy of particle weights or sample impoverishment, we use a *MH-within-particle filter*: Within each frame, we run an adaptive MH algorithm. Since the posterior pdf may drastically change between consecutive frames, we need to adapt the MH proposal distribution. We present a versatile proposal adaption scheme inspired by the covariance-matrix-adaptation evolution strategy (CMA-ES). CMA-ES is a competitive stochastic optimization method introduced by Hansen and Ostermeier (1996). It uses a Gaussian proposal and adapts its covariance matrix in order to robustly search the state space.

We validate the confidence estimates of our method on synthetic data at different SNR and present an application to the analysis of the intrinsically stochastic motion of astral microtubules during metaphase in yeast cells. Particle filtering without on-line confidence estimates has previously been successfully used to track microtubules in live cells (Smal et al., 2008).

In the following section we briefly introduce particle filtering and related problems. We then present the generic part of the present algorithm with a short discussion on convergence of adaptive schemes in subsection 6.3.1. In Sec. 6.4.2 we tailor the algorithm for microtubule length tracking by carefully designing a likelihood and a prior. We then numerically validate the algorithm and conclude with a short discussion in Sec. 6.6.

6.2 A short introduction to particle filtering

A comprehensive overview of particle filters (aka. *sequential Monte Carlo* or *condensation algorithm*) can be found in Cappé et al. (2007); Maskell (2001). From Maskell (2001) we adopt notation and the following relations.

Particle filters are sequential importance samplers. *Importance sampling*, of which the MH algorithm is one particular instance, is used is to approximate an unknown pdf π that can only be queried (evaluated). A particle cloud $\{x^i\}, i = 1 \dots N$, is drawn from a proposal distribution q and the target distribution is approximated as

$$\pi(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i), \quad w^i \propto \frac{\pi(x^i)}{q(x^i)}, \quad \sum_{i=1}^{N} w^i = 1, \tag{6.1}$$

where δ is the Dirac-delta function. The w^is are the particle weights, satisfying partition of unity. The set $\{x^i, w^i\}$ hence is a *random measure*.

In chapters 4 and 5 we used particles to represent a discrete approximation of the proposal distribution. Here, particles are weighted support points for a posterior pdf approximation. Formally, the posterior is approximated as:

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(\boldsymbol{x}_{0:t} - \boldsymbol{x}_{0:t}^i), \; w_t^i \propto \frac{p(\boldsymbol{x}_{0:t}^i|\boldsymbol{z}_{1:t}^i)}{q(\boldsymbol{x}_{0:t}^i|\boldsymbol{z}_{1:t}^i)}. \tag{6.2}$$

The vector $\boldsymbol{x}_{0:t}$ contains all states up to time t. The $\boldsymbol{x}_{0:t}^i$ are the support points with associated weights w_t^i, which together form a random measure that characterizes the posterior. The posterior $p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{1:t})$ incorporates the measurements $\boldsymbol{z}_{1:t}$ and priors up to time t. Assume that at time $t+1$ a new measurement becomes available. The algorithm incorporates this new information in the prior. The new state will therefore be *filtered* with this prior model. In order to adapt to the resulting new posterior, the particles adapt their weights and also move.

Using this Markovian property, Bayes' rule, and the assumption that the proposal at time $t-1$ only depends on $\boldsymbol{x}_{t-1}$ and the new measurement $\boldsymbol{z}_t$, one can derive the weight update (see (Maskell, 2001)):

$$w_t^i \propto w_{t-1}^i \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \cdot p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{z}_t)}. \tag{6.3}$$

The choice of proposal $q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{z}_t)$ is crucial for the algorithm's performance. Setting it equal to the prior is often a good choice. The prior $p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)$ describes the expected system dynamics. For the present appli-

cation, this is further discussed in Sec. 6.4.2.2. Prior and proposal usually can, by design, be efficiently evaluated. The particle likelihood $(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is often the most expensive part to compute. If it can be factorized, though, all likelihood marginals factor out when normalizing the weights.

In the present application of microtubule-length tracking in digital images, these factorable marginals correspond to pixel for which the pixel-wise likelihoods are identical across all particles. Hence, the likelihood calculation reduces to an energy-difference evaluation as discussed in chapter 3.

Degeneracy, resampling, and sample impoverishment In sequential schemes, the posterior changes over time. This change can include a movement of the distribution's mode and of its shape. This may render that particles misplaced, and Eq. (6.2) then only badly approximates the posterior. In practice, one observes that in such situations most particles have a weight close to 0. This is called the *degeneracy* problem. The posterior representation can be improved by resampling procedures. Resampling sets all particle weights to 1. In order to still represent the same distribution, particles are moved accordingly. This procedure has a of linear time-complexity. Resampling, however, causes particles to occupy coinciding state-space locations which may result in a bad posterior approximation due to insufficient particle support. This problem is referred to as *sample impoverishment*.

6.3 Proposed algorithm

We use the adaptive procedure outlined in Algorithm 9 to relax the sample impoverishment problem and allow for multi-scale likelihood functions. The algorithm starts by defining the initial set $\mathcal{P}$ of N particles (line 1 of Algorithm 9), which are later sampled from the proposal distribution q (line 4). This is followed by computation of the normalized particle weights (lines 6–7) and MCMC iterations (lines 8–18) comprising two phases: annealing (lines 10–13) and classical MCMC moves (line 17). The annealing phase provides a good starting point and proposal distribution for the classical MCMC moves.

Within the MCMC sub-routine (line 17), the particle positions are updated by a Metropolis-Hastings algorithm conditional on the value of the posterior pdf. This yields the new particle set $\mathcal{P}_u$ with unchanged weights. We use a Gaussian with a diagonal covariance matrix as the proposal distribution for the MCMC moves. Good proposal distributions should be similar to the desired stationary distribution (Chib and Greenberg, 1995). During annealing, the covariance matrix of $\mathcal{P}_u$ (line 9) approximates the covariance matrix of the stationary distribution. We estimate the covariance matrix as:

$$\tilde{\Sigma} = \sum_{i=1}^{N} w^i (x^i - \tilde{x})(x^i - \tilde{x})^T , \qquad \tilde{x} = \sum_{i=1}^{N} w^i x^i . \tag{6.4}$$

Since the initial proposal distribution is very broad, most particles are degenerate. In order to cluster the particles at positions of high likelihood, we perform a resampling operation (line 11). After resampling, Σ is adapted (annealed) in line 12. The parameter $0 < c < 1$ is the exponential learning factor of the adaptation of Σ. Annealing ends as soon as the adaptation becomes insignificant. We quantify significance by the Kullback-Leibler divergence D_{KL} between two normal distributions with means 0 and covariance matrices Σ and Σ' (line 14). If its derivative D'_{KL} is smaller than a user-defined threshold ϵ, annealing stops (lines 14–16).

As soon as a user-defined termination criterion (e.g., maximum number of iterations) is met (line 18), the desired estimates $\hat{x}$, $\hat{\Sigma}$ for the moments of the posterior are computed based on the union of all particle sets since the end of the annealing phase (line 19).

6.3.1 A note on adaptive proposals for MCMC

It is difficult to prove that the MH algorithm converges to the target distribution π when using adaptive proposals. Often, however, it is argued that such a proof is unnecessary in practice, because one can stop the adaptation at some point and freeze the proposal from then onward. Andrieu and Thoms (2008) counter that investigations on convergence properties

Algorithm 9 Particle Filter with Adaptive Annealing

Initialize sample set $\mathcal{P}_1 = \{x_0^i\}$
for $k = 1 \dots N_{\text{frames}}$ **do**
  $u = 1$, $annealing = true$
  Draw samples $x_k^i \sim q, w^i = 1/N$
  Initialize Σ using Eq. 6.4 on $\mathcal{P}_1$
  Compute weights $w^i = \mathcal{L}(I_k, x_k^i)p(x_k^i)$
  Normalize weights $w^i = w^i / \sum_j w^j$
  repeat
    Estimate Σ' using Eq. 6.4 on $\mathcal{P}_u$
    if *annealing* **then**
      $\mathcal{P}_u = \text{RESAMPLE}(\mathcal{P}_u)$
      Adapt $\Sigma = (1-c)\Sigma + c\Sigma'$
    if $D'_{\text{KL}}(\mathcal{N}(0,\Sigma)||\mathcal{N}(0,\Sigma')) < \epsilon$ **then**
      $annealing = false$, $u' = u$
    MCMC$(\mathcal{P}_u, \Sigma_{j,j})$, $u = u + 1$
  until *not*(termination criterion)
  Compute $\hat{x}$ and $\hat{\Sigma}$ using Eq. 6.4 on $\cup_{j=u'}^{u} \mathcal{P}_j$

of adaptive MCMC schemes are justified, because adaptation only works well when the proposal is already a good approximation of π.

Haario et al. (2001) presented an adaptive random-walk MH algorithm. They proved that their update scheme leads to an algorithm that converges to the target distribution while updating the proposal. Their result inspired research on adaptive proposals until today. Roberts and Rosenthal (2007) proved that *diminishing* (or *vanishing*) adaptation and *containment* together are sufficient for π-ergodicity, i.e., for the chain to converge to π. The latter condition is usually difficult to prove (Bai et al., 2009). A rigorous discussion of convergence for present rank-μ-based update scheme is beyond the scope of this work. We therefore freeze the proposal after some time and use preliminary runs as an optimization procedure. We verify that the adaptation did not diverge to "bad" proposals by inspection of the covariance entries.

6.4 Application to microtubule tracking

In this subsection we implement the above scheme with likelihood and prior for the application of microtubule tracking. We discuss in detailed the likelihood construction since the formation model differs slightly from the models discussed in Sec. 3.

3D digital videos were acquired using a confocal microscope and show the microtubule tip and Spindle Pole Body (SPB) proteins Spc72p and Bik1p, labeled with green fluorescent protein (Fig. 6.1a–c). The dynamics of the system are driven by microtubules randomly switching between phases of assembly and disassembly. As fluorescent proteins are constantly transported along the microtubules, they may cause object intensities to increase over time, even though the total intensity in the whole image decreases due to photobleaching.

6.4.1 State space

We model the spindle and microtubule tip with 3 connected, diffraction-limited objects. The three objects appear as scaled 3D PSFs. The state space $\Xi \subset \mathbb{R}^{12}$ is continuous. A state $\boldsymbol{x} \in \Xi$ is

$$\begin{aligned}\boldsymbol{x} = (&x_{\mathrm{SPB}_1}, y_{\mathrm{SPB}_1}, z_{\mathrm{SPB}_1}, c_{\mathrm{SPB}_1},\\ &x_{\mathrm{SPB}_2}, y_{\mathrm{SPB}_2}, z_{\mathrm{SPB}_2}, c_{\mathrm{SPB}_2},\\ &x_{\mathrm{Tip}}, y_{\mathrm{Tip}}, z_{\mathrm{Tip}}, c_{\mathrm{Tip}})\,.\end{aligned} \tag{6.5}$$

For each object, (x, y, z) are spatial coordinates and c is the object's intensity.

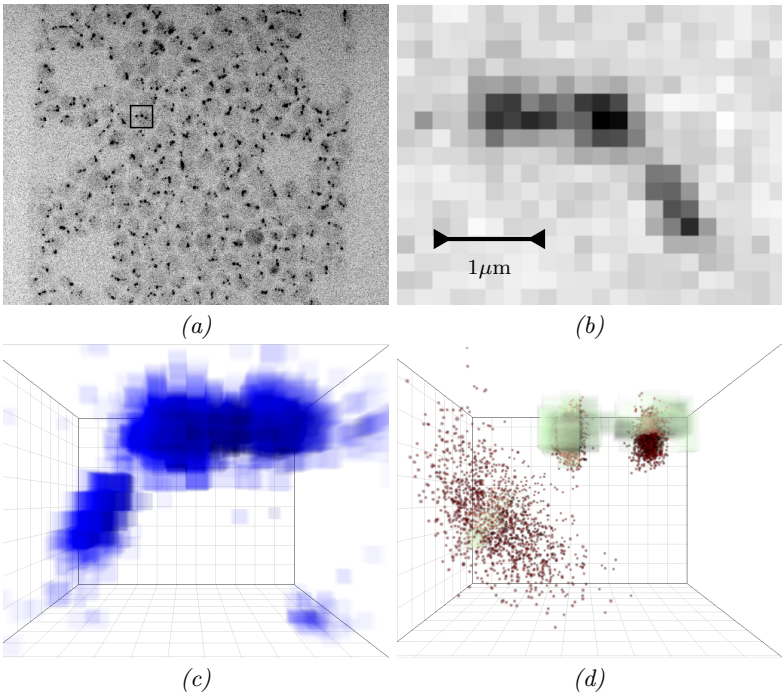


Figure 6.1: (a) An example micrograph of labeled yeast cells in maximum-intensity projection. Intensities are inverted for better visualization. (b) Maximum-intensity projection of the labeled microtubules in a single cell during mitosis. (c) 3D stack of the microtubules in (b). The spots from left to right are the microtubule tip, SPB 2 (old pole), and SPB 1 (new pole). (d) Image with overlaid proposal distribution. Each particle of the particle filter gives rise to 3 dots, corresponding to the 9 estimated position parameters.

6.4.2 Bayesian image model

6.4.2.1 Image acquisition and likelihood

We used an electron multiplying charge-coupled device (EMCCD) camera, where electrons exiting the CCD sensor are multiplied in a pipeline before the voltage is read out. The dominant noise sources are Poisson-distributed shot noise and excess noise (multiplication noise). Excess noise is introduced by the stochastic process of impact ionization in the electron-multiplying pipeline, while shot noise is imposed by the discrete photon counts. The pdf of the number e of electrons exiting the multiplying pipeline for a number E of electrons entering is modeled as (Hynecek and Nishiwaki, 2003):

$$p(e, E) = \frac{1}{\sqrt{2\pi F^2 G^2 E}} \exp \frac{-(n - GE)^2}{2F^2 G^2 E}\,, \tag{6.6}$$

where $G > 1$ is the linear digital gain and F the excess noise factor (ENF). The number E of electrons entering the pipeline can be determined from the gray-scale intensity values O in the image as:

$$E = (O - D)\frac{C}{G \cdot Q_\lambda}\,. \tag{6.7}$$

D is the mean of the "dark image", recorded using the camera system without any light source, C is a camera-specific constant provided by the manufacturer, and Q_λ the quantum efficiency of the CCD sensor at the recording wavelength λ.

Using this noise model, we formulate for each voxel $x \in I$ in the observed image I the marginal $p(I_v|J_v)$, the pdf of the pixel having intensity I_v given an expected intensity J_v. For a specific realization of J, i.e. an expected image, $p(J_v) = 1$, such that the marginal is given by the joint pdf in Eq. 6.6 with $E = J_v$. The expected image J (in the absence of noise) is computed from the current state vector $\boldsymbol{x}$ of the particle filter and the PSF of the

microscope as:

$$J(\boldsymbol{x}) = B + \sum_{i=1}^{2} c_{\mathrm{SPB}_i} \cdot K([x,y,z]_{\mathrm{SPB}_i}) + c_{\mathrm{Tip}} \cdot K([x,y,z]_{\mathrm{Tip}})\,. \qquad (6.8)$$

$K([x,y,z])$ is the microscope PSF centered at position (x,y,z). All pixels of the background image B are set to the most frequently occurring pixel intensity in the observed I. Construction of the marginals is completed by modeling the PSF of the microscope. As a first approximation, we use a Gaussian whose width is fitted to recorded images of point sources. In addition, we also measured the true PSF by imaging fluorescent beads, and generated a high-resolution look-up table for the PSF as described in appendix C. The two models are compared in Sec. 6.5.2.

Using the noise and PSF models outlined above, and assuming the noise in different pixels to be statistically independent, the likelihood function can be constructed by multiplying all marginals, thus:

$$p(\boldsymbol{z}|\boldsymbol{x}) = p(I|J(\boldsymbol{x})) = \prod_v p(I_v|J_v)\,. \qquad (6.9)$$

Evaluation of this likelihood function is computationally efficient since many voxels have identical intensity values and need not be considered separately.

6.4.2.2 Proposal and prior

We set the proposal equal to the prior. The prior distribution has, in our framework, the role of constraining the search space to regions of high likelihood. Before the annealing step of the algorithm, it is equal to the proposal distribution q of the particle filter. We choose a uniform distribution $q \sim \mathcal{U}[\mathcal{I}]$ over some interval $\mathcal{I}$. For the present application, we use cell-fixed spherical coordinates. The old SPB serves as the origin, the optical axis of the microscope corresponds to zero azimuthal angle, and the x-axis in image space is used as the reference for the polar angle. The interval $\mathcal{I}$ around x_k is specified in these coordinates. The resulting particle-represented proposal is illustrated in Fig. 6.1d and Fig.6.2a. Fig-

ure 6.2b shows how the MH algorithm concentrates the proposed particles around locations of high likelihood.

6.5 Validation and results

We use the mean of the particles to estimate the object positions. Furthermore, we assess uncertainty estimates by measuring the empirical variance of the approximated posterior distribution.

6.5.1 Validation data

We validate the error estimates on artificial feature point data. Time series of artificial images Z_i are generated according to Eq. 6.8 based on known state vectors $\boldsymbol{x}$ and $B = 50$. The feature point positions perform a random walk, yielding a movie of simulated Brownian motion. As a PSF, we use the measured PSF of the microscope determined from images of fluorescent beads. Assuming the PSF to be radially symmetric, the intensities are averaged along circles of different radii around the intensity centroid of the bead's image (Sbalzarini and Koumoutsakos, 2005). We average the PSF determined from 5 different images in order to reduce the nose. Since the width of the likelihood function decreases with increasing SNR, we test adaptation to different likelihood widths by simulating different SNRs. This is done by scaling the peak intensity $Z_{\max}$ of the PSF according to: $\mathrm{SNR} = (Z_{\max} - b)/(F \cdot \sqrt{Z_{\max}})$. The ENF is measured to be $F = 1.5$ for our equipment. Finally, we replace all pixel values Z_v in Z by Gaussian random numbers with mean Z_v and standard deviation $FG\sqrt{Z_v}$ (cf. Eq. 6.6) with $G = 40$. The resulting images correspond to a pixel size of $160 \times 160\,\mathrm{nm}$ and a distance between confocal planes (voxel depth) of $200\,\mathrm{nm}$.

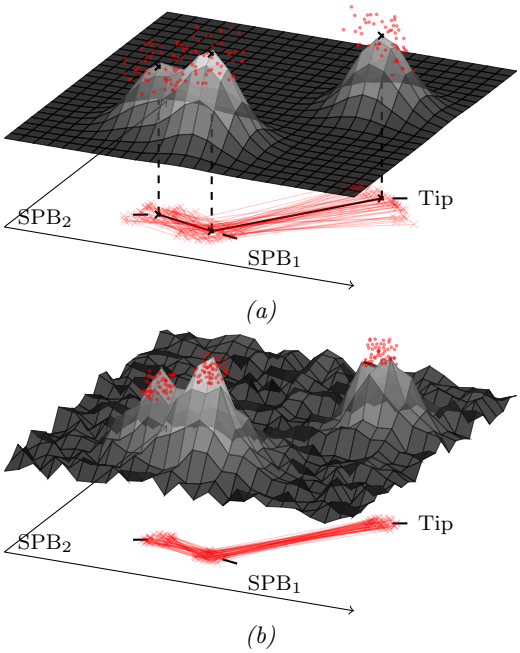


Figure 6.2: 2D illustration of the proposal and posterior pdfs with the images I and J. (a) The particles in red represent the proposal pdf. Each particle stores 3 space coordinates and 3 intensity values. The state space is sampled in the vicinity of the estimate of the previous frame. The gray-scale surface represents J based on one particular particle (indicated in black in the x-y plane). The weight of the particle is calculated based on the distance between the simulated image and the actually measured image. (b) Particle representation of the posterior pdf after adaptive MCMC (see text for details). The gray-scale surface represents the data image I.

6.5.2 Results

We use the present adaptive particle filter to track the Brownian motion of the feature points in the artificially generated movies and assess the quality of the confidence estimates (not of the tracking itself!). We compare the estimated standard deviation $\hat{\sigma}$ of the x and z positions (y is identical to x, not shown) – as determined by the $\hat{\Sigma}$ in Algorithm 9 – to the standard deviation with respect to the known true positions x_i of the validation data: $\sigma = \sqrt{\frac{1}{T}\sum_t^T (\hat{x}_t - x_t)^2}$. All estimates are averaged over $T = 200$ frames, leading to averaged true and estimated uncertainties $\langle\sigma\rangle$ and $\langle\hat{\sigma}\rangle$, respectively. We use $n = 40$ particles and 40 MCMC iterations after annealing. Figure 6.3 shows the results. The confidence estimates as determined by Algorithm 9 (solid lines) are always larger than the ground-truth standard deviations of the benchmark data (dashed lines), providing conservative uncertainty estimates. Using a Gaussian PSF model (squares), both the estimated and the true variance are larger than when using the true PSF (circles). For the true PSF, our algorithm is able to accurately estimate the tracking confidence with mean standard deviation differences $\Delta\sigma = |\langle\sigma\rangle - \langle\hat{\sigma}\rangle| < 6\,\text{nm}$ in the lateral and $\Delta\sigma < 14\,\text{nm}$ in axial direction. The standard deviation of the estimator decreases with increasing SNR and is low throughout ($<5\,\text{nm}$ laterally and $<20\,\text{nm}$ axially). In 99% of the frames, the confidence estimates are accurate within $\pm 3\hat{\sigma}$.

6.6 Conclusion

We have presented and validated an object tracking framework with on-line confidence estimates based on the *MH-within-particle filter* paradigm with adaptive proposal annealing. The framework provides on-line sub-pixel estimates of the tracking uncertainty at all SNRs tested. If the system and imaging models do not accurately describe the imaged process, these uncertainties are lower bounds.

The presented adaptive MCMC scheme handles well likelihood functions of different widths. On multimodal posterior distributions, however, the adaptation scheme leads to overly broad proposals. Consequently, this

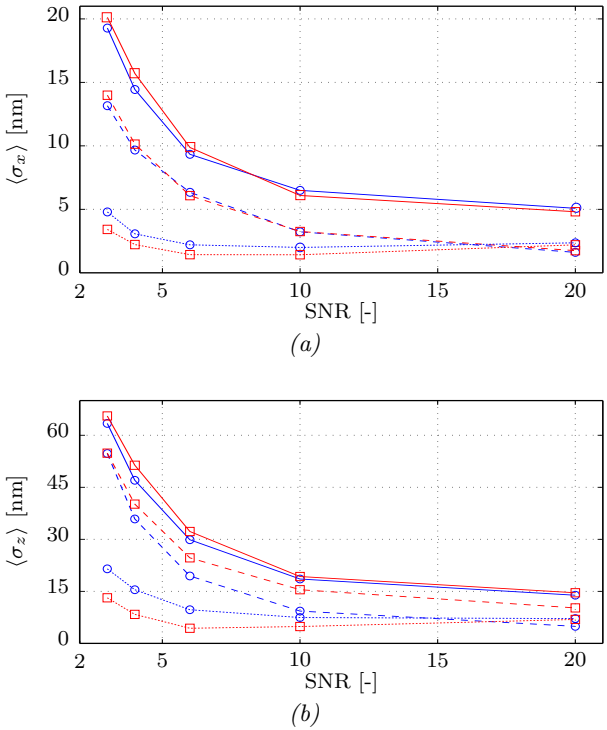


Figure 6.3: Means of real (dashed lines) and estimated (solid lines) tracking uncertainties at different SNRs using a Gaussian PSF approximation (squares) and the true, measured PSF (circles). The dotted lines represent the standard deviation of the estimator. (a) In the lateral x direction, the mean estimate deviates by $3.8 \ldots 6.0\,nm$ (< 0.04 pixel). (b) In the axial z direction, the difference of the mean errors is between 11.3 and $13.5\,nm$ (< 0.07 voxel depths).

leads to frequent rejection of MCMC moves and to bad estimation of the tracking uncertainty. In order to allow for the multi-hypothesis capability of particle filters, the adaptation scheme would need to be run on particle sub-populations. This could be, e.g., well done by using a clustering algorithm.

A more fundamental issue concerns the convergence properties. The present adaptation scheme needs to be disabled after the burn-in phase in order to guarantee π-ergodicity. Proving π-ergodicity for the present annealing scheme could be exploited significantly lower the computational cost.

The present framework, as well as the PSF estimation tool, are implemented as plug-ins to the open-source platform ImageJ. The tool has been used to process a large number cells for a global analysis of microtubule dynamics in S. cerevisiae (Rauch, 2012).

CHAPTER

SEVEN

Conclusions

We addressed the problem of unsupervised image segmentation by constraining a region to be a connected component. We presented an efficient topology-controlled optimizer that works with the above region definition. Furthermore, we presented a posterior sampling algorithm that assesses the segmentation robustness by reporting confidence estimates from the posterior. The resulting algorithm also robustifies the popular local optimization approach for deformable models using a *stochastic relaxation*.

All presented algorithms are based on Bayesian image models and allow incorporating prior knowledge about the image-formation process, as well as local and global shape prior. These algorithms were designed with focus on applications in fluorescence microscopy. However, they are also applicable to any other situation where an image-formation model can be formulated. Here, we have presented and implemented various models for piecewise constant and piecewise smooth images. We also considered image-formation models that involve a blurring process by a convolution with a PSF. We presented and implemented all models for images corrupted with Gaus-

sian noise, Gaussian noise with fixed variance, and Poisson noise. We did, however, not discuss the implementation of a deconvolving energy with a piecewise smooth image model. This is currently under development.

We presented a multi-region-competition image segmentation algorithm (RC) for black-box energies with unknown numbers of regions. The key assumption was that a region is a connected component. This amounts to a topological prior that locally regularized the problem of finding the number of regions in an image. We used a discrete explicit contour representation in order to make use of extended notions of digital topology. Characterizing topological events was intrinsically needed to ensure the definition of a region as a connected component. However, we took additional advantage of this topological control to enforce topological prior knowledge. RC proposes new contours using discrete particles as region-flip candidates. It optimizes the posterior by accepting particles in energy-rank order, while controlling the topology. The particle-based approach also allowed robustifying the local optimization scheme by approximating Sobolev gradients. This was achieved by energy-difference filtering on the particle cloud. The resulting contours are smoother and need less regularization. We benchmarked RC against PEARL, an EM algorithm using graph-cuts for geometry optimization. We benchmarked 2D and 3D images with PC and PS image models. RC performed well on expensive energies, because energy differences are only needed in the vicinity of the contour. The global geometry updates of PEARL resulted in low-energy solutions, provided that the algorithm had converged to the correct energy-landscape funnel. However, direct comparison was difficult since the two algorithms use different definitions of a region.

We identified RC's dependence on the initialization to be the major weakness of the algorithm. This undesirable property renders the usage of the algorithm non-trivial. Two less severe issues concern with contour-oscillation detection and accurate calculation of differentials. Both aspects are related and originate from RC's discrete character. The former causes the algorithm to perform unnecessary iterations. The latter leads to approximated energy gradients, which might cause contour oscillations.

In order to further improve segmentation robustness w.r.t. the initialization, we introduced the discrete region sampling (DRS) algorithm. DRS

is a Metropolis-Hastings algorithm that samples discrete shapes from the posterior distribution. DRS represents the proposal distribution using a discrete particle-based function approximation. Particles were mostly located in the vicinity of a contour and at large intensity-gradients. Using particle weights, we biased DRS' proposal toward smooth shapes. DRS relies on efficient energy-difference computations in order to sample a large number of perturbed shapes. Summary statistics of these samples then provide a mean segmentation and confidence intervals for a fixed number of regions. We proved that the algorithm converges to the correct target distribution. We validated and benchmarked DRS on artificial and real-world images in two and three dimensions. We benchmarked DRS against Gibbs-inspired Metropolis Hastings (GIMH) introduced by Chang and Fisher (2012). GIMH is a multi-region, level-set method that allows larger step sizes. These large steps enabled GIMH to quickly approximate the target distributions. However, we have shown that GIMH converges to a target distribution that is biased toward the mode. This is, if the target probability is below 50%, the approximated probabilities are too low. And the approximated probabilities are too high if the target probability is above 50%. We have shown in theory and practice that DRS converges to the correct target distribution.

We have further discussed limitations, possible future enhancements, and open problems of the presented approach. First, DRS needs to know the number of regions a-priori. Second, DRS (as well as GIMH) is a computationally involved sampling algorithm. The convergence time and hence also the runtime depend on the initialization and the complexity of the target distribution. However, we discussed various possibilities to speed-up the algorithm including parallelism or multi-resolution approaches. Third, like RC, DRS suffers from approximated energy differences based on approximated differentials. This sometimes causes the probability map to be pixelated.

Moreover, DRS' discrete explicit contour representation is not able to sample sub-pixel shapes. We hence presented a particle-filter framework to track objects that are smaller than the pixel size using a continuous explicit representation. In order to address the well-known problem of particle-weight degeneracy and sample impoverishment, we presented an adaptive MCMC algorithm that was applied at each iteration. The scheme adapts

the covariance of the proposal to the local likelihood scale and therefore provides better particle-based representations of the posterior. Convergence results for the adaptive Metropolis-Hastings scheme, however, have not been presented. Furthermore, the algorithm is currently limited to unimodal posteriors.

Ongoing research in convexification schemes has recently provided tight approximations to the global minimum of the joint geometric and photometric problem under the Chan-Vese image model (Brown et al., 2011). It has, however, also been proven that it is not possible to find exact convex relaxations for certain energies (Paul et al., 2013). For such energies, shape-sampling strategies such as GIMH and DRS provide an alternative. In our opinion, shape sampling methods moreover have the potential to replace local optimizers, such as active contours, in non-real-time applications due to their robustness.

Combining the topological prior of connected regions with the presented sampling approach is appealing. We found the topological prior useful for unsupervised segmentation, since it links the notion of a region to physical objects. Unlike approaches that penalize the number of regions in the energy, the presented topological prior only concerns the spatial location of regions. It therefore has a local character. Shape sampling with unknown numbers of regions, however, has not been addressed in this work. We consider this task difficult, since the state space grows tremendously. How to softly impose an equivalent prior in the internal energy in order to enable unsupervised segmentation using shape sampling, is unclear to us to date.

APPENDIX

A

Region competition on the command line

We implemented the RC and DRS algorithms presented in chapters 4 and 5 in the open-source image-processing library ITK (Ibanez et al., 2005). Image processing in ITK is performed in a pipeline where image data is processed stage-by-stage, starting at a source and ending in a sink. The source often is an image reader and the sink an image writer. In-between, *image-to-image filters* perform the actual processing, i.e., reconstruction, registration, or segmentation. RC and DRS are implemented as ITK image-to-image filters.

We also implemented a hierarchy of energy functions, including all functions presented in chapter 3. RC and DRS offer an interface to use any of these energy functions in any linear combination.

We also implemented a command-line tool called *RegionCompetition* that allows controlling all options of the algorithms. Additionally, the tool offers different intitializations and some pre-processing capabilities. In this section we discuss all options for both algorithms and describe how to configure the energies.

A.1 Basic usage

By default, the RegionCompetition tool segments the image provided in the first command-line argument. The tool then writes the label image into the working directory. Furthermore, a parameter file with the same name is written in the *json* format. Such parameter files can be reused via the `--params <path>` option. The `-o` option can be used to determine the output file names. For example[1]

```
./RC cells_A.tif -o cells_A_segm.vtk -i 200 -s 1 1 3
./RC cells_B.tif -o cells_B_segm.vtk --params cells_A.json
```

The option `-i <N>` determines the maximum iteration number. The `-s <x y [z]>` option expects a d-dimensional vector indicating the pixel or voxel sizes in physical units. In the example above, we set the voxel depth to be 3 times larger than the pixel size. This spacing determines the scaling of the *world coordinates*. `RC --help` gives an overview of all options.

A.1.1 Initialization

The initialization automatically determines the starting number of regions. Each connected component is interpreted as a region, unless the option `--continue` is used. The `--continue` option reads label values from an initialization image, which can be used for providing a custom an initialization or when using DRS.

Five different initialization modes are available. The mode is chosen as follows

```
--init_mode [sphere|rect|otsu|blob_det|file].
```

- `sphere` initializes a spherical region around each local intensity maximum after blurring the image with a Gaussian filter of width σ. This initialization is suited for blob-like structures with approximately

[1] The examples provided in this appendix assume execution in a bash shell.

equal sizes. The parameter σ can be specified by the option `--init_sigma`. A good value is approximately one sixth of the object diameter. The radius of the region can be set with `--init_sr`.

- `rect` initializes with a single rectangular region. We used this for primordial germ cells in Fig. 4.12a. The distance from the image border can be set with `init_rb <x> <y> [<z>]`
- `otsu` performs an Otsu thresholding. Values above the threshold are considered FG. Different connected components define the different intial FG regions.
- `blob_det` performs a scale-space blob detection (Lindeberg, 1998). This is suitable if blobs of different radii are to be segmented. Upper and lower scale can be set with `--init_blob_min <arg>` and `--init_blob_max <arg>`, respectively.
- `file` expects an external initialization image whose path must be set with `--init_image <path>`.

A.1.2 Preprocessing

Preprocessing options can be listed using the `--hp` option. Preprocessing includes Gaussian filtering, masking, simple background subtraction, discarding outlier pixel, and image normalization.

- `--no_normalization` prevents the image normalized to values in $[0, 1]$. Normalization has the advantage, that the energy hyper-parameters are independent of the image contrast. Since normalization changes the energy, it should be disabled when using DRS, except when using DRS for robust optimization.
- `--preproc_sigma <`σ`>` determines the Gaussian kernel's standard deviation σ in world coordinates. The kernel is isotropic.
- `--mask_image <path>` causes all pixels with non-zero entries in the mask image to be ignored. This is implemented by setting the forbidden region label at these locations. See Sec. 4.4.2.

- `--masking_sigma <σ>` and `--masking_ths <t>` are used together for automatic masking as follows: All values lower than the intensity percentile t after blurring with a Gaussian of width σ are ignored by setting their labels to the forbidden region label.

- `--bgs <s>` performes a simple background subtraction using high-pass filtering. The parameter `<s>` determines the high-pass scale.

- `--disc_lower` and `--disc_upper` can be used to discard "dead" pixels. Pixel that have a larger (or smaller) value than the 0.995 (or 0.005) quantile of the image histogram are set to a local mean.

A.2 Topology-related options

Region topology is controlled by the following three options:

- `--no_fusion` prevents regions from fusing during energy minimization (only in RC, since in DRS regions never fuse).

- `--no_fission` prevents regions from splitting into two connected components during energy minimization.

- `--no_handle` prevents regions from developing holes. This does, however, not fix the topology of a region that has been initialized with a hole.

A.3 RC-related options

The following options are used with RC only:

- `--gpu` computes energy differences on the GPU using OpenCL. In order to activate this option, one needs to set the corresponding compiler flag when compiling the program. Currently, $\mathcal{E}_{LS}^{PS}$ and the curvature flow described in Sec. 3.2.2.2 are implemented as GPU versions.

- `--fast_evolution` causes energies to be calculated only if particles or their neighborhood have moved in the previous iteration. If not, the energies of non-moving particles are evaluated only at increasing time intervals.

- `--sobolev_sigma <`E`>` activates the Sobolev gradient approximations for RC. The argument E is the length scale of the smoothness term considered in the inner product; see Eq. (4.3).

A.4 DRS-related options

All sampling options can be listed using the `--hm` option. The following options are available with DRS only. The sampler DRS is activated with the option `--mcmc <s>` where s determines the step size. A step size of 1 should be good for most applications.

- `burn <k>` modifies the burn-in phase length. The number of burn-in iterations is $k \cdot i$ where i is the total number of iterations, $k \in [0, 1)$.

- `--biased_proposal` enables the biased-proposal mode in order to propose smooth shapes, see Sec. 5.3.3.2.

- `--offboundary <`q_f`>` sets the off-boundary sampling probability q_f, see Sec. 5.3.1.3.

- `--temp` modifies the temperature of the Gibbs distribution. Note that altering the temperature also changes the posterior. Temperature modification can be used, for example, for *simulated annealing*.

A.5 Energy-related options

All energy options can be listed using the `--he` option.

A.5.1 EXTERNAL ENERGY OPTIONS

Image-formation model energies The external energy is selected using the `--energy <`$\mathcal{E}_{\texttt{data}}$`>` option. The influence of the external energy can be weighted with the option `--data <arg>`. Currently implemented energy types are:

- `pc` for $\mathcal{E}_{\text{LS}}^{\text{PC}}$, see Sec. 3.1.1.1.
- `pcGauss` for $\mathcal{E}_{\text{Gauss}}^{\text{PC}}$, see Sec. 3.1.1.1.
- `pcPoisson` for $\mathcal{E}_{\text{Poisson}}^{\text{PC}}$, see Sec. 3.1.1.2.
- `pcDec` for $\mathcal{E}_{\text{LS}}^{\text{PC,dec}}$, see Sec. 3.1.2.1.
- `pcDecGauss` for $\mathcal{E}_{\text{Gauss}}^{\text{PC,dec}}$, see Sec. 3.1.2.1.
- `pcDecPoisson` for $\mathcal{E}_{\text{Poisson}}^{\text{PC,dec}}$, see Sec. 3.1.2.2.
- `ps` for $\mathcal{E}_{\text{LS}}^{\text{PS}}$, see Sec. 3.1.3.1.
- `psGauss` for $\mathcal{E}_{\text{Gauss}}^{\text{PS}}$, see Sec. 3.1.3.1.
- `psPoisson` for $\mathcal{E}_{\text{Poisson}}^{\text{PS}}$, see Sec. 3.1.3.2.
- `psLi` for the energy functional by Li et al. (2008).

If a deconvolving energy is used, one must set the PSF using the option `--psf <path>`. For PS energies, one can set R, the radius used to collect local statistics, with the `-r <`R`>` option.

Data-dependent balloon potential Besides the standard external energies above, additional energy terms can be enabled: `-b <`β`>` enables the balloon force with coefficient β, see Sec. 3.1.4.

Region merging The region merging threshold θ can be set with the option `-t <`θ`>`.

A.5.2 Internal energy options

Local energies The internal energy can be selected using the `--internal <`$\mathcal{E}_{\text{int}}$`>` option. All internal energies are weighted with λ, which can be set with the `-l <`λ`>` option. Currently implemented internal energies are:

- `curv` uses the curvature regularization by Kybic and Kratky (2009) as discussed in Sec. 3.2.2.2. The mask radius R_κ used for curvature regularization can be set with `--cr <arg>`.
- `manhattanLength` uses the cut metrics by Boykov and Kolmogorov (2003) as discussed in Sec. 3.2.2.1. We use a 16-neighborhood grid in 2D and a 26-neighborhood grid in 3D.
- `springPotential` simulates a regularizing flow induced by the potential resulting from connecting particles with springs. The parameter λ is interpreted as the spring constant. (Not discussed in this thesis)
- `expPotential` simulates a regularizing flow induced by an attractive exponential potential acting between particles. (Not discussed in the thesis)

Global shape priors Using global shape priors requires the user to define a template shape using the option `--shape_image <path>`. The order k of moments considered in the energy can be specified with the option `--moment_order <`k`>`.

APPENDIX

B

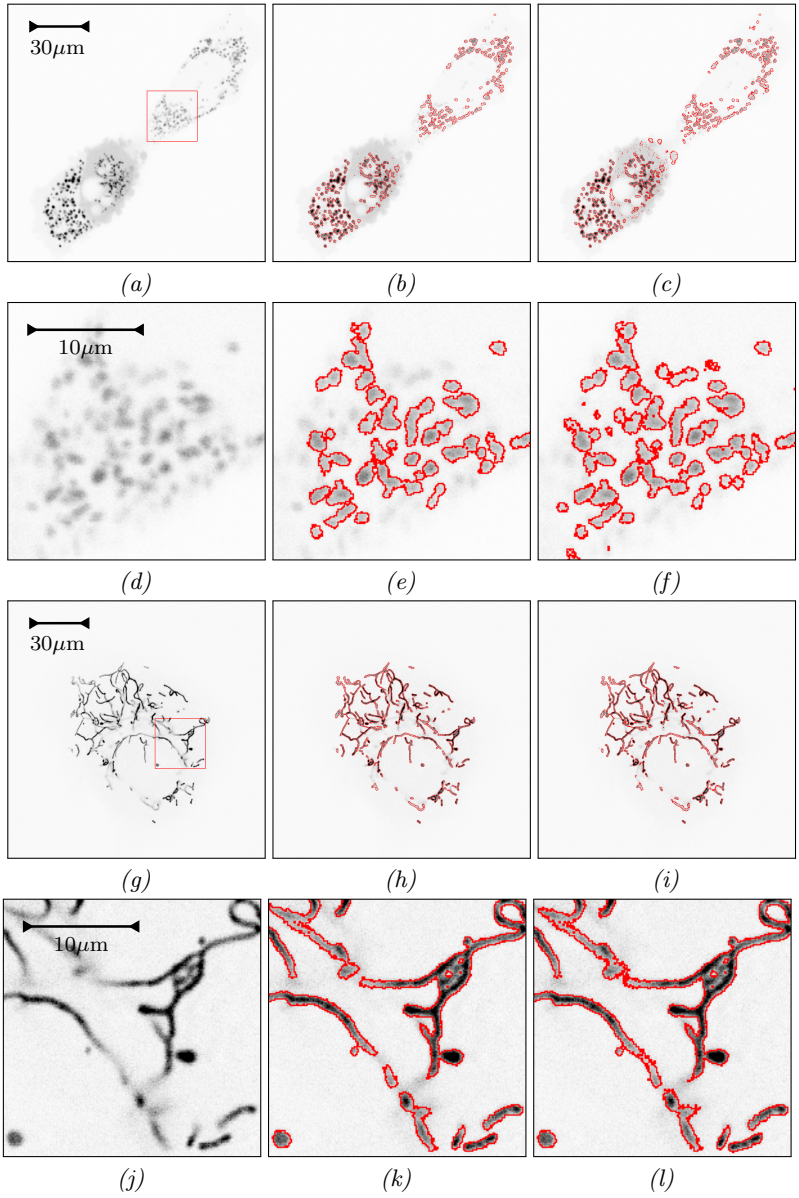
Additional results

B.1 Mitochondria

Figure B.1 compares DRS and RC segmentations using fluorescence microscopy images of mitrochondria in 2D. The figure complements the example in Sec. 5.4.5.2.

B.2 Proton emission patterns

We apply RC to images of proton emission patterns that occur when a high-energy laser beam hits a block of metal. Results are summarized in Fig. B.2. We initialized the RC algorithm with small bubbles at local intensity maxima. Since intensities are varying within filament patterns, we use a piecewise-smooth energy $\mathcal{E}_{\mathrm{LS}}^{PS}$. The images are almost noise free allowing for a small internal energy parameter λ. In order to minimize λ - and hence maximize the detection ability of dark filaments - we used Sobolev-gradient approximations.


30μm
(a)
(b)
(c)
10μm
(d)
(e)
(f)
30μm
(g)
(h)
(i)
10μm
(j)
(k)
(l)

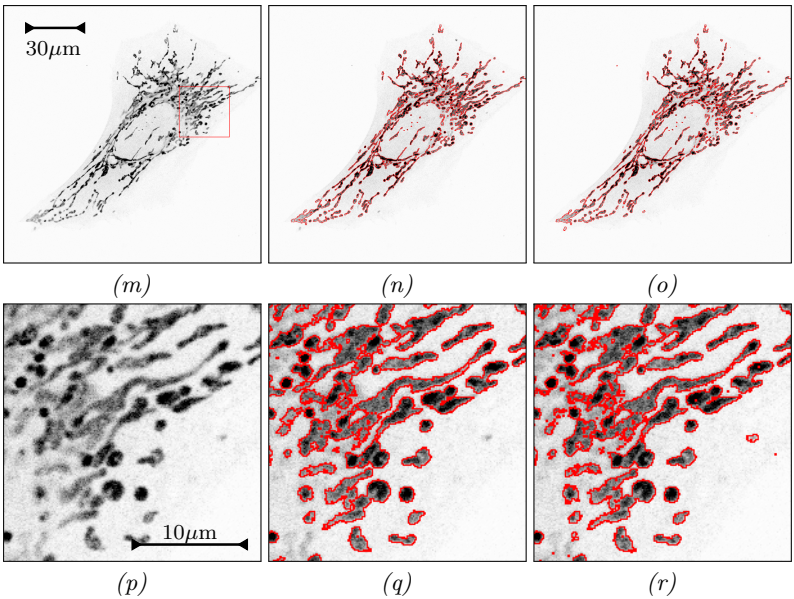


Figure B.1: Three comparisons of RC and DRS segmentations for images of fluorescently labeled mitrochondria in 2D. (a,g,m) The image data of size 1024×1024 *pixels (image: Kathy Ushakov, Dept. of Cell Physiology and Metabolism, University of Geneva Medical School). (b,h,n) Contours of the RC segmentation. (c,i,o) Contours corresponding to the 0.5 isolevel of the DRS probability map. (d,j,p) Close-up of the data image. (e,k,q) Close-up of RC's contours. (f,l,r) Close-up of the 0.5 DRS' confidence contour.*

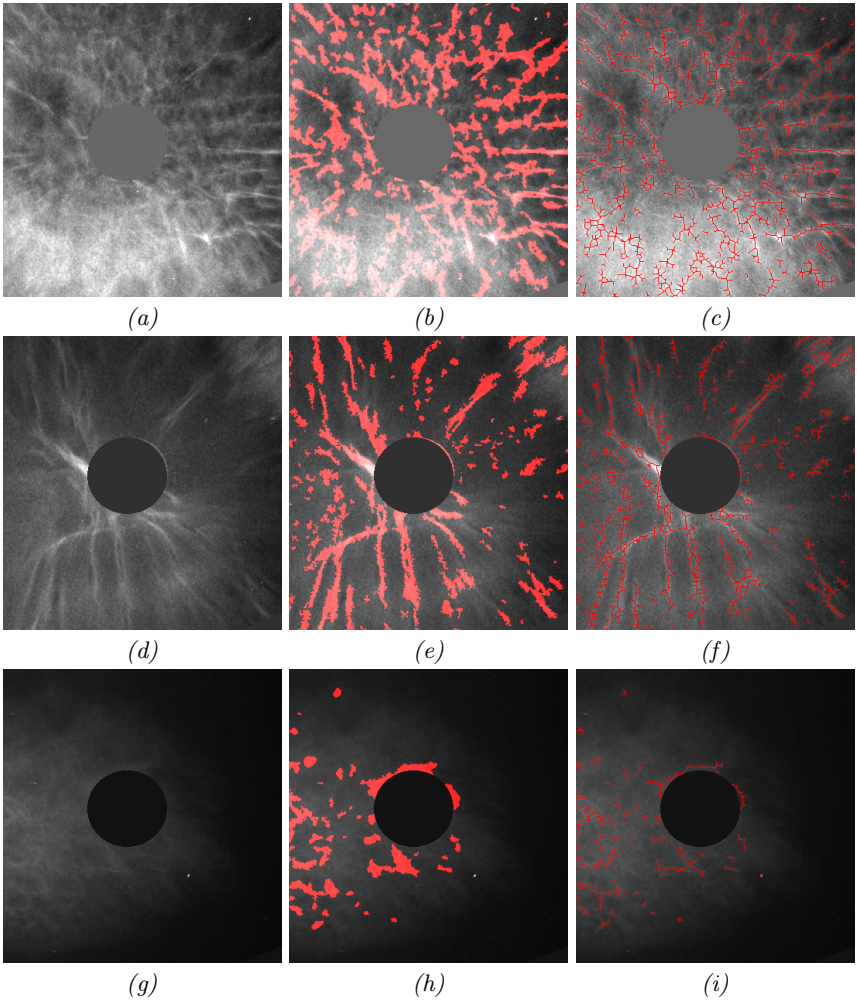


Figure B.2: Segmentations of proton emission patterns. (a,d,g) Raw data (image: Josefine Metzkes, Bussmann group, Helmholtz Center Dresden Rossendorf). (b,e,h) Piecewise smooth segmentations using RC. (e,f,i) Skeletons of (b,e,h).

APPENDIX

C

Point-spread function measurement

We measure PSFs from images of micro-beads. We first detect the beads in the image I using the centroid detection algorithm introduced by Sbalzarini and Koumoutsakos (2005). The image is then densely sampled in the vicinity of the centroid for different (r, z) pairs, where r and z are the distances from the centroid in lateral and axial directions, respectively. We therefore consider the PSF to be rotationally symmetric w.r.t. the axial direction. We aggregate values from I into a *PSF map*. Figure C.1 illustrates the intensity sampling procedure and shows an example result. If there are multiple samples for one (r, z)-entry, the values are averaged. Empty PSF-map entries are interpolated using bi-linear interpolation.

Note that in 3D fluorescence microscopy the PSF depends on the axial position of the emitting light source. This effect is neglected here.

The procedure has been implemented as a plugin for the open-source image-processing platform ImageJ (Schneider et al., 2012).

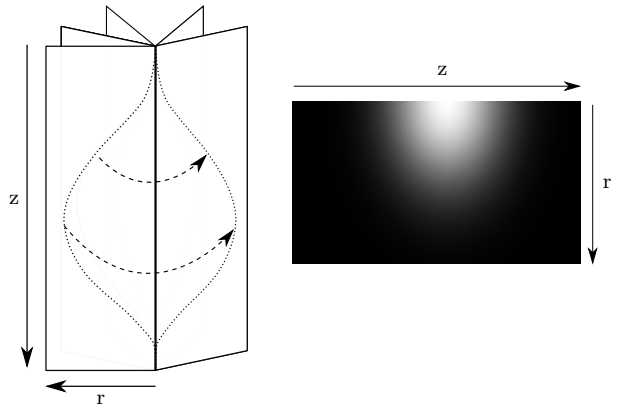


Figure C.1: Left: Illustration of the PSF measurement. We assume rotational symmetry around the optical axis. Right: an example of a high-resolution PSF map measured from images of fluorescent 100nm beads acquired with a confocal microscope.

Index

Bibliography

D. Adalsteinsson. *A fast level set method for propagating interfaces*. PhD thesis, University of California, 1994.

L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990. ISSN 1097-0312. doi: 10.1002/cpa.3160430805.

C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, December 2008.

I. Ayed and A. Mitiche. A region merging prior for variational level set image segmentation. *IEEE Trans. Image Process.*, 17(12):2301–2311, 2008.

Y. Bai, G. Roberts, and J. Rosenthal. On the containment condition for adaptive Markov chain Monte Carlo algorithms. *Open access*, 2009.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24 (4):509–522, 2002.

G. Bertrand, J. Everat, and M. Couprie. Image segmentation through operators based on topology. *Journal of Electronic Imaging*, 6:395–405, October 1997. doi: 10.1117/1.482584.

G. Bertrand. Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern Recognition Letters*, 15(10):1003 – 1011, 1994. ISSN 0167-8655. doi: DOI:10.1016/0167-8655(94)90032-9.

Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, volume 1, pages 26–33, Los Alamitos, CA, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.2003.1238310.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1222–1239, 2001. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/34.969114.

P. Bremaud. *Markov chains : Gibbs fields, Monte Carlo simulation and queues*. Springer, 1999. ISBN: 0-387-98509-3.

K. Bringmann and K. Panagiotou. Efficient sampling methods for discrete distributions. In A. Czumaj, K. Mehlhorn, A. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming*, volume 7391 of *Lecture Notes in Computer Science*, pages 133–144. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-31593-0. doi: 10.1007/978-3-642-31594-7_12.

S. P. Brooks. Markov chain Monte Carlo method and its application. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 47 (1):pp. 69–100, 1998. ISSN 00390526.

E. S. Brown, T. F. Chan, and X. Bresson. Completely convex formulation of the Chan-Vese image segmentation model. *Int. J. Comput. Vis.*, 2011. doi: 10.1007/s11263-011-0499-y.

T. Brox and D. Cremers. On the statistical interpretation of the piecewise smooth Mumford-Shah functional. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 203–213. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-72823-8_18.

T. Brox and J. Weickert. Level set based image segmentation with multiple regions. In *Pattern Recognition*, volume 3175 of *Lecture Notes in*

Computer Science, pages 415–423. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-28649-3_51.

F. Calderero and F. Marques. General region merging approaches based on information theory statistical measures. In *Image Processing (ICIP), 2008 15th IEEE International Conference on*, pages 3016 –3019, October 2008. doi: 10.1109/ICIP.2008.4712430.

O. Cappé, S. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899 –924, may 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2007.893250.

J. Cardinale, G. Paul, and I. F. Sbalzarini. Discrete region competition for unknown numbers of connected regions. *IEEE Trans. Image Process.*, 21(8):3531–3545, 2012.

V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vis.*, 22(1):61–79, 1997.

T. Chan and S. Esedoglu. Aspects of total variation regularized l 1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.

T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266 –277, feb 2001. ISSN 1057-7149. doi: 10.1109/83.902291.

J. Chang and J. Fisher. Efficient topology-controlled sampling of implicit shapes. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 493–496, 2012. doi: 10.1109/ICIP.2012.6466904.

J. Chang and J. W. Fisher III. Efficient MCMC sampling with implicit shape representations. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2081–2088. IEEE, IEEE, 2011.

G. Charpiat, R. Keriven, J.-P. Pons, , and F. O. Designing spatially coherent minimizing flows for variational problems based on active contours. In *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, volume 2, pages 1403–1408. IEEE, 2005.

S. Chen and R. Radke. Markov chain Monte Carlo shape sampling using level sets. In *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, pages 296 –303, 27 2009-oct. 4 2009. doi: 10.1109/ICCVW.2009.5457687.

S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995. doi: 10.1080/00031305.1995.10476177.

L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211 – 218, 1991. ISSN 1049-9660. doi: 10.1016/1049-9660(91)90028-N.

D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional. *Int. J. Comput. Vis.*, 50(3):295–313, 2002.

G. Danuser. Computer vision in cell biology. *Cell*, 147(5):973–978, Nov 2011. doi: 10.1016/j.cell.2011.11.001.

A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 0:2173–2180, 2010. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2010.5539897.

A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *Int. J. Comput. Vis.*, 2011.

N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia. Richardson-lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microscopy Research and Technique*, 69(4):260–266, 2006. ISSN 1097-0029. doi: 10.1002/jemt.20294.

J. Dieudonné. Dimension theory and separation theorems. In *A History of Algebraic and Differential Topology, 1900 - 1960*, Modern Birkhäuser Classics, pages 182–196. Birkhäuser Boston, 2009. ISBN 978-0-8176-4906-7. doi: 10.1007/978-0-8176-4907-4_8.

E. Ebrahim. Energy-based image segmentation using GPGPU. Master thesis, Technische Universität München & MOSAIC Group, ETH Zurich, 2011.

J. D. Eldredge, A. Leonard, and T. Colonius. A general deterministic treatment of derivatives in particle methods. *J. Comput. Phys.*, 180: 686–709, 2002.

A. C. Fan, J. W. Fisher, III., W. M. Wells, III., J. J. Levitt, and A. S. Willsky. MCMC curve sampling for image segmentation. In *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention*, MICCAI'07, pages 477–485, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-75758-9, 978-3-540-75758-0.

X. Fan, P.-L. Bazin, and J. Prince. A multi-compartment segmentation framework with homeomorphic level sets. *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.

C. Geyer. Markov chain Monte Carlo maximum likelihood. In *Computer Science and Statistics, 23th Symposium of the Interface*, pages 156–163, April 1991.

D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

G. R. Grimmet. A theorem about random fields. *Bulletin of the London Mathematical Society*, pages pp.81–84, 1973.

H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Unpublished manuscript, 1968.

X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:755–768, 2003. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2003.1201824.

N. Hansen and A. Ostermeier. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In *Proceedings of the 1996 IEEE Conference on Evolutionary Computation (ICEC '96)*, pages 312–317, 1996.

P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, 2006.

W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

L. He and S. Osher. Solving the Chan-Vese model by a multiphase level set algorithm based on the topological derivative. In *Proceedings of the 1st international conference on Scale space and variational methods in computer vision*, SSVM'07, pages 777–788, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-72822-1.

J. A. Helmuth, C. J. Burckhardt, U. F. Greber, and I. F. Sbalzarini. Shape reconstruction of subcellular structures from live cell fluorescence microscopy images. *J. Struct. Biol.*, 167:1–10, 2009.

J. A. Helmuth. *Computational Methods for Analyzing and Simulating Intracellular Transport Processes*. PhD thesis, Diss. ETH No. 19190, ETH Zürich, 2010.

J. A. Helmuth and I. F. Sbalzarini. Deconvolving active contours for fluorescence microscopy images. In *Proc. Intl. Symp. Visual Computing (ISVC)*, volume 5875 of *Lecture Notes in Computer Science*, pages 544–553, Las Vegas, USA, November 2009. Springer.

A. Henderson. *The ParaView guide : A Parallel visualization application*. Kitware, 2007.

R. W. Hockney and J. W. Eastwood. *Computer Simulation using Particles*. Institute of Physics Publishing, 1988.

J. Hynecek and T. Nishiwaki. Excess noise and other important characteristics of low light level imaging using charge multiplying ccds. *IEEE Trans. Electron Devices*, 50(1):239–245, 2003. ISSN 0018-9383. doi: 10.1109/TED.2002.806962.

L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide.* Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, 2nd edition, 2005.

H. N. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *Int. J. Comput. Vis.*, 2011.

C. Jordan. *Cours d'analyse de l'école polytechnique: Calcul intégral. Équations différentielles*, volume 3. Gauthier-Villars, 1887.

T. Kadir and M. Brady. Unsupervised non-parametric region segmentation using level sets. In *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, volume 2, pages 1267–1274, 2003.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vis.*, pages 321–331, 1988.

W. Kim and Y. Kim. A region-based shape descriptor using Zernike moments. *Signal Processing: Image Communication*, 16(1):95–102, 2000.

P. Koumoutsakos. Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.*, 37:457–487, 2005.

J. Kybic and J. Kratky. Discrete curvature calculation for fast level set segmentation. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3017 –3020, November 2009. doi: 10.1109/ICIP.2009.5414453.

J. Lamy. Integrating digital topology in image-processing libraries. *Computer Methods and Programs in Biomedicine*, 85(1):51 – 58, 2007. ISSN 0169-2607. doi: DOI:10.1016/j.cmpb.2006.08.006.

S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Trans. Image Process.*, 17(11):2029 –2039, nov. 2008. ISSN 1057-7149. doi: 10.1109/TIP.2008.2004611.

I. Larrabide, R. Feijóo, A. Novotny, and E. Taroco. Topological derivative: A tool for image processing. *Computers and Structures*, 86(13-14):1386–1403, 2008.

C. Li, C.-Y. Kao, J. Gore, and D. Z. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.*, 17(10): 1940 –1949, oct. 2008. ISSN 1057-7149. doi: 10.1109/TIP.2008.2002304.

J. Lie, M. Lysaker, and X.-C. Tai. A variant of the level set method and applications to image segmentation. *Mathematics of Computation*, 75: 1155–1174, 2006.

T. Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vis.*, 30(2):79–116, 1998.

J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):pp. 121–134, 2000. ISSN 01621459.

M. Liu, R. K. Yadav, A. Roy-Chowdhury, and G. V. Reddy. Automated tracking of stem cell lineages of Arabidopsis shoot apex using local graph matching. *The Plant Journal*, 62(1):135–147, 2010. ISSN 1365-313X. doi: 10.1111/j.1365-313X.2009.04117.x.

L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79, 1974. doi: 10.1086/111605.

J. Malcolm, Y. Rathi, A. Yezzi, and A. Tannenbaum. Fast approximate surface evolution in arbitrary dimension. In *Proc. SPIE 6914*, 2008.

A.-R. Mansouri, A. Mitchie, and C. Vazquez. Multiregion competition: A level set extension of region competition to multiple region image partitioning. *Computer Vision and Image Understanding*, 101(3):137 – 150, 2006. ISSN 1077-3142. doi: DOI:10.1016/j.cviu.2005.07.008.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, pages 416–423, Vancouver, BC, Canada, 2001.

S. Maskell. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. Technical report, Cambridge University Engineering Department, Cambridge, UK, 2001.

M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, January 1998. ISSN 1049-3301. doi: 10.1145/272991.272995.

K. L. Mengersen and R. L. Tweedie. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1):pp. 101–121, 1996. ISSN 00905364.

B. Mertzios and K. Tsirikolias. Fast shape discrimination using one-dimensional moments. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 2473–2475. IEEE, 1991.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087, 1953.

J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Comput.*, 19(14):1023 – 1040, 2001. ISSN 0262-8856. doi: 10.1016/S0262-8856(01)00064-6.

D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.

J. W. Neuberger. *Sobolev Gradients and Differential Equations*. Number 1670 in Lecture Notes in Mathematics. Springer, Heidelberg, Germany, 1997.

M. Nielsen, L. Florack, and R. Deriche. Regularization, scale-space, and edge detection filters. *Journal of Mathematical Imaging and Vision*, 7: 291–307, 1997. ISSN 0924-9907. doi: 10.1023/A:1008282127190.

R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/571647.571648.

S. Osher and J. A. Sethian. Fronts propagating with curvature- dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

G. Paul, J. Cardinale, and I. F. Sbalzarini. Coupling image restoration and segmentation: A generalized linear model/Bregman perspective. *Int. J. Comput. Vis.*, 2013.

T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *Proc. IEEE Intl. Conf. Computer Vision (ICCV)*, pages 1133–1140, October 2009. doi: 10.1109/ICCV.2009.5459348.

A. Rauch. *Microtubule Dynamics in S. cerevisiae*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 20622, 2012, 2012.

W. H. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62(1):55–59, Jan 1972. doi: 10.1364/JOSA.62.000055.

G. Roberts and J. Rosenthal. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of applied probability*, 44(2):458–475, 2007.

J.-L. Rose, C. Revol-Muller, D. Charpigny, and C. Odet. Shape prior criterion based on Tchebichef moments in variational region growing. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1081 –1084, November 2009. doi: 10.1109/ICIP.2009.5413477.

B. Sandberg, S. H. Kang, and T. Chan. Unsupervised multiphase segmentation: A phase balancing model. *IEEE Trans. Image Process.*, 19(1): 119 –130, jan. 2010. ISSN 1057-7149. doi: 10.1109/TIP.2009.2032310.

P. Sarder and A. Nehorai. Deconvolution methods for 3-d fluorescence microscopy images. *Signal Processing Magazine, IEEE*, 23(3):32 – 45, may 2006. ISSN 1053-5888. doi: 10.1109/MSP.2006.1628876.

I. F. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.*, 151(2): 182–195, 2005.

I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos. PPM – a highly efficient parallel particle-mesh library for the simulation of continuum systems. *J. Comput. Phys.*, 215 (2):566–588, 2006.

C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods*, 9(7):671–675, 2012.

B. Schrader, S. Reboux, and I. F. Sbalzarini. Discretization correction of general integral PSE operators in particle methods. *J. Comput. Phys.*, 229:4159–4182, 2010.

F. Ségonne. *Segmentation of Medical Images under Topological Constraints*. PhD thesis, Massachusetts Institute of Technology (MIT), December 2005.

Y. Shi and W. C. Karl. A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Trans. Image Process.*, 17(5):645–656, 2008.

J. Sibarita. Deconvolution microscopy. *Adv. Biochem. Eng. Biot.*, 95: 201–243, 2005. doi: DOI10.1007/b102215.

I. Smal, K. Draegestein, N. Galjart, W. Niessen, and E. Meijering. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis. *IEEE Trans. Med. Imag.*, 27(6):789–804, June 2008. doi: 10.1109/TMI.2008.916964.

A. F. M. Smith and G. O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(1):pp. 3–23, 1993. ISSN 00359246.

B. Song and T. Chan. A fast algorithm for level set based optimization. *CAM-UCLA*, 02-68, 2002.

J. Stone, D. Gohara, and G. Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(3):66, 2010.

G. Sundaramoorthi, A. Yezzi, and A. Mennucci. Sobolev active contours. In *Variational, Geometric, and Level Set Methods in Computer Vision*, number 3752 in Lecture Notes in Computer Science, pages 109–120, Heidelberg, Germany, 2005. Springer.

G. Sundaramoorthi, A. Yezzi, and A. C. Mennucci. Sobolev active contours. *Int. J. Comput. Vis.*, 73(3):345–366, 2007.

Z. Tu and S. Zhu. Image segmentation by data-driven Markov chain Monte Carlo. *IEEE T. Pattern. Anal.*, 24(Ieee Transactions On Pattern Analysis and Machine Intelligence):657–673, 2002.

G. M. P. van Kempen, L. J. Van Vliet, P. J. Verveer, and H. T. M. Van Der Voort. A quantitative comparison of image restoration methods for confocal microscopy. *Journal of Microscopy*, 185(3):354–365, 1997. ISSN 1365-2818. doi: 10.1046/j.1365-2818.1997.d01-629.x.

R. Veltkamp and M. Hagedoorn. 4. state of the art in shape matching. *Principles of visual information retrieval*, page 87, 2001.

L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis.*, 50:271–293, 2002. ISSN 0920-5691. 10.1023/A:1020874308076.

C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.

C. Vonesch and M. Unser. A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution. *IEEE Trans. Image Process.*, 17(4):539 –549, april 2008. ISSN 1057-7149. doi: 10.1109/TIP.2008.917103.

C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Process.*, 7(3):359 –369, mar 1998. ISSN 1057-7149. doi: 10.1109/83.661186.

C. Xu, J. Yezzi, A., and J. Prince. On the relationship between parametric and geometric active contours. In *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on*, volume 1, pages 483 –489 vol.1, 29 2000-nov. 1 2000. doi: 10.1109/ACSSC.2000.911003.

L. Yu, Q. Wang, L. Wu, and J. Xie. Mumford-Shah model with fast algorithm on lattice. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, page II, May 2006. doi: 10.1109/ICASSP.2006.1660434.

J. Zartman, S. Restrepo, and K. Basler. A high-throughput template for optimizing Drosophila organ culture with response surface methods. under revision in Development, 2012.

D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.

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, September 1996.

Publications

Refereed publications during PhD studies:

- Grégory Paul, **Janick Cardinale**, Ivo F. Sbalzarini, *Coupling image restoration and segmentation - A generalized linear model/Bregman perspective*, 2013, Int. J. Comp. Vis., accepted.
- **Janick Cardinale**, Grégory Paul, Ivo F. Sbalzarini, *Discrete region competition for unknown numbers of connected regions*, IEEE Trans. Image Process., 21(8), 2012, 3531-3545
- Grégory Paul, **Janick Cardinale**, Ivo F. Sbalzarini, *An alternating split Bregman algorithm for multi-region segmentation*, 2011, Proc. 45th IEEE Asilomar Conf. Signals, Systems, and Computers, 426-430
- **Janick Cardinale**, Alexander Rauch, Yves Barral, Gabor Székely, Ivo F. Sbalzarini, *Bayesian image analysis with on-line confidence estimates and its application to microtubule tracking*, 2009, Proc. IEEE Intl. Symposium Biomedical Imaging (ISBI), 1091-1094

Curriculum Vitae

Name:	Janick Oliver Cardinale
Born:	April 21th, 1983
Citizen of:	Berne, Switzerland

May 2002	Matura with focus on physics and applications to math
2002 - 2004	Foundational studies in Computer Science at EPFL, Lausanne, Switzerland
2004 - 2008	M.Sc. ETH in Computer Science with major in Computational Science and minor in Neuroinformatics, ETHZ, Zurich, Switzerland
2008 - 2013	PhD studies under the supervision of Prof. Dr. Ivo F. Sbalzarini at the MOSAIC Group, Institute of Theoretical Computer Science, ETH Zurich