Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Jan Hubený, September 21, 2007
Abstract

With the recent development of high-resolution microscopes and fluorescent probes together with tremendous development of information technologies, the cell biology has entered a new era. In these days, the biologists are able to observe sub-cellular components like chromosomes or even individual genes and proteins, the observations can be made in two or three dimensions and even dynamic processes in the living cell can be investigated. However, the research in this field depends more and more on imaging and computer vision methods and computer vision specialists, because such detailed observations and investigations produce huge image data sets which cannot be analyzed by hand.

A large number of computer vision methods have been developed in past two decades to support research in this field. However, the majority of successful methods is based on rather elementary image processing techniques. The image processing and computer vision have gone through a similar tremendous development as the cell biology in past two decades. The specialists in this field have developed a large amount of sophisticated mathematically well-founded image processing methods. The PDE-based (Partial Differential Equation) image processing techniques belong to such methods. However, these new methods are still not very often used in biological imaging, either for their complexity or for their computational time demands.

In this thesis, we present several applications of PDE-based image processing methods in fluorescence microscopy. We first provide a brief background on the fluorescence microscopy and its usage in the cell biology, then we review a selection of PDE-based image processing methods. As an own contribution, a segmentation algorithm based on two recent approximations of Chan-Vese active contour model is presented and tested. Further, two methods for the segmentation of interphase chromosome territories are designed, presented and tested. Finally, we present the application of state-of-the-art variational optic flow methods for motion estimation in time-lapse sequences from live-cell studies.
Acknowledgements

First of all, I would like to thank Michal Kozubek for being my supervisor and giving me the opportunity to work in his group. Second, I want to thank all colleagues from Centre for Biomedical Image Analysis for their comprehensive help and ideas. Further I would like to thank Pavel Matula who shows me the world of level set methods and PDE based image processing. David Svoboda, Vladimír Ulman, Petr Matula, Marek Kašík for fruitful discussions, helpful comments and for being good colleagues and friends. Stanislav Stejskal, Irena Koutrná, Vladan Ondřej for image data acquired on confocal microscope. Andrés Bruhn for his willingness to explain every detail of his multigrid implementation of variational optical flow methods.

Finally, I would like to thank my parents Hana Hubená and Stanislav Hubený for their support during my childhood and my wife Magdalena Hubená for her love, understanding and support in whatever I do.

The research presented in this thesis was partly supported by the Ministry of Education of the Czech Republic (Grant Numbers MSM-143300002, MSM-0021622419, LC-535) and by Grant Agency of the Czech Republic (Grant Number GD102/05/H050). Finally, I would like to thank METACentrum (http://meta.cesnet.cz/) for the computational time which I have used on their computers.
# Contents

1 Introduction ......................................................... 1

2 Fluorescence microscopy ........................................... 3
   2.1 Specimen preparation ...................................... 3
   2.2 Fluorescence microscopes .................................. 6
   2.3 Image data .................................................. 8
   2.4 Typical problems to solve .................................. 12
      2.4.1 Image preprocessing ................................. 12
      2.4.2 Image segmentation ............................... 13
      2.4.3 Image registration and motion estimation ....... 15

3 PDE-based image processing ....................................... 17
   3.1 Diffusion filtering ........................................ 18
      3.1.1 Physical motivation .................................. 18
      3.1.2 Gaussian diffusion filter ........................... 18
      3.1.3 Nonlinear diffusion filters ......................... 19
      3.1.4 Anisotropic nonlinear diffusion filters .......... 24
      3.1.5 Nonlinear diffusion filters with unbounded diffusivity ... 24
      3.1.6 Numerical approximations .......................... 26
   3.2 Implicit active contour models ............................ 30
      3.2.1 Level set framework ................................. 30
      3.2.2 Implicit active contour models .................... 32
      3.2.3 Fast marching method .............................. 39
      3.2.4 Region based active contours ....................... 41
   3.3 Variational optic flow ..................................... 44
      3.3.1 Methods with image-driven regularization ....... 45
      3.3.2 Methods with flow-driven regularization ......... 48
      3.3.3 Methods with robust data term ..................... 48
      3.3.4 Numerical approximation ............................ 51
   3.4 PDE in fluorescence microscopy ........................... 52

4 Fast and robust approximation of Chan-Vese model .............. 55
   4.1 Introduction ................................................ 55
   4.2 Fast approximations ....................................... 56
      4.2.1 Gibou-Fedkiw algorithm ............................ 57
      4.2.2 Esedoglu-Tsai algorithm ........................... 58
Chapter 1

Introduction

One of the biggest challenges for current cell biology is to fully understand the spatial organization of the genetic material in the cell nuclei, the structure of intracellular components and the complex mechanisms which occur in the cells. Results obtained in this research field have and probably will have an immense impact on the cure of such severe diseases like cancer or leukemia. The research in this field is done mostly with the help of fluorescence microscopy in these days. The fluorescence microscopy enables to visualize the transparent biological material which is labeled with the fluorescent dyes. There exist very specific fluorescent probes in these days so that the biologists are able to observe sub-cellular components like chromosomes or even individual genes and proteins with fluorescence microscopes. Further, the observations can be made in two or three dimensions and even dynamic processes in the living cell are commonly investigated. However, such detailed observations and investigations lead to huge image data sets which cannot be analyzed by hand. Therefore there is a natural demand on reliable image processing and computer vision methods.

A large number of computer vision methods have been developed for the analysis of image data from fluorescence microscopy. However, the majority of successful methods is based on rather elementary image processing techniques. This was partially caused by the fact that the sophisticated methods were not ready to be use at that time or it was impossible to use those methods on the former hardware. Nevertheless, the specialists in image processing and computer vision have developed a large amount of sophisticated mathematically well-founded image processing methods in past two decades. The PDE-based (Partial Differential Equation) image processing techniques belong to such methods. However, these methods are still not very often used for the analysis of image data from fluorescence microscopy. We suppose that it is due to their complexity and their computational time demands.

This thesis describes several applications of PDE-based image processing methods for analysis of biomedical image data which have its origin in fluorescence microscopy. We do not deal with one selected problem in this thesis. On the contrary, we describe our contribution to the solution of several practical problems which belong to typical tasks in this field. Namely, we will present a segmentation algorithm based on two recent approximations of Chan-Vese active contour model suitable for two-phase segmentation of low contrast and noisy biomedical data. We describe two methods for the segmentation of interphase chromosome territories. Finally, we present the application of state-of-the-art variational optic flow methods for motion estimation in time-lapse sequences from live-cell studies. We hope that the thesis will show to other researches that the PDE-based processing methods can be successfully used even for the
Introduction

analysis of large three dimensional data when using fast numerical methods and approximation algorithms.

The rest of the thesis is organized as follows: In the first chapter we describe the fundamental concepts and essential principles of fluorescence microscopy. We focus our attention on its usage in cell biology. Therefore, several specimen preparation techniques and microscope devices will be described. Further, we describe the typical image data which can be obtained in fluorescence microscopy and its properties. Finally, we describe the typical problems and tasks which the image processing and computer vision experts usually solve in this field. Furthermore, we make a summary of methods and algorithms which are usually used for image analysis of biomedical data obtained from fluorescence microscopy.

We provide a brief survey of state-of-the art of the PDE-based image processing in the second chapter. We describe this field from a view of practician and only the topics which will be used in the rest of the thesis will be discussed. Therefore, we describe the fundamentals of nonlinear diffusion filtering first. Then we discuss several active contour models for image segmentation. Further, we provide the overview of basic variational optic flow concepts in this chapter. Finally, we summarize previous successful application of the PDE-based image processing techniques in fluorescence microscopy.

Our first own contribution will be presented in the third chapter. We present fast and robust algorithm for minimization of Chan-Vese energy functional in this chapter. Proposed technique is based on two recently published approximations of this functional. We have realized that both those algorithms have their limitations and specific problems when processing low contrast biomedical data. Therefore we combine both approximation algorithms in order to preserve their individual advantages and avoid their limitations. We show that the proposed hybrid algorithm is suitable for the segmentation of image data from fluorescence microscopy.

The main aim of the fourth chapter is to propose two methods for the segmentation and reconstruction of chromosome territories in interphase cell nuclei. The proposed methods will use several PDE-based image processing techniques. The methods will be compared to each other and tested on both synthetic and real image data sets. Further, we describe a novel synthetic image generator in this chapter. However, our own contribution to this generator was only minor.

We describe a framework suitable for motion tracking in time-lapse sequences from live-cell studies in the fifth chapter. The framework is based on the state-of-the-art optic flow methods. We propose a pseudo-real image sequence generator which provides us with the artificial time-lapse sequences together with ground-truth displacement fields. We test the performance of several variational methods on the ground truth sequences. Finally, a simple motion tracking framework is described and tested on 2D and 3D, artificial as well as real time-lapse image sequences from live-cell studies.
Chapter 2

Fluorescence microscopy

We describe the fundamental concepts and essential principles of fluorescence microscopy in this chapter. Fluorescence microscopy is used for the observation of fluorescently labeled objects. It can be used in various fields, but we focus our attention on its usage in cell biology. We provide only elementary information which is necessary to understanding the rest of this thesis.

We explain the principle of fluorescence and several specimen preparation techniques first. Principles of fluorescence microscopes and several specialized confocal microscope devices, which serve for image data acquisition, will be described later. Further, we describe the typical input data and its properties. Finally, we discuss the common issues of image analysis in this field. Furthermore, we make a summary of methods and algorithms which are usually used for image analysis in fluorescence microscopy.

2.1 Specimen preparation

The transparent biological material has to be visualized in a certain way before we can observe it using any optical microscope. The cells or the intracellular components in specimens are visualized indirectly with fluorescent dyes in fluorescence microscopy. We describe most frequently used approaches for the visualization of specific objects in this section.

The fluorescence phenomenon is the fundamental element of each staining technique in fluorescence microscopy. Fluorescent dye (called fluorochrome) exposed to the light of certain wavelength range emits light of different wavelength range. The wavelength range of the emitted light is always longer because small amount of excitation light is transformed to heat (this phenomenon is described by the so-called stokes shift). Moreover, the difference between absorption and emission spectrum is large enough to distinguish the excitation light from the emitted light. This property is essential for the construction of the fluorescence microscopes. An example of absorption and emission spectrum of Rhodamine fluorochrome is shown in Fig. 2.1.

FISH staining technique The most frequently used and the most popular staining method in fluorescence microscopy is the FISH (Fluorescence In Situ Hybridization) method. This technique enables to visualize specific DNA and RNA sequences. Therefore, it is widely used for visualization of cell nuclei, chromosomes, cytoplasm or organelles which contain those acids.
2.1. Specimen preparation

The DNA and RNA sequences are labeled with so called probes. Probe is a DNA or RNA sequence complementary to the target sequence. The nucleic acids can be labeled and visualized in two ways. Either directly with fluorochrome labeled probes or indirectly with the biotin or digoxigenin molecules and fluorescent labeled antibodies. The probes and corresponding target sequences must be single stranded so that they can be combined and annealed together. Hence, both the biological material fixed on the microscopic slide and the probes are denaturalized first. In the second hybridization step, the probes are dropped onto the microscopic slide with the biological material and they are let to combine and anneal with the target sequences. Finally, the remaining redundant probes which did not anneal with any target are washed out in the post-hybridization washing step. The whole FISH staining process is illustrated in the Fig. 2.2.

FISH technique can be used for visualization of objects which greatly vary in size. It is possible to visualize relatively big objects like whole chromosomes. On the other hand, one can stain even small objects like individual genes. Further, it is possible to visualize different targets on one microscopic slide. One can use probes with different fluorochrome labels which have different excitation and emission spectra in one FISH experiment. Usually up to four different fluorochrome labels, which correspond to four different types of observed objects, can be used simultaneously. It is also possible to repeat the whole hybridization process if it is necessary to visualize more than four different targets or use labels with overlapping excitation or emission spectra. This process is called repeated hybridization. Nevertheless, the results from the two FISH experiments which were sequentially performed on one microscopic slide have to be registered and merged. More details about FISH staining technique and the biological background can be found in [Koz01].

Immunofluorescence staining technique Immunoﬂuorescence is another successful technique which suitably complements FISH. Whereas we can visualize only the DNA or RNA target
Fluorescence microscopy

2.1. Specimen preparation

Figure 2.2: The principle of FISH (Fluorescence In Situ Hybridization). The biological material and its internal structure is fixed in some way (e.g. on the microscopic slide) first. The probes are labeled (either directly or indirectly). After that, both the probes and the biological material are unrolled to their single-strand forms via the denaturalization process. Finally, the probes are dropped onto the microscopic slide where they anneal to the target sequences. Redundant probes which did not anneal with any target are washed away.

Image source: National Human Genome Research Institute (http://www.genome.gov/)

sequences with the FISH method, we can label and visualize particular proteins with the immunofluorescence technique. The Immunofluorescence is based on antibodies which are proteins produced by the immune system. It serves as a "first line" detector of a potential infection. There exist a vast amount of different antibodies. Each of them recognizes a specific target molecule which is called antigen. Owing to this specificity, it is possible to mark particular proteins with fluorescently labeled antibodies. There are two ways of fluorescent labeling in immunofluorescence. One can mark particular targets with antibodies which contain the fluorochromes. But, far better is to mark the target with primary blank antibodies and consequently mark the primary antibodies with secondary antibodies which carry the fluorophore. While in the first case only one antibody with fluorochrome can connect to the specific target, several secondary antibodies can connect to the primary antibody (see Fig. 2.3). Naturally, stronger fluorescent signal is obtained as a result.

We must note that the indirect FISH works on the same principle as indirect immunofluorescence. These two labeling techniques can be also successfully combined, when e.g. some interaction between certain proteins and genetic material inside the cell is the subject of the research.

Live-cell staining techniques Previous two staining techniques cannot be used for fluorescent labeling of living cells. The first step of both of them is the fixation of the cell in its current state on the microscopic slide. The cell is put to death as the byproduct of this step. Moreover, classical fluorochromes are phototoxic to the living cells. This means that illumination of living
2.2. Fluorescence microscopes

Fluorescence microscopes are used for observation of the fluorescently labeled specimens. The main difference between fluorescence and conventional optical microscope is that the fluorescence microscopes have some extra components and features added. We describe three most used fluorescence microscope setups in this section.
2.2. Fluorescence microscopes

Wide-field microscope Conventional wide-field microscope was developed in the end of 16th century in its simple form. It uses the light to illuminate the specimen. The light goes either through (transmission mode) or reflects (reflection mode) from the specimen into the objective and the magnified image of the specimen can be observed in the eyepiece. A fluorescence microscope provides the fluorescence mode by adding a set of emission and excitation filters to the light path (See Fig. 2.4). High-luminance mercury lamp is usually used as a source. It produces light with wavelength band from ultraviolet to infrared part of the spectra. This light goes through the excitation filter which ensures that only light in a particular wavelength band passes through. Filtered light continues on its way, it is reflected on the dichroic mirror, it passes through the objective and excites the fluorochromes in the specimen. The fluorochromes emit light of a longer wavelength range which goes back to the objective and passes through the dichroic mirror. The emission filter ensures that only light emitted by the fluorochromes passes through and goes to the eyepiece or onto the CCD chip of the camera.

The main drawback of the fluorescence wide-field microscopes is that it is difficult to perform three dimensional observations. The two dimensional image observed in the eyepiece or detected on the CCD chip contains partly the image of the specimen in focal plane, partly the blurred images of the planes above and below the focal plane. In other words, each point of the sample contributes independently to the light distribution in the image. The contribution can described by so-called PSF point spread function. Therefore, the two dimensional point in the observed image is a integration of contributions of the points in the three dimensional specimen. This limitation can be overcome by restoration of the three dimensional acquired image using the technique called deconvolution or by using the confocal fluorescence microscope which suppresses the contributions from non-focal planes already during the image acquisition.

Confocal scanning microscope Confocal laser scanning microscope (CLSM) is very similar to the classical wide-field microscope. First substantial difference is that the specimen is scanned point by point, line by line, one plane after another. Thus, the specimen cannot be observed directly in the eyepiece or by camera. The image is formed in the computer memory instead and can be displayed only when all points of the specimen are scanned. As the light detector is used so-called photomultiplier tube (PMT). The second modification adds a small pinhole in front of the detector. This pinhole effectively filters out the light contributions from the out of focus points (See Fig. 2.5). These two modifications enhance the overall resolution of the device and enable to acquire non-blurred high resolution three dimensional data.

The main disadvantages or constraints of CLSM result from its technical design. The three dimensional image acquisition is a little bit slower due to the point by point forming of the resulting image. The second constraint is caused by the pinhole. On one hand, it effectively filters out the out of focus contributions, on the other hand it blocks the majority of the emitted light and causes low light efficiency of the whole setup. Therefore, high intensity and usually expensive light source (like laser) must be used.

Confocal spinning disk microscope There is a way how to reduce the time needed for scanning and at the same time acquire the high resolution three dimensional image data comparable to those acquired using the classical scanning confocal microscope. One has to scan many points (typically thousands) in parallel. This task is usually performed using the rotating
Figure 2.4: Principle of optical fluorescence microscope. The light goes through excitation filter which lets only a specific band go through. Filtered (blue) light is reflected by the dichroic mirror, goes through the objective and illuminates the specimen. The fluorochromes in the specimen are excited by the emission light and emit the light (green) of a longer wavelength. This light goes back through the objective and dichroic mirror. The emission filter ensures that only emitted light finishes its way in the eyepiece or camera, where the specimen is observed.

disc with many pinholes (so-called Nipkow disc). The pinholes on the disc are arranged in a way ensuring that the specimen is scanned completely during just one rotation of the disc. Moreover, the pinholes are small and positioned relatively far from each other and the confocal image is obtained using this setup.

The main advantage of the confocal spinning disk microscope is lower price of the device (no laser light source is needed). The disadvantage is its poor light efficiency. Therefore, the image acquisition of fluorescently labeled data requires longer exposure times. This constraint was overcome by adding second disc with microlenses which focus and pass through the light from the local neighborhood of the pinholes inside them (see Fig. 2.7 and Fig. 2.6). The light efficiency of the whole setup is significantly improved by this modification. On the other hand, the laser light source has to be used in such microscope. The confocal spinning disk microscopes with microlenses are often used in live-cell imaging.

2.3 Image data

The fluorescently labeled specimens can be observed with fluorescent microscopes. We have described the specimen preparation techniques, acquiring devices and their properties in previous two sections. We describe the data which can be obtained and discuss their typical properties in this section.

Dimensionality Typical images acquired with fluorescence microscopes are either two or three dimensional (2D or 3D). Two dimensional images are frequently acquired with conven-
Fluorescence microscopy

2.3. Image data

Figure 2.5: Schematic diagram of modern laser scanning confocal microscope (LSCM). The light from the laser is filtered with the excitation filter (omitted in the schema). Further, it is reflected by the dichroic mirror, passes through the objective lens and illuminates one point with small neighborhood in the specimen. The emitted light goes back through the objective and dichroic mirror. The small pinhole in front of the detector filters out the emitted light from the out of focus planes (in red color). Only the contribution from the focal plane (green color) is filtered with the emission filter (omitted) and is detected by the detector (photo multiplier tube).

Traditional wide-field fluorescence microscopes (without confocal effect) because they have better light throughput than confocal ones. The three dimensional images are rather acquired with the confocal devices (scanning or spinning disk scanning) because they produce less blurred three dimensional data. The two dimensional data are often obtained by the so-called autofocus (maximal) projection along selected axis from the three dimensional data in image preprocessing phase. The time series of 2D or 3D images are acquired in live cell imaging. Some authors call this type of data four dimensional (4D in case of 3D image time series). However, the sampling in the time axis is more sparse than in the space, typically one gets only a few dozens of time samples. Therefore, more precise is to speak about 2D or 3D plus time [Rie04]. We have showed that fluorochromes emit the light of different wavelength ranges and it is possible to observe several targets labeled with different fluorochromes. Therefore the wavelength of the emitted light (color) is considered as another dimension in some publications.

Color Several different targets are labeled in a typical experiment in fluorescence microscopy. Only one wave band belonging to a particular fluorochrome is scanned by the microscope at a moment. One obtains several gray-scale images, each corresponding to particular target type, as a result of acquisition process. These images are often saved as channels of color image in
Figure 2.6: Schematic diagram of confocal spinning disk microscope with microlenses. The light from the laser beam is concentrated by the microlenses to the pinholes (many in parallel). It excites particular points of the specimen and the emitted light goes back through the objective. Further, it is reflected by the dichroic mirror to the CCD chip. 

one file, where the color of the channel approximately represents the color of the emitted light. However, these images do not contain the color information in its common sense. They are only pseudo-color and nothing more than only a multi-channel set of several gray-scale images.

**Image size** The size of the image data depends on the acquisition parameters (2D or 3D), size of the investigated objects and on the subsequent processing method. The raw volumetric (3D) image has typically around $1300 \times 1000 \times 60$ voxels. This rather big raw data are only rarely processed. Only the interesting regions are often cropped out from this big raw data before consecutive processing. The size of those sub-images depends on the size of the observed object. One can say that their size varies between $100 \times 100 \times z$ and $400 \times z$, where $z$ is the number of $z$-slices in case of 3D image data. Another important fact is that the number of voxels in $x$ and $y$ direction is a little bit larger than in $z$ direction in the three dimensional images.

**Bit depth** Modern CCD cameras and PMT detectors allow to acquire the image data even in 16-bit precision. However, only some specialized image processing algorithms (e.g. deconvolution) need to process images with such high bit depth to produce correct results. Only eight or twelve most significant bits of each image pixel are saved in common applications.

**Resolution** The word “resolution” is used in many different meanings in the context of fluorescence microscopy [Koz01]. One can speak about camera resolution, optical resolution or image resolution. The image resolution is most important for our purposes. It is determined by number of micrometers or nanometers per image pixel or voxel. The image resolution is the
Fluorescence microscopy

2.3. Image data

Figure 2.7: Effect of adding microlens disk to the standard spinning disk scanning microscope. (a) Left: Only a small fraction of the excitation light passes through the pinhole disc in the standard spinning disk microscope. (b) Right: The microlenses focus and concentrate the light into the small pinhole. Thus, greater portion of light can reach the specimen.

Properties of input data The quality of acquired image data depends on the setup of optical (microscope) and electronic (CCD camera, PMT detector) instruments, on the parameters of the acquisition process, on the used labeling technique and on the procedure of specimen preparation.

The first important property of image data acquired with fluorescence microscopes is that they are low contrast. This is caused by following facts. The important property of fluorescent dyes is that the intensity of their fluorescence is fading proportional to the light intensity of the light source and the exposure time. This effect is called photobleaching. One must also take into account the phototoxicity phenomenon in live cell imaging. The trend is to expose specimen to the light as little as possible in order to avoid those effects. Second important fact is that the confocal unit filters out a portion of emitted light too. These constraints are in the contradiction to the needs of the detector, which produce better data with longer exposure time. Therefore, a reasonable compromise between exposure time and light intensity is usually found for each particular application and the image data are often low contrast due to this compromise.

The noise is phenomenon which occurs in every real application imaging technique. The image data in fluorescence microscopy are often corrupted by several types of noise [Koz01] – readout, dark-charge and photon-shot noise. The dominant noise is the photon-shot noise which is caused by statistical fluctuations of the acquired signal level. This noise obeys the Poisson distribution. The signal level can be increased either by increasing the fluorescence light intensity or by increasing the exposure time. However, these two parameters are limited by the above mentioned constraints.

The size of examined object in the fluorescence microscopy is often comparable to the optical resolution of the microscope. Therefore the optical setup has a great influence on errors like chromatic aberrations, chromatic shifts and intensity variation [MP00], which should be taken into account when measuring small objects like genes.

Another common property of the images acquired using some fluorescence microscope is that they are more or less blurred. As we have shortly remarked, the conventional wide field microscopy produces blurred three dimensional images, because the intensity of each pixel is influenced by intensity of the neighboring pixels (most significantly along the z axis).
confocal microscopy does not suffer so much by the blurring effect, but the blurring is still present in the acquired image data. The amount of blurring depends on the optical setup of the microscope, but it can be never completely suppressed due to the laws of optics. The process, which tries to find deblurred image from the blurred input is called deconvolution [Ver98].

The preparation process of the specimen has great influence on the quality of acquired data. We described its principles in section 2.1 and one can suppose that they should work without any problems. But in the real world, some parts of target objects can be worse reachable for the fluorescent markers than others. Hence, target object labeling can lead to inhomogeneities in the object intensities. Further, it is not easy to wash out completely the unattached fluorescent markers from the specimen. Small amount of fluorescent dyes often remains at wrong places due to incomplete washing and produce false signal (usually weak, but visible). Finally, the foreign particles like dust can often get into the specimen and degrade the acquired image.

Several examples with image data acquired with various fluorescence microscope are included in the Appendix A.

2.4 Typical problems to solve

The acquired image data have to be further processed before reaching any reasonable or interesting conclusions. While the specimen preparation and image acquisition is usually made by the biologist researcher, the image analysis is a task for computer scientists. We discuss the common and specific task of image analysis in this section. We focus on the description of common techniques used by the imaging community in the field of fluorescence microscopy.

Fluorescence microscopy is an interdisciplinary research field where (among others) the biologist and computer scientist meet. We must specify the interests and goals of the biologists first in order to identify the main task of image analysis. The goal of current cell biology is to understand and describe the spatial organization of genetic and other material in the cell nucleus and in the cell. The specialists in this field want to find and understand the mechanisms, events and interactions which occur in the cell. Among the typical tasks which should be solved is the localization of particular objects, the analysis of their position within the cell (e.g. are the objects mostly near the border or in the center of the cell or cell nucleus) and the analysis of their interactions (e.g. is particular gene located close to some other gene). The objects of interest have to be founded in order to fulfill these assignments. Therefore, the most important task in this field is the segmentation. Several related problems should be solved as well, therefore we will discuss common image preprocessing methods, segmentation methods and task related to motion in this section.

2.4.1 Image preprocessing

When we take into account all the properties of image data acquired using fluorescence microscope, which were summarized in previous section, we have to admit, that the image preprocessing is the essential step of any further image analysis. We will now describe some common techniques used by the imaging community in the field of fluorescence microscopy.

Deconvolution The deblurred original image can be theoretically completely restored if we know the point spread function and no noise is present. However, the point spread function
2.4. Typical problems to solve

Fluorescence microscopy is dependent on the spatial position in the image, it even differs for various light wavelengths. The quality of deconvolution also depends on the noise which is present in the image. If the image is damaged by significant amount of noise, it can be difficult to deconvolve it. All these factors imply, that the best deconvolution algorithms are little bit tricky and they need a large amount of computational time for producing deconvolved image [SN06]. Therefore, the deconvolution is unfortunately not often used in the field of fluorescence microscopy as it probably should be.

Noise reduction Noise is naturally present in the image data acquired using fluorescence microscopy as was stated above. Image denoising and enhancement is therefore an indispensable step towards achieving high quality results of any further analysis. Gaussian and median filtering [Pra91], two basic denoising methods are commonly used for noise elimination by the fluorescence imaging community. They are usually sufficient for the suppression of statistical background noise (e.g. photon-shot noise, readout noise) in image data. The Gaussian filter reduces the noise perfectly, but it smooths the significant information like edges together with the noise. The Median filter removes spikes and preserves the edges in a better way, but it still modifies each pixel of the image and that’s why it adds some fuzziness and blurring to the image. One can use some sophisticated modifications of median filtering like Adaptive Median filtering [HH95, EM01] in order to reduce the blurring effect of the standard median filter. Advanced denoising algorithms based on statistical frameworks variational formulations [CS05] and PDE-based image processing have been extensively explored in recent years by the image processing community. However, they are not commonly used in fluorescence microscopy. The practitioners are just starting to use them in this field.

2.4.2 Image segmentation

Segmentation is the fundamental, perhaps the most difficult step of any image analysis. It is the art of automatically separating the image into different regions in a form that simulates the human visual system. The field of image segmentation is already quite well explored due to its importance. Nevertheless, some methods for segmentation of image data acquired with fluorescence microscopes have still their limitations and this research area is still open for new and better methods. In this subsection, we first formulate the demands on appropriate segmentation method and then we give a short list (certainly incomplete) of current segmentation methods used in the field of fluorescence microscopy.

Required properties of segmentation method We must be able to find exact location and accurate boundaries of objects in the biomedical images in order to perform the measurements and make some conclusions from those measurements. Moreover, sufficiently large number of measurements (from hundreds to thousands) needs to be performed to obtain statistically significant results. The measurements should be reproducible, so the segmentation process should be reproducible too. These demands allow us to formulate set of properties, which a segmentation method should have to be applicable in processing of images acquired using fluorescence microscope.

Such segmentation method should be: correct and complete – it should find all the visible objects with no over-segmentation or under-segmentation; as automatically as possible – the interaction with human supervisor should be minimized, because two distinct people can
produce two distinct segmentations; reasonably fast – the number of measurements should be as large as possible due to the statistic reasons.

**Current segmentation methods** We will now discuss the available segmentation methods according to the object of interest. The objects to be segmented vary in size and type. There is a need to find 3D boundaries of isolated cells or nuclei (e.g. blood cells), chromosomes, point-like objects like individual genes and boundaries of cells in thick tissue section.

The segmentation of the isolated cells or nuclei is a relatively simple task. The methods based on local thresholding has been successfully used to locate the boundaries of objects \[NYvV+97\]. After thresholding the remaining small objects and the incomplete objects on the boundary can be excluded from segmentation results. The watershed based algorithm can be used for separation of slightly touching objects \[MSV+97\].

For chromosome territories separation methods based on thresholding and mathematical morphology \[Koz01\] were proposed. In order to find the chromosome territories the image data were thresholded in the first step. The optimal threshold was found by the histogram analysis. The morphological closing was used to join some unconnected spots together to one larger spot, which represents the chromosome territory, in the second phase. Another method based on Voronoi tessellation was proposed in \[EBS+95, EDB+96\]. This method can be classified as the split and merge methods. Initial voronoi tessellation is iteratively split. At each step the variance of gray values in each voronoi polygon (polyhedron in 3D) is computed. If it exceeds some preset threshold the polygon is divided into two new smaller polygons. Resulting voronoi tessellation is then thresholded. All voronoi polygons which have the medium gray value above some threshold belong to the chromosome territories. The optimal threshold is determined by a parametric study. Another method based on fast marching method was proposed in \[MHK04\]. It should be noted that first two algorithms for chromosome territory reconstruction produce quite bumpy surfaces, while the fast marching based algorithm produces more smoother objects. All mentioned methods do not solve the splitting of joined chromosome territories.

Method for segmenting dot like objects based on watershed algorithm was successfully used in \[KKL+99\]. This method locates the dot signal very accurately and in an efficient way.

The automatic segmentation of cells in tissues is a more difficult task than the previous three. The main problem of this issue is that all objects(cells or nuclei) in tissues are touching each other or they overlap. The intensities of the objects often vary, which leads to over-segmentation. Third, the image background intensity is often uneven due to the fluorescence of the out of focus objects. The utilization of only one relatively simple method for such a complex problem leads to very poor results. Therefore the majority of segmentation methods for such problem consist of two or three phases. In the first phase, the foreground is distinguished from background and some preprocessing for the second phase is done. Thresholding (or local thresholding) is used for foreground segmentation. The second phase usually needs the seeds localized inside the objects. For this issue the Euclidean distance transform of foreground is often used, the foreground can be processed by suitable morphological operator, or some modifications of Hough transform are used to find the seeds. The vast majority of successful tissue segmentation methods use the watershed algorithm in the second phase \[AC01b, AC01a\]. An interesting application of dual active contours for precise localization of cell boundaries was presented in \[Svo04\]. The localization of seeds in the first phase is sometimes little tricky and there is hardly ever achieved the one to one mapping between real objects and proposed
seeds. Therefore, the over-segmented objects are connected or conversely the under-segmented clusters separated in the third phase. Hardly any present tissue segmentation method is fully-automatic, human supervisor often controls and corrects the results either after the first or second phase. The full three-dimensional segmentation of the tissue is often also time consuming task.

2.4.3 Image registration and motion estimation

Another tasks, which are commonly solved in image analysis in fluorescence microscopy, are related to the motion which can in various ways occur during image acquisition. Image registration is frequently needed as a post-acquisition step, e.g. in repeated FISH or in repeated immunofluorescence staining experiments. The motion estimation and even motion tracking in live cell imaging is a field of intensive current research. Various methods including image registration and active contours are used and examined for solution of this task. Finally, there are attempts to produce reference atlases of cells of one kind (similar to brain atlases in biomedical imaging) and to register acquired images of cells onto the atlas images.

Repeated acquisition of fixed cells  The FISH technique enables repeated staining of the same cells (the same microscopic slide). This procedure is used when more than four probes should be observed or when the used fluorochromes overlap in their emission spectra. The particular fields of view with cells of interest are acquired after the first hybridization. After that, the old probes are washed away and the new ones are added and hybridized with target DNA. In the next step, the same fields of view as in the first step are acquired. Although the cells on the microscopic slide are fixed, their positions and even their shapes are not same after the first and the second acquisition. Their images have to be registered before any further analysis. The common registration methods known from biomedical imaging [ZF03] can be used for this task. The point-based rigid registration (shift transformations) of the images from individual hybridizations was performed according to object weight centers as reference points in [KKEL+01]. The voxel based rigid registration can be used as well. Wählby et al. [WEBZ02] registered image data from repeatedly immunofluorescently stained specimen with that technique.

Live-cell imaging  Substantial improvements in optics hardware together with shift to digital imaging in last decades prepared the way for live cell imaging studies. Nevertheless, the most important thing for the growing popularity of live-cell studies was the discovery of green fluorescent protein (GFP) and the discovery of its expression mechanism. It has enabled to encode fluorescence into a specific gene of interest and this in turn enables to tag and visualize any protein of interest in living cells. These developments combined with time-lapse imaging have allowed to perform studies of cellular and even intracellular dynamics. Two types of motion can occur in image data from live cell imaging. Typical time intervals between acquisitions of images of the same cell are in the order of minutes. The live cell can move as a whole unit in the meantime, that is it performs some global motion. Further, particular objects inside the cell can perform local movements or some intracellular flows can happen in the same time. In some applications, it is sufficient only to estimate and suppress the global motion of observed cells. The point based registration method for fast alignment of live cells in two consecutive frames was developed in [MMKD06]. The voxel based registration technique for
alignment of cell nuclei was proposed in [RMDvV04]. Some studies and experiments require to track the objects of interest in the whole image series. This issue is more difficult because the objects can appear and disappear within the series. Sophisticated method based on active contours was used for tracking of whole cells [ZZD+06]. Tracking of particular proteins on intracellular level was discussed in [MSD06]. Manders et al. has used block-matching (BM3D) algorithm [dLvL02] in their study of chromatin dynamics during the assembly of interphase nuclei [MVK+03].

Atlas based registration The building of maps of 3D structure of human genome in interphase cell nucleus is a challenging task for modern cell biology. The aim is to relate geometric information to the genome function in order to achieve a better understanding of how the chromatin structure affects and is related to the gene expression. Therefore, one of the essential tasks is to gather that information from large amount of 3D microscopy images. It is obvious that the particular nuclei differ and it is necessary to “normalize” those images. The normalization is achieved by image registration, but one obviously has to use rigid as well as elastic registration methods. The work in this field is in experimental stage now, actually the experts are trying to develop suitable registration method [YKT+06b, YKT+06a].
Chapter 3

PDE-based image processing

The tremendous development of information technologies in last 60 years has initiated the foundation of many new research fields. Image processing and computer vision can be counted among those new fields. Along with enormous increase of the commonly available computer power, there is an apparent shift from the simple ad-hoc image processing methods to the more sophisticated mathematically well-founded methods in last two decades. The PDE-based image processing techniques belong to such methods. We want to provide a brief “state-of-the-art” survey of this vivid field in this chapter. Our goal is to familiarize the uninformed reader with this field and describe the fundamentals. Naturally, this chapter cannot give a complete overview about the field of PDE based image processing. We primarily describe those methods that will be used in following chapters in order to easily describe own contribution. Finally, we don’t want to provide a full theoretical review, we describe this field from the view of practitioner and we hope that it will be comprehensible for other practitioners.

We describe the fundamentals of nonlinear diffusion filtering in the first section of this chapter. Several implicit active contour models for image segmentation and their implementation in the level set framework will be discussed in the second section. Then we provide a brief overview of the variational optic flow methods for motion estimation. Finally, we summarize previous successful application of the PDE-based image processing techniques in fluorescence microscopy.

What PDE-based image processing is  The PDE is abbreviation for partial differential equation. Differential equation describes the relation between an unknown function and its derivatives. If the unknown function depends on several variables, partial derivatives can occur and therefore such equation is called partial differential equation. The images are considered as functions of two or more variables in PDE-based image processing. Hence, their transformations and operations over them can be described by the partial differential equations in such context. Certain PDE-based image processing algorithms are inspired by a physically based processes (e.g. nonlinear diffusion) whose mechanism is described by partial differential equation. Modern segmentation algorithms, denoising techniques or motion estimation techniques can be defined as a minimization problem of some energy functional. Nevertheless, the solution of such minimization problem leads to the partial differential equations as well. Therefore, PDE-based image processing inspired by physics or using the powerful mathematical tools can provide new possibilities and methods to obtain desired results.
3.1 Diffusion filtering

In this section, we discuss several nonlinear diffusion filters and their properties. We will describe the fundamental ideas of nonlinear diffusion filtering, particular filters and numerical methods for diffusion filtering. We will also discuss fast implementations of these methods, because they are strongly demanded when processing three dimensional image data.

3.1.1 Physical motivation

Diffusion can be intuitively described as a physical process which equilibrates the concentration differences of some substance while preserving its total mass (the mass is only diffused neither created nor destroyed). This intuitive definition can be described with following mathematical formulation:

The equilibration property is represented by the Fick’s law

$$j = -D \cdot \nabla u$$

where \(u\) denotes the concentration function of particular substance \(\nabla u\) denotes the concentration gradient of the substance, \(j\) denotes the flux that compensates this gradient. The relation between concentration gradient \(\nabla u\) and flux \(j\) is described by the diffusion tensor \(D\) which is a positive definite symmetric matrix. The mass preservation property is expressed by the continuity equation

$$\partial_t u = -\text{div} j$$

where \(t\) denotes the time and \(\text{div}\) denotes the divergence operator. Those two equations can be combined and the diffusion equation is obtained as a result

$$\partial_t u = \text{div}(D \cdot \nabla u)$$

This equation appears in many physical processes. In image processing, the concentration \(u\) is interpreted as gray value in a certain point. Depending on the tensor \(D\), this equation can describe a large family of image processing filters. If \(D\) is replaced with single scalar value for the whole image, one gets the classic smoothing filter. If \(D\) depends on the underlying evolving image \(u\) the corresponding diffusion equation describes a nonlinear diffusion filter. The nonlinear filters can be further divided to isotropic where the flux \(j\) is parallel to \(\nabla u\) and anisotropic filters where the flux \(j\) and \(\nabla u\) are not parallel.

3.1.2 Gaussian diffusion filter

One marvelous result of image processing theory shows that the most widely used method for image smoothing (low pass Gaussian smoothing filter) can be expressed in terms of partial differential equations [Wit83, AGLM93].

Let \(\Omega \subset \mathbb{R}^m\) denote the \(m\)-dimensional image domain and \(f : \Omega \rightarrow \mathbb{R}\) an initial gray-scale image. Let us consider \(u(x, 0) = f(x)\) as an initial condition of following diffusion equation

$$\partial_t u = \text{div}(l \cdot \nabla u)$$

where \(l\) denotes the identity matrix. This equation can be easily transformed to following form

$$\partial_t u = \Delta u$$
3.1. Diffusion filtering

where \( \Delta u = \partial^2_{\nu_1} u + \cdots + \partial^2_{\nu_m} u \) denotes the application of standard Laplacian operator on \( u \). The exact solution \( u(t) \) of this equation in time \( t > 0 \) is given by convolving \( f \) with the Gaussian
\[
G_\sigma = \frac{1}{(2\pi \sigma^2)^{m/2}} \cdot \exp \left( -\frac{|x|^2}{2\sigma^2} \right)
\]
(3.6)

where \( m \) denotes number of dimensions and the standard deviation \( \sigma = \sqrt{2t} \). Vice versa, the convolution with gaussian filter with some \( \sigma \) can be obtained as a solution of the diffusion equation (3.5) to the diffusion time \( T = \frac{1}{2} \sigma^2 \).

The theoretical properties of the linear diffusion filter have been investigated mainly by the means of the so-called linear scale space theory [Wit83, AGLM93]. Consider that the original image \( f(x) \) was embedded into a family of gradually simplified images \( \{ T_t f : t \geq 0 \} \) where each \( T_t f \) image is a particular solution \( u(t) \) of the equation (3.5) in time \( t \). Such family of images forms scale space. The scale space representation have to meet several requirements [AGLM93]. Generally, the operator \( T_t \) is desired to perform information reduction, smoothing and not to create new spurious details. Moreover, it is desirable that the operator \( T_t \) does not depend on the shift in gray values of the input data or translation of the input data. It has been shown that the gaussian scale space obtained by linear diffusion filtering is the only reasonable way to define linear scale space [Wei96].

It is well-known, that linear diffusion filtering has excellent smoothing results. The example of Gaussian smoothing is in the Fig. 3.1. However, one can see that the strength of diffusion process is the same on the whole image domain. It is important that the filter smoothes the undesirable local extrema (noise), but it also harms (with equal strength) the important structures and features of the image, particularly it blurs and dislocates the edges. The nonlinear diffusion filters were designed to overcome these drawbacks.

3.1.3 Nonlinear diffusion filters

Perona and Malik proposed a nonlinear diffusion filtering method [PM90] in order to avoid blurring and dislocation of edges, which is the main drawback of linear diffusion. The key idea of their pioneering work is to reduce the diffusion to minimum at locations with large likelihood to be edges.

The nonlinear diffusion filter proposed by Perona and Malik has following structure in \( m \) dimensions. Let \( \Omega \) again denote the \( m \)-dimensional image domain and \( f \) an initial gray-scale image. The nonlinear diffusion filter calculates a filtered image \( u = (x, t) \) of \( f(x) \) as a solution of diffusion equation
\[
\partial_t u = \text{div}(g(|\nabla u|)\nabla u)
\]
(3.7)

considering \( f(x) \) as initial condition
\[
u(x, 0) = f(x)
\]
with reflecting boundary conditions
\[
\partial_n u = 0 \quad \text{on} \quad \delta \Omega,
\]
where \( n \) denotes the outer normal to the image boundary \( \delta \Omega \), \( \nabla \) denotes the spatial gradient operator, \( \text{div} \) denotes standard vector divergence operator and \( g(s) \) denotes the diffusivity
3.1. Diffusion filtering

Figure 3.1: The impact of linear diffusion filtering. (a) Left: Example of three dimensional input image with HP1 protein domains inside the HL-60 cell nucleus. The image size is 400 × 400 × 38 voxels. xy, xz and yz cuts at position (250, 241, 21) are shown. The image was artificially degraded by Poisson and Gaussian noise. This image serves as the input image for the example of diffusion types. (b) Right: The example input image was filtered with Gaussian diffusion filter. The standard deviation $\sigma$ was set to 3.0. The noise is perfectly smoothed. However, the significant edges are blurred. Note that the voxel values from interval (0, 170) were mapped to interval (0, 256) and the gamma correction to $\gamma = 1.5$ was performed in both left and right images. The aim was to improve the visibility on the paper.

The edges should be localized in order to be able to prefer intra-region to inter-region smoothing. Perona and Malik proposed to use gradient magnitude $|\nabla u|$ of the evolving image as such edge detector. The diffusivity function $g(s)$ should be decreasing and nonnegative with $g(0) = 1$ and $g(s) = 0$ as $s \to \infty$ in order to smooth relatively homogeneous regions while preserving significant edges. Perona and Malik proposed following half bell-shaped $g$ functions:

$$
g(s) = \frac{1}{1 + (s/\lambda)^2}$$

(3.8)

and

$$
g(s) = e^{-(s/\lambda)^2}$$

(3.9)

where $\lambda > 0$. The diffusion in pixels with small gradient magnitude is almost linear due to the bell-shaped diffusivity function $g$ and vice versa, the diffusion is suppressed in points with large gradient magnitude. Therefore intra-region diffusion is favored over inter-region diffusion. The $\lambda$ plays here the role of contrast parameter. The diffusion filter will smooth the pixels with absolute value of the gradient magnitude lower than $\lambda$ and on the other hand Perona-Malik model enhances gradients in pixels, whose absolute value of the gradient magnitude is greater than $\lambda$.

This behavior can be more clearly described with flux function corresponding to particular diffusivity function $g(s)$ (See Fig. 3.2). Let $\Phi(s) = s \cdot g(s)$ denote such flux function. It is non-monotone and it satisfies $\Phi'(s) > 0$ for $s < \lambda$ and $\Phi'(s) < 0$ for $s > \lambda$ (see Fig. 3.2). For simplicity
3.1. Diffusion filtering

consider now only one dimensional diffusion equation (3.7). It can be written as

$$\partial_t u = \partial_x (\Phi_x(u)) = \Phi'(u)u_{xx}. \quad (3.10)$$

We observe that the change of $u$ in time depends on $u_x$. If $u_x$ is lower than $\lambda$ the forward diffusion is performed. It means that the diffused pixel looses more mass to its neighbors with smaller intensity value than it gains from the neighbors with larger intensity value. Therefore, edges are reduced. Vice versa, if $u_x$ is greater than $\lambda$ backward diffusion is performed. This means that a pixel can obtain more mass from its neighbors with larger intensity values than it looses to the neighbors with smaller intensities. Thus, the edges are enhanced.

Some alternative diffusivity $g$ functions, their influence on the filtering process and the connections between nonlinear diffusion and robust statistic is analyzed in [BSMH97] in detail.

Although the experiments with Perona-Malik filter (PM filter) are very impressive (see Fig. 3.3a), there are two problems to deal with. First, the PM filter can’t distinguish between edges and oscillations caused by the noise (see Fig. 3.3b). This problem is obvious from the definition of the nonlinear diffusion filter and the diffusivity function of the Perona-Malik filter. Moreover, the contrast enhancing effect of backward diffusion can even increase the oscillations caused by noise. The second problem has deeper theoretical roots. Although the implementations of the PM filter usually work very well without any problems, nobody was able to completely prove, that the solution of the underlying PDE is unique, till today (see chapter 15 in [HG99]). This means that we can get totally different results by evolution of two almost similar images theoretically.

These two problems were fixed by Catt et. al in 1992 [CLMC92]. They proposed a regularized version of Perona-Malik filter, which had unique solution and suppressed the noise in a better way. The CLMC (Catt, Lions, Morel, Coll) filter has following structure. The PDE equation (3.7) is slightly modified to

$$\partial_t u = \text{div}(g(|\nabla u|^2)\nabla u) \quad (3.11)$$

Figure 3.2: Graphs of the diffusivity functions (3.8,3.9) (blue line) and their corresponding flux functions (red line). (a) Left: Diffusivity function $g(s) = \frac{1}{1+A^2}$ with $\lambda = 2$. The pixels with $|\nabla u| < \lambda$ are diffused. The contrast in pixels with $|\nabla u| > \lambda$ is enhanced. (b) Right: Diffusivity function $g(s) = e^{-\frac{s^2}{\lambda^2}}$, again $\lambda = 2$. Note that the diffusion is almost stopped for $|\nabla u| > 10$. 

Some alternative diffusivity $g$ functions, their influence on the filtering process and the connections between nonlinear diffusion and robust statistic is analyzed in [BSMH97] in detail.
The gradient within the diffusivity $g$ in equation (3.7) is regularized by convolving it with Gaussian $G_\sigma$ with standard deviation $\sigma > 0$, thus $u_\sigma = G_\sigma * u$ (This regularization technique is widely used in other PDE based methods). For a diffusivity $g$ function they used

$$
g(s^2) = \begin{cases} 
1 & s^2 = 0 \\
1 - e^{-c_m s^2/\sigma^2} & s^2 > 0
\end{cases}
$$

The contrast parameter $\lambda$ and exponent $m$ must be set by the user of the filter, constant $C_m$ is computed in order to diffuse the pixels with gradient magnitude lower than $\lambda$. The size of $m$ influences the steepness of the cross-over between edge smoothing and preserving behavior. An example of the CLMC diffusivity function and its corresponding flux function are in Fig. 3.4.

Catt et. al proved that filter defined in such a way has always unique solution. Furthermore, the filter can suppress the noise artifacts of size smaller than $\sigma$. This partly suppress the limitations of Perona-Malik model, which interprets noise as edges (see Fig. 3.5a).

There are still some limitation of the CLMC filter, it can’t suppress the noise along the edges, because the diffusivity is almost zero there (due to edges). Moreover both PM and CLMC filters can preserve and enhance edges only in narrow scale slope. Weaker edges can be blurred and smooth ramp edges can become staircased according to the size of $\lambda$ parameter [KS02, Wei97] in PM filtering. The experiments show that CLMC filter is less affected by staircasing effect [NS92].
3.1. Diffusion filtering

Figure 3.4: Graph of diffusivity function (3.12) (blue line) and its corresponding flux function (red line) used in CLMC filter. The $m = 8$, $\lambda = 2$ and the $C_m$ parameter is equal to 3.31527. The $m$ parameter has influence on the steepness of the function in the neighborhood of $\lambda$.

Figure 3.5: The impact of CLMC and EED nonlinear diffusion filter. (a) Left: The input image Fig. 3.1a was filtered with CLMC filter. The image size was 400 x 400 x 38 voxels. $xy$, $xz$ and $yz$ cuts at position $(250, 241, 21)$ are shown. The CLMC filter with diffusivity (3.12), $\lambda = 2.0$, $m = 8.0$ and $\sigma = 0.8$ was run to time $t = 20.0$. (a) Left: The PM filter with diffusivity (3.8) and $\lambda = 2.0$ was run to time $t = 20.0$. (b) Right: The 21st $xy$ slice from the input image Fig. 3.1a was filtered with 2D EED filter. The image size was 400 x 400 pixels. The EED filter with diffusivity (3.8), $\lambda = 2.0$ and $\sigma = 0.1$ was run to time $t = 20.0$. Note that the voxel values from interval $(0, 170)$ were mapped to interval $(0, 256)$ and the gamma correction to $\gamma = 1.5$ was performed in both left and right images. The aim was to improve the visibility on the paper.
3.1.4 Anisotropic nonlinear diffusion filters

We have described that the PM and CLMC filters prefer intra-region to inter-region smoothing. They preserve the important edges and discontinuities. However, the noise in such regions is unfiltered too. Anisotropic nonlinear diffusion filters were designed to overcome this problem.

Anisotropic diffusion filters do not take into account the contrast of edges only, but also their directions. This can’t be done with the scalar diffusivity $g$ function anymore, the diffusion tensor has to be introduced [Wei97]. In following text we will describe the Edge-Enhancing diffusion (EED) filter of Weickert [Wei94, Wei96] and we will describe this filter in two dimensions for the sake of clarity. The generalization to 3D is straightforward. The diffusion equation (3.11) is modified to

$$\frac{\partial u}{\partial t} = \text{div}(D(\nabla u_\sigma) \cdot \nabla u),$$

(3.13)

where $D(\nabla u_\sigma)$ now denotes the symmetric positive definite diffusion tensor. In order to prefer smoothing along the edge to smooth across it, Weickert constructed the orthonormal system of eigenvectors $v_1, v_2$ of $D$ such that they reflect the edge structure:

$v_1 \parallel \nabla u_\sigma, \quad v_2 \perp \nabla u_\sigma$

One should also choose corresponding eigenvalues $\lambda_1$ and $\lambda_2$ such that

$$\lambda_1 = g(|\nabla u_\sigma|), \quad \lambda_2 = 1$$

where $g$ is the scalar diffusivity function as was defined in previous section (usually the diffusion same as in CLMC filter is used).

Weickert proved that such defined filter can eliminate the noise even at edges, where the diffusion along the edge (eigenvector $v_2$) is preferred to the diffusion parallel to image gradient. On the other hand strong diffusion with EED can lead to rounding of sharp corners and high curved objects in the image. An example of EED diffusion in in the Fig. 3.5b.

3.1.5 Nonlinear diffusion filters with unbounded diffusivity

All diffusion filters described in previous section rely on bounded diffusivity functions which depend on several parameters. From the practical experiences with PM, CLMC and EED filters it is known that they are sensitive to parameter settings ($\lambda, \sigma$), which should be tuned to enhance the edges. If this is not done carefully, some weak edges can become either blurred or staircased. Motivated by this observation there has been a growing interest in unbounded diffusivities, which become singular in zero [Bro05, ABCM01, KS02, WWS05]. These filters create simplified images with removed oscillations and well preserved edges. The most explored are filters which consider as a diffusivity function the following class of singular functions

$$g(|\nabla u|) = \frac{1}{|\nabla u|^p} \quad p > 0.$$  

(3.14)

There are some advantages of diffusion filters defined in such a way. The most obvious advantage is that there is no additional parameter to tune. Further, diffusion filters with unbounded diffusivity preserve edges over large scale slope [KS02] and for $p > 1$ they even enhance the edges.
Figure 3.6: The impact of Total Variation (TV) and Balanced Forward Backward (BFB) nonlinear diffusion filter. The input image Fig. 3.1a was filtered with TV and BFB filter. The image size was $400 \times 400 \times 38$ voxels. $xy$, $xz$ and $yz$ cuts at position $(250, 241, 21)$ are shown. (a) **Left:** The TV filter with diffusivity (3.15), $p = 2$ and $\epsilon = 0.001$ was run to time $\tau = 20.0$. (b) **Right:** The BFB filter with diffusivity (3.15), $p = 2$ and $\epsilon = 0.001$ was run to time $\tau = 40.0$. Note that the voxel values from interval $(0,170)$ were mapped to interval $(0,256)$ and the gamma correction to $\gamma = 1.5$ was performed in both left and right images. The aim was to improve the visibility on the paper.

The most explored are the Total Variation (TV) and Balanced Forward Backward (BFB) diffusion filters. The TV diffusion filter [ABCM01] is obtained by setting $p = 1$ in the diffusivity 3.14. This filter has following properties: it quickly removes the oscillations with a speed that depends on their scale, while it preserves the shape of large objects for a long time. Nevertheless, its application leads to constant signal (average gray value of the input image) in finite time. On the other hand, it does not enhance the edges as we can deduce from the flux function in the Fig. 3.7a. One can also see that the flux function is constant. This leads to evolution rules proposed by Brox [Bro05]. The regions (pixels or voxels in discrete case) which forms local extrema adapt their value to the values of its neighbors. All other regions do not change their value. Neighboring regions with same value form and behave as super-region during the evolution. The regions adapt their values with speed which is inverse proportional to their size. Therefore, the small scale details are removed prior to large scale features.

Although TV diffusion has many nice properties, it produces only smooth results, i.e. it does not enhance the edges (see Fig. 3.6a). If we set $p > 1$ in (3.14) the diffusion filter will allow backward diffusion, which enhance the edges. Although the every diffusion filter with $p > 1$ has this behavior, Keeling and Stollberger [KS02] proved that the best of them with respect to edge enhancement is the BFB diffusion filter where $p = 2$ (see Fig. 3.6b). The BFB filter has the similar behavior to TV filter that is it removes the oscillations and change mainly the values of extrema regions. In addition to that, the points on edges can immediately join a neighboring extrema super-region due to backward diffusion, which sharpen the edge, while with TV filter their value remains unchanged.

Unfortunately, besides those nice properties, there is a disadvantage of these filters: their numerical implementation is difficult due to the singularity of the $g$ function in zero. One can
use the following regularized version of the diffusivity to avoid implementation problems.

\[
g(|\nabla u|) = \frac{1}{(|\nabla u|^2 + \epsilon^2)^{p/2}}
\]  

(3.15)

Note that a small regularizing parameter \( \epsilon \) is introduced in the denominator of the diffusivity function. It should be mentioned that some blurring of edges is introduced [WWS05], when using regularized diffusivity (3.15).

For the sake of completeness we give here a short list of filters, which are related to nonlinear diffusion and we did not discuss them in this section. The foundations of geometry based diffusion filtering are described in [AGLM93] and other geometric based diffusion filters are described in [ALM92, MS96c, MS95, MS96b, WX01]. The field of nonlinear diffusion filtering is also closely related to total variation image restoration. There is a nice overview [CEPY, CS05], which summarizes current state of the art in Total Variation restoration.

### 3.1.6 Numerical approximations

We will discuss available numerical schemes for solving nonlinear diffusion filters in this section. The differential equations are usually solved with finite difference or finite elements methods. Methods based on finite differences are used more often by the image processing community, because they are relatively easy to implement because the digital image provides naturally the information on the Cartesian grid. Therefore, we restrict ourselves only to finite difference methods in this brief summary. Further, we take the CLMC filter (3.11) as a typical representative of nonlinear diffusion filter and present the most typical numerical schemes on it.

**Explicit finite difference scheme**

Explicit schemes are very simple to implement and therefore they are used as a starting scheme in vast majority of applications. The implementation simplicity is weighted with the fact that the small time step is needed for ensuring the stability of computation. We will now shortly
describe the explicit scheme for computation of two dimensional CLMC filter. Extension of
following discretization to three dimensions is straightforward. Let us remind now the CLMC
diffusion equation
\[ \partial_t u = \text{div}(g(|\nabla u|^2) \nabla u). \]

Further, we can expand this equation to
\[ \partial_t u = \partial_x(g(|\nabla u|^2) \partial_x u) + \partial_y(g(|\nabla u|^2) \partial_y u) \]

This continuous diffusion equation is discretized in following manner. The \( u_{i,j}^n \) denotes the
pixel at position \((x_i, y_j)\) and time \( t_n \) where \( x_i = i \cdot h, y_j = j \cdot h \) and \( t_n = n \cdot \tau, h \) denotes the spatial
grid size and \( \tau \) the time step.

**Time discretization** The vast majority of numerical schemes uses first order discretization of
time derivative
\[ \partial_t u \approx \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\tau} \]
The explicit (or Euler forward difference) scheme can be then written as
\[ \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\tau} = \text{div}(g(|\nabla u_{i,j}^n|^2) \nabla u^{n}) \]
Note that the unknown value \( u_{i,j}^{n+1} \) is computed from already known values \( u^n \). That’s why the
scheme is called explicit.

**Space discretization** The one sided or central differences are used for discretization of the
spatial derivatives. Namely, following discretizations are usually used
\[ \partial_x(g(s)\partial_x u(x)) \approx D_x^-(M_x^+(g_{i,j}^n)D_x^+(u_{i,j}^n)) \]
\[ \partial_y(g(s)\partial_y u(x)) \approx D_y^-(M_y^+(g_{i,j}^n)D_y^+(u_{i,j}^n)) \]
where the one sided differencing operators and averaging operators are defined as
\[ D_x^+(a_{i,j}) = \frac{a_{i+1,j} - a_{i,j}}{h}, D_x^-(a_{i,j}) = \frac{a_{i,j} - a_{i-1,j}}{h} \]
\[ M_x^+(a_{i,j}) = \frac{a_{i+1,j} + a_{i,j}}{2}, M_x^-(a_{i,j}) = \frac{a_{i,j} + a_{i-1,j}}{2} \]
and the estimation of diffusivity function is computed by
\[ g_{i,j}^n(|\nabla u_{i,j}|^2) \approx g \left( \left( \frac{u_{i,j+1,n}^n - u_{i,j,n}^n}{2h} \right)^2 + \left( \frac{u_{i+1,j,n}^n - u_{i,j,n}^n}{2h} \right)^2 \right) \]

When we substitute all the derivatives with their approximations, we get following scheme
\[ u_{i,j}^{n+1} = u_{i,j}^n + \tau (D_x^- (M_x^+(g_{i,j}^n)D_x^+(u_{i,j}^n)) + D_y^- (M_y^+(g_{i,j}^n)D_y^+(u_{i,j}^n))) \]
which can be finally expanded to

\[
\begin{align*}
    u_{i,j}^{n+1} &= u_{i,j}(1 - \frac{\tau}{2\delta_x}(g_{i-1,j}^n + g_{i,j}^n)) + \\
    &\quad u_{i,j}(\frac{\tau}{2\delta_y}(g_{i-1,j}^n + g_{i,j}^n)) + \\
    &\quad u_{i,j-1}(\frac{\tau}{2\delta_x}(g_{i,j-1}^n + g_{i,j}^n)) + \\
    &\quad u_{i,j+1}(\frac{\tau}{2\delta_x}(g_{i,j+1}^n + g_{i,j}^n)) + \\
    &\quad u_{i,j+1}(\frac{\tau}{2\delta_y}(g_{i,j+1}^n + g_{i,j}^n)) + \\
    &\quad u_{i,j}(g_{i,j}^n) + g_{i,j+1}^n + g_{i,j-1}^n + 2g_{i,j}^n + g_{i,j+1}^n + 2g_{i,j}^n + g_{i,j-1}^n + 2g_{i,j}^n + g_{i,j+1}^n)) + \\
\end{align*}
\]

(3.16)

This scheme is valid only for the pixel inside the image. The diffusion equation on boundary pixels has to be computed according to reflecting (Neumann) boundary conditions. The image must be enlarged by rows and columns of “ghost pixels”. The value in the ghost pixels is copied from the neighboring boundary pixel (see Fig. 3.8). This ensures that the Neumann boundary conditions will be fulfilled.

One can notice that the individual contributions to the diffusion between pixels in x and y direction in (3.16) are independent on each other. Therefore, if we arrange the pixels in particular order so that they form a column vector \( u^n \), the explicit scheme (3.16) can be written in matrix notation as

\[
\begin{align*}
    u^{n+1} &= (I + \tau A_x(u^n) + \tau A_y(u^n)) u^n
\end{align*}
\]

(3.17)

where \( I \) denotes the identity matrix, \( A_x(u^n) \) and \( A_y(u^n) \) denotes the matrices with diffusion coefficients. Note that the matrices with diffusion coefficients depend on \( u^n \). This notation will be used for explanation of AOS scheme in the following text. The explicit scheme is relatively easy to implement, however the time step \( \tau \) should be set to relatively small values in order to ensure numerical stability. Weickert in \cite{WtHRV98} shows that the explicit numerical scheme is stable only if

\[
    \tau < \frac{1}{2m}
\]

where \( m \) is number of dimensions, assuming that \( g(s) \leq 1 \) and \( h = 1 \). This limitation is quite severe, because the computation requires often too much iterations (due to the time step restriction) which leads to unacceptable computation times, especially for large three dimensional images.
AOS scheme

The semi-implicit Additive Operator Splitting (AOS) scheme for nonlinear diffusion filters was independently proposed by Weickert [WtHRV98] and Lu [LNT92] to overcome this barrier. The AOS scheme is unconditionally stable (no time step restriction). We will now shortly describe the key idea of this scheme.

Let us remind the matrix notation (3.17) of the explicit scheme discretization. The semi-implicit scheme has following form in two dimensions

$$u^{n+1} = u^n + (\tau A_x(u^n) + \tau A_y(u^n)) u^n + 1$$

(3.18)

Although the element of the matrices $A_x$ and $A_y$ are computed from the previous time step, the unknown $u^{n+1}$ has to be found by solving a system of linear equations

$$(I - \tau A_x(u^n) - \tau A_y(u^n)) u^{n+1} = u^n$$

(3.19)

Unfortunately, this system cannot be solved by some direct solver (Gauss elimination method, Thomas algorithm), because there exists no pixel ordering in two or three dimensions which leads to a system matrix with small bandwidth (i.e. $n$-diagonal). Therefore, iterative algorithms like Gauss-Seidel or multigrid method have to be used. Both this methods have its specific constraints. The AOS scheme was proposed in order to get around this constraint. Weickert [WtHRV98] and Lu [LNT92] independently suggest to separate the diffusion in individual directions by

$$u^{n+1} = \frac{1}{m} \sum_{i=1}^{m} (I - \tau A_i(u^n)) u^{n+1} = u^n$$

(3.20)

where $m$ denotes number of dimensions and $l = 1, \ldots, m$ denotes the direction index. The solution of this system is obtained by

$$u^{n+1} = \frac{1}{m} \sum_{i=1}^{m} (I - \tau A_i(u^n))^{-1} u^n$$

(3.21)

Each of the summands in (3.21) denotes the one dimensional diffusion process along the direction $l$. The overall result of one diffusion iteration is obtained as an average of those one-dimensional processes. The one-dimensional diffusions can be easily solved because their system matrix is tridiagonal and is diagonally dominant. Therefore, it can be solved using some direct solver (e.g. Thomas algorithm).

The AOS scheme is unconditionally stable (i.e. there is no time step restriction). It was experimentally found out that AOS scheme is about ten times faster than common explicit scheme. One can even perform the filters in real time using the parallel implementation of AOS scheme [BJF+04, WZtHRN97], because each row and column in the image is diffused separately. More accurate version of the AOS scheme was proposed in [BSIK03]. The AOS scheme was originally proposed only for nonlinear diffusion filters with bounded diffusivity function. However Brox [Bro05] shows, that we can put it to approximate even the filters with unbounded diffusivity function. Modified AOS scheme for anisotropic nonlinear diffusion filters was proposed in [Mrá01].
3.2 Implicit active contour models

In this section we get familiar with the basic ideas of implicit (also called geometric) active contour models for image segmentation. We will present the main ideas of level set methods first. Then we describe several geometric active contour segmentation models, which use level set framework for numerical computations. Finally we will discuss the difficulties of their implementation and some fast algorithms and approximations for their solution.

In the last two decades there was done an extensive research on so-called active contours (sometimes called also dynamic contours) in the area of image segmentation. The classical approach, which was originally proposed by Kass et al. [KWT87], is based on deformation of an initial contour (in two dimensions) towards the desired boundary of the object, which should be detected. The edge-driven deformation of the contour is obtained by trying to minimize following energy functional, which is designed to have its minimum at the boundary of the object.

\[ E_{\text{snake}} = \int_0^1 E_{\text{internal}}(C(s)) + E_{\text{image}}(C(s)) + E_{\text{constraints}}(C(s)) \, ds \]  

(3.22)

where \( C(s) \) is the parameterization of the initial contour. These active contours (called "snakes" by Kass) are only an example from the wide family of deformable models. The deformable models can vary in the formulation of the energy minimizing functional as well as in the choices for their representation and implementation.

Although there were published several successful parametric [KWT87, TWK88] and discrete [Del99] deformable models, there is a continually growing interest in the implicit deformable models, which use level set methods presented in the pioneering work of Osher and Sethian [OS88]. In the following text we will describe the basic ideas and properties of level set methods and the implicit (geometric) active contours models, which use the level set methods for the computation of the deformation.

3.2.1 Level set framework

The level set method was proposed by Osher and Sethian in [OS88] for tracking of the curve under mean curvature motion in time. The novelty of their approach was in the representation of the evolving curve. The moving contour is represented implicitly as a zero-level set of some function. This approach has several advantages over other parametric models and it has become very popular for curve representation in image analysis.

We describe now the level set framework. We restrict ourselves to two dimensions for the sake of simplicity. Let us consider a closed curve \( C(s) \) in Euclidean space \( \mathbb{R}^2 \) parameterized with the arc-length \( s \). Let \( C(s) \) bound an open region \( \omega \). The curve defined in such a way, can be represented by a so-called level set (or embedding) function \( \phi(x) \). The curve \( C \) itself is represented by the zero level set of the smooth function \( \phi \) i.e. \( C = \{ x | \phi(x) = 0 \} \). This can be easily done when using level set function with following properties:

\[ \begin{align*}
\phi(x) &< 0 \quad \text{for} \quad x \in \omega \\
\phi(x) &> 0 \quad \text{for} \quad x \notin \omega \\
\phi(x) &= 0 \quad \text{for} \quad x \in \partial \omega = C(s)
\end{align*} \]

This kind of curve representation allows to define describe the curve evolution. Let us consider the evolution of type

\[ \partial_t C = F n \]  

(3.23)
3.2. Implicit active contour models

Figure 3.9: Example of contour evolution using the level set framework. The contour $C$ is embedded as a zero level set of the function $\phi$. The contour is also projected onto a separate $xy$ plane for better visualization. 

(a) Left: Two circles are the initial contour. The initial contour $C$ (green color) is the zero level set of the function $\phi$ (red color). The circles expand in the normal direction during computation.

(b) Right: The contour $C$ later in the time is the zero-level set of the evolved function $\phi$. Note that the contour $C$ changes the topology naturally as the two circles merge during the computation.

where $n(s)$ denotes the outer normal of the curve $C$ at $s$ and $F(s)$ is a scalar force that force the curve to move in the normal direction. Osher and Sethian [OS88] proposed that this evolution can be described in the level set framework as well by following differential equation which acts on $\phi$ function

$$\partial_t \phi = F|\nabla \phi|$$

(3.24)

where $\phi_0(x)$ is given initial condition and defines the initial contour. The curve $C$ can be captured anytime only by locating the zero-level set of the function $\phi$. The evolution of the curve $C$ in time is embedded in the evolution of the whole function $\phi$ (see Fig. 3.9). The initial $\phi_0$ function is usually defined as $\phi_0(x) = \pm d$, where $d$ is the shortest distance from point $(x, y)$ to the initial contour $C_0$. The force $F$ can be some function of image gradient (if we are speaking in the context of image segmentation), it can depend on the geometry of the curve $C$, or it can be constant.

The level set approach offers several advantages:

- First advantage is that the level set formulation provides more convenient discretization. Function $\phi$ is defined on the fixed homogeneous grid and is independent on the evolution of its level lines. On the contrary, the discretized points of $C(s)$ moves during the evolution. This can lead to large differences in their mutual distances and their concentrations on the whole curve. Therefore, one should often resample the curve during the evolution (this process is also known as reparametrization).

- Second advantage is that the level set framework naturally represents topological changes. Moreover, sharp corners and kinks can be formed during the evolution on the zero level set (see Fig. 3.9).

- As a third advantage we consider the fact that in the level set formulation the implicit function represents also the regions of the curve. One can easily decode from the sign of $\phi(x)$ whether the point is inside or outside. Moreover, intrinsic geometric quantities
like normal and curvature of the interface (curve in 2D surface in 3D) can be easily extracted from the higher dimensional function \( \phi \). The normal is obtained as \( n = \nabla \phi / |\nabla \phi| \), mean curvature \( \kappa = \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \).

- Very important advantage for biomedical imaging is that everything can be directly extended to moving surfaces in three or more dimensions.
- A discrete grid can be easily used together with finite differences to devise a stable numerical scheme to approximate the solution.

It is obvious that the computations on \( m \)-dimensional structure \( C \) are substituted with computations on \( (m + 1) \)-dimensional structure \( \phi \). Therefore, the main drawback of the level set method is the additional computational complexity. The main properties of level set methods and their applications to various fields are extensively discussed in [OF03, Set99].

### 3.2.2 Implicit active contour models

The level set framework was used as a basic building block in several implicit active contour models. We will discuss several of them in this subsection. The models will be described mainly in two dimensions for the sake of clarity, but the conclusions and facts presented in this subsection are valid also for three dimensional case. Further, some difficulties that are hidden in the implementation of implicit active contour models will be presented.

**Front propagation – geometric active contour model** First implicit active contour model for image segmentation was independently proposed by Caselles et al. [CCCD93] and Malladi et al. [MSV95]. Their model is based on the ideas developed by Osher and Sethian [OS88] to model propagating solid/liquid interfaces with curvature dependent speed.

The minimization of the curve length in two dimensions can be performed with the mean curvature motion [AGLM93]. Let us remind the explicit and implicit formulation (3.23, 3.24) of the curve evolution in the normal direction. Let again the curve \( C \) be embedded as a zero level set of function \( \phi \). The implicit formulation of mean curvature motion can be then written as

\[
\frac{\partial \phi}{\partial t} = \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi| \quad (3.25)
\]

Let \( \kappa = \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \) denote the mean curvature. Previous equation can be then rewritten to

\[
\frac{\partial \phi}{\partial t} = \kappa |\nabla \phi| \quad (3.26)
\]

One can show that the mean curvature motion (MCM) minimizes the arc-length of the curve in two dimensions. Arbitrary curve evolved with the MCM is transformed to circle which collapses to a point and vanishes (see Fig. 3.10). In three dimensions the exact geometric meaning of the curvature motion depends on the used curvature (there exist several definitions of curvature for 3D). Nevertheless, the same MCM process in three dimensions reduces the surface area of the evolving surface when using the mean curvature.

The minimization of the boundary length between segmented regions belongs among the common assumption of segmentation models. Therefore, the mean curvature motion forms a
3.2. Implicit active contour models

Figure 3.10: Example of mean curvature driven motion. The initial contour at each point is deformed in the direction of its outer normal direction with the speed proportional to mean curvature of the curve in this point. This motion minimizes the arc-length of the curve. (a) Top Left: The contour in time $t = 25$. (b) Top Center: The contour in time $t = 25$. (c) Top Right: The contour in time $t = 50$. (d) Bottom Left: The contour in time $t = 100$. (e) Bottom Center: The contour in time $t = 200$. (f) Bottom Right: The contour in time $t = 275$.

part of many implicit active contour models for image segmentation. The front propagation model of Caselles et al. [CCCD93] and Malladi et al. [MSV95] combines the image driven front propagation in normal direction with the mean curvature motion. Let $\Omega \subset \mathbb{R}^m$ denote the m-dimensional image domain and $f : \Omega \rightarrow \mathbb{R}$ an gray-scale image, which should be segmented. The equation of motion of the implicit front propagation model is given by

$$
\partial_t \phi = g(c + \kappa) |\nabla \phi| \\
\phi(x, t = 0) = \phi_0(x)
$$

(3.27)

where $\phi_0$ embeds the initial position of the interface, $c$ is given constant, the term $\kappa = \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right)$ is the mean curvature. $g$ is decreasing nonnegative function of the segmented image $f$ and serves as the edge detector.

The initial interface (contour in 2D, surface in 3D) is driven by two forces in the outer normal direction. First, the interface is forced to move in the normal direction by the constant inflationary force $-c$ constant. Second, the interface is forced to stay smooth, because the mean curvature term forces it to smooth its shape in highly concave or convex parts. Finally, the image based stopping function $g$ reduces the movement near strong edges which are assumed to form the object boundaries. Several $g$ functions were defined in section 3.1. This functions were originally considered for stopping the diffusion at edges, so they can be used to stop the evolution of the interface on the same places. Anyway in [CCCD93, MSV95] another stopping function was defined

$$
g(\nabla f) = \frac{1}{1 + |\nabla G_p * f|^p}, \quad p = 1 \text{ or } 2
$$

(3.28)
The geometric active contour model for front propagation has all advantages of the level set methods which were described above. The main advantage over the classical parametric snakes is that the interface can change its topology naturally. Therefore it can segment an arbitrary number of isolated objects starting with only one initial contour which outlines all the objects.

Geodesic active contour model

Previous segmentation model has problems with locating the boundaries with high variations in the image gradient magnitude. Note that \( g > 0 \) and the evolving interface in fact never stops its movement, it is only slowed down and almost stopped near edges. Therefore, the evolving interface can pass through a weak edge and simultaneously stop on strong edges. The segmentation process can produce wrong results as a consequence.

The geodesic active contour model proposed by Caselles et al. [CKS97] was designed to overcome this problem by adding the weighting stopping function \( g \) to the mean curvature motion model. The underlying differential equation is defined by

\[
\partial_t \phi = \text{div} \left( g(\nabla f) \frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi| \quad (3.29)
\]

Caselles et al. [CKS97] have showed that the steady state of this equation minimizes in two dimensions the energy

\[
\int g(\nabla f) |C_s(s)| ds = \int g(\nabla f) ds \quad (3.30)
\]

The equation (3.30) nicely describes the meaning of the model. One can see that the geodesic active contour model locally minimizes the stopping function \( g \). If we consider the \( g \) function as an image which is formed by hills and valleys according to \( g \) value at particular pixel, the locally minimizing contour goes through the deepest valley which is enough close to the initial contour. Using the chain rule, the equation (3.29) can be rewritten to

\[
\partial_t \phi = g \kappa |\nabla \phi| + \nabla g^\top \cdot \nabla \phi \quad (3.31)
\]

where \( \cdot \) denotes the standard scalar product of two vectors. The equation consists of the mean curvature motion term weighted by stopping function \( g \). The second term attracts the evolving interface towards edges which are strong and close enough to the interface. Such presented geodesic active contour model converges slowly than other active contour models. Therefore, the combination of the front propagation and geodesic model is often employed in practice. The evolution equation of the modified geodesic model has following form

\[
\partial_t \phi = g c |\nabla \phi| + g \kappa |\nabla \phi| + \nabla g^\top \cdot \nabla \phi \quad (3.32)
\]

Note that the evolution equation now consists of three terms. The first term forces the interface to move in the normal direction, the second weighted mean curvature motion preserves the smooth shape of the interface and the last term attracts it towards strong edges.

Numerical approximations

Several catches and difficulties hidden in the implementation will be discussed in this section. We describe the simple Euler explicit scheme for numerical solution of implicit active contour models and then we discuss other implementation details. We will describe the numerical
Figure 3.11: Example of Segmentation with modified geodesic active contour (MGAC) model. The MGAC model was run with following parameters: $c = 1$, the strength of the MCM term was reduced by multiplication with constant $\epsilon = 0.025$ and the strength of the attraction term by multiplication with constant $\beta = 0.005$. (a) Top Left: The initial contour in time $t = 0$. (b) Top Center: The contour in time $t = 20$. (c) Top Right: The contour in time $t = 50$. (d) Bottom Left: The contour in time $t = 300$. (e) Bottom Center: The contour in time $t = 600$. (f) Bottom Right: The contour in time $t = 1000$. The constant front propagation term forces the contour to move towards the objects, while the MCM term ensures that the contour stays smooth. Finally, the attraction term forces the contour to stay on object borders.

scheme for the solution of the modified geodesic active contour model (3.32), because it contains all three terms (inflationary, mean curvature and attraction term). We again describe the numerical schemes only for two dimensional images, the extension to three dimensions is straightforward.

**Time discretization** The most used discretization of time derivative is the explicit first order discretization. Similar to the numerical discretization of nonlinear diffusion filters (see section 3.1.6) the forward Euler explicit scheme can be written as

$$\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^{n}}{\tau} = gc|\nabla \phi| + g\kappa|\nabla \phi| + \nabla g^\top \cdot \nabla \phi$$

**Space discretization** The modified geodesic active contour model consists of three terms. They differ in their nature, the inflationary and attraction terms have hyperbolic nature while the mean curvature term has parabolic nature. Therefore, their numerical discretization differs as well. We first describe the discretization of the inflationary term. The numerical scheme suitable for this task was primary developed for the computational physics [OS88, Set99, OF03]. The evolution equation has the following form:

$$\phi_t = cg|\nabla \phi|$$  \hspace{1cm} (3.33)
The equation (3.33) is usually discretized with Godunov’s scheme and forward Euler operator:

\[ \phi_{i,j}^{n+1} = \phi_{i,j}^n + \tau \left( \max(cg_{i,j}, 0) \nabla^+ + \min(cg_{i,j}, 0) \nabla^- \right) \]

where \( \nabla^\pm = \left[ \frac{\max(D_x^\pm(\phi_{i,j}^n), 0)^2 + \min(D_x^\pm(\phi_{i,j}^n), 0)^2}{\max(D_x^\pm(\phi_{i,j}^n), 0)^2 + \min(D_x^\pm(\phi_{i,j}^n), 0)^2} \right]^{1/2} \)

and the operators \( D_x^\pm \) have the same meaning as in section 3.1.6.

The mean curvature motion term is the second term which forms the modified geodesic model. Let us remind the mean curvature motion equation weighted with the stopping function

\[ \phi_t = g\kappa |\nabla \phi| \quad (3.34) \]

In order to implement this type of motion we have to discretize the \( \kappa \) first. The curvature can be computed directly from the level set function \( \phi \) in two dimension with

\[ \kappa = \frac{\phi_{yy}^2 - 2\phi_{xy}\phi_{xy} + \phi_{xx}^2}{|\nabla \phi|^3} \quad (3.35) \]

The mean curvature in three dimensions can be computed by

\[ \kappa = \frac{\phi_{yy}^2 - 2\phi_{xy}\phi_{xy} + \phi_{xx}^2 + \phi_{zz}^2 - 2\phi_{xz}\phi_{xz} + \phi_{xx}^2 + \phi_{yy}^2 - 2\phi_{yz}\phi_{yz} + \phi_{zz}^2}{|\nabla \phi|^3} \quad (3.36) \]

The discretized curvature is then computed with central and one sided differences. The partial derivative with respect to particular axis and the mixed derivatives are discretized with central differences

\[ \partial_x(\phi(x)) \approx D_x(\phi_{i,j}) \quad \partial_y(\phi(x)) \approx D_y(\phi_{i,j}) \]

where central differencing operator is defined by

\[ D_x(a_{i,j}) = \frac{a_{i+1,j} - a_{i-1,j}}{2h} \quad D_y(a_{i,j}) = \frac{a_{i,j+1} - a_{i,j-1}}{2h} \]

The second order partial derivatives with respect to particular axis are discretized with one sided differences

\[ \partial_x(\partial_x(\phi(x))) \approx D_x(D_x^\pm \phi_{i,j}) \quad \partial_y(\partial_y(\phi(x))) \approx D_y(D_y^\pm \phi_{i,j}) \]

Further, note that the denominator in the curvature is the gradient magnitude of \( \phi \) function powered to three. The same gradient magnitude multiplies the curvature in the equation (3.34). Therefore, it is convenient to abbreviate those two terms and discretize them together by

\[ K_{i,j}^n = \frac{D_x(\phi_{i,j}^n)^2 D_y(\phi_{i,j}^n)^2}{(D_x(\phi_{i,j}^n))^2 + (D_y(\phi_{i,j}^n))^2 + \epsilon^2} \]

where \( \epsilon \) is a small non-zero constant in order to avoid division by zero. The whole numerical scheme for weighted mean curvature motion is then given by

\[ \phi_{i,j}^{n+1} = \phi_{i,j}^n + \tau g K_{i,j}^n \quad (3.38) \]
The third term, which forms the modified geodesic model, is the attraction term. The motion equation for this term is given by
\[ \phi_t = \nabla g \cdot \nabla \phi \] (3.39)
if we denote the components of the \(g\) gradient vector by \(\nabla g = (g_1, g_2)\) in two dimensions, the equation (3.39) can be rewritten to
\[ \phi_t = g_1 \phi_x + g_2 \phi_y \] (3.40)

The so-called upwind schemes are used for a stable implementation of this equation. These schemes choose appropriate one sided difference operator according to the sign of \(g_i\). Further the \(\nabla g\) can be discretized with central differences. The explicit scheme in two dimensions is given by
\[
\phi_{i,j}^{n+1} = \phi_{i,j}^n + \tau \left( \min(D_x(g_{i,j}), 0)D_x^+(\phi_{i,j}) + \max(D_x(g_{i,j}), 0)D_x^-(\phi_{i,j}) + \right) \\
\min(D_y(g_{i,j}), 0)D_y^+(\phi_{i,j}) + \max(D_y(g_{i,j}), 0)D_y^-(\phi_{i,j}) 
\]
(3.41)

Finally, when we combine all three terms together they form the modified geodesic model (3.32) which can be implemented by a combination of previously described discretizations

The stability of such defined scheme is determined by the second mean curvature term which has the most restrictive stability conditions.

\[ \tau \left( \frac{2}{(h_x)^2} + \frac{2}{(h_y)^2} \right) < 1, \]
where \(h_x\) and \(h_y\) denotes the grid size in corresponding axes.

The implicit active contour models implemented with this scheme converge quite slowly especially for three dimensional segmentation. One reason is that the described explicit Euler schemes introduce an annoying limitation to the time step. The second problem is, that the \(\phi\) function is updated on the whole image domain in each step in the simple case. This is a time consuming operation.

Several ideas were suggested in order to speed up the computations. Some authors proposed to speed up the computations with so-called narrowband technique [AS95] or with the sparse field method [Whi98]. Note that the zero level set of the function \(\phi\) is the only "interesting" level set in the computations. Both narrowband and the sparse field methods limit the computations only into a narrow tube of grid points around the zero level set (see Fig. 3.12), which leads to the significantly reduced number of operations in one iteration. The main difference between narrowband and sparse-field method is the radius of the tube. The sparse field method keeps the tube as narrow as possible, the narrowband method uses tubes with bigger radius.

Other authors [NH03, DT02] proposed fast methods which use point-wise scheduling of interface propagation. In each iteration are updated only one or a few selected pixels which
have the biggest probability to move (either inside or outside from the interface). The whole
computation domain is not updated in one iteration step and the entire computation is sig-
ificantly sped up as a consequence. The geometric properties of the evolving interface (like
normal and curvature) were only roughly approximated in the method of Nilsson and Heyden
[NH03]. This enables even further speed up of the computations.

Finally, there were proposed some semi-implicit schemes in order to avoid the time step
restriction. A modification of the AOS scheme was proposed to overcome the time step restric-
tion of the simple explicit scheme in [GKRR99] and in [KWBE02].

Stopping of the evolution  One can also ask when to stop the process of interface evolution.
In practice, the evolution process is stopped when the interface reaches the steady state and
does not change a lot (e.g. in position, length or surface). The second obvious choice is to
perform given number of iterations.

Additional tasks  There are other two difficulties which complicate the implementation of
implicit active contour models. First, the level set function $\phi$ can become too flat or steep
during the computations, which leads to problems with numerical approximation and numer-
ical instability. It is desirable to reshape the $\phi$ function periodically to the signed distance
function during the computations, while keeping the zero level set unchanged. Second, the
stopping term $g$ derived from the image gradient is defined only for the zero level set of the $\phi$
function. It is necessary to extend the values of the stopping term to the whole computation
domain or at least to the narrow neighborhood of the zero level set (in case of the narrowband
technique). These two problems can be solved simultaneously. Sethian and Adalsteisson
proposed fast level set function redistancing and stopping term extension technique based on
the fast marching method in [AS99]. The fast marching method will be described in section
3.2.3. It should be noted that by using sparse field method the extension of the stopping term
$g$ is not needed anymore. Quite different approach based on solving redistancing differential
equation was proposed in [PMO+99]. The alternative to narrowband technique is described in this paper too.

3.2.3 Fast marching method

The implicit active contour models presented in previous section are very robust methods for image segmentation and object reconstruction. However, their main drawback is their computational complexity, even if the narrowband or some other acceleration technique is employed.

Sethian in [Set96] proposed an extremely efficient simplified variant of above active contour models. He used neither the mean curvature motion term nor the attraction term of the geodesic model. He only considered the positive (or negative) constant inflationary term weighted by the stopping term $\gamma$. These limitations lead to monotonic (one directional) motion of the interface (contour in 2D surface in 3D). Thus the motion equation of the implicit models is reduced to

$$\phi_t = cg|\nabla \phi|$$

(3.42)

Since the interface can enter the particular pixel only once, an arrival times function $T(x)$ can be defined. The values of arrival time, when the moving interface crosses the point $x$, are stored in this function. The arrival time function $T$ satisfies following differential equation

$$g|\nabla T| = 1$$

(3.43)

with $T(x) = 0$ on given initial interface and $c = 1$. Equation (3.43) simply says that the gradient of the arrival time function is inversely proportionally to the propagation speed $g$ of the front. Therefore, the fast marching method can be considered as a weighted distance transform where the $T(x)$ function stores the shortest distance weighted by $g$ from each pixel to the initial interface. An example segmentation with Fast marching method is shown in Fig. 3.13.

The main advantage of equation (3.43) is that its solution $T$ can be constructed in an efficient way. The algorithm proposed by Sethian is non-iterative and uses the min heap data structure. It constructs the solution $T$ from the initial data with $O(n \log n)$ execution cost (where $n$ is the number of image voxels). There were published even faster implementations of the fast marching algorithm recently. Yatziv et al. use the untidy priority queues instead of min heap data structure in [YBS06]. The proposed fast sweeping algorithm of Zhao [Zha01] uses the similarity of fast marching method to distance transforms.

On the other hand, fast marching algorithm computes the evolution very fast, but the price for this efficiency is the impossibility of integrating local properties of the evolving interface (mean curvature) to the speed term. This sometimes leads to problems with segmentation of noisy images or objects with high variations in gradient at edges.

Numerical approximation

We review the fast marching algorithm for computation of the arrival time function $T(x)$ [MS96a, Set96]. The algorithm differs from the previously described numerical schemes. It is non-iterative, the solution is computed in one sweep. We describe the algorithm again only in two dimensions for the sake of clarity, the extension to three dimensions is straightforward.
3.2. Implicit active contour models

Figure 3.13: Example of Fast marching method used for image segmentation in two dimensions. The arrival time function $T(x)$ is represented with the black wire-frame. The arrival time was set to zero in the four corners of the image and the rest of the arrival time function was computed with the fast marching method. The function $T(x)$ was multiplied by $-1$ due to the visualization purposes. The black level set on the function $T(x)$ denotes the arrival time which segments the spot in the image which is depicted in the base plane of the graph.

Fast marching algorithm

**Input**
- Discrete grid $G \subseteq \mathbb{Z} \times \mathbb{Z}$ of size $n_1 \times n_2 \in \mathbb{N}^2$
- Function $g : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$, $g(x) > 0$ for each $x \in G$
- Discretized initial contour $C \subseteq G$

**Output**
- Arrival times $T(x)$ for each $x \in G$

**Initialization**

1. Initialize three working point sets $Trial = \emptyset$, $Far = \emptyset$ and $Known = \emptyset$.

2. For each point $p \in G$ do
   - if $p \in C$ then
     - insert $p$ into $Trial$ and set $T(p) = 0$
   - else
     - insert $p$ in to $Far$ and set $T(p) = \infty$

**Main loop**

1. Find point $p \in Trial$ having the smallest value of arrival time $T(p)$,

2. Exclude $p$ from $Trial$ set and include it into $Known$ set,

3. For each immediate neighbor $q \notin Known$ of $p$ do
   - (a) If $q \in Far$ then include it into $Trial$,
3.2. Implicit active contour models

(b) Compute the new arrival time \( T_{i,j}^{\text{new}} \) for point \( q = (i, j) \) according to the following equation:

\[
\frac{1}{s_{i,j}^{-}} \left[ \max(D_{x}^{-}(T_{i,j}), 0)^{2} + \min(D_{x}^{+}(T_{i,j}), 0)^{2} + \max(D_{y}^{-}(T_{i,j}), 0)^{2} + \min(D_{y}^{+}(T_{i,j}), 0)^{2} \right] = \frac{1}{s_{i,j}^{-}},
\]

where \( D_{x}^{\pm} \) and \( D_{y}^{\pm} \) denote the one-sided differences. Equation (3.44) leads to solution of a quadratic equation. The smallest positive solution is considered as \( T_{i,j}^{\text{new}} \).

(c) If \( T_{i,j}^{\text{new}} < T_{i,j} \) then \( T_{i,j} := T_{i,j}^{\text{new}} \).

4. Break main loop and stop the algorithm while \( \text{Trial} \) set is empty.

Equation (3.44) was obtained from (3.43) using upwind finite difference scheme. The algorithm produces numerically stable solution and its correctness was proved in [Set96].

The suitable implementation of operations on \( \text{Trial} \) set is the key for the effective performance of the algorithm. Main loop is evidently performed at most \( n \) times, where \( n \) is the number of grid points \((n_1 \times n_2)\). If \( \text{Trial} \) set is implemented as a min-heap data structure, where finding the minimal node can be performed in constant \( O(1) \) time and inclusion and exclusion of a node in \( O(\log n) \) time, then the whole algorithm works in \( O(n \log n) \) time in the worst case.

3.2.4 Region based active contours

The implicit active contour models that we have described so far rely on the edge stopping function \( g \) which depends on the image gradient magnitude \( |\nabla f| \). Even the most robust geodesic active contour model can detect only objects with edges defined by the gradient. In practice, the size of image gradient is bounded. Moreover, the segmented image can be harmed by the noise and the Gaussian smoothing with big standard deviation \( \sigma \) must be employed in equation (3.28) in order to get good stopping \( g \) function. The consequence of those fact is that the edge stopping function \( g \) is never equal to zero on the edges. Therefore, the interface (curve or surface) may pass through the weak boundary or small gap in the edge, especially for front propagation model or evolution algorithms based on fast marching method.

Further, the edges are not fully reliable for extracting object boundaries. They do not always correspond to object boundaries. This situation can especially occur when segmenting textured objects. The edges can be also weak or blurred. Therefore, it will be useful to take into account also other properties that besides the edges characterize the segmented objects and regions.

Mumford and Shah in [MS89] proposed a variational formulation of image segmentation which take into consideration the interior of the segmented regions as well as the inter region boundaries. Their model simultaneously produces simplified version of segmented image and the set of the dividing edges. The energy functional of Mumford-Shah model is given by

\[
E_{\text{MS}}(u, \Gamma) = \lambda \int_{\Omega} (u(x) - f(x))^{2} d\Gamma + \beta \int_{\Omega - \Gamma} |\nabla u|^{2} d\Gamma + \mu |\Gamma|
\]

where \( \Gamma \) denotes the set of edges that separate the different regions, \( u \) is a smooth approximation of the input image \( f \) in those regions, and \( \lambda \), \( \beta \) and \( \mu \) are positive weighting parameters. The first term in functional (3.45) penalizes the deviations of \( u \) from input image \( f \), the second term penalizes the total variation of \( u \) within each segment of \( u \) and the third term penalizes the edge length \( \Gamma \). The Mumford-Shah variational model looks like a perfect segmentation...
model at first sight, however the way how to minimize it is left open until these days. Therefore several approximations and simplifications of energy functional (3.45) have been studied. Ambrosio and Tortorelli [AT90] proposed regularization of (3.45) where the boundaries have finite width. A simplified version of the Mumford-Shah functional that is called the “cartoon limit” is obtained when setting $\beta = 0$ and restricting the possible minimizers $u$ to be piecewise constant. The remaining energy functional

$$E_{CL}(u, \Gamma) = \lambda \int_{\Omega} (u(x) - f(x))^2 dx + \mu |\Gamma|$$

(3.46)

can be then minimized with a piecewise constant approximation $u$, and a boundary $\Gamma$ that separates the constant regions. Thus, the minimizing image $u$ consist of several regions with different constant gray values.

Chan and Vese developed in [CV01] an implicit active contour model which can be considered as a two phase formulation of cartoon limit (3.46) energy functional. They proposed functional which decomposes the image into foreground and background regions with respect to their mean intensity values and with respect to the length of the foreground boundary. Their model completely neglects the image gradient and that’s why they call it “active contours without edges”. Let $\Omega \in \mathbb{R}^m$ denote the $m$-dimensional image domain and $f : \Omega \to \mathbb{R}$ an initial gray-scale image. Further, let the image domain is split to two regions $\Omega_1$ and $\Omega_2$ with $\Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_1 \cap \Omega_2 = \emptyset$. Let $C$ is an initial given closed interface of codimension $\mathbb{R}^{m-1}$ bounding the foreground region $\Omega_1$. Chan and Vese introduced following energy functional

$$E_{CV}(c_1, c_2, C) = \mu L(C) + \lambda_1 \int_{\Omega_1} (f(x) - c_1)^2 dx + \lambda_2 \int_{\Omega_2} (f(x) - c_2)^2 dx$$

(3.47)

where $\mu \geq 0$, $\lambda_1, \lambda_2 \geq 0$ are given fixed parameters and $L(C)$ denotes the length of $C$. The segmenting contour $C$ which divides the image into two type of regions with mean average intensity $c_1$ and $c_2$ is obtained by minimizing defined functional in such a way. Chan and Vese have proposed to use the level set framework in order to minimize this functional. Let $\phi(x) \geq 0$ in $x \in \Omega_1$ and $\phi(x) < 0$ if $x \in \Omega_2$, then the zero level set determines the region boundary $C$ and (3.47) can be expressed as

$$E_{CV}(c_1, c_2, C) = \int_{\Omega} \left( \mu |\nabla H(\phi(x))| + H(\phi(x))(f(x) - c_1)^2 + (1 - H(\phi(x)))(f(x) - c_2)^2 \right) dx$$

(3.48)

where $H(s)$ denotes the Heaviside function that is $H(s) = -1$ for $s < 0$, $H(s) = 1$ for $s > 0$ and $H(0) = 0.5$. Minimizing (3.48) in the Euler-Lagrange framework leads to the following partial differential equation

$$\partial_t \phi = \delta(\phi) \left[ \mu \nabla \cdot \left( \nabla \phi \over |\nabla \phi| \right) - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2 \right]$$

(3.49)

$$c_1 = \frac{\int_{\Omega} f(x) H(\phi(x)) d\Omega}{\int_{\Omega} H(\phi(x)) d\Omega}, \quad c_2 = \frac{\int_{\Omega} f(x) (1 - H(\phi(x))) d\Omega}{\int_{\Omega} (1 - H(\phi(x))) d\Omega}$$

(3.50)

where $\phi$ is the level set function embedding interface $C$, $H(s)$ now denotes the regularized Heaviside function $\lim_{s\to-\infty} H(s) = 1$, $\lim_{s\to\infty} H(s) = -1$ and $\delta(s) = H'(s)$ is regularized Dirac function. The regularized Heaviside and Dirac functions are introduced due to the numerical approximation. The regularized Heaviside function is often derived from the arc tangent
3.2. Implicit active contour models

Figure 3.14: Example of Segmentation with Chan-Vese (CV) model. The CV model is independent on image-edges and is driven by the regional information. Therefore, the initial contour need not to be completely inside or outside of the objects of interest. The segmentation was run with $\mu = 10, \lambda_1 = \lambda_2 = 1$. (a) Top Left: The contour in time $t = 0$. (b) Top Center: The contour in time $t = 10$. (c) Top Right: The contour in time $t = 14$. (d) Bottom Left: The contour in time $t = 18$. (e) Bottom Center: The contour in time $t = 20$. (f) Bottom Right: The contour in time $t = 25$.

function (see [CV01]). One computes one iteration of equation (3.49) and the equations (3.50) in periodic manner in order to obtain the solution.

Active contour model defined in such a way can segment the objects with weak or ramp edges, it can find interior contours in objects with holes automatically without considering second initial contour. Again, the model allows automatic change of topology, due to the level set implementation. Furthermore, in opposite to previously described models, the initial contour need not surround the objects to be segmented (see Fig. 3.14 for example segmentation). On the other hand, described method is computationally expensive, when using the level set framework described in section 3.2.1.

Numerical approximations

The Chan-Vese model can be discretized using the similar discretization and corresponding operators as for previously described implicit active contour models (see section 3.2). The details will not be presented here again.

Fast implementations There were suggested several fast approximations of the Chan-Vese active contour model. Gibou and Fedkiw in [GF02] proposed approximation which replace the mean curvature motion in the equation (3.49) by preprocessing with nonlinear diffusion filter (Perona-Malik). Inspired by Gibou and Fedkiw, Esedoḡlu and Tsai proposed in [ET06] another fast technique for Chan-Vese model approximation. Their model replaces the mean curvature motion by Gaussian blur and simple thresholding. Another fast converging approximation was proposed in [SC02]. An hybrid algorithm based on Gibou-Fedkiw and Esedoḡlu-Tsai approximations will be proposed and analyzed in following chapter.
3.3 Variational optic flow

In this section we discuss the basic variational optic flow methods and their properties. We describe the ancestor of all variational optic flow method, the Horn-Schunck method first. After that, several variants and upgrades of this method will be described. Finally, we will discuss the basic numerical approximation and we mention a fast implementation of these methods.

Estimation of the motion in image sequences is one of the interesting tasks for the computer vision. Let two consecutive frames of image sequence be given. Optic flow methods compute the displacement vector field which maps all voxels from first frame to their new position in the second frame. Solution of this correspondence problem appears as the basic step of any further motion estimation (e.g. estimation of object motion, camera motion, object tracking, motion based segmentation ...).

Although several kinds of strategies exist for optic flow computation [BFB94], we take only the so-called variational optic flow methods into our considerations. They currently give the best results (in terms of error measures) [PBB06, BW05] and come out from transparent mathematical modeling (the flow field is described by energy functional) and the estimation of the flow field is realized via PDE. Finally, variational optic flow methods produce dense flow fields and are invariant under rotations.

The first prototype of variational optic flow method was proposed by Horn and Schunck in [HS81] almost thirty years ago. We will describe their method first, because even the most sophisticated methods available in these days are still based on the fundamental ideas of Horn-Schunck method. Horn and Schunck used two basic assumptions in order to estimate the optic flow between two frames. The first gray value constancy assumption which assumes that the gray value intensity of the moving objects remains the same. The second used assumption is the homogeneous regularization which assumes that the gray value intensity of the moving objects remains the same. The second used assumption is the homogeneous regularization which assumes that the flow is smooth across the image.

We describe in detail the Horn-Schunck method for optic flow estimation in two dimensional image sequences in the following text. It can be described also for three dimensional sequences, but we choose the first possibility in order to stay consistent with example images and also for the sake of clarity. Let \( \Omega_2 \subset \mathbb{R}^2 \) denote the 3-dimensional spatial-temporal image domain and \( f(x_0, x_1, x_2) : \Omega_3 \rightarrow \mathbb{R} \) a gray-scale image sequence, where \((x_0, x_1, x_2)^T\) is a pixel location within image domain \( \Omega_2 \subset \mathbb{R}^2 \) and \( x_2 \in [0, T] \) denotes time. Moreover, let’s assume that \( \Delta x_2 = 1 \) and \( u = (u_0, u_1, 1)^T \) denotes the unknown flow.

The gray value constancy assumption can be formalized in the following way

\[
 f(x_0 + u_0, x_1 + u_1, x_2 + 1) - f(x_0, x_1, x_2) = 0. \tag{3.51}
\]

The equation (3.51) is nonlinear in \( u_0 \) and \( u_1 \). Horn and Schunck further approximate this equation with its first order Taylor expansion and they obtained the so-called Optic flow constraint (OFC). It should be noted that this approximation is valid only when the displacement between two consecutive frames is rather small (up to several pixels). The optic flow constraint can be formulated as

\[
 f_{x_0} u_0 + f_{x_1} u_1 + f_{x_2} = 0, \tag{3.52}
\]

where \( f_{x_i} \) is partial derivative of \( f \) with respect to \( x_i \). The equation (3.52) with two unknowns has obviously more than one solution. Horn and Schunck assumed only smooth flows and they therefore employed the second assumption – the homogeneous regularization which penalizes the solutions which have large spatial gradient \( \nabla^2 u_i \) where \( i \in 0, 1 \) and \( \nabla^2 \) denotes the
PDE-based image processing

3.3. Variational optic flow

spatial gradient. Thus, the sum $\sum_{i=0}^{1} |\nabla^2 u_i|$ for every voxel should be as small as possible. We get following variational formulation of the problem if we combine these two considerations together:

$$E_{HS}(u) = \int_{\Omega} \left( f_{x_0}u_0 + f_{x_1}u_1 + f_{x_2} \right)^2 + \alpha \sum_{i=0}^{1} |\nabla^2 u_i|^2 dx \quad (3.53)$$

The optimal displacement vector field minimizes energy functional (3.53). Note that the OFC and the homogeneous regularization term are squared in order to penalize negative and positive deviations in the same way. The $\alpha$ parameter has the influence on the smoothness of the solution. The two terms, which represent the two assumptions, form the functional and are often called data and smoothness term.

Following the calculus of variations [GF00], the minimizer of (3.53) is obtained as a solution of the following Euler-Lagrange equations

$$0 = f_{x_0}^2 u_0 + f_{x_0} f_{x_1} u_1 - f_{x_0} f_{x_2} + \alpha \text{div}(\nabla^2 u_0)$$
$$0 = f_{x_0} f_{x_1} u_0 + f_{x_1}^2 u_1 - f_{x_1} f_{x_2} + \alpha \text{div}(\nabla^2 u_1) \quad (3.54)$$

with reflecting Neumann boundary conditions. $\text{div}(x)$ is the divergence operator. Note that the system (3.54) consists of two partial differential equations which are loosely coupled (via the $u_0$ and $u_1$) unknowns. The possible ways how to solve equations (3.54) numerically will be described in the end of this section.

The Horn-Shunck variational method was one of the first so-called global optic flow methods. The main advantage of the global optic flow methods to the local methods is that they produce dense flow fields. The local methods estimate the flow field only from the local neighborhood of each point. Therefore, the flow fields produced by local methods are sparse, because there is no reliable local information for the motion estimation in constant areas of the frames. To the contrary, the flow field in such locations is estimated with the help of the smoothness term when using a global optic flow method. The regularization term interpolates the flow from more reliable surrounding locations. Example results of Horn-Schunck method are in the Fig. 3.15.

The Horn-Schunck method has provided satisfactory results in many practical applications. However, one can immediately notice that the energy functional (3.53) does not model the movements which occur in real world time-lapse image sequences. The real-world objects have obvious borders. Therefore, the displacement fields which describe the object motion in image sequences contain sharp discontinuities more often than the smooth transitions. Further, the real world image sequences can contain larger movements that cannot be handled with simple Horn-Schunck method. Finally, the image brightness in the image sequence can change from one frame to the next frame and particular frames can be corrupted by noise. All phenomenons which we have described in this paragraph occur to some extent in the time-lapse image sequences from live-cell studies. Therefore, we discuss several other variational optical flow methods in the following text.

3.3.1 Methods with image-driven regularization

We describe two modifications of Horn-Schunck (HS) method in this section. Both methods will replace the homogeneous regularization term with a smoothness term which regularizes the flow with respect to the content of the underlying image frames.
Figure 3.15: Example of results of Horn-Schunck variational optic flow method. As an input we take two gray-scale frames of a time-lapse image sequence, which has been artificially generated. However, they simulate the real frames which can be obtained in live-cell studies. The image sequence contains moving HP-1 protein domains (white spots) inside HL-60 cell nucleus. Only the protein domains have moved. The average displacement of the spots was around 2.5 pixels. The displacement field between two consecutive frames was estimated with minimization of the energy functional (3.53), $\alpha = 1000$. (a) Top Left: The first input frame. (b) Top Right: The second input frame. (c) Bottom Left: The visualization of the ground truth flow field. The color denotes the direction of the movement, the color intensity codes the displacement size. (d) Bottom Center: The visualization of the flow field computed with the Horn-Schunck method. Note that the flow field is smooth across the image domain. There are no sharp discontinuities in the computed flow field in opposite to the ground truth. (e) Bottom Right: The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 3.0.
The most visible difference between the flow field computed with the HS method and the ground truth flow field in Fig. 3.15 is the smoothness of the HS flow. This effect is caused by the homogeneous regularization used in the smoothness term of energy functional (3.53). On one hand, the homogeneous regularization perfectly fills-in the flow in the location where the data term has no effect. On the other hand, it interpolates the flow from the neighborhood locations with the same weights and it produces smoothed flow fields as a result. In fact, this data term regularizes the flow in the same way as the Gaussian diffusion filter smoothes an input image. One can notice that the regularization is rather similar to the diffusion. We have described several nonlinear diffusion filters in the first section of this chapter. The key idea of those filters was to restrict the diffusion in location where the large gradients are located in the image. The image driven regularization in optic flow methods uses the same idea.

**Image-driven isotropic regularization** The first described image driven regularization treats the flow in the same way as the PM-filter diffuses the image. It down-weights the smoothness term at locations with large gradient. The energy functional of this method can be formulated as

\[ E_{II}(u) = \int_{\Omega} \left( (f_{x_0}u_0 + f_{x_1}u_1 + f_{x_2})^2 + \alpha g \left( |\nabla_2 f|^2 \right) \right) \sum_{i=0}^{1} |\nabla_2 u_i|^2 \, dx \]  

(3.55)

where \( g \) is a positive decreasing function. The Charbonnier function can be used for this purpose [Bru06].

\[ g(s^2) = \frac{1}{\sqrt{1 + s^2 / \lambda^2}} \]  

(3.56)

The \( \lambda \) has the role of the contrast parameter. The example flow obtained as a minimization of (3.55) is illustrated in the Fig. 3.16a.

**Image-driven anisotropic regularization** Nagel an Enkelmann proposed in [NE86] variational optic flow method with image-driven anisotropic regularization. Their method takes into account not only the presence of the edges in the underlying image, but their direction as well. The flow is smoothed mainly in the direction perpendicular to the image gradient in the presence of edges. Therefore, the flow fields with rather sharp discontinuities can be obtained with this method. The energy functional of Nagel-Enkelmann method is given by

\[ E_{IA}(u) = \int_{\Omega} \left( (f_{x_0}u_0 + f_{x_1}u_1 + f_{x_2})^2 + \alpha \sum_{i=0}^{1} \left( \nabla_2 u_i^T P_{NE}(\nabla_2 f) \nabla_2 u_i \right) \right) \, dx \]  

(3.57)

where \( P_{NE}(\nabla_2 f) \) is projection matrix perpendicular to \( \nabla_2 f \) defined as

\[ P_{NE} = \frac{1}{|\nabla_2 f|^2 + 2\varepsilon_S^2} \begin{pmatrix} f_{x_0}^2 + \varepsilon_S & -f_{x_0}f_{x_1} \\ -f_{x_0}f_{x_1} & f_{x_1}^2 + \varepsilon_S \end{pmatrix} \]  

(3.58)

The parameter \( \varepsilon_S \) is usually set to some small value (0.01) in order to avoid division by zero. The example flow minimizing the Nagel-Enkelmann functional (3.57) is shown in the Fig. 3.16b.
3.3. Variational optic flow

Figure 3.16: Example of results of variational optic flow methods with image driven regularization. The flow fields were computed for the same frame pair as in the Fig. 3.15. (a) Left: The visualization of the flow computed with the image-driven isotropic regularization method (3.55). The parameter settings: $\alpha = 3000.0$, $\lambda = 1.0$. (b) Center: The visualization of the flow computed with the Nagel-Enkelmann method (3.57). The parameter settings: $\alpha = 3000.0$, $\epsilon_s = 0.01$. (c) Right: The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 3.0.

3.3.2 Methods with flow-driven regularization

The optic flow methods with image-driven regularization can sometimes tend to produce artificial discontinuities in the flow field. This is caused by the fact that the smoothing term cannot differentiate the boundaries of moving objects from the edges in the object texture. The optic flow methods with flow driven regularization were proposed [Sch94] in order to overcome this limitation.

We describe the optic flow method with flow-driven isotropic regularization as an example of such methods. The smoothing of the flow field is disabled in the locations where the gradient in the flow arose. While the image-driven regularization restricts the smoothing of the flow always in the same way during the minimization process, the smoothing behavior of the flow driven regularization evolves during the minimization process. The energy functional of the optic flow method with flow-driven isotropic regularization can be formulated as

$$E_{FI}(u) = \int_{\Omega} \left( f_{x0}u_{0} + f_{x1}u_{1} + f_{x2} \right)^{2} + \alpha \Psi_{S} \left( \sum_{i=0}^{1} |\nabla_{2}u_{i}|^{2} \right) dx$$

(3.59)

where

$$\Psi_{S}(s^2) = \sqrt{s^2 + \epsilon_s^2}$$

(3.60)

where $\epsilon_s$ is the regularization parameter. An interesting connection to the diffusion filtering can be remarked at this place again. Similar to the image driven regularizations which have their counterparts in nonlinear diffusion filters, the regularization term (3.60) is similar to the TV diffusion. An example of a flow obtained as a minimization of (3.59) is illustrated in Fig. 3.17.

3.3.3 Methods with robust data term

The last group of variational optic flow methods which we describe in this section was designed with the accent to the noise robustness. This goal is achieved with modified data term.
Figure 3.17: Example of result of variational optic flow methods with flow driven regularization. The flow fields were computed for the same frame pair as in the Fig. 3.15. (a) Left: The visualization of the flow computed with the flow-driven isotropic regularization method (3.59). The parameter settings: $\alpha = 40.0$, $\epsilon_s = 0.01$. (b) Right: The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 3.0.

All methods discussed so far have the same data term which penalizes the deviations from the constancy assumption in the quadratic way. Therefore, the methods which use this penalization are quite sensitive to noise, because they severely penalize the constancy assumption outliers. However, such outliers can be caused by the noise. The noise sensitivity of flow-driven method which uses the quadratic data term is illustrated in Fig. 3.18a. The variational optic flow with non-quadratic penalizer was proposed by Bruhn et al. [BWS05a] in order to overcome this problem. They used a linear penalization function for this purpose and they proved that the such data is more robust to outliers. The energy functional of their method is given by

$$E_{RDFL}(\mathbf{u}) = \int_{\Omega} \Psi_D \left( f_{x_0} u_0 + f_{x_1} u_1 + f_{x_2} \right)^2 + \alpha \Psi_S \left( \sum_{i=0}^{1} \left| \nabla_2 u_i \right|^2 \right) dx$$

(3.61)

where

$$\Psi_D(s^2) = \sqrt{s^2 + \epsilon_D^2}$$
$$\Psi_S(s^2) = \sqrt{s^2 + \epsilon_S^2}$$

(3.62)

Note that both data and smoothness term are regularized with the TV penalization function. The example output of this method is illustrated in Fig. 3.18b (the sequence corrupted by the noise) and in Fig. 3.19a (original sequence).

We have described the methods with image driven and flow driven regularization and a flow driven method with robust data term so far. However, it is also possible to combine the robust data term with the image driven smoothness term. An example of such method is the following combination of robust data term and image-driven anisotropic regularization

$$E_{RDIA}(\mathbf{u}) = \int_{\Omega} \Psi_D \left( f_{x_0} u_0 + f_{x_1} u_1 + f_{x_2} \right)^2 + \alpha \sum_{i=0}^{1} \left( \nabla_2 u_i \right)^T P_{NE} (\nabla_2 f) \nabla_2 u_i dx$$

(3.63)

The example output of this method is depicted in the Fig. 3.19b.
3.3. Variational optic flow

Figure 3.18: The impact of the data term regularization on the noise sensitivity of the optic flow method. The flow fields were computed for the same frame pair as in the Fig. 3.15, but the input frames were corrupted with gaussian noise with $\sigma = 10.0$. (a) **Left:** The visualization of the flow computed with the flow-driven isotropic regularization method (3.59). The parameter settings: $\alpha = 40.0$, $\epsilon_s = 0.01$. (b) **Center:** The visualization of the flow computed with the flow-driven isotropic regularization method with robust data term (3.61). The parameter settings: $\alpha = 20.0$, $\epsilon_s = 0.01$, $\epsilon_D = 0.01$. (c) **Right:** The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 3.0.

Figure 3.19: Example result of variational optic flow methods with robust data term. The flow fields were computed for the same frame pair as in the Fig. 3.15. (a) **Left:** The visualization of the flow computed with the flow-driven isotropic regularization method with robust data term (3.61). The parameter settings: $\alpha = 20.0$, $\epsilon_s = \epsilon_D = 0.01$. (b) **Center:** The visualization of the flow computed with the image-driven anisotropic regularization method with robust data term (3.63). The parameter settings: $\alpha = 300.0$, $\epsilon_s = \epsilon_D = 0.01$. (c) **Right:** The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 3.0.
3.3.4 Numerical approximation

We briefly describe the basic numerical scheme which is suitable for the minimization of the variational optic flow functionals in this section. We describe the scheme on the basic Horn-Schunck method. Finally, we provide reference to the literature where more sophisticated and faster techniques are described.

Each optic flow computation has one preliminary step. The input frames are presmoothed with the Gaussian filter. This step obviously removes some amount of noise from the sequence, but the filtering has another possibly more important reason. The input images are often quantized to discrete levels, e.g. the gray value of particular pixels is in the set \{0, 1, \ldots, 255\}. However the input image sequence \(f\) should be \(f \in C^\infty\) in order to guarantee the well-posedness of the methods. Therefore, the presmoothing is necessary.

We will present the numerical scheme for the Horn-Schunck method in the following text. The basic ideas of this scheme can be used also for the rest of the described methods. Let us remind the Euler Lagrange equation (3.54)

\[
0 = f_x^2 u_0 + f_x f_y u_1 + f_x f_z u_2 - \alpha \text{div}(\nabla_2 u_0)
\]

We describe the Gauss-Seidel iteration schema for this system in the following text. We can expand the linear system of elliptical equations to

\[
0 = (\partial_x f)(\partial_x f)u_0 + (\partial_x f)(\partial_y f)u_1 + (\partial_x f)(\partial_z f)u_2 - \alpha (\partial_x(\partial_x u_0) + \partial_y(\partial_y u_0))
\]

This system is discretized in the following manner. Let \([u_{ij}],_{i,j} \in \{0,1\}\) denote the pixel of the flow vector component at position \((x_0, [x_1])\) where \((x_0) = i \cdot h, [x_1] = j \cdot h\) and \(h\) denotes the spatial grid size. Then the discretized system is given by

\[
0 = (D_x(f)) \left[ u_{i,j} + D_x(f)D_x(f)u_0 + D_x(f)D_y(f)u_1 + D_x(f)D_z(f)u_2 - \alpha (D_x(D_x(u_0)) + D_y(D_y(u_0))) \right]
\]

\[
0 = D_y(f)) \left[ u_{i,j} + D_y(f)D_x(f)u_0 + D_y(f)D_y(f)u_1 + D_y(f)D_z(f)u_2 - \alpha (D_y(D_y(u_0)) + D_z(D_z(u_0))) \right]
\]

where the \(D^\pm\) denotes the central, forward and backward differences (see previous sections for their definition). If we further expand the forward and backward operators which discretize the smoothing term we get the following system

\[
0 = (D_x(f)) \left[ u_{i,j} + D_x(f)D_x(f)u_0 + D_x(f)D_y(f)u_1 + D_x(f)D_z(f)u_2 - \alpha \left( [u_0]_{i-1,j} - 2[u_0]_{i,j} + [u_0]_{i+1,j}) + ([u_0]_{i,j-1} - 2[u_0]_{i,j}) + [u_0]_{i,j+1}) \right] \right)
\]

(3.64)

\[
0 = D_y(f)) \left[ u_{i,j} + D_y(f)D_x(f)u_0 + D_y(f)D_y(f)u_1 + D_y(f)D_z(f)u_2 - \alpha \left( [u_1]_{i-1,j} - 2[u_1]_{i,j} + [u_1]_{i+1,j}) + ([u_1]_{i,j-1} - 2[u_1]_{i,j}) + [u_1]_{i,j+1}) \right] \right)
\]

(3.65)
3.4. PDE in fluorescence microscopy

PDE-based methods, that we have described in previous sections, have been successfully applied in processing of image data from various fields. We will now describe the main and most interesting results obtained by their application in the field of fluorescence microscopy. This short review will be incomplete for sure. We will discuss the results in the rough chronological order as they appear in the literature.

Sarti et al. proposed in [SOLM98, SSLM00] complex method for three dimensional segmentation of cell nuclei in thick tissue section. They used PDE based methods in preprocessing phase for input image enhancement, then they segmented the nuclei with threshold and watershed based algorithm. In the post-processing phase they used several variants of active contour models for smoothing of nuclear surfaces and for further cluster segmentation. The filter used in the article for image smoothing was the following variant of geodesic active contour model

\[ u_t = g \kappa |\nabla u| + \beta \nabla g \cdot \nabla u \]  

(3.68)
where $u$ is the smoothed image and $\beta$ is a constant parameter which control the strength of the attract term. From another point of view, this filter is a variant of mean curvature motion filtering, which reduces the smoothing of image level sets near edges and even attracts the level set towards edges. Sarti et al. showed that this filter produces highly smoothed intra-nuclei regions while preserving the edges on their places and reducing the noise significantly.

The surface of cell nuclei is often boxy due to the properties of thresholding and watershed algorithm. Sarti et al. uses the classic geodesic model to smooth the nuclei surfaces

$$\phi_t = g\kappa|\nabla\phi| + \beta\nabla g \cdot \nabla\phi$$  (3.69)

Here $\phi$ embeds the smoothed surface as its zero level set. The model smooths the nuclei surface very well and even reconstructs them in sub-pixel accuracy.

The Perona-Malik nonlinear diffusion filter was used for preprocessing of two dimensional input data in [TBM+99] and [LJT01]. A spatio-temporal modification of nonlinear diffusion filter was used by Uttenweiler et al. [UWJ03] for enhancing the images from fluorescence time-lapse microscopy. They consider the two dimensional image sequence with rather dense sampling in time axis as a one three-dimensional image. The authors stated that signal-to-noise ratio was significantly improved when using a modification of Edge Enhancing Diffusion. Their new anisotropic nonlinear diffusion filter prefers the intra-region object oriented smoothing with respect to the temporal axis to the inter-region smoothing with respect to the spatial axes.

The cell nuclei are often visualized by staining the whole volume with a fluorescent dye in fluorescence microscopy. The common segmentation methods often fail to segment the nuclei in the case that they are too close to each other. Solorzano et al. [SMLL01] used geodesic active contour model for segmenting cells and nuclei labeled with membrane related protein markers. Only borders of object of interest were visualized by fluorescent labeling of the cell membranes and cell nuclei with specific fluorescent proteins. It could seem easy to segment image defined in such a way. However in the real world the membrane is never visualized completely, high variations in the intensities and even quite large gaps in the boundaries are presented. Solorzano et al. uses Hough transform like algorithm for finding initial interfaces inside each object and then they used the modified geodesic active contour model for localization of the boundary.

$$\phi_t = -g|\phi| + g\kappa|\nabla\phi| + \beta\nabla g \cdot \nabla\phi$$  (3.70)

Here $\epsilon$ is the parameter which controls the strength of the smoothing term. It was experimentally shown that the model is capable to reconstruct the object boundary with properly set parameters. Solorzano et al. conducted all experiments only in two dimensions and they claimed that their method is computationally expensive and not suitable for real-time applications even if using narrowband technique.

Nilsson and Hayden proposed in [NH01, NH05] sophisticated method for segmentation of cell clusters in two dimensional bone marrow samples. Their algorithm locates the boundaries of nuclei in the first phase, then locates the boundaries of the whole cells and in the third phase it assembles the oversegmented cells. The fast marching method is used in the first phase as the preprocessing stage of the watershed algorithm. The weighted distance map is computed moving towards the nuclei centers starting from the coarse segmented nuclei boundaries (obtained by thresholding). As the $g$ stopping function they use

$$g(x) = \frac{u_{nuc}}{u_{nuc}(x)}$$  (3.71)
where $\bar{u}_{\text{nuc}}$ is the mean intensity in the nuclei and $u_{\text{nuc}}(x)$ is the intensity in the pixel $x$. The nuclei boundaries obtained after watershed transform are moved outwards to locate the cell boundaries. Nilsson and Hayden use fast algorithm approximating geodesic active contour model proposed by themselves in [NH03] for this task. This algorithm uses point-wise scheduling of propagation (similar idea is used in [DT02]) using a heap sort queue. The geometric properties of the evolving interface (like normal and curvature) are only roughly approximated in order to enable very fast computations. In the third phase the oversegmented cells are joined by solving combinatorial optimization problem. Nilsson and Hayden claimed accurate segmentation results, while the computation are performed almost in realtime. However, the computations were performed only on relatively small two dimensional image data.

A slight modification of the Chan-Vese active contour model [ZOM05] was used for segmentation and tracking of fluorescently labeled cells by Dufour et al. [DST+05]. The method uses multiple active surfaces coupled by a penalty for overlaps. Additionally, a volume conservation constraint that improves outlining of cell/cell boundaries is incorporated in order to better separate touching cells. The tracking is realized by simple assumption that the segmentation results from one frame are used as a initialization for next frame. Obviously, such technique can get into problems when particular cell is moving too fast and its images do not overlap in neighboring frames.

The optic flow methods were not used very often for processing of the image sequences from fluorescence microscopy. Nevertheless, a combination of variational optic flow and nonlinear diffusion was used for tracking the actin filaments in [RSJU02] and [SS01]. The authors of these works claim that they were able to track the filaments in the very noisy 3D image sequences. The problem of motion estimation and object tracking in image sequences from fluorescence microscopy is reviewed in [Miu05]. The author of this book has described and evaluated the methods and solutions which can be used for this task in these days. The variational optic flow methods are included in the list of possible methods. However, only the Horn-Schunck and methods with image-driven regularization are considered.
Chapter 4

Fast and robust approximation of Chan-Vese model

This chapter presents fast and robust algorithm for minimization of Chan-Vese energy functional. Proposed technique is based on recently published k-Means level set [GF02] and threshold dynamics [ET06] approximations of Chan-Vese functional. The approximation algorithms are combined in order to preserve their individual advantages and avoid their limitations. Hence, the proposed hybrid algorithm is robust and converges reasonably fast to steady state and is suitable for the two-phase segmentation of low contrast biomedical data. A simple numerical scheme for threshold dynamics method is derived in the chapter. Results of the hybrid algorithm that are better than results of both k-Means level set and threshold dynamics methods employed individually are presented. Finally we show that the hybrid algorithm is suitable for the segmentation of image data from fluorescence microscopy.

4.1 Introduction

The interest in variational approaches to biomedical image analysis problems is steadily increasing because they allow computation of the optimal result in a particular sense that is given by an energy functional. The examples of such defined functionals have been presented in Chapter 3. We have described the variational formulation of image segmentation proposed by Chan and Vese [CV01] which allows object detection both with and without gradient at object boundaries. The formulation is robust even in case of presence of noise and is applicable to low contrast data.

There are two main obstacles preventing from practical usage of this method in real biomedical applications. First, the functional minimization can be relatively time consuming for large amount of data (especially for 3D images). Second, the result may depend on initialization especially when processing low contrast data. Therefore, it is not surprising that many papers were published to overcome these problems.

Several authors independently proposed solutions which try to reduce the computational demands by employing semi-implicit numerical schemes for numerical approximation of the mean curvature term in (3.49). Kimmel [Kim03] extended the Chan-Vese model with further geometrical measures and proposed semi-implicit locally one dimensional (LOD) scheme for the solution of his extended model. Jeon et al. [JAPP05] used additive operator splitting (AOS)
Fast and robust approximation of Chan-Vese model

4.2. Fast approximations

We have briefly described the main ideas of region based active contours in previous chapter. The Chan-Vese two-phase model of the Mumford-Shah cartoon limit model (3.46) was explained there in detail. Here we just shortly remark that the Chan-Vese model (3.47) decomposes an input image to foreground and background regions with respect to the mean intensity values of considered regions and with respect to the length of foreground boundary. The minimization of the Chan-Vese functional (3.47) is similar to other implicit active contour models performed in the level set framework. Let’s here remind the Euler-Lagrange equation (3.49) for minimization of (3.47) in the level set framework

\[
\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ \mu \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2 \right]
\]

(4.1)

\[
c_1 = \frac{\int_{\Omega} f(x)H(\phi(x)) \, d\Omega}{\int_{\Omega} H(\phi(x)) \, d\Omega}, \quad c_2 = \frac{\int_{\Omega} f(x)(1 - H(\phi(x))) \, d\Omega}{\int_{\Omega} (1 - H(\phi(x))) \, d\Omega}
\]

(4.2)

scheme to minimize Euler-Lagrange equation (3.49) of the Chan-Vese model. Weickert and Kühne proposed the AOS scheme for the approximation of the geometric front propagation model (3.27) which also contains the mean curvature motion term. Unfortunately, these approaches are still relatively slow because many iterations may be required. The novelty of those methods is only in using the semi-implicit AOS or LOD scheme for numerical computations and therefore they are still sensitive to initialization. On the other hand, these splitting schemes are suitable for parallel computation [JAP05, WZtHRN97].

Song and Chan proposed a fast algorithm for Chan-Vese functional optimization that do not need to solve Euler-Lagrange equation of the underlying variational problem [SC02]. Instead, the energy is directly calculated and checked if it is decreased when the foreground point is changed to background or vice-versa. The algorithm is fast but strongly depends on initialization. The segmented image must be relatively close to the expected two-phase result.

Different approach which can speed up the minimization was proposed by Gibou and Fedkiw [GF02]. Their algorithm consists of three steps (nonlinear diffusion filtering, k-mean clustering and mean curvature motion). It is fast and insensitive to initialization. On the other hand the last step of the algorithm assumes no data fitting term and therefore the result needs not fit data well.

Esedoglu and Tsai published fast threshold dynamics technique for Chan-Vese model approximation [ET04, ET06]. Their algorithm is based on very simple and elegant approximation of the mean curvature motion. Therefore, it is quite fast. Moreover, it produces reasonably accurate results. However, we found out that it strongly depends on initialization when processing real low-contrast image data acquired with the fluorescence microscope.

This chapter describes a hybrid of the approach by Gibou and Fedkiw and the approach by Esedoglu and Tsai. We combine those algorithms to get an algorithm which inherits the advantages of its ancestors and avoids their limitations.

The text is organized as follows. Chan-Vese model is briefly reminded in Section 4.2. The Gibou-Fedkiw and Esedoglu-Tsai approximation algorithms are described in Section 4.2.1 and 4.2.2. The proposed hybrid algorithm and the new numerical scheme for solution of Esedoglu-Tsai algorithm are described in Section 4.3. Section 4.4 is devoted to the experiments and results obtained for synthetic data and real biomedical data.

4.2 Fast approximations

We have briefly described the main ideas of region based active contours in previous chapter. The Chan-Vese two-phase model of the Mumford-Shah cartoon limit model (3.46) was explained there in detail. Here we just shortly remark that the Chan-Vese model (3.47) decomposes an input image to foreground and background regions with respect to the mean intensity values of considered regions and with respect to the length of foreground boundary. The minimization of the Chan-Vese functional (3.47) is similar to other implicit active contour models performed in the level set framework. Let’s here remind the Euler-Lagrange equation (3.49) for minimization of (3.47) in the level set framework

\[
\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ \mu \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2 \right]
\]

(4.1)

\[
c_1 = \frac{\int_{\Omega} f(x)H(\phi(x)) \, d\Omega}{\int_{\Omega} H(\phi(x)) \, d\Omega}, \quad c_2 = \frac{\int_{\Omega} f(x)(1 - H(\phi(x))) \, d\Omega}{\int_{\Omega} (1 - H(\phi(x))) \, d\Omega}
\]

(4.2)
The equation (4.1) is standardly solved with finite difference schemes. However, (4.1) is expensive to solve as was published in many articles (see e.g. [GKRR99], [Kim03], [WK03] [JAPP05], [ET06]), because of the mean curvature motion term $\nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right)$ in its right hand side. The computational demands are still very high even if semi-implicit schemes are used like in [JAPP05].

### 4.2.1 Gibou-Fedkiw algorithm

Gibou and Fedkiw [GF02] proposed following three step hybrid k-means level set algorithm for approximation of Chan-Vese model. In the first step the image $f$ is preprocessed with Perona-Malik nonlinear diffusion filter [PM90, CLMC92]. The diffusion should simplify the input image $f$ and suppress the noise, therefore it plays a role of the length term in (3.47) and substitutes the computational expensive MCM term. The amount of the nonlinear diffusion indirectly approximates the strength of the length term in Chan-Vese model (3.47). In the second step the MCM term in (4.1) is omitted and $\delta_e(\phi) = 1$ is preset. This leads to following simple ordinary differential equation (ODE):

$$\frac{d\phi}{dt} = -\lambda_1(f - c_1)^2 + \lambda_2(f - c_2)^2$$

(4.3)

Gibou and Fedkiw showed that the segmentation results obtained as a steady state solution of this ODE are equivalent to k-Means clustering [Har75] (with two means). Nevertheless, they propose following scheme for solving (4.3):

**Algorithm 1** k-Means level set.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>gray-scale image $f : \Omega \rightarrow \mathbb{R}$</td>
<td>two-phase function $\phi : \Omega \rightarrow {-1, 1}$</td>
</tr>
<tr>
<td>initial two-phase function $\phi_0 : \Omega \rightarrow {-1, 1}$</td>
<td>mean values $c_1, c_2 \in \mathbb{R}$</td>
</tr>
<tr>
<td>$\lambda_1, \lambda_2$ parameters</td>
<td></td>
</tr>
</tbody>
</table>

**Main loop**  Let $R(x)$ denotes the right hand side of (4.3)

1. $n = 0$, $\phi_0$ and image $f$ are given
2. Compute the mean values $c_1$ and $c_2$ (using $f$ and $\phi_n$)
3. For each voxel $x$ do:
   $$\phi_{n+1}(x) = \begin{cases} 
   1 & \text{if } R(x) \geq 0 \\
   -1 & \text{otherwise}
   \end{cases}$$
4. if $\phi_{n+1}$ does not converge to steady state, set $n = n + 1$ and go to 2.

Finally, the boundary between two phases is evolved by mean curvature motion for a particular amount of time in order to smooth the segmentation results in the third step.

Gibou and Fedkiw showed experimentally on several examples that their model gives similar segmentation results to the full Chan-Vese model. They also showed that their model converges quickly to steady state only in several iterations (the difference in computational efficiency is in orders of magnitude in comparison with original Chan-Vese algorithm).
4.2. Fast approximations

4.2.2 Esedoglu-Tsai algorithm

Esedoglu and Tsai proposed [ET04] and recently published [ET06] fast threshold dynamics technique for Chan-Vese model approximation. Their model is motivated by work of Merriman et al.[MBO92] and uses linear diffusion and thresholding for approximation of MCM term in (4.1). We first describe the basic idea of Merriman model for mean curvature motion approximation and then we describe the Esedoglu-Tsai scheme.

Merriman, Bence and Osher [MBO92, MBO94] introduced a simple algorithm for approximation of motion by mean curvature. They investigated following diffusion-reaction equation

\[ 0_t = 2\varepsilon \Delta \phi - \frac{1}{\varepsilon} W'(\phi) \] (4.4)

with initial condition given by

\[ \phi_0(x) \in \{0, 1\} \]

where \( \varepsilon > 0 \) is small and \( W(s): \mathbb{R} \to \mathbb{R} \) is a so-called double well potential with two equidepth wells at 0 and 1. A simple choice for \( W(s) \) is \( W(s) = s^2(1-s)^2 \). It has been well known that as \( \varepsilon \to 0^+ \) the solutions of equation (4.4) approximates motion by mean curvature (3.26) of the evolving interface that separates 0 and 1 phase of the solutions. Merriman et al. proposed an elegant scheme for solution of (4.4). They split its solution into two parts. Assume that we have an iteration step \( \phi^n \) and then first solve the linear diffusion equation

\[ 0_t = 2\varepsilon \Delta \psi \] (4.5)

for some time and then solve the reaction equation

\[ 0_t = -\frac{1}{\varepsilon} W'(\phi) \] (4.6)

until the steady state is reached and finally assign the result to \( \phi^{n+1} \) and repeat the splitting. Merriman et al. found out that the reaction step can be surprisingly approximated with the simple thresholding with the threshold equal to 0.5. Consequently, the motion of the interface under mean curvature (3.26) can be approximated with repeated Gaussian smoothing and thresholding. An example of approximation of the mean curvature motion via Merriman et al. scheme is depicted in Fig. 4.1.

Esedoglu and Tsai proposed the following diffusive interface approximation of (3.47):

\[ E_{ET}(c_1, c_2, \phi) = \int_{\Omega} |\nabla \phi|^2 + \frac{1}{\varepsilon} W(\phi) + \lambda \left[ \phi^2(c_1 - f)^2 + (1 - \phi)^2(c_2 - f)^2 \right] dx \] (4.7)

where \( f: \Omega \to \{0, 1\} \) is an initial gray-scale image, \( \phi \) denotes the piecewise constant level set function with the values from the set \( \{0, 1\} \), \( \lambda \geq 0 \), \( \varepsilon \in (0, 1) \) are given parameters and \( W(\varepsilon): \mathbb{R} \to \mathbb{R} \) is the double well potential. They showed that (4.7) converges to (3.47) as \( \varepsilon \to 0^+ \). Note that the voxel intensities of the initial image \( f \) are restricted to closed interval \( \{0, 1\} \). Esedoglu and Tsai derived the gradient descent PDE from (4.7) with respect to \( \phi \) (for detailed derivation see [ET04, ET06]).

\[ 0_t = 2\varepsilon \Delta \phi - \frac{1}{\varepsilon} W'(\phi) - 2\lambda \left[ \phi^2(c_1 - f)^2 + (\phi - 1)^2(c_2 - f)^2 \right] \] (4.8)

Consecutively, they suggested following threshold dynamics scheme for minimization of (4.7) similar to the work of Merriman et al. [MBO92, MBO94]:

58
Fast and robust approximation of Chan-Vese model

4.2. Fast approximations

Algorithm 2  Threshold dynamics.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>gray-scale image $f : \Omega \to {0,1}$</td>
<td>two-phase function $\phi : \Omega \to {0,1}$</td>
</tr>
<tr>
<td>initial two-phase function $\phi_0 : \Omega \to {0,1}$</td>
<td>mean values $c_1, c_2 \in {0,1}$</td>
</tr>
<tr>
<td>$\lambda, \delta t, \tau$ parameters</td>
<td></td>
</tr>
</tbody>
</table>

Main loop  Let $n = 0$ and $\phi_0$ is given initial piecewise constant level set function.

1. Compute the $c_1$ and $c_2$ constants according to $\phi_n$ and $f$.

2. Compute the solution $\psi$ of the following partial differential equation in some fixed time $\delta t$, with the initial condition $\psi_0 = \phi_n$ and with time step equal to $\tau$.

   $$ \frac{\partial \psi}{\partial t} = \Delta \psi - \frac{\lambda}{\sqrt{\pi \delta t}} (\psi (c_1 - f)^2 + (\psi - 1) (c_2 - f)^2) $$  \hspace{1cm} (4.9)

3. Set

   $$ \phi_{n+1}(x) = \begin{cases} 
   0 & \text{if } \psi(x) \leq \frac{1}{2} \\
   1 & \text{otherwise}
   \end{cases} $$

4. If $\phi_{n+1}$ does not converge, set $n = n + 1$ and go to 1.

Esedoğlu-Tsai claimed that their model converges in a few tens iterations and approximates the original Chan-Vese model very accurately. Note that there is only one $\lambda$ parameter which affects the data fitting term.
4.3 Hybrid algorithm

Both Gibou-Fedkiw (GF) and Esedoglu-Tsai (ET) approximations are very fast in comparison to original Chan-Vese (CV) algorithm. On the other hand, they both have some advantages and limitations. We describe typical properties of GF and ET algorithms and propose a hybrid algorithm for CV functional approximation in this section. Our algorithm has advantages of previously mentioned approximation techniques and on the contrary it avoids their limitations.

The main advantage of the GF algorithm is that it converges to the correct segmentation from almost any initial two-phase partitioning already after the first two steps. On the other hand, there is no data fitting term in the third smoothing step, which is performed via mean curvature motion. Therefore, the boundary between two phases can be inaccurate, if one wants to obtain smoother results. Note that one should not omit the third step, because the segmentation results after the first two steps can be jagged because of noise when processing real low-contrast and noisy biomedical data.

ET algorithm has one major advantage over GF algorithm. The whole segmentation is done in one single iterative process and it produces accurate and smooth boundaries simultaneously. On the contrary, we observed that the speed and sometimes even the results of ET algorithm quite strongly depend on the initial partitioning, especially when processing low-contrast biomedical data. This limitation is rather inconvenient in situations where some type of automatic initialization is used. We give examples of above described properties in Section 4.4.

We propose a hybrid algorithm, which is based on previous argumentation. We combine the GF and ET algorithms in the following way. We perform first two steps of GF algorithm and then we replace its third step with the full ET algorithm. It is obvious that the GF algorithm provides an excellent initialization for the ET algorithm. On the other hand, the ET algorithm smoothes the initial boundary with respect to the underlying image data. We describe our hybrid algorithm more formally:

**Algorithm 3** Hybrid k-Means, threshold dynamics.

**Input**
- gray-scale image \( f : \Omega \to (0,255) \)
- initial two-phase function \( \phi_0 : \Omega \to \{-1,1\} \)
- \( \lambda, \delta t, \tau \) parameters

**Output**
- two-phase function \( \phi : \Omega \to \{0,1\} \)
- mean values \( c_1, c_2 \in (0,1) \)

1. Perform a nonlinear diffusion filtering with an appropriate filter. The image \( f \) is transformed to \( f' \) by this filtering.
2. Run the Algorithm 1 with parameters \( (f', \phi_0, \lambda_1 = \lambda_2 = 1) \). Save the output to \( \phi_{km} \).
3. Map the values of the input image \( f \) to closed interval \( (0,1) \), convert \( \phi_{km} \) values of \((-1)-\)phase to 0.
4. Run the Algorithm 2 with parameters \( (f, \phi_{km}, \lambda, \delta t, \tau) \). Save the output to \( \phi_{td}, c_1, c_2 \).
5. Put \( \phi_{td}, c_1, c_2 \) to output.
4.3.1 Implementation details

We use either the Balanced-Backward-Forward (BFB), Total Variation (TV) diffusion filter (3.14) or the regularized version of Perona-Malik filter (3.11) in the first step of our hybrid algorithm. We approximate these filters with implicit additive operator splitting (AOS) scheme (see Section 3.1.6). Therefore, we can compute only several iterations (usually not more than ten) with larger time steps.

Further, we proposed a locally one dimensional (LOD) implicit scheme [MM05] for solving (4.9), which is simple and straightforward to implement. We were motivated by the fact that (4.9) is slightly modified linear diffusion equation and that there exist successful implicit LOD and ADI (alternating direction implicit) schemes for the basic linear diffusion equation (see [WtHRV98] and [MM05] for detailed description).

Esedoglu and Tsai rewrote (4.9) to the following form:

$$\partial_t \psi = \Delta \psi - A(x) \psi + B(x),$$

where

$$A(x) = \lambda [(c_1 - f(x))^2 + (c_2 - f(x))^2]$$
$$B(x) = \lambda (c_2 - f(x))^2$$
$$\lambda = \frac{\lambda_0}{\sqrt{\pi \delta t}},$$

(4.11)

They proposed following unconditionally stable numerical scheme with forward Euler discretization in time and implicit discretization in space

$$\frac{\psi^{n+1} - \psi^n}{\tau} = \Delta \psi^{n+1} - C \psi^{n+1} + (C - A(x)) \psi^n + B(x),$$

(4.12)

with reflecting boundary conditions

$$\partial_n \psi = 0 \quad \text{on} \quad \delta \Omega,$$

where $n$ denotes the outer normal to the image boundary $\delta \Omega$ and $C$ is a constant which should be chosen large enough compared to $A(x)$. We set

$$C \approx \max_{x \in \delta \Omega} (A(x))$$

(4.13)

in our computations. One can calculate the $C$ constant either only once after the initialization of the computation or always after estimating the $c_1$ and $c_2$ means in the Step 1 of Algorithm 2. We observed that the interface can stuck when the $C$ constant is set too large and the means $c_1$ and $c_2$ are almost equal (i.e. the dividing interface is far from the two-phase segmentation). Therefore, we found out that it is suitable to periodically recompute the $C$ constant according to (4.13) because the right hand side of (4.13) is decreasing during the computations as the dividing interface converges to the desired solution. We recomputed the $C$ constant after each 10 iterations in our experiments, because we found out that it is not changing rapidly. Further, we discretized the Laplacian of $\psi$ in (4.12) with standard five point stencil (seven points in three dimensions). The Laplacian of $\psi$ can be written in three dimensions as

$$\Delta \psi = \partial_x (\partial_x \psi(x)) + \partial_y (\partial_y \psi(x)) + \partial_z (\partial_z \psi(x))$$

(4.14)
where spatial derivatives are approximated with one sided difference operators (see Section 3.1.6 for their definition)
\[
\begin{align*}
\partial_x(\partial_x \psi(x)) &\approx D_x^+ (D_x^+ (\psi_{i,j,k}^{n+1})) \\
\partial_y(\partial_y \psi(x)) &\approx D_y^+ (D_y^+ (\psi_{i,j,k}^{n+1})) \\
\partial_z(\partial_z \psi(x)) &\approx D_z^+ (D_z^+ (\psi_{i,j,k}^{n+1}))
\end{align*}
\]

Consequently the discretized Laplacian of \( \psi \) can be written as
\[
\Delta \psi_{i,j,k}^{n+1} \approx D_x^+ (D_x^+ (\psi_{i,j,k}^{n+1})) + D_y^+ (D_y^+ (\psi_{i,j,k}^{n+1})) + D_z^+ (D_z^+ (\psi_{i,j,k}^{n+1}))
\]

and equation (4.12) for particular voxel at position \((i, j, k)\) is discretized by
\[
(1 + \tau C)\psi_{i,j,k}^{n+1} - \tau \Delta \psi_{i,j,k}^{n+1} = [1 + \tau (C - A(x))]\psi_{i,j,k}^n + \tau B(x)
\]

It is obvious that equation (4.16) cannot be solved directly for particular voxel due to the implicit discretization of the Laplacian of \( \psi \). Instead, the linear system should be solved in each iteration. If we consider a particular ordering of voxels so that they form a column vector, the implicit scheme (4.16) can be written in matrix notation as
\[
(dI - M^A)\psi_{i,j,k}^{n+1} = [1 + \tau (C - A(x))]\psi_{i,j,k}^n + \tau B(x),
\]

where \( d = 1 + \tau C, I \) is a unit matrix and the Laplacian coefficients are included in \( M^A \) matrix. Unfortunately, system defined in such a way cannot be solved directly, because the matrix on the left hand side has large bandwidth (this is similar to numerical approximation of nonlinear diffusion see section 3.1.6). Note, that the \( M^A \) is constructed from contributions in each spatial axis i.e.

\[
M^A = \sum_{l=1}^{m} M_{i}^A
\]

where \( m \) is number of dimensions and the matrix \( M_{i}^A \) corresponds to derivatives along the \( l \)-th coordinate axis. In order to derive suitable LOD scheme we can approximate the matrix \((dI - M^A)\) as
\[
(dI - M^A) \approx (dI - \sum_{l=1}^{m} M_{i}^A) \approx \left[ \prod_{l=1}^{m-1} \left( I - \frac{1}{d} M_{i}^A \right) \right] (dI - M_{m}^A),
\]

It holds that
\[
\left[ \prod_{l=1}^{m-1} \left( I - \frac{1}{d} M_{i}^A \right) \right] (dI - M_{m}^A) = (dI - \sum_{l=1}^{m} M_{m}^A + O(\tau^2))
\]

Therefore, we can finally propose a first order multiplicative splitting scheme for solving (4.17)
\[
\psi_{i,j,k}^{n+1} = (dI - M_{m}^A)^{-1} \left( \prod_{l=m-1}^{1} \left( I - \frac{1}{d} M_{i}^A \right)^{-1} \right) ([1 + \tau (C - A(x))]\psi_{i,j,k}^n + \tau B(x))
\]

The linear system (4.20) can be decomposed into \( k \) (\( k \) is number of rows along the \( l \)-th axis) tridiagonal systems with the same system matrix \([\text{WtHRV98}]\) in step \( l \). This systems can be solved effectively with well-known Thomas algorithm \([\text{MM05}]\).
4.4 Results

In this section, we highlight and illustrate the limits of GF and ET algorithms, which arise while processing low contrast and noisy image data. Furthermore, we show that our hybrid algorithm gives better results than both mentioned methods. We present the results on both two and three dimensional data of synthetic as well as real image data from fluorescence microscopy. Finally, we discuss the computational time demands and the setting of parameters.

Fig. 4.3 shows an experiment with a synthetic 2D image. The input image 4.2 was created in the following way: the gray level intensity of the background was set to 100, while the gray level intensity of the foreground objects was set to 120. Next, the image was corrupted by Gaussian noise with standard deviation \( \sigma = 20 \). We took a regular grid of circles as an initial partitioning (see Fig 4.3a,b). This way of initialization is often used in real applications because it can be done automatically without user interaction.

Figure 4.3c,d shows contours obtained by the first two steps of GF algorithm. We preprocess the image with 15 iterations of BFB nonlinear diffusion filter with time step set to 4.0. The segmentation results in Fig. 4.3c,d were computed with the Algorithm 1. We can see that the boundary is quite jagged and there are some noise artifacts as well.

Apparently, the results could be better after application of the third MCM smoothing step of the GF algorithm. On the other hand, there is no data fitting term in the third MCM step and the results can easily be over-smoothed (see Fig. 4.3e,f). It should be also noted that the noise artifact can be suppressed much better by the diffusion in the the first preprocessing step. However, it is not easy to determine the exact amount of diffusion and the nonlinear diffusion process is still quite time demanding while processing large 3D image data.

Figure 4.3g,h shows contours obtained by the ET algorithm with \( A = 50, \delta t = 12.0 \) and \( \tau = 2.0 \). We find out that the parameters need to be fine-tuned in order to get these nice results. If the parameters are set less carefully, the ET algorithm either sticks near the initial contour or the evolving interface completely disappears (the dependency of ET algorithm on initialization is further discussed in our second experiment). The parameters \( \lambda \) and \( \delta t \) can be set with less care if the constant \( C \) (see Section 4.3.1) is equal to zero. However, this solves the problem only partially because the problems with numerical stability appear and the \( \tau \) parameter must be set small.

Figure 4.2: Artificial low contrast image used as input in the first two experiments. It has the size of 256 \( \times \) 256 pixels. The average gray level intensity of the background is equal to 100. The gray level intensity of the foreground objects is 120. The image is corrupted by Gaussian noise with standard deviation \( \sigma = 20 \).
4.4. Results

Figure 4.3ij shows contours obtained by the hybrid Algorithm 3. The computations were made with $\lambda = 30$, $\delta t = 6.0$ and $\tau = 2.0$, the input image was preprocessed with 5 iterations of BFB nonlinear diffusion filter with time step equal to 3.5. The contour obtained by the GF algorithm was used as an initialization of ET algorithm. The result is then computed much faster and the setting of parameters for ET algorithm is not so critical to obtain reasonable steady state result. We could use only three iterations to compute the solution of (4.9) in contrast to six iterations needed for the ET algorithm running alone.

Now, we describe a correctness measure which describes more precisely the quality of the segmentation results. We compute the sensitivity ($\text{Sens}$) and specificity measures of the segmentation results after each run with respect to a priori known ground truth:

\[
\text{Sens}(f) = \frac{TP}{TP + FN}, \quad \text{Spec}(f) = \frac{TN}{TN + FP},
\]

where $f$ is the two-phase segmentation, $TP$ (true positive) is number of pixels correctly assigned to foreground, $FP$ (false positive) is number of pixels incorrectly assigned to foreground, $TN$ (true negative) and $FN$ (false negative) are defined in a similar way with respect to background. We describe the correctness of the segmentation result with following measure:

\[
\text{Cor}(f) = \text{Sens}(f) \times \text{Spec}(f)
\]

The “visually” correct segmentation results have $0.9 < \text{Cor}(f) \leq 1.0$.

**Sensitivity of ET and hybrid algorithm to initialization** We demonstrate that the ET algorithm depends on the initialization when processing low contrast and noisy images in our second experiment. Further, we show that the hybrid algorithm is not sensitive to initialization. We again try to segment the Figure 4.2. We fixed the parameters $\lambda$, $\delta t$, $\tau$ and run the both algorithms 50 times, each run with different initial partitioning. The partitioning consists of the regular grid of circles with the radius $r$. The radius $r$ was increased by one after each run of the algorithms. We start this parametric study with $r = 10$. The distance between circle centers was set to $3r$. The results of this experiment are shown in Fig. 4.4. Note that the small change in the initial partitioning (i.e. difference between $r = 18$ and $r = 19$) leads to entirely different segmentation results when using ET algorithm. Moreover, the results of ET algorithm are different when we slightly change the remaining parameters ($\lambda$, $\delta t$, $\tau$). It is obvious that it is not easy to find the combination of ET algorithm parameters and initial partitioning which certainly leads to correct segmentation. On the other hand, it should be noted that this limitation of ET algorithm appears only when processing the low contrast data and ET algorithm works well with any initialization when processing high contrast data\(^1\). The proposed hybrid algorithm does not depend on the initialization because the GF algorithm provides good initial guess for the ET algorithm. Moreover, the segmentation results are closer to the ground truth.

**Innsensitivity of hybrid algorithm to parameter settings** We will show that the hybrid algorithm is insensitive to parameter settings in the third experiment. We perform two parametric studies on Fig. 4.2. In the first case we segment the image using the hybrid algorithm with

\(^1\)e.g. the well-known cameraman.tif image
Fast and robust approximation of Chan-Vese model

4.4. Results

Figure 4.3: (a,b) Top Left: The initial mask and initial contour superimposed on the input image. (c,d) Top Right: The mask and the contour computed in the first two steps of GF algorithm. 15 iterations with time step $\tau = 4$ of BFB filter were used for image preprocessing. (e,f) Middle Left: The boundary can easily be over-smoothed by the third step of GF algorithm because there is no data fitting term in this step. (g,h) Middle Right: The mask and the contour computed by the ET algorithm. The parameters were set to $\lambda = 50$, $\delta t = 12.0$ and $\tau = 2.0$ and had to be carefully fine-tuned in order to get this result. The solution was computed by 80 iterations of Algorithm 2. (i,j) Bottom: The mask and the contour computed by the hybrid algorithm. 5 iterations with time step $\tau = 3.5$ of BFB filter were used for preprocessing. The parameters of ET algorithm were set to $\lambda = 30$, $\delta t = 6.0$ and $\tau = 2.0$. The solution was computed in 19 iterations of Algorithm 2. There was no problem with the parameter choice and the results remained almost the same for different settings.
Fast and robust approximation of Chan-Vese model

4.4. Results

Figure 4.4: The segmentation of Fig. 4.2 was performed with ET and hybrid algorithm. Different radius $r$ of initial partitioning circles was used. Note that the correctness $Cor(f)$ of the results obtained by ET algorithm strongly depends on the initial partitioning. On the other hand, the hybrid algorithm produces correct results. The input image was preprocessed with 5 iterations of BFB filter with time step $\tau = 3.5$ in the hybrid algorithm.

incrementally increasing $\lambda$ parameter. The other parameters were fixed ($\delta t = 6.0, \tau = 2.0$). The input image was diffused with 5 iterations of the BFB filter with time step equal to 3.5 in the GF algorithm. It is shown in the Fig. 4.5 that the hybrid algorithm always correctly segmented the image. Moreover, the graph shows that the $\lambda$ parameter controls how accurate the results fitted to the image data are. We fixed the ($\tau = 1.0$ and $\lambda = 25$) parameters in the second case and incrementally increase the $\delta t$ parameter. The other parameters were the same as in the first case. We demonstrate that the $\delta t$ parameter has only a minor influence on the segmentation results which can be oversmoothed when using high values of $\delta t$.

Experiment with real 3D data  Fig. 4.6 shows an experiment with real biomedical data acquired using a confocal microscope. The nucleus of a HL-60 cell is shown in maximal projections to $xy$, $xz$, and $yz$ planes in Fig 4.6a. We took a regular grid of ellipsoids as an initial partitioning (similar to the two dimensional case). The surface in Fig. 4.6b was computed with the first two steps of GF algorithm. We use the BFB filter for preprocessing (10 iterations of AOS scheme with the time step set to 3.5) again. Note that there are almost no noise artifacts, but on the other hand the boundary is still jagged. The ET approximation (see Fig. 4.6c) requires careful fine-tuning of parameters (especially $\delta t$ parameter) as in the 2D example.

Fig. 4.6d shows the result computed by the hybrid algorithm. The parameters of non-linear diffusion preprocessing were used the same as in Fig. 4.6b. The parameters of the ET approximation were set to $\lambda = 50.0$, $\delta t = 6.0$, $\tau = 2.0$. Also in this experiment, a half number of iterations in the third step of Algorithm 2 was needed in comparison to the ET algorithm running alone.
Fast and robust approximation of Chan-Vese model

4.4. Results

Figure 4.5: The segmentation of Fig. 4.2 was performed with hybrid algorithm. (a) Top: the influence of $\lambda$ parameter on the segmentation results. Note that small values of $\lambda$ leads to results which are not fitted well to underlying data and has smooth boundary. (b) Bottom: the influence of $\delta t$ parameter on the segmentation results. The hybrid algorithm with more inner steps in the ET method produces relatively over-smoothed results.

Level of detail We have shown in previous experiment that the hybrid algorithm is suitable for segmentation of images from fluorescence microscopy. We show in the following experiment that one can segment the images of cells or cell nuclei with varying level of detail with the hybrid algorithm. The cell nucleus of fibroblast cell is in the Fig. 4.7a. The chromatin in the nucleus was stained with DAPI. The cell nucleus contains two nucleoli. They form two dark regions inside the nucleus, because they do not contain chromatin and therefore they are not stained with DAPI. One can segment such image with the hybrid algorithm in two ways. An accurate segmentation of the cell nucleus is depicted in the Fig. 4.7b,c. The hybrid algorithm was run with following parameters: the BFB filter was run with 10 iterations with the time step 3.0, the ET algorithm was run with $\lambda = 10, \tau = 2.0, \delta t = 6.0$. Note that the nucleoli are located quite well. In the real application one is often interested in the mask of the whole nucleus including the nucleoli. This type of segmentation can be easily obtained when setting $\lambda$ to lower values (e.g. $\lambda = 1$). The model is still well fitted to the underlying image data, but the length term forces the algorithm to fill the holes in the mask. The resulting segmentation is depicted in the Fig. 4.7d,e.

4.4.1 Discussion

Selection of the diffusion filter The choice of an appropriate nonlinear diffusion filter in the first step of GF algorithm was neglected a little bit in previous text. Actually, the nonlinear diffusion step loses its importance in the proposed solution. Only several iterations of appropriate (CLMC, TV, BFB, ...) filter need to be carried out. We can even omit this preprocessing step completely if the input data are corrupted by the noise moderately. The second step of GF algorithm (algorithm 1) provides sufficient initialization for ET approximation in this situation.
Figure 4.6: (a) Top Left: An image of a HL-60 cell nucleus. The image has size of 160 × 150 × 60 voxels. The maximum projection in xy, xz and yz planes. (b) Top Right: Results computed by the first two steps of GF algorithm. 10 iterations with time step \( \tau = 3.5 \) of BFB filter were used for preprocessing. (c) Bottom Left: The surface produced by the ET algorithm. The parameters were set to \( \lambda = 50, \delta t = 12.0 \) and \( \tau = 2.0 \). The ET algorithm need 70 iterations to converge. (d) Bottom Right: The surface computed by the hybrid algorithm, the parameters of GF algorithm were the same as in (b), The parameters of ET were algorithm were set to \( \lambda = 50, \delta t = 6.0 \) and \( \tau = 2.0 \). The hybrid algorithm needed 27 iterations to converge.

Nevertheless, we suppose that it is generally better to use the filters with unbounded diffusivity functions in the first step of the hybrid algorithm for several reasons. First, the only parameter of the TV or BFB filter to tune is the total diffusion time which has influence on the total amount of diffusion. On the contrary, the user of CLMC filter have to tune the \( \sigma \) regularization parameter and the \( \lambda \) contrast parameter. Second, the principle of diffusion filters with unbounded diffusivity is that they remove the oscillations in the signal. Small scale details (e.g. noise artifacts) are removed prior to large scale objects. Therefore, the edges of important objects remain unaffected for a long time. On the other hand, the nonlinear diffusion filters with bounded diffusivities perform the forward diffusion which can blur the weak edges of large object. Gibou and Fedkiw make use of this fact in order to simulate the length term of the Chan-Vese functional (3.47). However, the length term of the Chan-Vese model is better approximated inside the Esedoglu-Tsai step in the hybrid algorithm. Therefore, it is desirable to start the ET algorithm with a fine detailed segmentation which can be obtained when BFB or TV filter is used in the first step of GF algorithm. Several example initializations obtained with the first two steps of GF algorithm are shown in Fig 4.8.
Fast and robust approximation of Chan-Vese model

4.4. Results

Figure 4.7: The impact of the $\lambda$ parameter of the hybrid algorithm on the segmentation results. (a) Top Left: 2D image of fibroblast cell nucleus. Two nucleoli (dark circles) are located inside the nucleus. The image has size of $430 \times 300$ voxels. The hybrid algorithm was performed with the following parameters: 10 iterations of the BFB filter with time step set to $3.0$, $\lambda = 10$, $\tau = 2.0$ and $\delta t = 0$. (b) Top Center: The resulting contour superimposed onto the input image. (c) Top Right: The binary segmentation result. (d,e) Bottom: The hybrid algorithm was run with same parameters except $\lambda$ which was set $\lambda = 1$. The contour superimposed on the input together with binary mask is presented again.

Remark to object reconstruction  The 3D object reconstructions in Figs. 4.6c,d, 4.9b,c were created with the well-know Marching-Cubes algorithm [LC87]. The object models in the figures are quite smooth, without sharp creases. However, the ET as well as the proposed hybrid algorithm produce binary image as the result of the segmentation. Therefore, the boundary surfaces produced by the Marching-Cubes algorithm should be apparently boxy, but they are not. This is caused by the application of a post-processing step which is not mentioned in the description of the hybrid Algorithm 3. The example of 3D reconstruction from the original segmentation results is depicted in the Fig. 4.9b,d. We performed one more diffusion step of the ET algorithm (second step in Algorithm 2) after getting the segmentation results. The sharp edge between the 0-level and the 1-level of the $\phi$ function is blurred by this step (see Fig. 4.1b). Therefore, the Marching-Cubes algorithm can now produce smoother reconstructed surface. The example 3D reconstruction from the diffused segmentation results is depicted in the Fig. 4.9c,e.

From another point of view, the 3D reconstruction from original segmentation results creates surfaces with voxel accuracy because each voxel is either in or out of the surface. On the contrary, the surfaces with sub-voxel accuracy are obtained from the diffused segmentation results. Finally, note that the same procedure can be applied in two dimensions. Nevertheless, the "pixelization" of the dividing contour is not so apparent in presented figures because the contour is one pixel wide.

Typical parameter settings  We found out that one must set $\lambda \geq 10$, when reasonably accurate and simultaneously smooth boundary is demanded. On the other hand, the $\delta t$ and $\tau$ should
Figure 4.8: The impact of the diffusion filtering on the results of k-Means level set algorithm. (a) **Top:** The real input image to be segmented. The image contains the methylated histons in the cell nucleus labeled with FITC fluorochrome. The image has size of 460 x 390 pixels. **Top Left:** Raw gray scale image. **Top Right:** The same image – inverted. **Second Row:** The image was diffused with particular diffusion filter and segmented with Algorithm 1. The resulting contour was superimposed onto the diffused inverted input image. The images in each column were diffused with different filters, the diffusion time is the same \( t = 50 \). **Columns Left to Right:** CLMC filter \( \sigma = 0.5, \lambda = 3.0 \); TV filter \( p = 1.0 \); Unbounded diffusivity filter \( p = 1.5 \); BFB filter \( p = 2.0 \). **Third to Last Row:** Diffusion time \( t = 100, 300, 999 \). The preprocessing with CLMC filter together with k-Means level set algorithm produce the most smooth resulting contour. On the other hand, the BFB filter preserves sharp edges for a long time and the k-Means filter algorithm produces still quite detailed segmentation.
Fast and robust approximation of Chan-Vese model

4.4. Results

Figure 4.9: The enhancement of 3D object reconstruction. (a) Top: The three dimensional image of fibroblast cell nucleus. The autofocus projections are used in xy, xz and yz cuts. The image has size of 370 x 260 x 40 voxels. (b) Middle Left: Hybrid algorithm produced binary segmentation. 3D reconstruction of the cell from this segmentation is in the figure. Note that is is a little bit “voxelized”. (c) Middle Right: The binary segmentation results were additionally diffused with second step of algorithm 2. The 3D object reconstruction from such modified data is presented here. (d) Bottom Left: Detail of 3D reconstruction from binary segmentation results. (e) Bottom Right: Detail of 3D reconstruction from diffused segmentation results.

be rather set small (i.e. $\delta t \leq 6.0$ and $\tau \leq 2.0$), because we need only to tune the initialization provided by the previous step. In these situations, the proposed hybrid algorithm converges fast to a steady state. The Algorithm 3 needs seldom more than ten iterations of Algorithm 1 and more than 70 iterations of algorithm 2.

Computational demands The run times of hybrid algorithm are comparable with the run times of GF algorithm because the hybrid method replaces several MCM steps by several iterations of the ET algorithm in the third step. On the other hand, we found out that the hybrid algorithm is 2 to 5 times faster than the ET algorithm, when processing real low contrast data. The particular speed-up depends on the parameter settings and on the segmented image itself. The ET algorithm needs many iteration steps (tens to hundreds) to move the initial boundary towards the “correct” boundary. On the other hand, the hybrid algorithm moves the initial
Fast and robust approximation of Chan-Vese model

4.5 Summary

We have presented a fast and robust hybrid algorithm for Chan-Vese based segmentation suitable for low contrast and noisy biomedical data in this chapter. Presented algorithm combines the ideas of the Gibou-Fedkiw (GF) and Esedoḡlu-Tsai (ET) approximations of Chan-Vese functional. We have shown that none of these two algorithms is well suited to the segmentation of noisy low-contrast images, which are common in fluorescence microscopy. However, the hybrid algorithm overcomes its ancestors.

boundary towards the “correct” boundary with the fast GF algorithm and then only fine-tune the results with several iterations of ET algorithm.

The examples of hybrid algorithm run times are shown in Fig. 4.10. We find out that the $\lambda$ parameter has the biggest influence on the run times in our parameter study. The smaller the value of $\lambda$ is the more iterations the ET part of the hybrid algorithm needs to converge. The dependency of the runtime on image size is shown in Fig. 4.11. The testing image with one cell nucleus was sampled to different sizes and the hybrid algorithm was run to steady state with same parameter settings on each of the resampled images. In the Fig. 4.11a,b are shown the runtimes. In the 4.11c,d is depicted that the ET part of the algorithm needed more iterations to reach the steady state for bigger images.

Figure 4.10: The examples of hybrid algorithm run times. We use a common PC workstation (Pentium 4 2.6 GHz, Linux 2.6.x) in our experiments. (a) Top: 2D image, size $256 \times 256$ pixels. The hybrid algorithm was run with the following parameters: The image was preprocessed with 5 iterations of BFB filter, the $\delta t$ was set to 12, $\tau$ to 2.0. The $\delta t$ parameter was set bigger in order to emphasize the influence of $\lambda$ onto the computational times. (b) Bottom: 3D image, size $173 \times 161 \times 59$ voxels. The image was again preprocessed with 5 iterations of BFB filter, the $\delta t$ was set to 12, $\tau$ to 2.0. The graph of the run times was smoothed with Bezier curve in both cases. It is obvious that for increasing size of $\lambda$ the hybrid algorithm needs less iterations in the ET part to converge. It should be noted that the GF part of the algorithm affected the total execution time only by a fraction which was constant (0.2 seconds for 2D image, 4 seconds for 3D image).
The hybrid algorithm uses ET threshold dynamics algorithm in the final step and therefore both algorithms produce equivalent results. However, the computational time of the hybrid algorithm is much shorter because the initial guess for ET algorithm is prepared using fast GF k-Mean clustering algorithm. It is also much easier to set the suitable minimization parameters for our method than for ET algorithm. Therefore, our method is better suited to the real biomedical applications, especially, if the interaction with a user is not desired or possible. We have shown on practical examples that the hybrid algorithm is suitable for the segmentation of the image data from fluorescence microscopy. We have discussed the parameter settings and their influence on resulting segmentation results.

Finally, an effective locally one dimensional scheme was derived for the computation of the Esedoḡlu-Tsai approximation in this chapter. This numerical scheme quite simple to implement and can be parallelized in a natural way in dimension by dimension manner (either on single multiprocessor machine or on a cluster).
Figure 4.11: The impact of image size on hybrid algorithm run times. The experiment was performed on a common PC workstation (Pentium 4 2.6 GHz, Linux 2.6.x). The hybrid algorithm was run with 10 iterations of BFB filter, $\lambda = 10$ (blue color) or $\lambda = 20$ (red color), $\tau = 2.0 \delta t = 6.0$. The testing image with one cell nucleus in its center was sampled to different sizes and the hybrid algorithm was run until the steady state was reached. (a) Top: The experiment performed with 2D image set. (b) Upper Center: The experiment performed with 3D image set. In the last two graphs we show that the ET algorithm need more iterations for reaching steady state for bigger images. The same image sets with different sampling and same parameter settings were used. (c) Lower Center: The dependency depicted on 2D image set. (d) Bottom: The dependency depicted on 3D image set.
Chapter 5

3D interphase chromosome reconstruction

The main aim of this chapter is to propose two methods for the reconstruction of chromosome territories in interphase cell nuclei. The proposed methods make use of two implicit active contour models for segmentation and territory reconstruction and nonlinear diffusion filtering for image preprocessing which were described in previous two chapters.

In the first section we describe the biological background and motivation of this chapter. We define the goal which the methods proposed later on should solve. Then we describe the way, how the input data were acquired and the first common phase of presented algorithms. After that we present two methods for 3D reconstruction of interphase chromosome territories. Finally, the test on synthetic data and results on real image data will be presented.

5.1 Problem assignment

Each person has unique genetic information that determines how his or her body looks and functions. Genetic information is coded by double stranded DNA, spiral-shaped molecule that's found inside the nucleus of each body cell. Specific segments of DNA that contain the instructions for making specific body proteins are called genes. Current estimate is that human DNA carries about 20,000 genes. Genes are found in specific segments along the length of human DNA, neatly packaged within structures called chromosomes. Every human cell contains 46 chromosomes, arranged as 23 pairs, with one member of each pair.

The large chromosomes in eukaryotic nucleus are packaged by histon proteins into chromatin in order to fit the genetic material inside nucleus. The structure of chromatin significantly varies during the cell cycle. It is compacted to the well-known 4-arm structure during the cell division. On the other hand it is in the unrolled and loose form in the interphase nucleus in order to enable the transcription of the genetic information.

Recent evidence has demonstrated that chromosomes in interphase cell nucleus occupy distinct domains, called chromosome territories [CC01, CKZ '93]. Each territory can be considered as a connected, variably-shaped, three-dimensional structure which is mutually exclusive from other territories. Nevertheless, the organization of individual genes inside the chromosome territories is not known so far. Moreover, neither the mutual positioning of chromosome territories is known.
That's why the morphological characteristics and 3D mutual position between two different objects (chromosome territories, individual genes inside chromosome territories) need to be studied by the experts in cell biology. For example the typical position of chromosome pair in the cell nucleus or position of a specific gene (imaged as a bright point or a very small spot) in the chromosome territory are studied. The objects of interest have to be segmented and reconstructed to be able to perform such measurements and analysis. Further, one must analyze a large number of objects in order to get statistically significant results.

If we take all these requirements into account, we can conclude, that reasonably fast, precise and automatic methods for 3D chromosome territories reconstruction are needed nowadays in the field of fluorescent microscopy. We propose and analyze two methods for the segmentation and reconstruction of chromosome territories in this chapter.

5.1.1 Input data

The methods which we further present were studied on the following material. Targets in biological material (HL-60 cells — blood cells) were visualized by fluorescence in situ hybridization (FISH technique). The chromatin of cell nuclei (occupies the whole volume of the nuclei) was stained by DAPI (blue color). The chromosome pair 22 was stained by FITC marker (green color).

The images of visualized targets were acquired using fully automated high-resolution cytometry system in the Centre for Biomedical Image Analysis, Masaryk University Brno [KKEL+01] (Zeiss Axiovert 100S inverted fluorescence microscope equipped with a CARV confocal module based on a Nipkow spinning disc). Specimen was observed through a PlanApochromat 63×/1.4 oil immersion objective.

A stack of 40 2D images (parallel optical sections) was captured with a high-quality digital CCD camera for ten chosen stage positions (fields of view) for each color. The CCD chip of the camera had 1300x1030 pixels; pixel size was 6.7μm. The dynamic range of the camera was 12 bits but only 8-bit integer was used for pixel intensity storage. The axial (z) step between two optical sections was 0.3μm. The lateral (x, y) step is given by the magnification power of the objective and the pixel size of the CCD chip and it was 0.123μm.

Each field of view typically contained tens of cells. For each 3D image also a maximal intensity projection image in the axial direction over all slices, called autofocus (AF) image, was computed (see Fig 5.1).

5.1.2 Common preliminary step

Both proposed methods have common preliminary step. Particular cell nuclei are located in the images the whole field of view first. Small sub-images which contain the cell nuclei are cropped from the large 3D input images covering the whole microscope field of view in the second step.

The particular cell nuclei are located within the big field of view with a simple algorithm for cell nucleus segmentation based on a local thresholding [KKEL+01]. The cell nuclei of HL-60 line have relatively round shape. Therefore the local thresholding can be applied on the two dimensional autofocus chromatin images (blue channel). This greatly reduces the computational demand of the local thresholding.
5.2 Fast marching based 3D reconstruction

The first proposed method uses the nonlinear diffusion and fast marching algorithm. The diffusion filtering is used for the image preprocessing and the fast marching method for the localization of the territories boundary. The main ideas of those methods were described in Chapter 3.

The proposed method consists of three steps. Each input sub-image is enhanced and simultaneously simplified with nonlinear diffusion filter in the first step. The fast marching algorithm is then run on the simplified image data in the second step. The image with arrival times of the “marching” surface is obtained as the result. Finally, the “optimal” arrival time of the dividing surface is estimated. We now describe the three steps of the method in detail.

Image enhancement  Each sub-image of chromosome territories was diffused with nonlinear diffusion filter before the reconstruction step performed by the fast marching algorithm.
From the wide family of nonlinear diffusion filters we used the BFB (balanced forward backward filtering - see Section 3.1.4) [KS02] filtering because of its properties: it removes oscillations, it preserves and even enhances significant edges, it converges rather fast to the simplified versions of input image, and there is no additional contrast parameter for the diffusivity \( g \) function to tune. Note that the third property (no extra contrast parameter) is quite important, because we want the method to be as automatic as possible.

We approximate the BFB filtering with the AOS scheme (briefly described in Section 3.1) although it was not proposed for filters with unbounded diffusivity function originally. Similarly to Brox [Bro05], we have experimentally verified, that for the BFB with regularized diffusivity function (3.15) the approximation with AOS numerical scheme is numerically stable and produces correct results. The only thing, which has the influence on the stability is the regularizing parameter \( \epsilon \) (see eq. 3.15). If it is set too small (i.e. \( \epsilon^2 < 10^{-6} \)), the approximation of the filtering can be unstable for time steps \( \tau > 5 \). Therefore we set the \( \epsilon \) parameter to larger values (0.0001 < \( \epsilon^2 \) < 0.001) so as to ensure the stability of computations for arbitrary large time steps.

We can perform only several iterations of BFB filter in order to get sufficiently diffused and simplified results. We used 10 iterations with \( \tau = 4.0 \) and \( \epsilon^2 = 0.001 \) in our computations usually. The example of BFB filtering of real image data is illustrated in Fig. 5.2.

**Fast marching algorithm** It is a-priori guaranteed that each input sub-image contains just one cell nucleus. The corresponding chromosome territories are somewhere near the center of the sub-image. We use this a-priori given knowledge for the initialization of the reconstruction process. Only the eight corners of the input bounding box were considered as the starting marching surface \( C(0) \) in order to initialize the surface outside of the objects of interest (see Fig. 5.3a). Note, that such initialization can be performed automatically.

We used equation (3.28) as a speed function with \( p = 1 \), the variance \( \sigma \) of the gaussian smoothing in the denominator was set to \( \sigma = 1.2 \). This speed function ensures that as the surface marches through the data, it slows down its evolution in the voxels with high image gradient magnitude and waits there till the “faster” surface which passes through voxels with small gradient magnitude reaches some high gradient magnitude pixel. We point out at this place again, that the marching surface changes naturally the topology during the computation and we take benefits from the advantages of implicit active contour models. We start the evolution with eight seed points (eight independent surfaces – Fig. 5.3a) and we finish the evolution with (optimally) two surfaces (shapes) of chromosome territories (see Fig. 5.3d).

**Optimal arrival time** In previous paragraph we described the initialization and the evolution of the initial interface and the influence of the speed function, which stops the marching surface in the voxels with high gradient magnitude. However, there is one drawback of the fast marching algorithm. There is no explicit representation of the marching surface in some time \( t \) during the computation. The fast marching algorithm computes only the arrival time function \( T(x) \) in which the surface reaches particular voxel. We can consider the arrival time \( T(x) \) as the weighted distance from voxel \( x \) to the nearest point of the initial surface \( C(0) \). We have to determine the particular time \( t \) at which the whole surface closely encircles the objects of interest (the surface \( C(t) \) is the \( t \) level set of the arrival time function \( T \)).
Figure 5.2: The impact of BFB filtering on the real image data. Three-dimensional chromosome images are filtered with three-dimensional BFB filter in the first step of improved fast marching reconstruction. The results and the key properties of the filtering are demonstrated and visualized on one slice of the input data. (a) Top Left: One slice from 40 slices of a typical input sub-image. (c) Bottom Left: The same slice after diffusion with BFB filter (10 iterations with time step $\tau = 4$ and $\epsilon^2 = 0.001$). The colors in (a) and (c) are inverted because of visualization purposes. (b) Top Right: Isometric representation of the slice before the diffusion filtering. (d) Bottom Right: Isometric representation of the slice after filtering. Note, that the diffusion filter reduces the noise, small scale detail, while preserving significant information.

Figure 5.3: Example of fast marching reconstruction: (a) Left: The initial surface $C(0)$ in eight corners of the bounding box. (b) Center Left: The intermediate marching surface $C(5)$ in time $t = 5$. (c) Center Right: The intermediate surface $C(25)$ in time $t = 25$. (d) Right: The final surface $C(t = 54)$, the optimal arrival time $t$ was determined by histogram analysis procedure. Note that the surface changes its topology several times during the computation.
We analyze the histogram of function $T(x)$ in order to determine the optimal arrival time in our method. The one dimensional histogram was constructed from the integer part (floor) of function $T(x)$. Such constructed histogram maps integer values $t$ to the number of grid points which were visited between time $t$ and $t+1$. The histogram was convolved with the one dimensional Gaussian kernel of variance $\sigma = 2.0$ and radius $3\sigma$ in order to smooth the oscillations. The number of visited points characterizes the size of the marching surface $C(t)$ in time $t$. The first derivative of the histogram describes the change of the surface size (or contour length in 2D), the second derivative describes the speed of this change. The aim of the histogram analysis was to find a level where the surface movement was very slow. If the surface movement is very slow, the first derivative of histogram doesn’t vary a lot in time. Therefore, the second derivative is almost zero in this case. The consequence from this reasoning is following: when the marching surface reaches the borders of the objects, the absolute value of the histogram second derivative gets stuck near zero. We now present another property of histogram’s second derivative. The surface size is growing very fast in the beginning as it starts from eight seed points and moves towards the objects of interest. In some time, the surface fully surrounds the objects and it begins to lose its area. The second derivative of histogram has a maximum in this turning point. This is illustrated in Fig. 5.4a. Thus we choose the first level after this peak point, which has the absolute value lower than some predefined constant (see Fig. 5.4b) as the optimal arrival time $t$. The resulting 3D shape $C(t)$ is then visualized on the 3D model, which is computed using the marching cubes algorithm [LC87], see Fig. 5.3d.

## 5.3 Fast Chan-Vese based reconstruction

We describe the Chan-Vese based reconstruction method in this section. The first preliminary step, which cuts out the cell nuclei from the big data stack, remains the same as in the first
method, but the reconstruction step uses the Chan-Vese two phase segmentation model, which was described in previous chapter.

We try to use this technique because of following motivation: The image data of chromosome territories are often of very low contrast. The edge information in these images is sometimes weak, even if we use the image enhancement techniques like the nonlinear diffusion filtering in previous method. In the contrary to the fast marching method and geodesic or geometric active contour model, the Chan-Vese model does not use the gradient of the input image at all. The properties of this model were briefly described in Chapter 3 and a fast algorithm which approximates the Chan-Vese model was presented in previous chapter. Here we just recall that the Chan-Vese model tries to divide the voxels of the input image to two groups (foreground and background), where each voxel should have similar gray intensity value as the mean intensity of his group. The Chan-Vese model is noise insensitive, because there is a length restriction onto a dividing contour, and therefore it takes into account also the geometry of the reconstructed objects. We suppose this segmentation model to be suitable for the reconstruction of chromosome territories due to its properties.

The main drawback of Chan-Vese segmentation model is its computational complexity. Fortunately, we overcome this problem by using the fast algorithm which we proposed in previous chapter.

Method overview The segmentation of the input sub-image with chromosome territories was performed by the three step Hybrid Chan-Vese based algorithm (see Chapter 4). We only slightly modify the halting criterion of the Esedoglu-Tsai algorithm.

The input image is filtered with ten iterations of BFB filter with time step equal to 4.0 in the first step. The filtered image is then segmented with the k-Means level set algorithm in the second step. The initial partitioning of the k-Means algorithm can be almost arbitrary, but the initial foreground region should cover at least some part of object of interest. As the foreground region we propose the regular grid of ellipsoids, which covers the whole image domain (see Fig. 5.5a). The k-Means algorithm produces coarse segmentation results (see Fig. 5.5b) which serve as the input segmentation for the Esedoglu-Tsai (ET) algorithm. Finally, the fine segmentation is obtained by running the ET algorithm to the steady state (see Fig. 5.5d). The iteration process of the ET algorithm was stopped by modified halting criterion. Usually, the iteration process is stopped when the foreground and background regions in step \( n + 1 \) are identical to the regions obtained by the previous step \( n \). We use a weaker form of this steady state criterion. We compute the number of voxels which form the foreground regions after each iteration step \( n \). Then we compute the difference of foreground size between iteration step \( n \) and \( n - 1 \). The iteration process is stopped when the change in foreground size is smaller than a predefined threshold. We stop the iteration process of ET algorithm when the change in foreground size was smaller than 100 voxels in our computations.

5.4 Synthetic image data generator

We describe the generator of three dimensional artificial image data in this section. The synthetic data set used for testing the proposed method was created using this generator.

The synthetic images with an a-priori known ground truth are needed for the evaluation of the proposed methods. Obviously, the ground truth images are not available for the real
3D interphase chromosome reconstruction

5.4. Synthetic image data generator

Figure 5.5: Example of Chan-Vese based reconstruction: (a) Left: The initial artificial division to foreground and background regions (b) Center Left: The intermediate results of the reconstruction – output of k-Means level set algorithm. (c) Center Right: The result of threshold dynamic approximation of Chan-Vese model. (d) Right: Smoothed final results of 3D reconstruction.

images acquired using fluorescence microscope. Therefore, we have developed a generator of synthetic image data [SKM'07], that is able to simulate a process of 3D image data acquisition using the fluorescence microscope. The proposed generator produces the synthetic images together with the ground truth images. Our image generating tool is rather unique, because other existing tools [LSR'05] produce only 2D synthetic images.

Our simulation tool has been originally intended for obtaining artificial three-dimensional images of cell nuclei, especially nuclei of HL-60 cell line. Nevertheless, it can be used for the generation of images with chromosome territories as well. The whole synthetic image generator consists of several image filters and operations ordered in a sequence. Each of the filters represents specific phenomenon which appears when acquiring the image and storing it into the computer hard drive. The following paragraphs are ordered in the way corresponding to the acquisition process, i.e. when reading the text one can easily understand their meaning and track the order of the simulated events.

**Specimen selection** In the very beginning, the shape of the objects which should be simulated is selected. We suppose that the chromosome territories can be initially generalized as two ellipsoids. These two simple geometric objects, which represent the image “foreground” are placed into the empty volume, which represents the image “background” (see Fig. 5.6a). The intensity of both foreground objects is set to the same gray value. The background intensity is set to other value. The basic objects represented by the ellipsoids are too simple and regular. Since the aim is to simulate real objects (chromosome territories) a certain amount of irregularity is required. We have successfully reduced the regularity of foreground objects using the geodesic active contour model (3.32) which was modified to

\[ \partial_t \phi = g |\nabla \phi| + \alpha g \kappa |\nabla \phi| + \beta |\nabla g^T \cdot \nabla \phi| \]  

where \(\alpha\) and \(\beta\) are the given parameters and the other symbols were described in Section 3.2.2. Further, we use a randomly generated Perlin procedural texture [Per85] as a stopping function \(g\). The procedural texture in location \((x, y, z)\) is defined as a sum of several Perlin’s noise functions:

\[ \text{texture}(x, y, z) = \sum_{i=0}^{N-1} \frac{\text{noise}(x, y, z) \cdot b^i}{d^i}, \]

82
where \( b \) parameter controls flickering of the texture, \( a \) parameter is responsible for smoothness, and \( N \) controls whether the result is still coarse (for \( N < 5 \)) or fine enough. We modify such generated texture so that one half of the voxels contained values lower than zero and the other half values greater or equal than zero. This modification ensures that the interface can be deformed in both outer and inner normal direction. The boundaries of the foreground objects were considered as an initial interface. The surface was evolved using the Nilsson and Heyden's approximation algorithm [NH03] for a certain amount of time. The objects after the deformation are depicted in Fig. 5.6b. This image data have served as a ground truth images in the segmentation tests.

**Specimen preparation** The intensity of the signal is not the same within the whole chromosome territory because the stain does not attach homogeneously to the chromosome. We simulate this fact by adding the procedural texture (5.2) to the foreground objects. The illustration of procedural texture addition is depicted in Fig. 5.6c.

**Incomplete stain washing** We have remarked in Section 2.3 that the unattached fluorescent labels are not completely washed out from the specimen sometimes. We took this unpleasant fact into account and tried to simulate it. We used the Perlin procedural texture model again for this purpose. First, we modified the generated procedural texture in the following way. We mapped its intensity values into the interval \((0, 1)\). After that we raised each voxel intensity to preset exponent and multiplied it by predefined constant. This ensures that only several patches have influence on the generated image. Finally, the texture which imitates the wrongly washed staining is added to the background (not to the foreground). The image data after this step is shown in the Fig. 5.6d.

**Optical system** Each optical system can be described by a point spread function (PSF) which is the impulse response of this system determining the amount and the way of input image blur. The PSF can be either empirically measured (real PSF) or estimated (theoretical PSF). We used theoretical PSF which was generated using the image analysis software Huygens\(^1\). The theoretical PSF is usually based on the prior knowledge of the optical system properties (confocal/widefield microscope, optical lens, wavelength of light, \ldots). The generated PSF is subsequently used as a convolution kernel expressing the influence of the optical system to the image passed (see Fig. 5.6e).

**Plane scanning** The image generated so far had the same resolution in lateral (\(x\) and \(y\)) and in axial (\(z\)) directions. The real image data in fluorescence microscopy have often worse resolution in the axial direction. Therefore, the image with isotropic resolution is resampled to the image with anisotropic resolution in this step. The classic Nearest-Neighbor algorithm was used for this purpose. The subsequent images in the pipeline have the desired resolution starting from this point.

**CCD chip properties** The charge-couple device (CCD) detector, commonly used in microscope imaging, introduces an error to the acquired signal. This detector noise is typically

\(^1\)Huygens® Essential – http://www.svi.nl/products/essential/
table 5.1: The real and synthetic image dataset image descriptor comparison. Each column corresponds to the mean of selected measure evaluated over the whole image datasets. The real image dataset contained 25 images, synthetic image dataset contained one hundred images. Notice the similarity in each column.

<table>
<thead>
<tr>
<th></th>
<th>average intensity</th>
<th>central moments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2nd</td>
<td>3rd</td>
</tr>
<tr>
<td>real data</td>
<td>3.84</td>
<td>7.71</td>
</tr>
<tr>
<td>synthetic data</td>
<td>4.01</td>
<td>7.56</td>
</tr>
<tr>
<td></td>
<td>13.89</td>
<td>15.25</td>
</tr>
</tbody>
</table>

modelled as the photon-shot noise with Poisson distribution (see Fig. 5.6f):

$$ h_\lambda(x) = \frac{e^{-\lambda} \lambda^x}{x!} $$

(5.3)

where $x$ is the number of occurrences of photons and $\lambda$ is the expected number of occurrences that occur during the given interval.

A/D converter Similarly to CCD detector, the analogue-digital (A/D) converter produces a certain amount of noise. This type of noise is modelled with additive Gaussian noise controlled by Gaussian distribution (see Fig. 5.6g):

$$ h_{\mu,\sigma}(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} $$

(5.4)

where $\mu$ and $\sigma$ is the mean and the variance of the noise, respectively.

Simulator output The example output of our synthetic image data generator is shown in Fig.5.6g. The real image data is depicted in the Fig. 5.6h in order to enable a visual comparison. The plausibility of the synthetic images can be assessed in many ways. The most common and straightforward way is to visually compare the synthetic and real image data. Three examples of real and synthetic images are in Fig. 5.7. On one hand, this approach is important for coarse estimate when deciding whether the given image is similar to the class of simulated real images. On the other hand, this way of verification is quite cumbersome and surely not exact. Anyway, it is very difficult or nearly impossible to recognize all the image features with a naked eye. For this reason we decided to utilize some well-defined image similarity metrics common in image databases. These are image descriptors [CJ01] from which we selected the intensity histogram and the central moments. We evaluated these measures for 100 synthetic images and for 25 real images. The similarity of real and synthetic data with respect to image descriptors is summarized in Fig. 5.8 and Tab. 5.1.

5.5 Results

We present the segmentation and reconstruction results of the proposed methods in this section. We describe the test on synthetic and real image data. We discuss the influence of parameter settings on the results. We compare the methods to each other. Finally, we present their demands on computation time.
Figure 5.6: Example of the synthetic data generation process. All images are in the autofocus projections. The $xy$, $xz$ and $yz$ planes are shown. (a) **Top Left**: The chromosome territories are represented by ellipsoids in the beginning. The voxel intensities inside the ellipsoids are set to particular value. The background is set to another value. (b) **Top Center**: The initial ellipsoids after the deformation with modified geodesic active contour model. The regularity of the objects is significantly reduced. (c) **Top Right**: The Perlin procedural texture is added to the foreground objects. This step simulates the inhomogeneous staining of the chromosome territories. (d) **Middle Left**: Simulation of the incomplete specimen washing. The fluorescent markers are rarely completely washed out from the specimen. A small amount often remains in improper places. The phenomenon is simulated with the modified procedural texture. (e) **Middle Center**: The image after convolution with theoretical PSF. This step simulates the distortion caused by the optical setup. (f) **Middle Right**: The image is resampled to the desired resolution. The Poisson noise which simulates the photon-shot noise caused by the CCD is added. (g) **Bottom Left**: The final result of the simulation. The Gaussian noise which simulates the noise produced by the A/D converter was added. (h) **Bottom Right**: The real image data. One can compare them with the synthetic image data.
3D interphase chromosome reconstruction

5.5. Results

Figure 5.7: Visual comparison of the real and synthetic image data. All 3D images are depicted in the $xy$, $xz$ and $yz$ autofocus projections. (a) Top: Three real images. (b) Bottom: Three synthetic images.

Figure 5.8: Comparison of intensity histograms of real and synthetic image data. The average frequency and standard deviation is depicted in the histogram. The logarithmic histograms were normalized. (a) Left: Real data. The average histogram was computed from 25 images. (b) Right: Synthetic data. The average histogram was computed from 100 images.
5.5.1 Tests on synthetic data

We describe the tests which were performed on a synthetic data set in this section. We test the accuracy of the fast marching and Chan-Vese based reconstruction method. Further, the parameter settings which lead to the best results with respect to segmentation accuracy will be discussed.

**Synthetic data set** We set up a synthetic data set containing one hundred images for our tests. Each image was created with the synthetic data generator which we have described in previous section. The size, position and gray level intensity of chromosome territories randomly varies in each sub-image in order to generate non-uniform data set. The voxel size and spacing of synthetic images is the same as in the real image data set. The similarity of real and synthetic data can be visually compared in Fig. 5.7, the comparison by image descriptors in summarized in Fig. 5.8 and Tab. 5.1.

**Segmentation accuracy measures** We describe three well-known co-localization measures used in following tests in this paragraph. We use them for comparison of a-priori known binary ground truth images and binary output of the segmentation. We choose overlap coefficient and so-called Manders coefficients [MVK+03] for measurement of segmentation accuracy.

The overlap coefficient can be described by the following equation

\[
r(A, B) = \frac{\sum_{i=0}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=0}^{n} A_i^2 \cdot \sum_{i=0}^{n} B_i^2}}
\]  

where \( A \) and \( B \) denote the images tested for mutual overlap, \( n \) denotes the number of voxels in tested images and \( A_i \) or \( B_i \) is particular voxel of the image. The overlap coefficient returns values between 0 and 1. The greater the value is the better the co-localization of input images is.

The Manders coefficients were proposed in [MVK+03] and are described by following equation

\[
M(A, B) = \frac{\sum_{i=0}^{n} \tilde{A}_i}{\sum_{i=0}^{n} A_i}
\]

where

\[
\tilde{A}_i = \begin{cases} A_i & \text{if } B_i > 0 \\ 0 & \text{otherwise} \end{cases}
\]

and all remaining symbols have the same meaning as in equation (5.5). The equation (5.6) describes the co-localization rate of image \( A \) with respect to \( B \). Therefore, it in fact describes two coefficients, because we can either measure the co-localization of image \( A \) with respect to \( B \) and vice versa co-localization of \( B \) with respect to \( A \). The value range of Manders coefficients is again in interval \([0, 1]\).
Fast marching based method test We tested the segmentation accuracy of fast marching based method in the first test. The fast marching based method except the optimal arrival time estimation was run on each image from the synthetic image data set. The resulting arrival time image was then thresholded with all potential thresholds instead of estimating the optimal arrival time with the proposed algorithm. The three co-localization measures were computed on ground-truth image and segmentation image corresponding to particular threshold. Finally, the average and standard deviation of overlap and Manders coefficients for each threshold were computed. The test description in algorithmic notation is attached bellow

1. For each image from the dataset do:
   (a) Compute the arrival time image $T$ with the fast marching method.
   (b) For each arrival time $t$ do:
      i. Threshold the image $T$ with threshold $t$ and save the results to image $S$.
      ii. Compute and save the coefficients $r(S, GT)$, $M(S, GT)$ and $M(GT, S)$ using the ground truth image $GT$.

2. Compute the average and standard deviation of overlap and Manders coefficients for each arrival time $t$.

The results of the test are summarized in the Fig. 5.9. The graph shows that the three coefficient functions intersect roughly in one point for arrival time equal to 57. The fast marching based method have produced the best results with respect to ground truth for arrival times around this value. The images were under-segmented for smaller values of arrival time and over-segmented for bigger values. Further, the graph shows that the fast marching method was not able to produce results equal to ground truth images. We realized that the results are over-segmented in $xy$ planes and under segmented in the direction of $z$ axis when considering the optimal arrival time. This fact is illustrated in Fig. 5.10.

The test with synthetic data shows us that the fast marching based method is not able to perfectly recover the ground truth and we get a knowledge that the optimal arrival time for the synthetic data should be somewhere between 50 and 60. The second fact helped us with the fine tuning of the algorithm for optimal arrival time estimation. The method has one parameter – the predefined threshold which stops the searching for optimal arrival time (see Section 5.2). We set this threshold to 1.0 when we inspected the segmentation results only visually. The algorithm underestimated the optimal arrival time with this settings (it returns the values around 30). The experiment with ground truth images and co-localization measures shows us that the threshold should be smaller in order to get better estimates (values between 0.1 and 0.2).

Chan-Vese based method test The segmentation accuracy of Chan-Vese based method was measured in the second test. The test was similar to the first test. The segmentation accuracy was measured with respect to $\lambda$ parameter of the Chan-Vese method, because it has the biggest influence on the segmentation results (see tests in Chapter 4). The Chan-Vese based method was run on each image from the synthetic image data set several times with increasing $\lambda$ parameter. The three co-localization measures were computed on ground-truth image and segmentation image corresponding to particular $\lambda$ parameter. Finally, the average and standard
5.5. Results

Figure 5.9: The test of fast marching based method segmentation accuracy. The method was run on one hundred synthetic images. The arrival time image $T$ was computed for each image from data set. The binary segmentation image $S$ and the co-localization measures with respect to ground truth image $GT$ were computed for each particular arrival time for each image $T$. The average values and standard deviations of overlap and Manders coefficients over the whole data set are summarized in the graph. The fast marching method produced the best segmentation results for the arrival times between 50 and 60 (the intersection of three coefficient curves).

Figure 5.10: Visual comparison of the fast marching based method segmentation results and synthetic ground truth. The 3D images are depicted in the $xy$, $xz$ and $yz$ autofocus projections. (a) Left: The segmentation results produced by the fast marching based method. The arrival time was set with respect to the co-localization measures. The chromosome territories are in red color. (b) Center: The corresponding ground truth image. The chromosome territories are in green color. (c) Right: The ground truth images imposed upon the segmentation results. The results are under-segmented in $xy$ planes and over segmented in the direction of $z$ axis.
5.5. Results

Figure 5.11: The test of Chan-Vese based method segmentation accuracy. The method was run on one hundred synthetic images. The Chan-Vese based method was run on each image from the synthetic image data set several times with increasing $\lambda$ parameter. The co-localization measures for each binary segmentation image $S$ and corresponding ground truth image $GT$ were computed. The average values and standard deviations of overlap and Manders coefficients for each $\lambda$ across the whole data set are summarized in the graph. It is clear, that the Chan-Vese method had the best segmentation results for $\lambda = 12$ (the intersection of three coefficient curves).

deviation of overlap and Manders coefficients for each $\lambda$ were computed. The test description in algorithmic notation is attached bellow

1. For each image from the dataset do:
   (a) For $\lambda$ in \{0, 1, 2, ..., 50\} do:
   (b) Compute the segmentation image $S$ with the Chan-Vese based method.
   (c) Compute and save the coefficients $r(S, GT)$, $M(S, GT)$ and $M(GT, S)$ using the ground truth image $GT$.

2. Compute the average and standard deviation of overlap and Manders coefficients for each $\lambda$.

The results of the test are summarized in the Fig. 5.11. The graph shows that the three coefficient functions intersect roughly in one point for $\lambda$ equal to 12. The Chan-Vese based method produced the best results with respect to ground truth images for $\lambda$ around this value. This fact corresponds to our conclusions in Chapter 4. The images were over-segmented for smaller $\lambda$ values and under-segmented for bigger $\lambda$ values. Further, the graph shows that the Chan-Vese based method was not able to produce results equal to ground truth images similar to the fast marching based method. We realized that the results are over-segmented in $xy$ planes and under segmented in the direction of $z$ axis when considering the optimal arrival time. This fact is illustrated in Fig. 5.12.
Figure 5.12: Visual comparison of the Chan-Vese based method segmentation results and synthetic ground truth. The 3D images are depicted in the $xy$, $xz$ and $yz$ autofocus projections. (a) Left: The segmentation results produced by the Chan-Vese based method with $\lambda = 12$. The $\lambda$ value was set with respect to the co-localization measures. The chromosome territories are in red color. (b) Center: The corresponding ground truth image. The chromosome territories are in green color. (c) Right: The ground truth images imposed upon the segmentation results. The results are under-segmented in $xy$ planes and over segmented in the direction of $z$ axis.

Conclusions from synthetic data tests  The results of the first two tests shows us that both proposed methods are comparable with respect to synthetic image data. The highest possible accuracy was roughly the same. We demonstrated that their parameters can be tuned in order to get reasonable results. On the other hand, both methods cannot completely recover the ground truth images and their segmentation results are similar (see Figs. 5.10, 5.12).

5.5.2 Visual test on synthetic and real image data

We visually inspected the performance of proposed methods on synthetic as well as on real image data set in the third test. Both proposed methods, which were described in Section 5, were successfully applied to the real data (section 5.1.1). We ran both methods on each sub image from both synthetic and real datasets and compared their results.

The segmentation and reconstruction results for each sub-image were examined by an expert in the field. The expert had to decide how many chromosome territories were present in the nucleus according to the input image (usually two) and whether the algorithm found accurate boundaries of the territories. The expert could study the input data, superposition of the final model onto the input data (two dimensional contours on each 2D cut) and 3D computer model of chromosome territories.

We have realized, that the reconstruction (based upon the segmentations) results of proposed methods can be divided into several categories:

- The method found the chromosome territories correctly.
- The method located the territories accurately, but it found some small additional objects too.
- The method found only one joined object, instead of two mutual exclusive territories. However, the boundaries of the territories were located accurately (except for the connected parts).
- The segmentation results were not accurate.
5.5. Results

Figure 5.13: Example of the reconstruction: (a) Left: Correctly reconstructed chromosome territories. (b) Center Left: Correctly reconstructed chromosome territories with additional objects. (c) Center Right: Joined reconstructed territories. (d) Right: Inaccurate results – the right most territory is over-segmented. It is wrongly divided to two parts. It should have similar size to the left territory.

We discuss the results of proposed methods in more details according to these four categories now. Further, we distinguish the results on synthetic and real data set. See also Tab. 5.2 for the resume of following discussion.

Correct reconstruction The fast marching based method (Section 5.2) reconstructed the territories without the need of any further processing in 87 of 100 sub-images of synthetic data set. In comparison, the Chan-Vese based method (section 5.3) analyzed 92 of 100 synthetic testing inputs without any problems.

Further, the fast marching based method reconstructed correctly 19 of 25 real images. The Chan-Vese based method analyzed 20 of 25 real testing inputs. An example of a typical final 3D reconstruction of chromosome territories of a cell nucleus is shown in Fig. 5.13a.

Correct reconstruction with additional objects The fast marching based method sometimes found some additional objects together with correctly segmented territories. It should be noted, that the additional objects were always much smaller (in orders of magnitude) than the correctly located chromosome territories in the same image and they can be easily removed in the post-processing phase. The fast marching methods produced reconstructions with additional objects in the case of 4 synthetic images and in the case of 3 real images. The Chan-Vese based reconstruction did not have not these problems, because small objects have big curvature and they quickly disappear during the iteration process. An example of reconstructed territories with additional objects is shown in Fig. 5.13b.

Joined territories Both methods had problems with correct reconstruction of closed or even touching territories. The fast marching based method can't distinguish two territories in 3 synthetic inputs and 1 real input. The Chan-Vese based method had a little bit worse performance. It was not able to distinguish two territories in 5 synthetic and 3 real images. The worse performance of Chan-Vese based method is related with its better performance in previous case. It can efficiently eliminate the small additional objects, but on the other hand it is more susceptible to joining two close objects into one object. The example of joined territories is illustrated in Fig. 5.13c.
Incorrect reconstruction  The fast marching method failed to correctly locate the chromosome territories in 6 synthetic sub-images and in 2 real sub-images. The Chan-Vese based method failed to correctly locate the chromosome territories in 3 cases in the case of synthetic and 2 cases in the case of real data set. The input images were over-segmented in all cases due to the weak signal in data. The example of wrongly segmented and reconstructed territories is illustrated in Fig. 5.13d.

Summary  To sum it up, we can say, that the fast marching based method and the Chan-Vese based method give comparable results (see Tab. 5.2). Both methods have their advantages and disadvantages. The fast marching method sometimes segments some additional object besides the chromosome territories. Both methods cannot distinguish touching chromosome territories, but the Chan-Vese method joined several closed territories in the testing data, which were not joined by the fast marching method. While testing the methods on synthetic data, we have observed that both methods under-segment the territories along the direction of z axis (see Figs. 5.12, 5.10) with respect to the ground truth images. However, this effect was a little bit more apparent on the results of the fast marching method. On the other hand, both methods over-segment the territories in the xy planes, but this effect was more apparent on the Chan-Vese based method results. Finally, there were several images in the synthetic as well as real image data set, which were not correctly segmented by any method. The signal in those images was quite weak (even for the human observer) and the results were over-segmented in all cases.

5.6 Parameter settings

In this section we summarize the parameter settings which have been used for our tests of the proposed methods. We also discuss their influence on the computation time of the methods.
**Fast marching based method**  The nonlinear diffusion filtering is the first step of the fast marching based method pipeline. As we stated before, we use the BFB filter for this task, because it has no contrast parameter. We obtain the best results, while diffusing the input sub-images with 10 iterations of the BFB filter with time step $\tau = 4.0$ and $\epsilon^2 = 0.001$. When we perform slightly more iterations, the significant edges get sometimes diffused. If we perform less iterations the reconstructions have sometimes more additional objects. The spacing between voxels was set $h = 1$ for all axes, while computing the BFB diffusion.

The fast marching algorithm makes up the core of the first method. The voxel spacing $h$ and the variance $\sigma$ of the Gaussian filter are the only parameters, which can be set during the computation of the fast marching algorithm. The voxel spacing $h$ is used in the equation (3.28) for the gradient computations. Fast marching algorithm uses this $h$ spacing parameter too. We find out, that the spacing $h$ should be quite small (we use $h = 0.01$ for all axes in our computations). The speed function does not stop the moving contour, when the $h$ parameter is larger, because the gradient magnitude in the denominator of equation 3.28 is lower when $h$ is too big and vice versa. The $\sigma$ parameter affects smoothing degree of the Gaussian filter in the denominator of (3.28). We set $\sigma = 1.2$ in our computations in order to suppress the possible stair-casing effect which can be incorporated to the image by the BFB filtering (see Section 3.1).

The algorithm for optimal arrival time estimation has two parameters. The first parameter is the $\sigma$ variance of the one dimensional Gaussian smoothing filter. We usually set $\sigma = 2.0$. The second parameter is the threshold for the selection of optimal arrival time with respect to second derivation of surface area. The best estimations of the optimal arrival time were obtained when this threshold was set to 0.1.

**Fast Chan-Vese based reconstruction**  There are several parameters to tune in the fast Chan-Vese based reconstruction algorithm. We again set the time step parameter $\tau$ in BFB the nonlinear diffusion filtering step to four and spacing between voxels to $h = 1$. Ten iterations of the BFB filter were used of the image enhancement.

We set $\lambda_1 = \lambda_2 = 1$ in the k-Means level set computations (GF algorithm – see Chapter 4), because there is no need to favor either the foreground or the background regions. We set the $\lambda$ parameter of the threshold dynamics algorithm (ET algorithm see Chapter 4) equal to 12.0. The method returned the best results with respect to the ground truth images with this settings. The $\tau$ parameter of the ET algorithm was set to 2.0 and we perform 3 inner steps in all computations of the ET algorithm.

### 5.7 Implementation and computation time

All algorithms which we have presented and tested in this Chapter were implemented using the C++ programming language and included into the i3dlib\textsuperscript{2}.

We use common PC workstation (Pentium 4 2.6 GHz, Linux 2.6.x) for all computations and tests presented in this chapter. We now discuss the execution times of particular algorithms. The first common preliminary step needs only about ten seconds per one field of view. The average size of the sub-images in our test database was $96 \times 96 \times 40$ voxels. The fast marching based method needed 1.8 seconds for computation of the BFB filtering and 1.14 seconds for fast

\textsuperscript{2}i3dlib - multi-platform library for manipulating 2D and 3D images. The library is continuously developed in Centre for Biomedical Image Analysis, Faculty of Informatics, Masaryk University Brno, Czech Republic
marching algorithm for one sub-image in average. The execution of Chan-Vese based algorithm took again 1.8 seconds for BFB filtering, 0.1 seconds for the k-Means algorithm and 3.8 seconds for the threshold dynamics algorithm for one sub-image in average. It is evident, that both methods needed approximately the same time for the computations and that the computational demands are not the bottleneck of presented methods.

5.8 Summary

In this chapter we aimed our effort at proposing some new methods for the specific problem of segmentation and 3D reconstruction of interphase chromosome territories. The main contribution of this chapter was the design of new chromosome territory segmentation–reconstruction techniques, which gained from the robustness of PDE-based filtering and implicit active contour models. The parallel goal was to design methods, which could be easily used for automatic image analysis. We proposed two methods to fulfill this task. The *fast marching based method* and *Chan-Vese based method*. Both methods are reasonably fast and non-interactive. We discussed in detail their properties, advantages and disadvantages. Neither method was superior to the other in our tests. It becomes clear from our tests, that the methods are suitable for automatic chromosome territory segmentation and reconstruction, because their parameters can be set at the beginning of computations, they are reasonably fast and the methods do not need any interaction with the user during the reconstruction process. On the other hand, the proposed methods have still their limitations. Specifically, they can’t correctly separate touching chromosome territories and they were not able to correctly segment the input in rare cases.

Second interesting contribution in this chapter was the generator \(^3\) of synthetic three dimensional images. Up to our best knowledge, there exists no similar generator which is able to synthesize *three* dimensional images of objects (cell nuclei, chromosome territories) common in fluorescence microscopy. We suppose, that the tests on synthetic data which were produced with our generator became more reliable. Finally, taking in mind the overall topic of this thesis, there is a nice application of geodesic active model in the synthetic data generator.

\(^3\)It should be noted that the author of this thesis was only a co-author of this generator partly responsible for level set based deformation module and wrong staining module.
Chapter 6

Variational optic flow in live-cell imaging

The main aim of this chapter is to describe and test in a systematic way a motion tracking framework based on advanced variational optic flow method. We show that the proposed relatively simple framework is suitable for the motion tracking in time-lapse sequences from live-cell studies. First, we propose a pseudo-real image sequence generator which provides us the artificial time-lapse sequences together with ground-truth displacement fields. We test the performance of several variational methods for large displacement on the ground truth sequences in the second part of this chapter. Finally, a simple motion tracking framework which uses the tested optic flow methods is described and tested on 2D and 3D, artificial as well as real time-lapse image sequences from live-cell studies.

6.1 Introduction

There is a steadily growing interest in live cell studies in modern cell biology. The progress in staining of living cells together with advances in confocal microscopy devices has allowed detailed studies of the behavior of intracellular components including the structures inside the cell nucleus. The typical number of investigated cells in one study varies from tens to hundreds because of statistical significance of the results. One gets time-lapse series of three or two dimensional images as an output from the microscope. It is very inconvenient and annoying to analyze such data sets by hand (especially for 3D series). Moreover, there is no guarantee on the accuracy of the results. Therefore, there is a natural demand for computer vision methods which can help with the analysis of these time-lapse image series. Estimation or correction of global as well as local motion belongs to main tasks in this field. The suitability of the state-of-the-art optic flow methods for estimation of local and combined local-global motion will be studied in this chapter.

We remark the main aspects and features of live-cell imaging and the typical properties of the image data obtained from these studies in the following two paragraphs. The details were described earlier in the Chapter 2. The live-cell studies are mainly performed using the confocal spinning disc microscopes in these days. These microscopes provide far better 3D image data (less blurred) than the classical wide-field microscopes. The main disadvantage of confocal
microscopes is their lower light throughput. This causes larger exposure times as compared to
the wide-field mode.

Transparent biological material is visualized with fluorescent proteins in live-cell imaging. Living
specimen usually does not contain fluorescent proteins. Therefore, the living cells are forced to
produce those proteins in the specimen preparation phase [CTE 94]. The image of the living cells in
the specimen on the microscope stage is acquired periodically. The cells can move or change their
internal structure in the meantime. The interval between two consecutive acquisitions varies in the
range from fractions of second up to tens of minutes. It would be convenient to acquire snapshots
frequently in order to have only small changes between two consecutive frames. But, the interval
length cannot be arbitrary small mainly because of photo-toxicity (the living specimen is harmed
by the light) and photo-bleaching (the intensity of fluorescent markers fades while being exposed
to the light). However, it is usually possible to find a reasonable compromise between those
restrictions and adjust the image acquisition so that the displacement of objects between two
consecutive snapshots is not more than ten pixels.

There are two types of tasks to be solved in this field. First, the global movement of objects
like cell or cell nucleus should be corrected before subsequent analysis of an intracellular
movement. This goal is often achieved using common rigid registration methods [ZF03]. A fast
3D point based registration method [MMKD06] was recently proposed for the global alignment
of cells.

The second task is to estimate local changes inside the objects. This task is more complex.
The objects inside the cells or nuclei can move in different directions. One object can split
into two or more objects and vice versa. Moreover, an object can appear or disappear during
the experiment. Therefore, this task requires computation of dense motion field between two
consecutive snapshots. Manders et. al. had used block-matching (BM3D) algorithm [dLvL02]
for this purpose in their study of chromatin dynamics during the assembly of interphase nuclei
[MVK+03]. Their BM3D algorithm is rather slow. It is similar to basic optic flow methods but
it does not comprise any smoothness term.

The main goal of this chapter is to describe and test a motion tracking framework suitable
for tracking intracellular objects which perform large local movements inside the cell or cell
nucleus during the live-cell study. The framework will be able to process 2D as well as 3D
image sequences from live-cell studies. Further, we have to do some additional work in order
to be able to fulfill this main goal.

Therefore, we describe the pseudo-real image sequence generator in the first section. The
proposed generator allows for the evaluation of optic flow computing methods. The primary
aim is to create a pair of new grayscale images similar to the given real-world image (e.g.
Fig. 6.3a) together with appropriate flow field. The generator is flexible: it can handle global
cell motion together with independent local motions of selected intracellular structures which
is a phenomenon often observed in the field of fluorescence microscopy. The generator is
accurate: the created images perfectly resemble the real-world images and the created flow
field describes the movements that are displayed in the image data. It should be noted that the
ground truth image generator was proposed in collaboration with Vladimir Ulman, who made
the major part of work.

We study the latest optic flow methods [Bru06] for the estimation of intracellular movement
in the second section of this chapter. Up to our best knowledge, nobody investigated the
application of these state-of-the-art methods in live-cell imaging. The simple ancestors of
these methods, which can reliably estimate small motion (around one pixel), were successfully used for the analysis of protein movements in 2D live cell sequences [Miu05]. The examined methods are able to reliably estimate the flow larger than one pixel. They can produce piecewise smooth flow fields which preserve the discontinuities in the flow on object boundaries. These properties are needed for the estimation of local divergent motion which occurs in live-cell imaging. We have extended state-of-the-art optic flow methods into three dimensions. Especially, we focused on 3D extension of recently published optic flow methods for large displacements [PBB+06]. We tested these methods on synthetic as well as real data and compared their behavior and performance. Our experiments identify the optic flow methods which can be used in live cell imaging.

We propose and test a simple motion tracking framework, which uses the variational optic flow methods for motion estimation, in the third section. We show on examples that this simple framework can provide very good results when using the accurate advanced variational optic flow methods for motion estimation.

Finally, we want to remark that all tests, examples and experiments which are described in this chapter are performed on time-lapse sequences which study the dynamic of HP1 protein domains in the HL-60 cell nucleus. However, we feel confident to claim that our results and conclusions presented in this chapter are applicable at least to similar data obtained from other live-cell studies.

### 6.2 Pseudo-real image sequence generator

We describe the framework for generation of pseudo-real image sequences that are similar to the real ones obtained from live-cell studies in this section. First, we describe how the framework produces the pseudo-real image sequence together with the ground truth displacement fields. Second, we analyze its behavior and present few sample artificial image sequences which imitate the real image sequences from live-cell studies.

#### 6.2.1 Motivation

Basically, there are just two possible approaches to obtain image sequences with ground-truth flow fields. One may inspect the real data and manually determine the flow field. Despite the bias [WHH+03] and possible errors, this usually leads to a tedious work, especially, when inspecting 3D image sequences. The other way is to generate sequences of artificial images from scratch by exploiting some prior knowledge of a generated scene. This is most often accomplished by taking 2D snapshots of a changing 3D scene [GMN+98, BFB94, BB95]. The prior knowledge is encoded in models which control everything: the shape of objects, movements, generation of textures, noise simulation, etc. [LSR+05, You96]. This may involve a determination of many parameters as well as proper understanding of the modeled system. Once the two consecutive images are created, the information about movement between these two can be extracted from the underlying model and represented in the flow field. We have adopted this approach in the proposed framework.

**Basic idea of the framework** Based one real input image, our generator produces two artificial frames and the displacement field which maps the first frame onto the second one.
The generation process runs in the backward manner. The second frame is generated first. The ground truth displacement field from the first frame (not existing at this point) to the second frame is generated in the second step. Finally, the first frame is computed via the backward transformation \([LB94]\) of the second frame. These three steps can be repeated recursively in order to generate arbitrarily long artificial sequence.

The nature of generated images and flow fields

The generator takes three images as its input. The real gray valued input image \(I\), the mask of the background \(m_{bg}\) and the mask of the foreground objects \(m_{fg}\). Every created artificial image consists of background and foreground layer. The bottom background layer contains an artificially generated background, which usually represents the area of the whole cell. The foreground layer contains exact copies of object regions from the sample input image. The object in the foreground layer usually represents some cellular components which can move independently of each other and of the background. Therefore, we restrict ourselves to generate following type of movements. The background region can be a subject of global movement only. The foreground regions perform the same global movement as the background. Moreover, they can additionally perform independent local movements. The ground-truth flow field is then a composition of these movements. Thus, the ground-truth flow field can contain sharp discontinuities, because we mix independent global and local movements.

The problem with backward transformation

Since both background and foreground are given only by mask images, we accomplish the movements using the backward transformation technique \([LB94]\). Let us denote \(f_{out} = \text{BackT}(f, u)\) and define that the image \(f_{out}\) is the image \(f\) transformed according to the flow field \(u\). Basically, the backward transformation (sometimes called registration) technique translates \(f(x + u(x))\) to \(f_{out}(x)\) for every \(x\), which means that voxel values move “against” the flow field (see Fig. 6.1a for example). However, we realize, that some artifacts can occur when considering displacement fields with sharp discontinuities and non-smooth transitions. This is illustrated in Fig. 6.1. Therefore, we considered to use the forward transformation technique \([LB94]\) and to generate the whole sequence in forward manner. However, the forward transformation produces similar artifacts as the backward transformation. See Fig. 6.2 for the example of forward transformation artifacts. Further, the computation of forward transformation is more time demanding than the computation of backward transformation and it is more difficult to implement the forward transformation. Therefore, we developed the generator framework based on backward transformation technique and we improved the basic idea of the algorithm in order to avoid the limitations of backward transformation.

6.2.2 Detailed description of the framework

In this section, we describe in detail the generation process of the two artificial frames and a displacement field which maps one frame onto the next.

The input and output of the framework

The framework’s input was a sample input image \(I\), the background mask \(m_{bg}\) and the foreground mask \(m_{fg}\). The output consisted of images \(f_{1st}\), \(f_{2nd}\) and ground-truth flow field \(u_{gt}\) between \(f_{1st}\), \(f_{2nd}\) which denoted the first and the second
Variational optic flow in live-cell imaging

6.2. Pseudo-real image sequence generator

Figure 6.1: The basic idea of backward transformation. **(a) Left:** The grid represents voxels and their boundaries. The position of coordinate of voxel A is illustrated by dashed lines. The transformation moves value from vector's end to its beginning. In the case of real-valued vector (as is the one originating from voxel B), the moved value is the weighted sum of the nearest voxels' values with weights given by the portion of the gray area (values of voxels F, G, H and I). More vectors from distant places (as demonstrated with vectors originating from C and D) may fetch almost the same value when the flow is not smooth enough. This drawback results in the “copy” effect. **(b) Center:** An example of input image with non-smooth flow field. **(c) Right:** A result of the backward transformation with the “copy” effect.

Figure 6.2: The forward transformation. **(a) Left:** The grid represents voxels and their boundaries. The position of coordinate of voxel A is illustrated by dashed lines. The transformation moves value from vector’s beginning to its end. In the case of real-valued vectors (as are those originating from voxels F, G, H and I), the value for particular voxel must be searched for in the close vicinity (illustrated by gray region around voxel B). The value is then weighted with weights being the distance of the nearest vectors' ends in each quadrant of the marked area. More vectors from distant places (as demonstrated with vectors originating from D and E) may end up in almost the same location when the flow is not smooth enough. This drawback results in the “averaging” effect. **(b) Center:** An example of input image with non-smooth flow field. **(c) Right:** A result of the forward transformation with the “averaging” effect.
6.2. Pseudo-real image sequence generator

Variant optic now in live-cell imaging

Figure 6.3: Example of an image formation. (a) Top Left: The input image $f$ of fluorescently marked HP1 proteins in the HL60 cell. (b) Top Center Left: The intensity histogram of the input image at the bottom, binary image above displays voxels with intensities in the white strip of the histogram. (c) Top Center Right: Outcome of the rand($r$) generator. (d) Top Right: $f_{fg}$ from (6.4). (e) Bottom Left: $f_{2nd}$ from (6.5). (f) Bottom Center Left: The weights of the extended foreground mask, brighter intensity shows higher weights, see section 6.2.3. (g) Bottom Center Right: $f_{2nd}$ from (6.6). (h) Bottom Center Left: The map of intensity differences between $f$ and $f_{2nd}$, the maximal brightness shows the value of 30. All images were enhanced for the purpose of visualization.

Background characteristics extraction The preliminary step of the algorithm is to prepare a pool $r$ of voxel intensities. Only voxels $x$ satisfying $m_{bg}(x) > 0$ and $m_{fg}(x) = 0$ (simply the background voxels) are copied into the pool. The mean value $\mu$ is computed within $r$ since we have observed that histogram of the background voxels resembles Gaussian-shaped distribution (Fig. 6.3b). Because of that, voxels with intensities $i$ for which $i \notin (\mu - \sigma, \mu + k\sigma)$ are removed from the pool. We have chosen $\sigma = 11$ and $k = 3/2$ to fit the histogram better. This interval is shown as the white strip in Fig. 6.3b.

Artificial background generation An artificial background is generated in two steps. First, the foreground regions are filled with interpolated values. The interpolation is done in the following way. First, we find the nearest background for each foreground voxel in direction of each axis. That is, for each $x = (x, y, z)$, where $m_{fg}(x) > 0$ we search for coordinates $x_1$ and $x_2$ such that

\[
\begin{align*}
  x_1 &= \max \{ x' < x \text{ and } x' = (x', y, z) \text{ and } m_{fg}(x') = 0 \} , \\
  x_2 &= \min \{ x' > x \text{ and } x' = (x', y, z) \text{ and } m_{fg}(x') = 0 \}
\end{align*}
\]

and set $v_{x_1} = f(x_1, y, z)$ and $v_{x_2} = f(x_2, y, z)$ (Note that the $\{ x, y, z \}$ subscripts denotes here the affiliation to particular axis, not the partial derivation). If there are no such $x_1$, which can
Variational optic flow in live-cell imaging

6.2. Pseudo-real image sequence generator

happen exceptionally only when a mask touches the image border, then set \( v_{x_1} = \mu \) and \( x_1 \) is equal to the leftmost coordinate in \( f \). The \( x_2 \) is treated in the similar fashion. The value for \( f(x) \), proportionally along \( x \)-axis, is given by

\[
v_x = (v_{x_2} - v_{x_1}) \cdot \frac{x - x_1}{l_x + 1} + v_{x_1}
\]  

with \( l_x + 1 = x_2 - x_1 \). The \( v_y \) and \( v_z \) values can be obtained in the similar fashion. The replacing of foreground is finished by assigning \( f_{fg}(x) \) as in Fig. 6.3D:

\[
\forall x: f_{fg}(x) = \begin{cases} 
  \frac{l_x + v_x + l_y + v_y + l_z + v_z}{l_x + l_y + l_z} & \text{if } m_{fg}(x) > 0 \\
  f(x) & \text{otherwise} 
\end{cases}
\]  

The new artificial background is generated in the subsequent step. The \( f_{fg} \) image was convolved with separable averaging kernel. We used the filter \( \frac{1}{9}(1,1,1,1,1,1,1,1,1) \) for each axis. The new background image was then computed as

\[
\forall x: f_{2nd}(x) = \text{rand}(r) + (f_{fg}(x) - \mu)
\]  

where \( \text{rand}() \) is a generator of random numbers obeying uniform distribution. The effect of this term in (6.5) is to uniformly choose intensity values from the pool \( r \). This ensures that the generated background shares similar statistics, including intensity fluctuations and noise. The last term in (6.5) enables to display intracellular structures in the background, e.g. nucleolus as in Fig. 6.3e. Finally, the generation of the background layer is finished with blurring the image with the Gaussian filter with \( \sigma = 0.7 \).

**Addition of the foreground objects**  The foreground regions are overlaid over the artificial background image in order to finish the output image \( f_{2nd} \):

\[
f_{2nd} = \text{Copy}(f, m_{fg})
\]

where the notation \( f_{out} = \text{Copy}(f, m_{fg}) \) is used to state that only a regions given by mask \( m_{fg} \) are copied from \( f \), the rest of \( f_{out} \) remains untouched.

At this point, we have completed the generation of the second artificial frame of the sequence. It remains to generate the ground-truth flow field and the first frame of the sequence.

**Generation of the ground-truth flow field and first frame**  The ground-truth flow field for global movement of the whole image is created into \( u_{gt} \). We utilize an arbitrary rotation around arbitrary center together with arbitrary translation. The flow field is created regardless of masks \( m_{fg} \) and \( m_{bg} \). In fact, any flow field can be used provided that it is reasonably smooth. Further, the images are transformed with the global movement in order to generate the first frame:

\[
f' = \text{BackT}(f, u_{gt}),
\]

\[
m_{bg} = \text{BackT}(m_{bg}, u_{gt}),
\]

\[
m_{fg} = \text{BackT}(m_{fg}, u_{gt})
\]

We repeat the process of artificial background generation in the new position with \( f' \) instead of \( f \). The result is stored into \( f_{1st} \). Note that the same intensity pool \( r \) is used for the generation.
Addition of local movements to the ground truth flow and first frame  
A random translational vector $v_i$, is assigned to each component $i$ of the mask $m_{fg}$. For each $i$, we create local flow field $u_i$ and mask image $[m_{fg}]_i$:

\[
\forall x: \quad [u]_i(x) = v_i, \quad (6.10)
\]

\[
\forall x: \quad [m_{fg}]_i(x) = \begin{cases} 
1 & \text{if } x \text{ belongs to component } i \\
0 & \text{otherwise} .
\end{cases} \quad (6.11)
\]

Note that $u_i$ is uniformly filled what guarantees a smooth flow field. Independent local movements are embedded into $u_{gt}$ by computing the following equations:

\[
\forall x: \quad u_{gl}(x) = u_{gt}(x), \quad (6.12)
\]

\[
\forall i: \quad [m_{fg}']_i = \text{BackT}([m_{fg}]_i, u_i), \quad (6.13)
\]

\[
\forall i: \quad f'_i = \text{BackT}(f', u_i), \quad (6.14)
\]

\[
\forall i: \quad u_{gt} = \text{Copy} (\text{BackT}(u_{gl}, u_i), [m_{fg}'])_i), \quad (6.15)
\]

\[
\forall i: \quad u_i = \text{Copy} (0, 1 - [m_{fg}']), \quad (6.16)
\]

\[
\forall x: \forall i: \quad u_{gl}(x) = u_{gt}(x) + u_i(x) \quad (6.17)
\]

with the following interpretations: backup $u_{gt}$ (6.12), translate the patch corresponding to each component in $u_{gt}$ (6.15) together with its mask (6.13) according to its flow field, zero the component’s flow field outside of its moved mask (6.16) and add the result to the $u_{gt}$ (6.17).

Equations (6.15) to (6.17), in fact, concatenate global and local flow fields since the movement of the foreground consisted of global (6.9) and then local (6.13) movement. The image of each component is separately moved according to its flow field (6.14). Moving the entire image $f'$ according to the final $u_{gt}$ would produce image corrupted by the “copy” effect of non-smooth flow field.

Finally, the output image $f_{1st}$ is computed:

\[
\forall i: f_{1st} = \text{Copy} (f'_i, [m_{fg}'])_i) . \quad (6.18)
\]

The Copy() operation just overlaid the moved foreground regions over the artificially generated background. Optionaly, the ground-truth flow field can be trimmed:

\[
u_{gt} = \text{Copy} (0, 1 - m_{bg}) \quad . \quad (6.19)
\]

Generation of long sequences  
The presented framework also allows for the generation of an arbitrary long time-lapse image sequence. Due to the property of the backward transformation technique, the generation proceeds from the last image $f_{nth}$ of the sequence, given some $n \geq 2$, towards the first image $f_{1st}$. Clearly, the last image is the artificial substitute for the sample input image and so $f$ can be used as a sample without any modification. For the other images in the generated sequence, $f$ must be transformed to the actual position. Instead of iteratively moving the image, we suggest to hold the flow fields that prescribe the transformations required to get $f$ to the demanded position. Two flow fields should be enough. Let $[u_{gt}]_{k,i}, i < j$ denote the flow between images $f_{nth}$ and $f_{jth}$. For the purpose of consecutive generation of $f_{kth}$, $k = n - 2 \ldots 1$, compute:

\[
\forall x: \quad [u_{gl}]_{k,n}(x) = [u_{gt}]_{k,n+1}(x) + \text{BackT}([u_{gl}]_{k+1,n}, [u_{gt}]_{k+1,n+1})(x), \quad (6.20)
\]

\[
\forall x: \quad [u_{gl}]_{k,n}(x) = [u_{gl}]_{k+1,n}(x) + \text{BackT}([u_{gl}]_{k+1,n}, [u_{gl}]_{k+1})(x) \quad (6.21)
\]
where \([u_{gl}]_{k,k+1}\) is the flow field corresponding to just the global component of \([u_{gt}]_{k,k+1}\). Then, repeat the second part of the proposed framework from (6.7), as if \(f_{1st}\) should be created, with the following exceptions: for the background generation use \([u_{gl}]_{k,n}\) in (6.7) while for the foreground movements prepare \(f'\) with \([u_{gt}]_{k,n}\) in (6.7) and set \(u_{gt} = [u_{gt}]_{k,k+1}\). Also start the background generation from its second step with the convolution of \(f_{ig}\).

6.2.3 Results

We verify that the artificially generated image sequences are similar to the real ones in this section. We provide both examples and qualitative verification. Further, we describe several details of mask generation, remark a pitfalls of long sequence generation. Finally, we discuss the computational time demands of the generator framework.

Artificial data fidelity verification The generator was tested on several different 2D real-world images and one such 3D image. All generated images were inspected. Since every generated image arose from some supplied sample image \(f\), we could compare \(f\) and \(f_{2nd}\). For each pair, we computed the correlation coefficient (Corr.), average absolute difference (Avg. diff.) and root mean squared difference (RMS). The results are summarized in Table 6.1. The generator achieved minimal value of 0.98 for correlation, see Fig. 6.3h. This quantitatively supports our observations that generated images were very similar to their originals. A few 2D frame pair examples are shown in Fig. 6.4. An example of 2D artificial image sequence together with flow fields is illustrated in Fig. 6.5. Example of generated 3D ground truth frames is in Fig. 6.6. Decent improvement was observed when artificial background of 3D images was formed in a slice-per-slice manner what is also acknowledged in Table 6.1.

The image \(f_{1st}\) could not be evaluated quantitatively for the obvious reason. Nevertheless, the ratio would be definitely worse since all moved images are blurred a little. This is a feature of both backward and forward transformations when processing flow fields containing vectors with non-integer elements. In order to make both output images appear the same, we suggest to let \(f_{2nd}\) image perform the translation along vector \((0.5,0.5,0.5)\) and modify the \(u_{gt}\) correspondingly.

Mask generation details Both background and foreground masks were generated by thresholding of sample input image with manually chosen threshold. The jagged borders of the thresholded output were smoothed out with several iterations of the hybrid Chan-Vese algorithm which we have presented in Chapter 4.

Inappropriately created foreground mask may emphasize the borders of extracted foreground when inserted into artificial background. We replaced the Copy() operation in eqs. (6.6) and (6.18) by the following sequence of operations: extend the foreground mask by several dilations (the “Ext.” column in Table 6.1), compute the distance transform (we used [ST94]) on the mask and threshold it (see Fig. 6.3f), insert the foreground according to the weights (for details refer to [Ulm05]). We generally observed visually better results with this modification. According to Table 6.1, just 2 dilations achieved qualitatively better results in comparison to overlaying of foreground driven by unmodified input mask \(m_{ig}\).

We also tried the local movements mask which permitted the foreground to translate only inside this mask. This should prevent the structures from moving into the regions where there were not supposed to be, i.e. outside the cell. The masks are simple to create, for example by
Figure 6.4: Examples of generated pseudo-real frame pairs. The $f_{1st}$, the $f_{2nd}$ and the sample input image $f$ are shown in rows (A), (B) and (C), respectively. Notice the similarity between rows (B) and (C) in columns (1), (2) and (3). Images (A4) and (C4) should be similar too. Foreground objects (the white spots) in each cell were subject to additional local movements. (1) An example of the cell rotated 9 degrees clock-wise around its edge. (2) An example of another cell rotated 9 degrees clock-wise around its center. (3) An example of a similar cell with more foreground objects and with no global motion. (4) The same as (3) but the generator based on the forward transformation was used. All images were enhanced for the purpose of visualization.
Figure 6.5: Example of image sequence generated with the proposed framework. (a) Top Half: 15 artificially generated frames are in the top half of the figure. The whole cell is performing global counter clock-wise rotational movement while four foreground regions perform a local movement too. (b) Bottom Half: The displacement field between two consecutive images is visualized using the same notation as in Chapter 3.
Variational optic flow in live-cell imaging  

6.2. Pseudo-real image sequence generator

Figure 6.6: Example of 3D artificial frame generation. The whole cell performs slow transformation and rotation. Individual foreground objects performs locally independent movements. (a) **Left**: The 5th artificial frame of the sequence. (b) **Center**: The 7th artificial frame. (c) **Right**: The 9th artificial frame.

<table>
<thead>
<tr>
<th>Ext.</th>
<th>Corr.¹</th>
<th>Avg. diff.¹</th>
<th>RMS¹</th>
<th>Corr.²</th>
<th>Avg. diff.²</th>
<th>RMS²</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0.989</td>
<td>3.87</td>
<td>5.13</td>
<td>0.989</td>
<td>3.87</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.989</td>
<td>3.80</td>
<td>5.03</td>
<td>0.989</td>
<td>3.85</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.989</td>
<td>3.73</td>
<td>4.94</td>
<td>0.989</td>
<td>3.82</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.989</td>
<td>3.68</td>
<td>4.90</td>
<td>0.989</td>
<td>3.83</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0.992</td>
<td>2.76</td>
<td>3.83</td>
<td>0.992</td>
<td>2.77</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.992</td>
<td>2.62</td>
<td>3.69</td>
<td>0.992</td>
<td>2.74</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.993</td>
<td>2.41</td>
<td>3.46</td>
<td>0.992</td>
<td>2.62</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.993</td>
<td>2.33</td>
<td>3.40</td>
<td>0.992</td>
<td>2.64</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0.980</td>
<td>3.67</td>
<td>4.79</td>
<td>0.980</td>
<td>3.67</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.980</td>
<td>3.73</td>
<td>4.89</td>
<td>0.980</td>
<td>3.81</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.981</td>
<td>3.53</td>
<td>4.69</td>
<td>0.981</td>
<td>3.70</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.981</td>
<td>3.42</td>
<td>4.59</td>
<td>0.981</td>
<td>3.66</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0.982</td>
<td>3.15</td>
<td>4.16</td>
<td>0.982</td>
<td>3.16</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.983</td>
<td>3.07</td>
<td>4.08</td>
<td>0.982</td>
<td>3.13</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.983</td>
<td>3.00</td>
<td>4.03</td>
<td>0.983</td>
<td>3.11</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.984</td>
<td>2.92</td>
<td>3.96</td>
<td>0.983</td>
<td>3.10</td>
</tr>
</tbody>
</table>

Table 6.1: Comparisons of images $f$ and $f_{2nd}$. The column heading “Ext.” shows the number of dilations performed on the foreground mask $m_{fg}$. The mask controlled the foreground extraction as well as its plain overlaying¹ or weighted merging² (explained in section 6.2.3). A) and B) Comparisons based on two 2D images. C) Comparison based on a 3D image. D) Comparison based on the same 3D image, separate pools of voxel intensities were used for each 2D slice during the formation of the artificial background.
extending the foreground mask into demanded directions. The generated images became even more real.

**Remark on long sequence generation** We argue against further iterations of the framework to get \( f_{k \text{th}} \) from \( f_{(k+1)\text{th}} \). When proceeding towards smaller \( k \), transforming images iteratively leads to worse quality images because of the smoothing effect (Fig. 6.4A) of both transformations. Our suggested solution guarantees not more than two transformations of sample input image when creating \( f_{k \text{th}} \) for arbitrary \( k \in (1, n - 1) \).

**Implementation, computational demands** The image sequence generation framework which we have presented and tested in this section was implemented using the C++ programming language. We use common PC workstation (Pentium 4 2.6 GHz, Linux 2.6.x) for all computations and tests presented in this section. The backward variant of the ground-truth generator was able to generate two 2D artificial frames together with ground truth displacement field in 0.5 seconds in average. The size of generated images was 400 \( \times \) 400 pixels. The average speed of ground generator in three dimensions was 290 seconds per frame pair when generating images of 300 \( \times \) 300 \( \times \) 80 voxels. We confirm that the forward variant is up to two orders of magnitude slower than backward variant for 2D images. This is mainly because of greater complexity of forward transformation in contrast to backward transformation.

### 6.3 Optic flow methods for large displacement

We describe our experiences and experiments with motion estimation in live-cell image sequences in this section. First, we provide a motivation why to use the state-of-the-art optic flow methods in live-cell imaging. Then we describe and extend to three dimensions four advanced optic flow methods which we carefully choose from the pool of available state-of-the-art methods. Finally, we test those methods on artificial frame-pair data sets and evaluate them with respect to ground-truth flow fields.

#### 6.3.1 Motivation

We have stated earlier in this thesis (see Chapter 2) and in the introduction of this chapter, that the estimation and exact measurement of the motion in the image-sequences from live-cell studies is an important and interesting task for computer vision. In these days, the live-cell studies are mainly performed with fluorescence microscopes and one gets a time-lapse image sequence as a result of observation.

However, these sequences are more similar to a series of static snapshots than to a video sequence (see Fig. 6.7 for real example) due to several limitations of the acquisition process (see Chapter 2). Therefore, the motion estimation in these sequences is not an easy task, because the scene can change too much between two consecutive frames. The process of motion estimation in such sequences is usually divided to two consecutive steps. First, the global motion of the observed cells is estimated and corrected. Several image registration techniques were proposed and applied in order to correct and fix the global motion in live-cell image sequences (see Section 2.4.3). Second, the local motion inside the objects of interest is estimated. Common image processing techniques like thresholding, pattern matching or Gaussian fitting together with
Variational optic flow in live-cell imaging

6.3. Optic flow methods for large displacement

Figure 6.7: Example of two consecutive frames from live-cell study. The dynamics of the HP-1 protein domains in HL-60 cell nucleus was the subject of the study. The cells in the image perform global movement and in addition to that, the bright spots inside moves locally within the cell. Moreover, the cell partly change its shape. (a) Left: First frame. (b) Right: Second frame of the study.

manual tracking or trajectory recovering methods were used for this task in previous works [GME03, EA03, Miu05]. These methods have one thing in common. The objects of interest are found first in each frame and then the correspondence between two consecutive frames is established. The motion is estimated from the correspondence information. To the contrary, the variational optic flow methods can provide a different type of information, because they directly map each voxel from one frame to some other voxel in next frame. Therefore, the intracellular flows and local motion of intracellular objects can be easily estimated with these methods.

Nevertheless, there were hardly any attempts to estimate motion in the time-lapse sequences from fluorescence microscopy with the variational optic flow methods. One can say that the application of those methods to this field was overlooked [Miu05]. We suppose that it is due to two reasons. First, the classic variational optic flow methods (see Chapter 3) are not able to estimate large motion which is presented in some image sequences from live-cell studies (see Fig. 6.8d). Second, the computation of variational optic flow methods is quite computationally demanding task when using the basic numerical techniques. However, these two restrictions can be overcome in these days. First, it is possible to estimate large motion with the recently published methods, which make use of the original non-linearized gray value constancy assumption [PBB+06, BW05]. Second, the computation of variational optic flow methods can be performed even in real-time (for rather small 2D frames), when using the advanced multigrid method [BW05, BWS05b, Bru06].

We show in this section that the state-of-the-art variational optic flow methods can be used for estimation of large local motion in time-lapse sequences from fluorescence microscopy (see Fig. 6.8e). Further, we show that it is possible to estimate rather precisely combined global and local motion in such sequences with those methods. Finally, we show that the motion estimation in 3D image sequences can be accomplished in reasonable time when using bleeding edge numerical methods.
Figure 6.8: Comparison of classic variational optic flow methods with the variational optic flow methods for large displacement. The demonstration is performed on two artificial frames which simulates the dynamics of HP-1 proteins domains in HL-60 cell nucleus. The whole cell performs global movement and the protein domains perform local movement. The images were generated with our generator framework. (a) **Top Left:** First frame of the sequence. (b) **Top Center:** The second frame of the sequence (in red color channel) over-imposed onto the first frame (green color channel). (c) **Top Right:** The ground truth displacement field. (d) **Bottom Left:** The displacement field estimated with classic optic flow method (3.63). Note, that the motion mainly inside the objects was underestimated. (e) **Bottom Center:** The displacement field estimated with optic flow method for large displacement. A “large displacement” variant of (3.63) was used. Note, that the motion is not underestimated. (f) **Bottom Right:** The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 10.0.
6.3.2 Optic flow for large displacement

We describe the four variational optic flow methods for large displacement in this section. The methods were chosen from a larger pool of available methods with respect to their suitability for our data. The description will be provided for three-dimensional image sequences.

**Combined local global method**  The combined local-global (CLG) method for large displacements proposed in [PBB+06] is the first method which we have tested. This method produces one among the most accurate results [Bru06] in these days. It can produce flow fields with discontinuities and is robust to noise. Therefore, we assumed that it will be suitable for our data.

The CLG method is still similar to its Horn-Schunck precursor (other current state-of-the-art variational optic flow methods are also similar to HS method). The energy functional of CLG method still consists of *data* and *smoothness* term. Let \( \Omega_4 \subset \mathbb{R}^4 \) now denote the 4-dimensional spatial-temporal image domain and \( f(x_0, x_1, x_2, x_3) : \Omega_4 \rightarrow \mathbb{R} \) a gray-scale image sequence, where \((x_0, x_1, x_2, x_3)^T\) is a voxel location within image domain \( \Omega_3 \subset \mathbb{R}^3 \) and \( x_3 \in [0,T] \) denotes time. Moreover, let’s assume that \( \Delta x_3 = 1 \) and \( u = (u_0, u_1, u_2, 1)^T \) denotes the unknown flow. Then, the energy functional of CLG method can be defined as:

\[
E_{\text{CLG}}(u) = \int_{\Omega} \Psi_D(|f(x + u) - f(x)|^2) + \alpha \Psi_S \left( \sum_{i=0}^{2} |\nabla_3 u_i|^2 \right) \, dx
\]

(6.22)

where

\[
\Psi_D(s) = \sqrt{s^2 + \epsilon_D^2} \quad \Psi_S(s) = \sqrt{s^2 + \epsilon_S^2}
\]

and \( \epsilon_D, \epsilon_S \) are reasonably small numbers (e.g. \( \epsilon_D = 0.01 \)). The CLG method integrates several concepts. The data term consists of the original *non-linearized* gray value constancy assumption (3.51). This allows to correctly estimate the large displacements. Moreover, the CLG method uses the non-quadratic penalizers \( \Psi_D(s) \) and \( \Psi_S(s) \) and therefore it is robust with respect to noise and outliers.

**Image driven method for large displacement**  The second tested method consists of robust data term with non-linearized gray value constancy assumption and anisotropic image driven smoothness term [NE86].

We denote this method LRDIA. Up to our best knowledge, this method was never explicitly published, but it is only a straightforward combination of several known concepts. The energy functional of LRDIA method can be formulated as:

\[
E_{\text{LRDIA}}(u) = \int_{\Omega} \Psi_D(|f(x + u) - f(x)|^2) + \alpha \sum_{i=1}^{3} \left( \nabla_3 u_i \right) P_{\text{NE}}(\nabla_3 f) \nabla_3 u_i \, dx
\]

(6.23)

where \( P_{\text{NE}}(\nabla_3 f) \) is projection matrix perpendicular to \( \nabla_3 f \) defined as

\[
P_{\text{NE}} = \frac{1}{2|\nabla_3 f|^2 + 3\epsilon^2} \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}
\]

(6.24)
Variational optic flow in live-cell imaging  6.3. Optic flow methods for large displacement

where

\( a = f_{x_1}^2 + f_{y_1}^2 + \epsilon^2 \)
\( b = -f_{x_0}f_{x_1} \)
\( c = -f_{x_0}f_{y_2} \)
\( d = f_{x_0}^2 + f_{y_2}^2 + \epsilon^2 \)
\( e = -f_{x_1}f_{x_2} \)
\( f = f_{x_0}^2 + f_{y_1}^2 + \epsilon^2 \)

The \( \epsilon \) is a reasonably small number (e.g. \( \epsilon = 0.01 \)). Note that the only difference between CLG and LRDIA method is in fact the smoothing term. LRDIA method smooths the flow with respect to underlying image. The image data in live-cell imaging are often low contrast due to the limitations of the optical setup. Therefore, we assume that this smoothing term can help with processing of such data.

**Gradient constancy assumption**  The third and fourth tested methods are variants of the previous two. The image sequences from fluorescence microscopy have in common that the gray value intensity of the observed objects fades out with increasing time. Therefore, we add another constancy assumption to the data term in addition to the gray value constancy assumption. The added assumption is the gradient constancy assumption, which assumes that the gradient value remains the same in the corresponding voxels of two consecutive frames. This should provide us better results on image sequences which fade out with increasing time. The energy functional of CLG method with gradient constancy assumption is defined as

\[
E_{CLG_G}(u) = \int_{\Omega} \Psi_D(|f(x+u) - f(x)|^2 + \gamma(|\nabla f(x+u) - \nabla f(x)|^2)) + \alpha \Psi_S \left( \sum_{i=0}^{3} |\nabla u_i|^2 \right) \, dx \tag{6.25}
\]

The variant of LRDIA method with gradient constancy assumption is defined as

\[
E_{LRDIA_G}(u) = \int_{\Omega} \Psi_D(|f(x+u) - f(x)|^2 + \gamma(|\nabla f(x+u) - \nabla f(x)|^2)) + \frac{\alpha}{2} \sum_{i=0}^{3} (\nabla u_i^T \nabla f) \, dx \tag{6.26}
\]

where \( \gamma \) is the parameter which controls the influence of gradient constancy assumption. Note that the gradient constancy assumption is again included in the non-linearized form.

**Remark on numerical approximation**  The detailed description of the used numerical approximation is far beyond the scope of this thesis. Nevertheless, we want to remark that the non-linearized constancy assumptions contained in selected methods make the minimization quite complex. We use the multi-scale based approach, which was proposed in [PBB+06] for the minimization of (6.22,6.23,6.25,6.26). The image pyramids of both frames which contain frames in different scales are computed first. The optic flow is then computed on the coarsest level. The second frame on the finer level is then warped according to the displacement field computed on the coarsest level. The optic flow is computed on the finer level after that. This procedure is recursively repeated on finer and finer levels until the final flow is computed. The optic flow on each warping level was computed with the multigrid numerical method [BHM00]. An extensive description of optic flow computation with the so-called “warping” technique which uses multigrid framework on each warping level can be found in [Bru06, BW05].

112
Variational optic flow in live-cell imaging

6.3. Optic flow methods for large displacement

6.3.3 Results

In this section, we test the behaviour of CLG, LRDIA, CLG\(_G\) and LRDIA\(_G\) methods on artificial and real image data. We measure their performance on live-cell image sequences with large local displacements. Moreover, we present the results of the methods on image sequences with combined global rigid (translation, rotation) and local displacements. We present the results on both two and three dimensional data. Finally, we discuss the computational time and storage demands.

Large local displacement test

In the first test, we tested how accurately can the described methods estimate the large local displacement in the time-lapse sequences from fluorescence microscopy. The first experiment was performed on artificial two dimensional data. We used our artificial image sequence generator and prepared a data set of artificial images with large local displacements. The data set consisted of six different frame couples. The size of input frames was 400 \(\times\) 400 pixels. The number of objects which moved inside the nucleus varies from seven to eleven. The size of the individual translations vectors varies from 1.2 to 11.3 pixels. According to the literature, the input frames were filtered with Gaussian blur filter with standard deviation \(\sigma = 1.5\). We performed a parametric study over \(\alpha\) parameter for four tested methods over whole data set. The results were compared with respect to average angular error (AAE) \([FJ90]\) where angular error is defined as

\[
\arccos\left(\frac{(u_0)_{gt}(u_0)_e + (u_1)_{gt}(u_1)_e + 1}{\sqrt{((u_0)_{gt}^2 + (u_1)_{gt}^2 + 1)((u_0)_e^2 + (u_1)_e^2 + 1)}}\right)
\]  

where \((u_i)_{gt}\), \((u_i)_e\) denote the \(i\)-component of ground-truth and estimated vector, respectively. There were two coupled goals behind this experiment. We wanted to identify the best method by the mean of AAE and at the same time find suitable setting of \(\alpha\) parameter. The results are presented in Fig. 6.9. The AAE was computed for each run of particular method with particular \(\alpha\). Then, the averages of AAE over the frame pairs in the data set were computed. These averages are depicted in the graphs in Fig. 6.9. The AAE was computed only inside the moving objects. We can see that all methods perform reasonably well. The CLG method is outperformed by the LRDIA method. The gradient variant of CLG provided better results than the simple CLG method. The LRDIA\(_G\) method is not depicted in the graph because we noticed that its results depended on data and the average value was biased. It was slightly better than simple LRDIA method in some cases. But, it was slightly worse in other cases.

Combined global and local displacement test

The goal of the second experiment was to examine the performance of the tested methods on sequences with combination of global and local movement. We again prepared a dataset with artificial frame pairs. Image size and other input sequence properties were the same as in the previous experiment. The cell nuclei were transformed with global translation (up to 5 pixels) and rotation (up to 4 degrees). After that the local displacements were applied on the foreground object inside. It becomes clear from our experiments that we should divide the data into two groups. The results of the computations were influenced by the following fact. If the majority of objects inside the cell nucleus moved in the direction similar to the global translation the results fell into the first group and vice versa.
6.3. Optic flow methods for large displacement

Variational optic now in live-cell imaging

Figure 6.9: The dependency of average angular error on $\alpha$ parameter. The methods were tested on artificial image data with large local displacements up to 11.3 pixels. (a) Top: CLG and CLG$_C$ method. (b) Bottom: LRDIA method.

The AAE was computed on the cell nucleus mask (see the Fig. 6.8c) in this experiment. The results for both groups are illustrated in Fig. 6.10.

Test on real 3D data We tested the methods with best parameter settings on real three dimensional data in the third experiment. We computed the displacement field of two input frames of human HL-60 cell nucleus with moving HP1 protein domains. There are global as well as local movements in the frame pair (see Fig. 6.11). We estimated the flow with LRDIA method, $\alpha$ was set to 100. The size of the input frames was $276 \times 286 \times 106$. The results are illustrated in Fig. 6.11.

Implementation, computational demands The variational optic flow methods were implemented in C++ language and tested on common workstation (Intel Pentium 4 2.6 GHz, 2 GB RAM, Linux 2.6.x). We use the multigrid framework [Bru06] for numerical solution of tested variational optic flow methods. The computations on two 3D frames of size $276 \times 286 \times 106$ took from 700 to 920 seconds and needed 1.5 GB of RAM. Computations on two 2D frames of size $400 \times 400$ pixels took from 13 to 16 seconds and needed 13 MB of RAM.

Discussion We found out that the LRDIA methods produce slightly better results than the CLG methods (with respect to AAE) in both experiments on synthetic live-cell image data. Moreover, LRDIA methods are less sensitive to $\alpha$ parameter setting. It became clear that the smoothing term of LRDIA method is more suitable for the low contrast image sequences. The computed flow field can be easily oversmoothed by the CLG methods, because they consider
Variational optic flow in live-cell imaging 6.3. Optic flow methods for large displacement

Figure 6.10: The dependency of average angular error on $\alpha$ parameter. The methods were tested on artificial image data with global movement (translation up to 5 pixels, rotation up to 4 degrees) and large local displacements up to 11.3 pixels. (a) Top: CLG and CLG$_C$ method. (b) Upper Center: LRDIA and LRDIA$_C$ method. Majority of local displacements have the same direction as compared to global movement (top, upper center). (c) Lower Center: CLG and CLG$_C$ method. (d) Bottom: LRDIA and LRDIA$_C$ method. Majority of local displacements has different direction than global movement (lower center, bottom).
6.4 Simple motion tracking framework

We describe simple motion tracking framework which makes use of the variational optic flow in this section. The framework is very straightforward, it uses no new ideas or revolutionary algorithms. However, the experiments with the framework have provided quite impressive results. We first describe the framework and then we present results on both 2D and 3D image sequences and on both artificial and real image sequences.

Figure 6.11: Experiment with real 3D data. Frame size $276 \times 286 \times 106$. $xy$, $xz$ and $yz$ cuts on position $(138, 143, 53)$ are shown. (a) Left: First input frame. (b) Center: First input frame (red channel) superimposed on second frame (green channel). Correlation is 0.901. (c) Right: The LRDIA method with $\alpha = 100$ computes the flow field. Backward registered second frame (green channel) is superimposed onto the first frame (red channel). Correlation is 0.991.

The moving objects to be outliers in the data by particular parameter settings (larger $\alpha$). See Fig. 6.12 for the example of flow fields computed with tested methods. The oversmoothing effect of CLG method is illustrated in Fig. 6.12e.

The use of "gradient" variants of tested method can slightly improve their performance on live-cell image data. On the other hand, there is no warranty that the result will be always better. Actually, the LRDIA method seems to be more sensitive to $\alpha$ parameter when using its "gradient" variant. Surprisingly, the expected improvement of "gradient" variants was not significant even for fading out sequences. We suppose that the gradient constancy assumption does not help a lot, because the decrease of the intensities between two consecutive frames is not big. To sum it up, the 2D tests show that the LRDIA method is the most suitable for our data. Therefore, we used it for the experiment on the 3D real image data. The backward registered results show perfect match on $xy$-planes. The match in $xz$ and $yz$ planes is a little bit worse. This is caused by the lower resolution of the microscope device in the $z$ axis.

The bleeding-edge multigrid technique allowed us to get the flow fields in reasonable times even for 3D (up to 15 minutes for one frame pair). Small data sets which consist of only tens of frames of several cells can be analyzed in the order of days on one common PC. Larger data sets as well as parametric studies in 3D should be analyzed on a computer cluster.
Figure 6.12: Typical results of tested methods. The demonstration is performed on two artificial frames which simulates the dynamics of HP-1 proteins domains in HL-60 cell nucleus. The whole cell performs global movement and the protein domains perform local movement. The images were generated with our generator framework. (a) Top Left: The second frame of the sequence (in red color channel) over-imposed onto the first frame (green color channel). (b) Top Center: The ground truth displacement field. (c) Top Right: The displacement field estimated with LRDIA optic flow method. \( \alpha \) was set to 80. (d) Bottom Left: The displacement field estimated with CLG optic flow method. \( \alpha \) was set to 8. (e) Bottom Center: Example of oversmoothing effect of CLG method. The \( \alpha \) parameter was set to 18. (f) Bottom Right: The legend to the flow field visualizations. The color denotes the direction of the movement, the color intensity codes the displacement size. Maximum displacement (indicated with most intensive colors) was set to 10.0.
6.4.1 Motion tracking framework description

**Input** The input of the motion tracking framework consists of two components. The time-lapse sequence \( f \) of gray-value images is the first component (see Fig. 6.13a). The given list of points which mark the position of objects of interests in the first frame of the sequence is the second component. The point list is provided by the user of the framework.

**Tracking process** The actual tracking process is performed in two separate steps. First, the displacement field between all consecutive frame pairs of the sequence is estimated (see Fig. 6.13b). This can be done with one of the variational optic flow methods for large displacement, which were tested in previous section. We use the same optic flow method with identical parameter settings for all frame pairs in the sequence.

The trajectory of each point from the input list \( I \) is estimated in the second step (see Fig. 6.13c). Each displacement field between two consecutive frames is averaged with the Gaussian blur filter first in order to remove possible oscillations. The Gaussian filter is run on each vector component image separately. Then, for each point from the input list we denote its position in the first input frame. Further, the position of particular point in the following frame with respect to the computed flow field is estimated. The new position is gained by simple addition of the displacement vector corresponding to the voxel to its actual position. The coordinates of the new position are rounded to integer numbers. The previous steps are recursively applied until the end of image sequence is reached.

**Output** The main output of the motion tracking framework are the trajectories of each input point. The point positions can be denoted in the input image frames (see Fig. 6.13c). Further one can save the trajectories in separate file and visualize them in several different ways (see following section).

6.4.2 Results

We present several motion tracking examples in this section. We use the LRDIA method in all examples, because it identified as the most suitable method in previous section.

**Experiments on artificial 2D sequences** We generate several artificial 2D image sequences with our generator and test the behavior and performance of the proposed tracking framework in the first group of tests. Results from motion tracking in two such sequences are presented in this paragraph.

There were considered only large local displacements (up to ten pixels) of four selected objects in the first testing sequence. There was no global motion of the whole cell nucleus. The sequence consisted of 50 frames each of which had size 400 \( \times \) 400 pixels. The initial positions of the moving objects in the first frame were set by hand. We used the LRDIA method for the computations of the displacement fields between two consecutive frames. The parameters of the method were set to \( \alpha = 90, \sigma = 1.0 \) and \( \epsilon_D = 0.01 \). The results of the motion tracking are depicted in Fig. 6.14a. The "trace" image based on the estimated motion is in Fig. 6.14b. One can compare this image with the "trace" image based on the ground-truth motion which is depicted in Fig. 6.14c. It can be seen that the "trace" images are almost the same. The location of particular positions differed at most by one pixel. This behavior was caused by the fact, that
Figure 6.13: Scheme of the motion tracking framework. (a) Top Row: The input image sequence $f$. First, second and third frame (from left to right) are shown. (b) Middle Row: The displacement field between each two consecutive frames is computed with particular variational optic flow method in the first step. The parameter setting is the same for all computations. The displacement field is visualized with vectors in this image. (c) Bottom Row: Each point from the input list is tracked with respect to computed displacement (flow) field in the second step. The estimated position in each input frame is denoted with the green cross in each frame.
we rounded each real valued position to integer valued position. Therefore, the position could differ, when the estimated displacement field was not exactly the same as the ground-truth field. The three dimensional movement reconstruction in time and space is visualized in Fig. 6.14d.

We combine global rotational motion with the large local displacement of particular objects in the second test. The center of the rotational movement was set to the center of the image and the rotation degree was set to one. The large local displacements up to ten pixels were added to selected objects. The sequence again consisted of 50 frames and size of each frame was $400 \times 400$ pixels. The initial positions of the moving objects were set again by hand. We use the LRDIA method for the computations of the displacement fields between two consecutive frames. The parameters of the method were set to $\alpha = 100, \sigma = 1.0$ and $\epsilon_D = 0.01$. The results are summarized in Fig. 6.15. It should be noted that motion was again nearly perfectly estimated.

**Experiments on real 2D sequences**  We felt confident to perform experiments on the real time lapse sequences, motivated by the fact that the motion was estimated nearly perfectly with the proposed framework in the artificial sequences. Results from two real sequences are presented in this paragraph.

Both real image sequences captured the motion dynamics of the HP-1 protein domain in the HL-60 cell nucleus during apoptosis. The first sequence consisted of eleven frames of size $400 \times 400$ pixels. The local movements presented in the sequence were not so large as in the artificial sequences. However, the cell nucleus collapses during the evolution and changes its shape and this is challenging for the motion estimation. The motion between each consecutive frame pair was computed with LRDIA method with $\alpha = 100, \sigma = 1.0$ and $\epsilon_D = 0.01$. The results are summarized in Fig. 6.16. It should be noted that motion was estimated quite good. Even the rather big motion between last two frames was partially estimated. However, one can notice that the motion of several objects was underestimated in the last two frames (see Fig. 6.16a).

The second real image sequence consisted of nine frames of size $400 \times 400$ pixels. This sequence was challenging from several reasons. First, the cell nucleus again changed its appearance during the evolution. Second, the gray value intensity fades out during the evolution. Finally, some objects joined into one object and again divided into two objects during the evolution. The motion in the sequence was estimated with the LRDIA method. The parameters of the method were set to $\alpha = 100, \gamma = 6, \sigma = 1.0$ and $\epsilon_D = 0.01$. The results are summarized in Fig. 6.17. One can notice, that the motion framework was not confused by the changing appearance of the cell nucleus. Further, the motion was estimated quite well even for the last frames whose intensity is strongly fading out. However, the motion of two objects which joined and divided during the evolution was not estimated correctly.

**Experiments on 3D sequences**  We present the results of motion tracking in one artificial and one real 3D image sequence in this paragraph. We do not provide the same amount of examples as for 2D sequences, because of several reasons. First, there is no principal or implementation difference between the framework in 2D and in 3D and all the results which we presented in 2D examples hold in 3D as well. Second, the visualization of motion tracking in 3D sequences is a little bit problematic on 2D paper.
Figure 6.14: Example of the results of the motion tracking in 2D. The objects in artificially generated sequence perform large local movements (up to 10 pixels). The artificial sequence consisted of 50 frames. The size of each frame was 400 \times 400 pixels. The motion between each consecutive frame pair was computed with LRDIA method with \( \alpha = 90, \sigma = 1.0 \) and \( \epsilon_D = 0.01 \). (a) Top Five Rows: From left to right top to bottom: The motion of four selected objects was tracked according to computed displacement. The positions in each frame were denoted with colored cross. Each second frame is shown. (b) Bottom Left: The computed object positions in the sequence summarized in one “trace” image. (c) Bottom Center: The same image as in (b), but the “trace” image was computed using the ground truth flow. Note that (b) and (c) are almost the same. (d) Bottom Right: The 3D movement reconstruction in time and space. Time is represented by the third spatial coordinate.
Variational optic now in live-cell imaging

6.4. Simple motion tracking framework

Figure 6.15: Example of the results of the motion tracking in 2D. The objects in artificially generated sequence perform large local movements (up to 10 pixels). The whole cell nucleus performs a rotational movement in clockwise direction. The artificial sequence consists of 50 frames. The size of each frame is 400 × 400 pixels. The motion between each consecutive frame pair was computed with LRDIA method with $\alpha = 100$, $\sigma = 1.0$ and $\epsilon_0 = 0.01$. (a) **Top Four Rows:** From left to right top to bottom: The motion of four selected objects is tracked according to computed displacement. The positions in each frame was denoted with colored cross. Each third frame is shown. (b) **Bottom Left:** The computed object positions in the sequence summarized in one “trace” image. (c) **Bottom Center:** The same image as in (b), but the “trace” image was computed using the ground truth flow. Note that (b) and (c) are almost the same. (d) **Bottom Right:** The 3D movement reconstruction in time and space. Time is represented by the third spatial coordinate.
Figure 6.16: Example of the results of the motion tracking in real 2D time-lapse sequence. The real sequence consisted of 11 frames. The size of each frame was 400 × 400 pixels. The motion between each consecutive frame pair was computed with LRDIA method with $\alpha = 100$, $\sigma = 1.0$ and $\epsilon_D = 0.01$. (a) Top Three Rows: From left to right, top to bottom: The motion of twenty one selected objects was tracked according to computed displacement. The positions in each frame was denoted with colored cross. Each frame is shown. (b) Bottom Left: The computed object positions in the sequence summarized in one “trace” image (c) Bottom Right: The 3D movement reconstruction in time and space. Time is represented by the third spatial coordinate. Note that even the big motion between the last two frames was partially estimated. However, the motion of some objects was underestimated between the last two frames.
Figure 6.17: Example of the results of the motion tracking in real 2D time-lapse sequence. The real sequence consisted of 9 frames. The size of each frame was $400 \times 400$ pixels. The motion between each consecutive frame pair was computed with LRDIAG method with $\alpha = 100$, $\gamma = 6$, $\sigma = 1.0$ and $\epsilon_D = 0.01$. We use the gradient version of the LRDIAG method because of the fade out effect at the end of the sequence. (a) **Top Three Rows:** From left to right, top to bottom: The motion of eleven selected objects was tracked according to computed displacement. The positions in each frame was denoted with colored cross. Each frame is shown. (b) **Bottom Left:** The computed object positions in the sequence summarized in one "trace" image (c) **Bottom Right:** The 3D movement reconstruction in time and space. Time is represented by the third spatial coordinate. Note that the motion of the two objects in the center of the cell was not correctly estimated. However, motion of the remaining objects was estimated correctly.
We generated an artificial 3D time-lapse sequence for the first experiment. The image sequence had 15 frames of size $276 \times 286 \times 80$ voxels. The cell nucleus performed both translational and rotational global movement. Additionally, the objects inside the cell nucleus performed independent local movements. The initial positions of the moving objects in the first frame were set by hand. We used the LRDIA method for the computations of the displacement fields between two consecutive frames. The parameters of the method were set to $\alpha = 90$, $\sigma = 1.0$ and $\epsilon_D = 0.01$. The results of the tracking are depicted in Fig. 6.18. The autofocus projection of computed and ground-truth "trace" image is depicted in Fig. 6.18a and in Fig. 6.18b, respectively. The autofocus projection (maximum intensity projection) of the first and the last frame of the sequence with denoted positions of each point of interest are depicted in Figs. 6.18c,d. Finally a combined 3D visualization which consisted of volumetric visualization of first frame and trajectories of points of interest. The results on artificial 3D image sequence were comparable to those results in 2D. The object trajectories were nearly the same as the ground truth trajectories.

We track the motion in the real image sequence in the second 3D experiment. The sequence consisted of eleven frames of size $310 \times 310 \times 70$. Actually, the first example of motion tracking in 2D real image sequence worked on autofocus projections of this 3D sequence. Therefore, one can easily compare whether the motion tracking in 3D has similar outputs to motion tracking in 2D. The initial positions of the moving objects in the first frame were set by hand. We used the LRDIA method for the computations of the displacement fields between two consecutive frames. The parameters of the method were set to $\alpha = 90$, $\sigma = 1.0$ and $\epsilon_D = 0.01$. The results are summarized in Fig. 6.19. It should be noted that motion was estimated quite good similar to the 2D version of this sequence. The big motion between last two frames led to the same motion underestimation.

**Implementation, computational demands** The motion tracking framework was implemented in C++ language and tested on common workstation (Intel Pentium 4 2.6 GHz, 2 GB RAM, Linux 2.6.x). The vast majority of the processor time and computer memory was consumed by the computation of the variational optic flow method. The own tracking process formed only a small fraction of the total time and memory demands. Therefore, one can say that the processing of 2D sequence took up to 15 minutes (considering 50 frame in the sequence) and the processing of 3D sequence took up to four hours (considering around 15 frames in the sequence).

**6.5 Summary**

The overall goal of this chapter was to test and prove whether the modern variational optic flow methods for large displacement can be used for motion tracking in time-lapse sequences from fluorescence microscopy. We feel confident to claim that we have shown that the motion in the image sequences from live cell studies can be reliably estimated and even tracked using those modern methods. Our work was divided into three parts. We have proposed a generator of the artificial image sequences first. Then we have tested and compared the performance of several modern variational optic flow methods. Finally, we have proposed a simple motion tracking framework which uses those optic flow methods for motion estimation. We summarize the details of each part in following paragraphs.
Variational optic flow in live-cell imaging

6.5. Summary

Figure 6.18: Example of the results of the motion tracking in artificial 3D time-lapse sequence. The artificial sequence consisted of 15 frames. The size of each frame was $276 \times 286 \times 80$ voxels. The cell nucleus performed both translational and rotational global movement. Additionally, the objects inside the cell nucleus performed independent local movements. The motion between each consecutive frame pair was computed with LRDIA method with $\alpha = 90, \sigma = 1.0$ and $\varepsilon_D = 0.01$. (a) Top Left: The computed object positions in the sequence summarized in one “trace” image. The trace image is in the autofocus projection (maximum intensity projection). (b) Top Right: The ground-truth object positions in the sequence summarized in one “trace” image. (c) Middle Left: The initial positions of the points of interest. The image is in the autofocus projection. (d) Middle Right: The last image of the sequence with the denoted position of the tracked points. (e) Bottom: The 3D movement reconstruction. The image consisted of volumetric rendered first frame (pseudo-colored). The reconstructed trajectories are incorporated to the model.
Figure 6.19: Example of the results of the motion tracking in real 3D time-lapse sequence. The real sequence consisted of 11 frames. The size of each frame was $310 \times 310 \times 70$ voxels. The motion between each consecutive frame pair was computed with LRDIA method with $\alpha = 90$, $\sigma = 1.0$ and $\varepsilon_D = 0.01$. (a) Top Left: The computed object positions in the sequence summarized in one “trace” image. The trace image is in the autofocus projection (maximum intensity projection). (b) Top Center: The initial positions of the points of interest. The image is in the autofocus projection. (c) Top Right: The last image of the sequence with the denoted position of the tracked points. (d) Bottom: The 3D movement reconstruction. The image consisted of volumetric rendered first frame (pseudo-colored). The reconstructed trajectories are incorporated to the model.
Pseudo-real image sequence generator We have proposed a framework for generating time-lapse pseudo-real image data. It allows for automatic synthesis of unbiased sequences of 2D and 3D images. By suppling real-world sample image we could force images in the sequence to look more realistic. The background mask of the cell and the foreground mask of selected intracellular structures were supplied too. This gave us a layered control over the regions where global and local movements should occur. The aim was to automatically generate a vast amount of data together with corresponding flow field, that we called ground-truth, in order to evaluate methods for foreground tracking as the next step. We may conclude that it generated images very similar to the sample image. The foreground was a copy from the sample image which implicitly assured its quality. The background voxels posed the same statistics since they were generated to do so. Theoretically, the presented framework has ambitions to work reliably on arbitrary data exhibiting of unimodal background distribution.

On the other hand, the proposed pseudo-real image sequence generator have one limitation. It is not possible to generate image sequences where the shape of the background object is deformed during the evolution. However, this feature can be added to the generator in the future.

Optic flow methods for large displacement in live-cell imaging We studied state-of-the-art variational optic flow methods for large displacement for motion tracking of fluorescently labeled targets in living cells. We focused on 2D as well as 3D images. Up to our best knowledge, we tested those methods first time in the literature for three dimensional image sequences.

We showed that these methods can reliably estimate large local divergent displacements up to ten pixels. Moreover, the methods can estimate the global as well as local movement simultaneously. Surprisingly, the variants of CLG and LRDIA methods with gradient constancy assumptions did not bring significant improvement for our data. The LRDIA method produced the best results in our experiments with respect to the ground-truth flow fields. From the practical point of view, we have achieved reasonable computation times (even for three dimensional image sequences) using the warping technique together with full bidirectional multigrid numerical method.

Simple motion tracking framework We have described a simple motion tracking framework in the last part of this chapter. The framework made use of the modern variational optic flow methods. The tracking was performed in two steps. First, the optic flow (displacement) field between each consecutive frame pair was estimated. Second, the trajectories of several given points were estimated. We have obtained very promising results both for 2D and 3D sequences. We have shown that the motion tracking in live-cell time-lapse sequences can be done even with such simple framework when using the advanced variational optic flow methods.
Chapter 7

Summary

We have described several successful applications of PDE-based image processing methods in the field of fluorescence microscopy in this thesis. The fundamental concepts of fluorescence microscopy and PDE-based image processing methods were described and summarized in the first two introductory chapters. Our own contribution was then described in the following three rather independent chapters.

**Fast and robust approximation of Chan-Vese model – Chapter 4**  We have presented a fast and robust hybrid algorithm for Chan-Vese based segmentation suitable for low contrast and noisy biomedical data in Chapter 4. Presented algorithm combines the ideas of two recently published approximations of the Chan-Vese model. We have shown that both Gibou-Fedkiw (k-Means level set) and Esedoglu-Tsai (threshold dynamics) approximations have its specific limitations when processing low contrast and noisy images, which are common in fluorescence microscopy. The main limitation of the Gibou-Fedkiw approximation is its third step. The boundary of segmented objects can be easily over-smoothed with this step. On the other hand, we have shown that the segmentation of low contrast images can completely fail when using only the Esedoglu-Tsai approximation.

We have combined both approximations into one hybrid algorithm in order to eliminate their limitations. The proposed three-step hybrid algorithm uses the first two steps of k-Means level set algorithm to get rough segmentation of the input image. The hybrid algorithm only fine-tunes the segmentation results with the threshold dynamics algorithm in the final step. Therefore, the boundary of segmented objects can be neither easily over-smoothed like with Gibou-Fedkiw approximation nor completely wrong like with the Esedoglu-Tsai approximation. The computational time of the hybrid algorithm is comparable to the computational time of its ancestors because the initial guess provided by the k-Means level set algorithm reduces the number of iterations which the threshold dynamics algorithm needs to reach the steady state. Further, we have shown that it is much easier to set the suitable minimization parameters for our method than for the ancestor algorithms. Therefore, the hybrid algorithm is better suited to the real biomedical applications, especially, if the interaction with a user is not desired or possible. We have shown on practical examples that the hybrid algorithm is suitable for the segmentation of the image data from fluorescence microscopy. We have discussed the parameter settings and their influence on resulting segmentation results.
Summary

Finally, we have derived an effective locally one dimensional scheme for the computation of the Esedoglu-Tsai approximation and we have proposed a small modification of this approximation in order to produce smooth reconstruction of both 3D and 2D objects.

3D interphase chromosome reconstruction - Chapter 5  We have aimed our effort at proposing two new methods for the specific problem of segmentation and 3D reconstruction of interphase chromosome territories in Chapter 5. The main contribution of this chapter was the design of new chromosome territory segmentation-reconstruction techniques, which gained from the robustness of PDE-based filtering and implicit active contour models. The parallel goal was to design methods, which could be easily used for automatic image analysis. We have proposed two methods to fulfill this task. The fast marching based method and Chan-Vese based method. We have shown that both methods are reasonably fast and non-interactive. The detailed analysis of method properties, advantages and disadvantages was presented as well. We have found out that neither method is significantly superior to the other in our tests. Nevertheless, both methods correctly segmented about 90% of input images from the artificial image data set as well as from the real image data set.

Therefore, we suppose that the methods are suitable for automatic chromosome territory segmentation and reconstruction, because their parameters can be set at the beginning of computations, they are reasonably fast and they do not need any interaction with the user during the reconstruction process. On the other hand, we have shown that the proposed methods have still their limitations. Specifically, they can't correctly separate touching chromosome territories and they were not able to correctly segment the input in rare cases.

Second interesting contribution described in Chapter 5 was the generator of synthetic three dimensional images. Up to our best knowledge, there exists no similar generator which is able to synthesize three dimensional images of objects (cell nuclei, chromosome territories) common in fluorescence microscopy. We want to remark on this place again that our contribution to this generator was minor. We suppose, that the tests on synthetic data which were produced with our generator are more reliable than any other possible tests. Further, the tests on synthetic data and the comparisons with ground-truth images helped us to find the best parameter settings for both proposed methods.

Variational optic flow in live-cell imaging - Chapter 6  We have built and tested in a systematic way a motion tracking framework based on advanced variational optic flow methods in Chapter 6. We have shown that the modern variational optic flow methods for large displacement can be used for motion tracking in time-lapse sequences from fluorescence microscopy. Moreover, we have shown that the motion in the image sequences from live cell studies can be reliably estimated and even tracked using those modern methods.

Our work in Chapter 6 was divided into three parts. First, we have proposed a framework for generating time-lapse pseudo-real image data. The proposed generator allows for the evaluation of optic flow computing methods. The primary aim was to create a pair of new gray-scale images similar to the given real-world image together with appropriate flow field. The generator can generate sequences with global cell motion together with independent local motions of selected intracellular objects, which is a phenomenon often observed in the field of fluorescence microscopy. On the other hand, the proposed pseudo-real image sequence generator has one limitation: Only global and local translations and rotations were considered.
as possible movements of the background and foreground objects. It is not possible to generate image sequences and ground truth flow fields where the shape of the background object is deformed during the evolution. However, this feature can be added to the generator in the future. We want to remark on this place that the ground truth image generator was proposed in collaboration with Vladimir Ulman, who made the major part of work.

We studied state-of-the-art variational optical flow methods for large displacement in the second part of Chapter 6. We extended recently published methods to 3D and tested their performance on artificial sequences with respect to the average angular error of the computed displacement field. We showed that these methods can reliably estimate large local divergent displacements in live-cell sequences. Moreover, the methods can estimate the global as well as local movement simultaneously. It became clear from our test that the best results on our data can be obtained when using the LRDIA method. From the practical point of view, we have achieved reasonable computation times (even for three dimensional image sequences) using the warping technique together with full bidirectional multigrid numerical method.

Finally, we have described a simple motion tracking framework in the last part of the Chapter 6. We have obtained very promising results both for 2D and 3D sequences. We have shown that the motion tracking in live-cell time-lapse sequences can be done even with such a simple framework when using the advanced variational optic flow methods.

What can be done further  We have presented own contribution in three rather independent chapters. It is our personal opinion, that the most promising results were obtained in Chapter 6. However, we have tested only a very simple tracking framework. The tool can be probably more powerful if, for example, the occlusions and interaction of the moving objects were handled.
Appendix A

Example images

Figure A.1: Pseudo-color 24-bit image representing three dimensional image of two fibroblast cell nuclei (blue color) with methylated histons (green color). The image dimension is 750 × 550 × 15. The voxel size is 0.1 μm in x and y direction, 0.3 μm in z direction. **Top Left**: xy cut. **Top Right**: yz cut. **Bottom**: xz cut. The short line markers on the borders of the cut planes denote the cut positions. The chromatin in cell nuclei is stained with DAPI and has blue color. The histons in the nuclei are immunolabeled with FITC and have green color.
Figure A.2: Pseudo-color 24-bit image representing three dimensional tissue scan (colon tissue). The image dimension is $1300 \times 1030 \times 12$. **Top Left:** $xy$ cut. **Top Right:** $yz$ cut. **Bottom:** $xz$ cut. The short line markers on the borders of the cut planes denote the cut positions. Chromatin is stained with *PI* and has red color.
*Example images*

**Figure A.3:** Pseudo-color image representing three dimensional image which contain HL-60 cell nuclei (blue channel) with chromosome pair 22 inside (green channel). The image dimension is $1300 \times 1030 \times 40$. The voxel size is $0.065 \, \mu m$ in $x$ and $y$ direction, $0.59 \, \mu m$ in $z$ direction. The image is visualized in the autofocus projection.
Example images

**Figure A.4:** Example of the 3D image sequence from the live-cell study. Three frames from the sequence are presented. The dynamics of the HP-1 protein domains inside the HL-60 cell nucleus is the subject of the study. Each frame consists of $400 \times 400 \times 15$ voxels. The voxel size is $0.065 \mu m$ in $x$ and $y$ direction, $0.59 \mu m$ in $z$ direction. Each frame is visualized in the autofocus projection. **Left:** First frame. **Center:** Second frame. **Right:** Third Frame.

**Figure A.5:** Another example of the 3D image sequence from the live-cell study. Three frames from the sequence are presented. The telomere dynamics inside the cell nucleus is the subject of the study. Each frame consists of $80 \times 70 \times 30$ voxels. The voxel size is $0.36 \mu m$ in $x$ and $y$ direction, $0.2 \mu m$ in $z$ direction. The 2D cuts on specific positions are shown in each frame image. The short line markers on the borders of the cut planes denote the cut positions. **Left:** 1st frame. **Center:** 3rd frame. **Right:** 5th Frame.
Appendix B

Publications


Appendix C

Structure of included CD

The CD included to this thesis has the following directory structure:

root
  |- bin
    |- README
    |- *.exe
    |- *.dll
  |- data
    |- chapter4
    |- chapter5
    |- chapter6
  |- text
    |- attachments
  |- viewer3d
    |- setup.exe
  |- README

In each directory there is a README file which contains the detailed description of each file and directory on particular level. All executables which were used for computations of all examples in this thesis are in the bin directory. Although all experiments in this thesis were performed on Linux machine, the executable files on the CD were compiled for Windows XP operating system. They are provided with all build dependencies (libraries). The executables are mainly command line utilities. Each of them provides the help when executed without parameters. Some image data which were presented in this thesis are in the data directory. The image data are saved in an independent directory for each contribution chapter. The d.pdf file with electronic version of the thesis text with clickable bookmarks is in the directory text. There are also some attachments like cv and publication list in this thesis. An application for viewing 3D images is in the directory viewer3d. The directory contains the self installing executable for Windows XP operating system.
Glossary

Dictionary

**actin** – is globular structural protein which can be found in many eukaryotic cells. Actin has two main functions in cells. In all cells it forms the thinnest part of the cytoskeleton, which allows motility, while in muscle cells it also forms the contractile apparatus.

**antibody** – is a protein used by the immune system to identify the foreign objects. The antibody is constructed to identify particular target called antigen.

**antigen** – is a foreign element (e.g. a specific molecule) which stimulate the immune system to respond.

**apoptosis** – is a process of self-suicide by a cell in a multicellular organism. It is one of the main types of programmed cell death. This process is usually started either by the cell itself or by the surrounding environment of the cell when the cell is somehow damaged. The process prevents the cell from developing to cancer cell.

**autofocus** – is a type of projection of three dimensional image data to two dimensional image data. The pixel value in the resulting 2D image is maximum of voxel values along a projection axis (either x, y or z).

**biotin** – is a water-soluble B-complex vitamin. It is a standard immunochemical marker used in fluorescence microscopy.

**CCD** – charge-coupled device. It is an image sensor which consists of an integrated circuit which contains an array of linked or coupled light-sensitive capacitors. It serves for the acquisition 1D or 2D gray-scale or color image.

**cell nucleus** – is the main organelle in eukaryotic cells. It contains almost whole genetic information of the cell. This information is encoded in multiple long double stranded DNA molecules. This molecules form the chromosomes.

**chromatin** – is the complex of DNA and proteins which forms chromosomes. The structure of chromatin is different in different phases of the cell cycle. It forms a network of long thin threads in the interphase nucleus. It forms the well known 4-arm structure during the cell division.

**chromosome** – is a single long molecule of DNA. It contains many individual genes. The DNA which encodes the genetic information is organized into chromosomes in the cell nucleus.
cytoplasm – is a substance which fills the cell. It consists mainly of water, organic molecules, enzymes, salts etc..

DAPI – 4', 6-diamidino-2-phenylindole is a fluorescent stain that binds strongly to DNA. It is used extensively in fluorescence microscopy. Since DAPI will pass through an intact cell membrane, it may be used to stain both live and fixed cells.

denaturation – is a structural change in macromolecules which is caused by external conditions such as heat. The transformation of DNA from its natural double stranded form to two single-stranded molecules is often meant by this term in the context of FISH imaging.

dichroic mirror – also called “chromatic beam-splitter”. It is a semi-transparent bandpass filter that reflects light shorter than a specific wavelength and transmits light that is longer than that wavelength.

digoxigenin – a specific plant steroid hormone. It is a standard immunochemical marker used in fluorescence microscopy.

DNA – deoxyribonucleic acid. In this acid the genetic information of vast majority of living organisms is encoded.

double well potential – is a function with two global same global minima with two points.

eyepiece – the part of the microscope which is closest to he observer eye. It is together with the objective responsible for the magnification of the observed object.

FISH – Fluorescence In Situ Hybridization. A technique in which the fluorescently labeled DNA probes are used for visualization of specific cell objects (nuclei, chromosomes, genes, etc.). The specimens are visualized using the fluorescence microscopes.

FITC – also called fluorescein. It is a fluorochrome used in fluorescence microscopy.

fluorescence – The fluorescent substances being exposed to the light emit light with longer wavelengths. Most fluorescent substances produce very little heat and fluorescence has been therefore referred as “cold light”.

fluorochrome – is a part of molecule which makes the molecule fluorescent. The electrons in fluorochrome absorb energy from light of specific wavelength and are excited to higher energy level. The new energy level is not stable and the electrons quickly return to the original energy level. The electrons emit the light with longer wavelength (lower energy) when returning to the stable energy level. The emitted light is responsible for the fluorescence of the fluorochrome. The whole process of absorption and emission takes about $10^{-9} - 10^{-6}$ second.

gene – it is a segment of nucleic acid (DNA or RNA) which contains the necessary information for the production of particular protein.

GFP – is an acronym for green fluorescent protein. It is a protein which naturally exist in bodies of jellyfish Aequorea victoria. It fluoresces green light when exposed to blue light. Derivatives of this protein are extensively used in live cell imaging.
**Glossary**

**hybridization** – is the process of joining of two complementary strand of nucleic acid (either DNA or RNA) to form double-stranded macromolecule.

**immunofluorescence** – is a technique which allows the visualization of specific proteins in cells by binding particular antibodies containing fluorochromes to the antigens of examined protein.

**interphase** – is a phase of the cell cycle, defined only by the absence of cell division. The cell spends majority of its live in this phase.

**noise – dark-charge** – is one type of noise which occurs in fluorescence microscopy. This noise is caused by statistical fluctuations of the dark charge signal i.e. the signal generated even in the case when no light illuminate the detector.

**noise – photon-shot** – is one type of noise which occurs in fluorescence microscopy. It is caused by statistical fluctuations of the acquired signal level.

**noise – readout** – is one type of noise which occurs in fluorescence microscopy. It is caused by the a/d converter which converts the analog electronic signal to the digital levels.

**nucleolus** – is a organelle located in the cell nucleus. The main function of the nucleolus is to produce components of ribosomes. Nucleoli are roughly spherical. They are surrounded by a layer of condensed chromatin. The nucleoli themselves do not contain chromatin.

**objective** – is a part of microscope which is responsible for the image magnification of the observed specimen. This part is closest to the observed object.

**PDE** – partial differential equation. Differential equations describe the relations between an unknown function and its derivatives. When the unknown function depends on several variables, the partial derivatives can occur and therefore the equation is called partial differential equation.

**photobleaching** – is the destruction of fluorochromes caused by excitation light. The fluorochromes become more chemically reactive as they have been excited. They can undergo reactions that lead to permanent changes, by which they loose their fluorescence property. This can complicate the observation in the fluorescence microscopy and is especially problematic in live-cell imaging.

**photoxicity** – is a phenomenon which occurs in live-cell imaging, where illumination of the particular fluorochrome in the cell puts this cell to death.

**PI** – Propidium iodide is a fluorescent biomolecule that can be used to stain DNA as well as RNA.

**plasmid** – is a DNA molecule which is not a part of chromosomal DNA. It is circular and double stranded. It is usually far shorter than chromosomal DNA. There can be many copies of one particular plasmid in the cell. It can be usually found in bacteria.

**PMT** – photomultiplier tube. It is a very sensitive 0D-light detector.
Glossary

**probe** – is a DNA or RNA sequence complementary to the target sequence. The probes can be labeled with fluorochromes or with molecules of biotin or digoxigenin.

**proteins** – compounds of amino-acids and are essential parts of all living organisms. They participate in every process within cells.

**PSF** – point spread function. Also known as system impulse response. It describes the response of an imaging system to a point light source or point object.

**RNA** – ribonucleic acid. This acid is mainly used as a messenger in protein production process of eukaryotic cells. It can also encode genetic information of viruses.

**stokes shift** – is the difference in wavelength between positions of the band maximum of the absorption and fluorescence emission spectra of the same electron transition. The shift occurs because the emitted wavelength is always longer due to the energy conservation law. A small part of the excitation energy is converted to heat.
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bonus

The attentive reader who carefully read this thesis through to this point will obtain a cup of coffee made personally by the author, whenever they meet :-).