Segmentation, suivi et visualisation d'objets biologiques en microscopie 3D par fluorescence: Approche par modèles déformables

Thèse de doctorat

présentée et soutenue publiquement le **4 décembre 2007** par

Alexandre Dufour

en vue de l'obtention du titre de

Docteur de l'Université Paris Descartes

Composition du jury

Rapporteurs :	Pr. Françoise Préteux	Institut National des Télécommunications
	Pr. Jean Sequeira	Université de la Méditerranée, Aix-Marseille II
Examinateurs :	Pr. Nicole Vincent	Université Paris Descartes
	Pr. Georges Stamon	Université Paris Descartes
	Dr. Jean-Christophe Olivo-Marin	Institut Pasteur
	Dr. Auguste Genovesio	Institut Pasteur Korea

Résumé

Nous nous intéressons à la détection et au suivi d'objets biologiques divers (cellules, noyaux, etc.) dans des images et séquences tridimensionnelles acquises en microscopie par fluorescence. L'observation de phénomènes biologiques in situ étant de plus en plus cruciale pour les experts, il est nécessaire, en plus de l'analyse quantitative, d'effectuer un rendu volumique 3D de la scène et des objets qui y évoluent. De plus, pour certaines applications, l'automatisation des techniques d'acquisition d'images requiert un haut niveau de reproductibilité et induit des contraintes de temps de calcul que nous nous efforçons de prendre en compte.

En analyse d'images, les modèles déformables, autrement connus sous le nom de contours actifs, font actuellement partie des méthodes de pointe en segmentation et suivi d'objets grâce à leur robustesse, leur flexibilité et à leur représentation sémantique hautniveau des entités recherchées. Afin de les adapter à notre contexte, nous devons faire face à deux types de difficultés. Tout d'abord, la majorité des méthodes existantes se réfèrent aux variations locales d'intensité ou gradients de l'image pour détecter le contour des objets recherchés. Cette approche est inefficace en microscopie tridimensionnelle par fluorescence, où les gradients sont très peu prononcés selon l'axe de profondeur de l'image. Ensuite, nous devons gérer la détection et le suivi simultanés d'objets multiples dans une même scène en évitant la confusion d'objets entrant en contact au cours du temps.

Dans un premier temps, nous pallions à ces problèmes en proposant un modèle de segmentation et de suivi multi-objets basés sur la méthode des lignes de niveaux ou level sets utilisant l'approche par région de Mumford et Shah. La méthode proposée donne des résultats satisfaisants en terme de qualité, en revanche elle ne se prête pas efficacement au rendu 3D de la scène, pour lequel nous sommes tributaires d'algorithmes dédiés à la reconstruction 3D, comme, par exemple, la méthode des "Marching Cubes". Ces algorithmes sont coûteux en temps de calcul et peuvent induire des erreurs visuelles d'approximation, pouvant résulter en une mauvaise interprétation des résultats.

Ainsi, dans un deuxième temps, nous proposons une reformulation de la méthode en employant des maillages triangulaires, très populaires dans le domaine de la conception assistée par ordinateur (CAO). La méthode obtenue repose sur la même approche région et produit donc des résultats similaires. Cependant, la représentation par maillage réduit, d'une part, la complexité du problème (et donc le temps de calcul) et permet, d'autre part, un rendu 3D parallèle au processus de segmentation, ce qui permet d'éliminer l'étape de reconstruction et les approximations qu'elle induit.

Les performances des deux méthodes proposées sont d'abord évaluées puis comparées sur un jeu de données simulées reproduisant le mieux possible les caractéristiques des images réelles. Ensuite, nous nous intéressons plus particulièrement à l'évaluation de la méthode par maillages sur des données réelles, en évaluant la robustesse et la stabilité de quelques descripteurs de forme simples sur des expériences d'imagerie haut-débit. Enfin, nous présentons des applications concrètes de la méthode à des problématiques biologiques réelles, réalisées en collaboration avec d'autres équipes de l'Institut Pasteur de Corée.

Abstract

We focus on the detection and tracking of various biological objects (cells, nuclei, etc.) in 3D images and sequences acquired in fluorescence microscopy. With the growing importance of *in situ* observation of biological phenomena, it is necessary to provide efficient and intuitive 3D visualization of the scene in addition to its quantitative analysis. Moreover, for some applications, the automatization of the image acquisition process requires a high level of reproducibility and induces time consumption constraints that we attempt to account for.

In the field of image analysis, deformable models, also known as active contours, are currently part of the cutting edge techniques for objects segmentation and tracking thanks to their robustness, their flexibility and their high level of semantic description of the target entities. In order to adapt these techniques to our context, we must face two sets of difficulties. Firstly, most exisiting methods rely on local intensity variations or gradients of the image in order to detect the target boundary. This approach is inefficient in 3D fluorescence microscopy, where gradients are very weak along the depth axis of the images. Secondly, we must deal with the simultaneous detection and tracking of multiple objects evolving in the same scene, while ensuring that objects touching through time are distinguished correctly without confusion.

In the first part of this work, we attempt to solve these problems by proposing a multi-object segmentation and tracking method employing the level set formalism and the region-based approach of Mumford and Shah. This method yields quite satisfying results, however it is not suitable for efficient 3D rendering of the scene, for which we must rely on 3D reconstruction algorithms, such as the "Marching Cubes". Such algorithms are computationally expensive and may induce visual approximation errors, potentially resulting in a wrong interpretation of the results.

Thus, in the second part, we propose a reformulation of the method using triangular meshes, very popular in the field of computer assisted design (CAD). The obtained method relies on the same region-based approach and therefore produces similar results. Nevertheless, the triangular mesh representation offers two main advantages. Firstly, it reduces the method complexity and, hence, its computational cost. Secondly, it allows a simultaneous 3D rendering of the scene during the segmentation process, thus eliminating the time cost and approximation errors of the reconstruction step mentioned earlier.

The performance of both methods are first evaluated and compared on a set of simulated data, reproducing as best as possible the characteristics of real images. Then, we focus more particularly on the evaluation of the triangular mesh method on real data. We perform this evaluation on trial high-thoughput screnning experiments using a small set of shape descriptors. Finally, we present practical applications of the method to real biological problems, performed in collaboration with other teams of Institut Pasteur Korea.

Contents

Chapter 1
Introduction

Chapter 2
State-of-the-Art: deformable contours in computerized image analysis

Chapter 6
Conclusion

Contents

vi

List of Figures

List of Tables

1

Introduction

This chapter describes the context, motivations and goals of our work. We start by giving an overview of the image analysis problem in the biological field. Then, we consider particular aspects of the problem to pinpoint the applicative needs and isolate the major difficulties to be encountered. Finally, we outline the scope of this thesis by introducing the solutions that we propose to solve these difficulties.

Contents

1.1 Preamble

1.1.1 Human vision vs. Computer vision

The human vision system has been the focus of extensive research studies over the past decades. Indeed, we are seamlessly able to recognize people, objects, shapes, colors, textures, motion, distances, etc., thanks to a powerful stereo-vision system backed up by a visual memory and a magic black box: the visual cortex. Whereas the operating mechanisms of the human eye are nowadays relatively well understood, how the information is stored and/or processed by the brain is still a mystery that is not likely to be discovered in the near future.

Computer vision deals with similar recognition problems through the world of digital images. Whether the knowledge about human vision should be used (or not) as starting point for computer vision systems is still nowadays an open debate. While the formation process of digital images is globally identical to that of real images in the human eye, their analysis and interpretation have very little knowledge to start with. This is probably the main reason why image processing, analysis and understanding is a huge and growing research topic in computer science, joining efforts from very diverse domains.

1.1.2 Computer vision: why and what for ?

There are two main motivations to develop computer vision systems. The first reason is to assist or replace individuals for repetitive tasks of everyday life involving visual control. Applications may relate either to professional tasks (e.g., quality control, fingerprint authentication, medical diagnosis, architecture restoration, military drones) or to personal comfort amenities such as face recognizing door locks, self-driving cars, robot assistants/companions etc.. This wide range of applications leads to an ever-growing market that will continuously follow the leading-edge of computer vision science. The second reason motivating computer vision research is the exponentially growing amount of data coming from digital cameras and video systems. Managing such a colossal amount of data cannot be done manually, and thus requires automated systems to perform indexing, classification, recognition and information extraction.

With these considerations in mind, we focus on the recognition and information extraction aspects of computer vision, and more particularly on their applications to current challenges in biological microscopy, as we shall see below.

1.1.3 Computer vision in biology

Over the past decade, biology has certainly become one of the fields that depends the most upon imaging, due to substantial progress in microscopy imaging. In this context, fluorescence microscopy has become the major source of information, initiating the migration from subjective visual inspection to robust quantitative analysis. We first give below a brief historical overview fluorescence imaging techniques, for which further details can be found in [Claxton et al.,] and [Amos and White, 2003].

Fluorescence imaging in biology

The principle of fluorescence imaging is to exploit the capacity of organic and non-organic specimens to absorb and subsequently re-emit light. Fluorescence was discovered in 1852 by Sir George Stokes, who noticed that a fluorspar mineral emitted red light through ultra-violet excitation. Similar observation was then made on many other specimens, stimulating the development of fluorescence microscopes to produce digital images of this phenomenon.

The first fluorescent microscopes appeared in 1904, and were first applied to biology in the 1930s with the introduction of fluorescent-labeled antibodies. The explosion of fluorescent imaging started in the late 70s, after antibodies were adapted to normal proteins, and thereby to cellular and subcellular structures.

With the increasing specificity of fluorescent probes, traditional fluorescence microscopes (also called *widefield* microscopes) started to show their limit: images are disturbed by out-of-focus light mostly due to the thickness of the specimen. Although many researchers avoided the problem by focusing their studies exclusively on thin objects, others were interested in imaging a broader range of objects with higher detail. Focus has then turned to confocal microscopes.

The principle of confocal microscopes was invented by Minsky in 1955. The idea is to illuminate a point of the specimen with a diffraction-limited spot, and place an aperture in front of the detector that rejects out-of-focus light coming back from the medium. The operation is then repeated by sweeping the focal point over the whole domain to produce the final image, yielding the name of *point-scanning* microscope. This technique induces a slicing effect that eliminates almost completely the blur induced by traditional microscopes, suggesting its suitability for efficient three-dimensional imaging via motorized Z-step devices. The drawback of point-scanning systems is that acquisition is much slower than in the widefield case, since image pixels are acquired one at the time. The need for faster acquisition lead to several enhancements such as the *spinning disk* confocal microscope [Egger and Petran, 1967] and the line-scanning confocal microscope [Sheppard and Mao, 1988]. The former illuminates the medium at several locations simultaneously, and each spot is focused upon by a specific aperture in a Nipkow-type spinning disk, imitating multiple confocal microscopes working in parallel. The latter acquires a whole line of pixels at once using a line-shape illumination and a slit detector.

Throughout the 90s, confocal microscopes benefited from advances in optical and electronic components, improving their efficiency, reliability and resistance to noise. Meanwhile, fluorescent probes tailored to match laser excitation wavelengths were introduced. Coupled with the ever-increasing power and storage capabilities of modern computers, confocal microscopes thus gained tremendous interest in countless applications. Although more expensive than conventional microscopes, recent distribution of personal confocal microscopes has decreased the price of low-end systems and increased the number of individual users, making them a "must-have" tool in any biological research facility.

Biological questions

Joint efforts in biology and microscopy have allowed to image a broad range of cellular and subcellular functions, both *in vitro* and *in vivo*, in two and three dimensions at different wavelengths (colors), eventually with time-lapse imaging to investigate cellular dynamics.
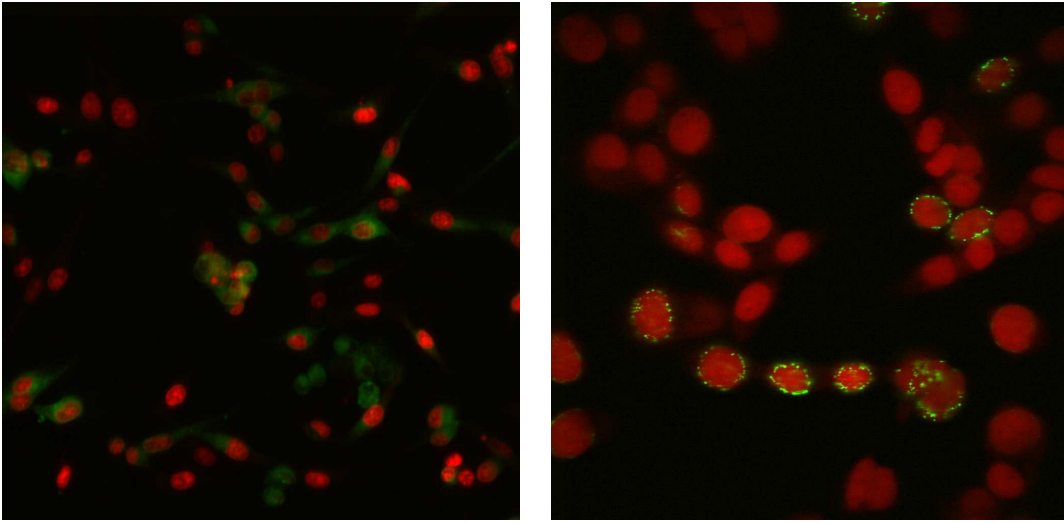
Figure 1.1: Two-channels Confocal microscopy images of cells under two different conditions using different optical magnification factors.

Yet, 2D imaging has some limitations, in particular for the study of objects that either move or exhibit a heterogeneous shape along the depth axis, e.g., cells and nuclei. In the following, we shall focus on the following applications:

- Nuclear morphology
  The cell nuclear morphology is a topic of active research in biology. A large array of biological functions is accompanied by major changes in the geometry of the nucleus [Leman and Getzenberg, 2002]. Determining exactly how geometric characteristics relate to cellular function requires accurate morphological information. Therefore, one has to turn to 3D images in order to analyze the entire structure of cellular and sub-cellular compartments [Vonesch et al., 2006].

- Cell motility
  Cell migration is central to several fundamental biological processes, many of which have important medical implications. Understanding the mechanisms of cell migration and how they can be controlled to prevent or cure disease is thus also an important goal for biomedical research.

1.2   Problems and challenges

In this work we address the problems of segmentation, tracking and interactive visualization of multiple biological objects in 3D fluorescence microscopy. As we have seen above, these tasks are recurrent in many biological applications, and require automated, fast and robust algorithms in order to be applied to high-content screening applications. The main difficulties we will have to face are the following:

- Segmentation
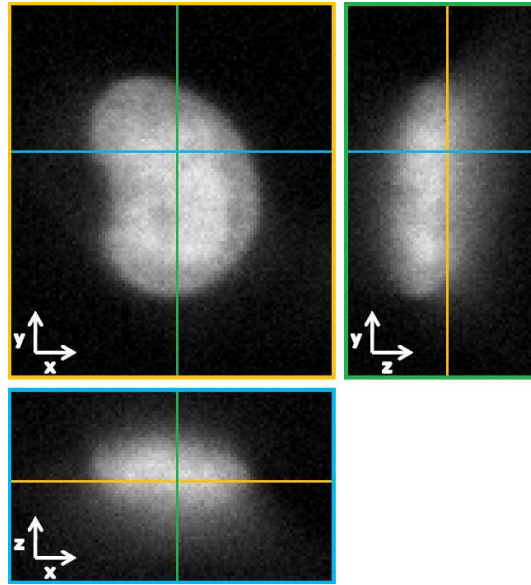
Figure 1.2: Axis-dependent slices of a 3D image showing a cell nucleus. (top) XY slice of the image. (right) YZ slice of the image. (bottom) XZ slice of the image. One can notice the significant blur on the nuclear membrane along the Z axis compared to the X and Y axes, suggesting the inefficiency of gradient-based techniques on 3D fluorescence microscopy data.

Throughout this work we will have to deal with fluorescence images coming from both widefield and confocal microscopes. Although it is commonly agreed that confocal microscopy images are less difficult to analyze than widefield images, both techniques present similar disturbances, the strongest one being the point-spread function (PSF) of the microscope. This function induces an anisotropic blur to the image, which is stronger along the depth axis (see figure 1.2). Other sources of image degradation are the auto-fluorescence of the medium, the acquisition noise induced by the camera detector, and various other imperfections of the optical system.

- Tracking

  The human eye performs tracking by focusing on an object of interest and matching it within the continuously evolving environment, involving huge amounts of memory-based prior-knowledge on the object's potential variations (direction, speed, deformation, etc.). Computer-based tracking, unfortunately, usually possesses very little knowledge about the object movement, and must cope with the non-continuous nature of video sequences. In the current context, we wish to track multiple objects evolving simultaneously in the same scene. We will thus have to maintain a correct association between the objects from a sequence frame to the next, while preventing confusion between objects that come close to each other through time.

- Visualization

  Visualization of 3D data is a topic of active research. It seems clear that visual feedback could be profitable in an image analysis context, for instance to adjust the parameters of the segmentation and tracking algorithms, or to produce elegant visual outputs of the models, either for visual inspection or for educational purposes and public presentations under various forms (articles, pictures, movies, etc.). Since this

step is not crucial to the quantitative result, visualization should be performed at reasonable cost compared to the analysis process.

1.3 Proposed solutions

There are two possible approaches to address these problems. The first approach is to deal with each task separately, using appropriate methods in each case. Although satisfactory results may be obtained, we stress that the combination of these methods may yield technical difficulties that might increase either computational load or even yield wrong results. We adopt the second approach, which consists in dealing with all the problems at the same time by integrating them into a common framework. We choose to do so through deformable models. Deformable models are among the cutting-edge techniques in computer vision, thanks to their flexibility and adaptability to any given problem. They are particularly well suited for segmentation and tracking, since they can accomplish both tasks at the same time by re-using a model from a sequence frame to the next without re-initializing its characteristics.

The outline of this work is as follows. In chapter 2, we review the literature on deformable models applied to image analysis. During this review, we distinguish the two main families of deformable models, in order to introduce our contributions in each of these families in chapters 3 and 4. We evaluate the proposed frameworks in chapter 5, first on synthetic data, and then on trial screening experiments. We also present current efforts on a real biological application. Finally, we conclude this thesis in chapter 6 by summarizing our contributions and proposing several extensions to the proposed work, motivated by future challenging applications.

2

State-of-the-Art: deformable contours in computerized image analysis

This chapter focuses upon the existing literature on deformable models applied to image segmentation, and is organized as follows. Section 2.1 first recalls the origins and principles of deformable models originating from the early 80's. Then, the two main families of deformable models are presented in sections 2.2 and 2.3, along with their application to computerized 3D image analysis.

Contents

2.1 Introduction

Deformable contours belong to the more general family of deformable models, i.e., "interactive models that can be deformed through user or automated intervention". In computer vision, they relate to the notion of *model matching*, where one wishes to match a target in the observed scene using an initial model, eventually serving as reference. In our case, we wish to segment objects, i.e., extract each object individually from the others and from the scene background. To do so, we need to find the outline of each object boundary, defined by a closed curve in 2D and a closed surface in 3D. For easier reading, these terms will be commonly referred to as N-dimensional *contours*. A diverse range of low-level vision methods and algorithms have been proposed for this purpose (see [Rosenfeld and Thurston, 1971], [Canny, 1986], [Geman et al., 1990], [Beucher, 1990], [Glasbey, 1993], and also [Haralick and Shapiro, 1985] for a general review). The power of deformable contours over these low-level approaches is their flexibility. Indeed, deformable models are potentially able to solve any problem related to segmentation in the sense that they can combine any type of information that might improve robustness and/or efficiency. This information may come either from the observation itself or from prior knowledge on the target defined by the user. Throughout this chapter, we shall describe how this can be done for deformable contours, and which problems can thereby be solved.

Deformable contours in image analysis originate from the early work of [Burr, 1981], and were made popular several years later thanks to [Terzopoulos, 1987] and [Kass et al., 1988], under the names of *active contours* or *snakes* models. This latter terminology originated from the idea of bending an initial line or contour until it coincides with a target, similarly to the undulations of a snake (see Figure 2.1). This deformation is commonly expressed as the minimization of a user-defined *energy* or *cost* functional E that should have a global minimum when the contour Γ matches the target perfectly. Hence, the contour should satisfy in its final steady state

$$\Gamma_{final} = \arg \min_{\Gamma} E(\Gamma). \tag{2.1}$$
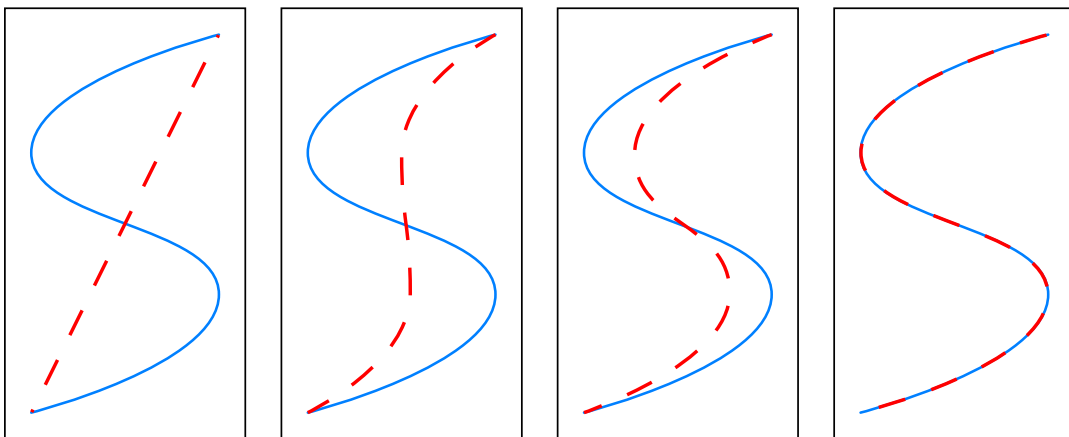
Figure 2.1: Principle of a *snake*. The snake (dashed red line) is iteratively deformed until it fits the target (solid blue line).

Before going further, it is important to remind that image segmentation is, as many other tasks in computer vision, a typical *inverse* problem. An inverse problem basically

consists in finding a certain set of theoretical descriptors of a phenomenon given only experimental observations. This is a very delicate task, since the estimated descriptors may be very far from expected theoretical values, depending on the observation conditions and the estimation method employed. Let us illustrate the principle of inverse problems with a short example illustrated in figure 2.2. Assume that we wish to determine the location and value of the minimum of an analytical function (plotted on the left) given an experimental measurement (plotted on the right). Depending on the chosen approach, one has to deal with local minima (pointed out with red arrows) and eventually several global minima having the same value.
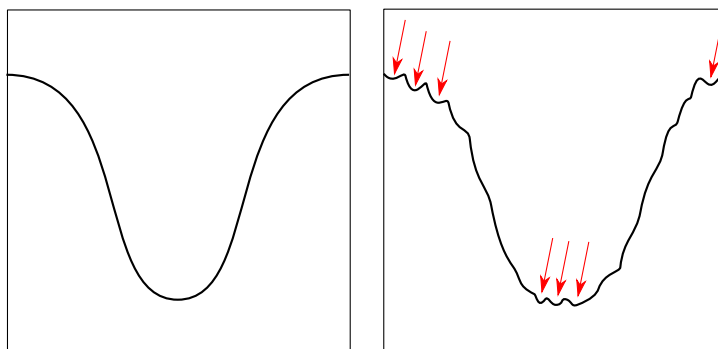
Figure 2.2: Illustration of an inverse problem: one wishes to determine the value and location of the minimum of the function on the left, given the experimental measure on the right. Depending on the chosen approach, local and global minima may greatly disturb the estimation of the solution.

Due to the variability of the resulting estimation, most inverse problems are said to be *ill-posed* and should be transformed into well-posed problems before they can be solved. [Poggio et al., 1985] suggested to regularize the model through physical constraints. In an energy min

As mentioned before, the great advantage of deformable contours over low-level techniques is that there is no particular constraint on the energy functional, other than it should be minimal when the desired result is reached. Therefore, one can incorporate and combine an infinity of energy terms of diverse nature in order to obtain optimal results (e.g., describing the target or constraining the deformation). These terms are commonly grouped into three categories:

- **Image**-based terms, also known as *external* terms, link the contour to the image data. Their minimization attracts the snake toward the target based on its intensity or color characteristics. Hausdorff stated some time ago in [Hausdorff, 1978] that any closed set in the plane could be decomposed into two disjoint sets: the boundary and the interior. Hence, the influence of the image on the contour deformation is transcribed through two different sets of terms naturally called boundary or *edge-based* terms, and interior or *region-based* terms. Edge-based terms usually aim at finding local discontinuities or gradients in the image, located on the target boundary as illustrated in figure 2.3. Region-based terms describe the global intensity distribution of the target, and may express diverse information such as texture or region smoothness.

- **Regularization** terms, also called *internal* terms, maintain the stability of the contour deformation in the sense of the regularization scheme. They ensure that the
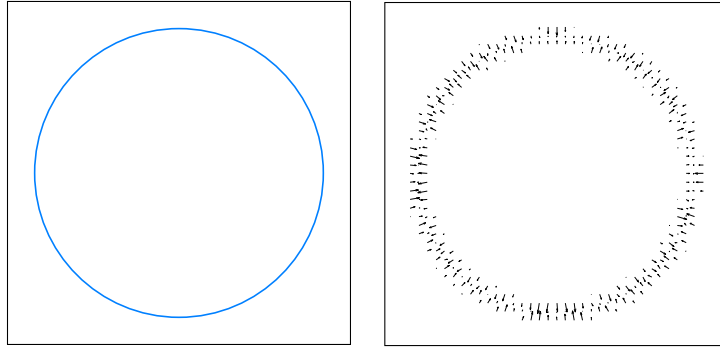
Figure 2.3: Image information used by edge-based models. The target contour (left), induces on the digitized image high intensity variations, yielding a gradient map shown on the right. The vectors in this map will be used to pull the deformable model toward the target.

contour undergoes a smooth deformation, thus approaching the solution in a robust way, hopefully avoiding local energy minima that would otherwise stop the contour on its way. These terms are essential in real applications, where the energy functional exhibits many local minima due to the various sources of image disturbance.

- **Model**-based terms, or *prior* terms, regroup all additional user-information that may help the contour finding the target in a faster or more efficient way, especially when the terms above are not sufficient. These terms usually describe prior knowledge about the target, therefore they vary greatly from an application to another. For instance, some terms may tweak the contour deformation by pulling it toward points of interest belonging to the target. These points can either be set manually or through a feature point extraction method such as the Harris detector [Harris and Stephens, 1988] or wavelet transforms, as done in [Dorval et al., 2003]. Other terms describe the geometrical properties of the target, constraining the contour deformation according for instance to shape knowledge, as suggested for deformable templates in [Yuille et al., 1989] and applied to deformable contours in [Staib and Duncan, 1992],[Ray et al., 2002],[Cremers et al., 2002],[Paragios, 2003].

Combined together, the general energy functional can be written

$$E(\Gamma) = \lambda \ E_{external}(I_0, \Gamma) + \mu \ E_{internal}(\Gamma) + \gamma \ E_{prior}(\Gamma), \tag{2.2}$$

where I_0 is the original image and μ, λ, γ are non-negative weighting parameters. The solution is obtained when the energy is minimum, i.e., when the contour Γ reaches a steady state. Tweaking the weight parameters may help improving the convergence speed, or reducing the risk of being trapped in a local energy minimum. However, these parameters are hard to find and usually determined empirically, although many attempts have been made to determine these parameters automatically. We do not describe these methods here, also we refer the reader to chapter 2 in [Rousselle, 2003] for a review on parameter-estimation for active contours.

2.1.1 Energy minimization

Once the energy functional has been defined, a suitable numerical scheme must be chosen in order to find its minimum (at least local, ideally global). Three main approaches appear in the literature, namely the *variational*, the *dynamic* and the *statistical* approach.

- **Variational** formulation

  This approach consists in finding a minimum of the energy using a time-dependent gradient-descent framework. The most widely-used tools for this purpose are the Euler-Lagrange equations, which state that an energy minimum is found when the contour reaches a steady state, i.e.

$$\frac{\partial \Gamma}{\partial t} = \nabla F(I_0, \Gamma) = 0, \tag{2.3}$$

  where F is a potential force computed from the minimization of equation (2.2). This approach is the most popular, since the Euler-Lagrange framework ensures mathematical convergence to a minimum, at least local, of the energy functional. Unfortunately, the energy is often non-convex and exhibits many local minima, although they can be reduced by adjusting empirically the parameters μ, λ and γ.

- **Dynamic** formulation

  The variational formulation can be viewed as a particular case of the more general dynamic formulation (also referred to as mass-spring system), where the contour is defined time-varying and follows the Lagrangian laws of motion under the influence of physical forces analogous to F, yielding

$$m\frac{\partial \Gamma}{\partial t} + d\frac{\partial^2 \Gamma}{\partial t^2} + \nabla F(I_0, \Gamma) = 0, \tag{2.4}$$

  m and d weighting the contour mass and damping densities, respectively. The final solution is again obtained when the system reaches a steady state, i.e.

$$\frac{\partial \Gamma}{\partial t} = \frac{\partial^2 \Gamma}{\partial t^2} = 0. \tag{2.5}$$

  This approach introduces the notions of mass and damping density, i.e., physical properties of the contour, and is therefore preferred in applications related to physical systems modeling, e.g. in computer-assisted surgery [Cotin et al., 1999].

- **Statistical** formulation

  Another range of methods expresses the problem as a fitting process in a probabilistic framework. This idea was developed by [Szeliski and Terzopoulos, 1991] and [Staib and Duncan, 1992]. The contour is redefined as set Θ of parameters and their probability $p(\Theta)$. This probability is then defined high for low energy contours (this is usually done using a Gibbs distribution).

  Then, assuming the probability of the potential force $p(P)$ and that of P knowing the initial contour $p(P/\Theta)$ are known, the probability of Θ knowing P can be obtained through to the Bayes theorem

$$p(\Theta/P) = \frac{p(P/\Theta)p(\Theta)}{p(P)}. \tag{2.6}$$

  This probability is then iteratively maximized to find an optimal estimator θ describing the final contour. This solution is called the *maximum a posteriori*. This framework can also be extended into a time-varying model, assuming a time-varying prior model which can be viewed as the system model of a Kalman filter. This idea

lead to the *Kalman snake* proposed by [Terzopoulos and Szeliski, 1992], where the system model is obtained from the evolution equation of the variational approach.

Probabilistic approaches are well suited to incorporate statistical-based prior knowledge (e.g. resemblance to a reference model) and may eventually give an uncertainty measure of the final fitting, however the probability $p(\Theta)$ is usually hard to define and is based on assumptions that may either be wrong or yield a complex model for which optimization algorithms become computationally very expensive.

All these numerical schemes are somewhat similar, i.e., most problems can be written in either of the three forms. The choice usually depends on their suitability to the constraints imposed by the user (and also on their preference).

2.1.2   Organization of the review

Throughout the literature, two main categories of deformable contours have been proposed, depending on its mathematical representation: *explicit* models, where the contour is expressed in the same dimensionality as the image, i.e., a curve in 2D and a surface in 3D, and *implicit* models, where the contour is defined as the zero level set of a higher-dimensional scalar function defined over the image domain. We shall thus review separately these two categories in sections 2.2 and 2.3 respectively.

In a second step, we shall also distinguish models relying on gradient information and models relying on region information. Indeed, although image-based energy terms are usually not bound to the chosen representation, practical discretization and implementation may vary considerably from one family to the other. Therefore we will also distinguish edge-based and region-based approaches in order to highlight the practical advantages and drawbacks of each family regarding our applications.

2.2   *Parametric* or *explicit* models

Explicit models can be further categorized in two families. The first is based on deformable splines, where a polynomial spline function is fitted to a set of control points using physical elastic properties. The second is based on discrete geometrical models, where no assumption is made on the physical properties of the contour, therefore the control points are simply connected by straight lines. The discrete approach allows more freedom to the contour and usually performs faster since no spline interpolation is involved. It should be noted that, in practice, these two families are equivalent in the sense that discrete models can be derived as approximations of spline models, as it has been shown in [Cohen and Cohen, 1993] and [McInerney and Terzopoulos, 1995a] using finite elements [Huebner et al., 2001].

2.2.1   Contour representations

Deformable splines

Deformable spline models are bound to a parametric formulation, inheriting from the theory of Lagrangian dynamics for multi-dimensional deformable models for model matching in images, introduced in [Terzopoulos, 1986] and [Terzopoulos, 1987]. Contours may be closed or not and are defined by a finite set of control points on which a basis spline

polynomial function of arbitrary degree is fitted to obtain a smooth final contour. The movement of the spline is then driven by the movement of each control point, and the displacement of each control point induces a change in the interpolation process, i.e., a change in the global contour shape. This propagation process can be computationally intensive, and its complexity increases with the number of control points as well as the polynomial degree of the basis function. A simpler interpolation process was proposed by [Cipolla and Blake, 1990], where the spline is replaced by a linear combination of local-support *B-splines* (acronym for short basis splines). The equivalence between these two representations with respect to degree and smoothness has been proved in [de Boor, 2001]. Hence, with local support, the movement of one control point only affects the control points belonging to the same support, yielding faster interpolation. For the same reason, B-splines and their variants (Bezier, Nurbs etc.) are very popular in other domains such as computer graphics, especially in computer-assisted design [Farin, 1992].

From a mathematical point of view, a contour Γ evolving in an image domain Ω is thus represented in 2D by a curve $\mathcal{C}(p)$ or in 3D by a surface $\mathcal{S}(p,q)$ such that

$$
\begin{aligned}
\mathcal{C} : \quad & p \in [0,1] & \longmapsto \Omega \subset \mathbb{R}^2, \\
\mathcal{S} : \quad & (p,q) \in [0,1]^2 & \longmapsto \Omega \subset \mathbb{R}^3,
\end{aligned}
$$

where p, q cover the parametric domain. The curve and surface are closed under the following conditions:

$$
\begin{aligned}
\mathcal{C}(0) &= \mathcal{C}(1) \\
\mathcal{S}(p,0) &= \mathcal{S}(p,1) & \forall p, \\
\mathcal{S}(0,q) &= \mathcal{S}(0,r) & \forall (q,r), \\
\mathcal{S}(1,q) &= \mathcal{S}(1,r) & \forall (q,r),
\end{aligned}
$$

and open otherwise. During the initialization step, a set of control points is defined to represent the parametric domain. Conversely, the energy terms in equation (2.2) express the contour deformation in the same parametric domain, thereby inducing the spline deformation iteratively through its control points. For visualization purposes, the spline basis function has to be fitted after every iteration in order to be rendered on screen.

The first deformable contour method inheriting this theory is the well-known *snake* model introduced by [Kass et al., 1988] for 2D images, and by [Terzopoulos, 1987] for 3D images. The internal energy in equation (2.2) expresses the two first Tikhonov regularizing terms, i.e., the two first contour derivatives, yielding the following energy functional:

$$
E(\mathcal{C}) = \int_0^1 w_1 \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 + w_2 \left| \frac{\partial^2 \mathcal{C}}{\partial p^2} \right|^2 - P(\mathcal{C}) \; dp \tag{2.7}
$$

in the 2D case, and

$$
E(\mathcal{S}) = \int_0^1 w_1 \left| \frac{\partial \mathcal{S}}{\partial p} \right|^2 + w_2 \left| \frac{\partial \mathcal{S}}{\partial q} \right|^2 + w_3 \left| \frac{\partial^2 \mathcal{S}}{\partial p^2} \right|^2 + w_4 \left| \frac{\partial^2 \mathcal{S}}{\partial q^2} \right|^2 + w_5 \left| \frac{\partial^2 \mathcal{S}}{\partial p \partial q} \right|^2 - P(\mathcal{S}) \; dp \; dq \tag{2.8}
$$

in the 3D case. Through this formulation, the internal energy describes a "membrane - thin plate" physical system, where w_i are nonnegative weights adjusting its mechanical properties. The first order derivative terms make the contour act like a membrane, taking

high values when it is stretched. The second order derivatives make the contour act like a thin plate, taking high values when it is bended. The remainder is a potential force P computed from the image and prior energy terms, taking low values when the contour is far from its goal and high values in the other case.

Following a variational formulation, the problem of minimizing (2.7) or (2.8) is thus equivalent to solving the associated Euler-Lagrange equations

$$\frac{\partial \mathcal{C}}{\partial t} = w_1 \frac{\partial^2 \mathcal{C}}{\partial p^2} - w_2 \frac{\partial^4 \mathcal{C}}{\partial p^4} + \nabla P(\mathcal{C}) = 0 \tag{2.9}$$

in 2D, and

$$\frac{\partial \mathcal{S}}{\partial t} = w_1 \frac{\partial^2 \mathcal{S}}{\partial p^2} + w_2 \frac{\partial^2 \mathcal{S}}{\partial q^2} - w_3 \frac{\partial^4 \mathcal{S}}{\partial p^4} - w_4 \frac{\partial^4 \mathcal{S}}{\partial q^4} - w_5 \frac{\partial^4 \mathcal{S}}{\partial p^2 \partial q^2} + \nabla P(\mathcal{S}) = 0 \tag{2.10}$$

in 3D. Note that the potential force can basically hold any kind of information, denoting the genericity of the framework.

Discrete models: a simpler alternative

It can be argued that the spline interpolation process has several drawbacks for image segmentation. Indeed, a spline defines a shape with certain elastic properties that do not necessarily apply to the target. Moreover, the shape of the final smooth contour depends directly on the polynomial basis function, and may not even reflect the real shape of the segmented target if wrong degree or weights are chosen. The same problem arises if one wishes to change the number of control points during the evolution, so as to increase or decrease the precision in given regions of the image (process called *reparametrization*). Changing the number of control points usually implies to change the weights of the basis function, yielding substantial genericity and reproducibility issues.

For many of the reasons stated above, several methods abandoned spline-based models and work at all times with a discrete contour, i.e., a set of points joint by straight lines. Mathematically speaking, this is equivalent to working under a discrete formalism with local deformation influence. A 2D curve is thus represented by a polygon, and a 3D surface by a *polygonal mesh*. Without interpolation, there is no more difference between the real and visible contour. Although 2D discrete models perform equivalently to spline versions, the complexity and speed gain is mostly significant in 3D, yielding popular interest in applications where real-time visual feedback and/or user-interaction is required, e.g. for simulated surgery [Cotin et al., 1999].

The contour is defined only in the image domain instead of the parametric domain, and the control points are renamed to *vertices*, by analogy with the computer graphics terminology. The deformation is somewhat similar to spline-based models, since the minimization of the energy functional induces a deformation force that is applied to the vertices. The main difference is that the energy is defined at all times in a discrete fashion as the sum over all vertices of an energy functional specific to each vertex v_i. This gives a functional very similar to the general equation (2.2)

$$
\begin{aligned}
E(\Gamma) &= \sum_{v_i \in \Gamma} E(v_i), \\
E(v_i) &= \lambda\, E_{external}(I_0, v_i) + \mu\, E_{internal}(v_i) + \gamma\, E_{prior}(v_i).
\end{aligned}
$$

External and priori energy terms are similar to the parametric case, since they will induce the same deformation on the contour vertices than they do on spline control points. However, since the contour is not defined on a parametric domain anymore, the regularization scheme employed above has to be transcribed into a discrete formalism that does not necessarily resemble to that of the continuous domain, especially if no elastic property is given to the model.

2.2.2 Edge-based models

The original *snake* model

In the original model of [Kass et al., 1988], the potential force P is based on gradient information and is given by

$$P(\mathcal{C}) = \lambda \; g(I_0(\mathcal{C})), \tag{2.11}$$

where λ is a weighting parameter and g is an edge detector, expressed by any positive non increasing function of a gradient map of the image. A frequently used candidate is

$$g(I_0) = \frac{1}{1 + \alpha |\nabla(G_\sigma * I_0)|}, \tag{2.12}$$

where α is a nonnegative parameter and G_σ is a Gaussian kernel of standard deviation σ. Its convolution with the original image is used to smooth small noise artifacts that could otherwise be interpreted as object boundaries. This function acts as an edge detector since it takes high values when the contour sits on high image gradients, and thus will stop the contour when it reaches boundaries in the image. An illustration of the model on the circle image is presented in figure 2.4.
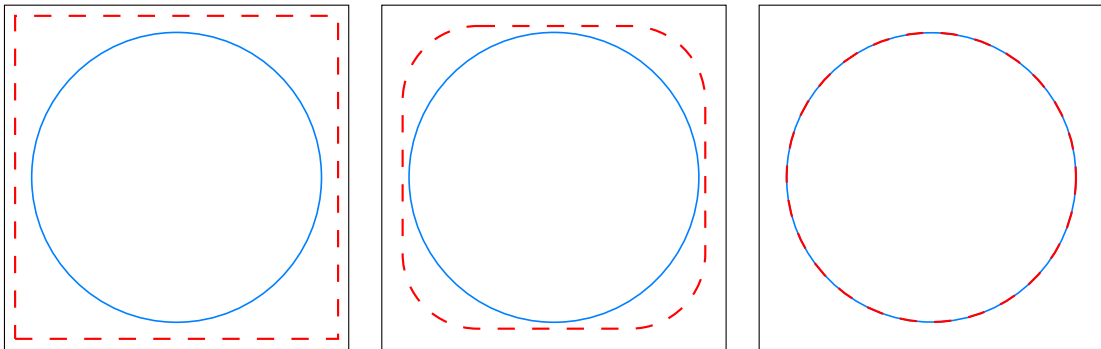
Figure 2.4: Evolution of the snake model of [Kass et al., 1988]. The initial contour (outer most dashed line) progressively shrinks due to the regularization term until the image gradients (depicted in figure 2.3) attract the contour toward the target. At this point, if the weight of the image force is chosen stronger than that of the regularization force, the contour will stop.

Although the edge detector relies on a gradient map of a smoothed image, the contour may still be attracted by spurious edges and small artifacts in the image. Also, the external energy term depends directly on the gradient magnitude, thus high gradients will move the contour faster than low gradients, causing it eventually to oscillate around the optimal solution.

In [Dinten and Trouve, 1992] and [Bossart et al., 1997], it was suggested to replace the gradient detector g by a Laplacian of the image, allowing the user to impose whether the

contour should be attracted by interior or the exterior of the detected edges. It was also suggested to include a directional operator that aims at matching an elongated roof-shaped rectangular template resembling to a natural edge, thereby improving the discrimination between real and spurious edges in the image. The drawback of this operator is the computational complexity in its incorporation to the model. Finally, the appealing property of the approach is to embed the segmentation process in a multi-scale environment. The contour is first deformed on a strongly smoothed image in order to find approximately the most significant edges. The deformation is then iteratively repeated on less smoothed version of the same image, which provides a good initialization and improves the contour localization while reducing the influence of surrounding noise.

Despite its efficiency, edge information only has a local influence, meaning that the contour cannot be attracted by distant edges and has to be initialized quite close from the target edge. Originally, this model was proposed as a tool for manual assisted segmentation, where the user would pull the control points toward the target, until the contour gets close enough so that gradient forces would "lock" onto to the target. In a context where automatization is required, the contour needs thus to be initialized close to the target, requiring a complex pre-processing step.

The *Balloon* model

The *Balloon* model was introduced in [Cohen, 1991] and originated from the idea of inflating the contour as would be done with a balloon. This inflation offers two advantages over the original *snake* model. First, the contour would be less trapped in local energy minima, especially on noisy images. Then, the contour would be able to reach target features far away from its initial position with more robustness regarding contour initialization.

The inflation is expressed by a new term in the energy functional that pulls the contour inward or outward (i.e., following its local curvature). Additionally, they normalized the external energy potential such that the contour movement does not depend anymore on the gradient magnitude. The evolution equation (2.9) becomes

$$\frac{\partial \mathcal{C}}{\partial t} = w_1 \frac{\partial^2 \mathcal{C}}{\partial p^2} - w_2 \frac{\partial^4 \mathcal{C}}{\partial p^4} + w_3 \frac{\nabla P(\mathcal{C})}{|\nabla P(\mathcal{C})|} + k \overrightarrow{N}(\mathcal{C}) = 0, \qquad (2.13)$$

with $\overrightarrow{N}(.)$ being the normal unit vector to the contour pointing outward and k being a scalar parameter determining the strength of the balloon force, while its sign determines whether the contour will inflate or deflate. Since the snake model has an intrinsic deflating capability, the balloon is usually employed to inflate the contour. An illustration of this force is given in figure 2.5.

The advantage of this model is that the contour may now be initialized almost anywhere in the image. It can however not be used in an automated manner. Indeed, the balloon energy term is not linked to the image data, inducing two drawbacks. Firstly, the direction of the balloon force has to be manually adjusted according to the position of the initial contour relatively to the target: if the contour is initialized outside (resp. inside) the target, k must be given a negative (resp. positive) value. Secondly, if the contour is initialized across the target (a portion would be inside and another outside), the balloon force will pull one of the portions toward the target, while the other portion, following the same force, will move far away from its goal.
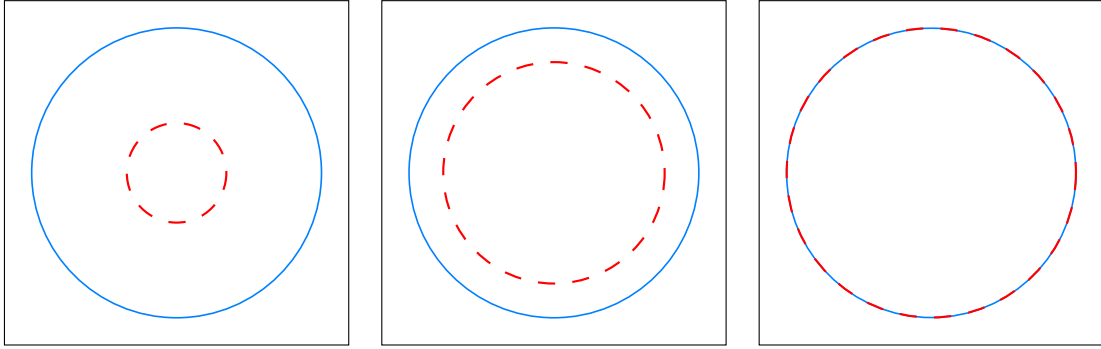
Figure 2.5: Impact of an inflating balloon force on the contour. Since the contour is initialized inside the target, image and regularization forces are not sufficient to attract the contour in the right direction. The balloon force helps inflate the contour until the image forces stop its evolution as the desired location.

The *Geodesic* active contour model

[Caselles et al., 1995] argued that the Snake model had three major drawbacks, and proposed a model that corrected these problems: the *Geodesic Active Contour* model.

- They first stated that the second regularization term in equation (2.7) forbids second-order discontinuities to the contour, hence, the contour cannot fit cusps and corners correctly. Moreover, they showed that this term was redundant with the first since both minimize the contour curvature. Hence, similar results can be obtained by setting $w_2 = 0$, allowing second-order discontinuities to the contour while keeping the contour smooth.

- They showed that the segmentation result depends on the contour parameterization (the choice of the control points). Hence, a different parameterization of the same initial contour may yield a different result. To avoid this dependency, they used *Maupertuis principle* [Dubrovin et al., 1991] to prove that the solution of the snake model is a geodesic curve in a Riemannian space induced by the image I_0, i.e., a minimal distance path between given points of the image. Their energy functional reads

$$E^{\text{GAC}}(\mathcal{C}) = \int_0^1 g(I_0(\mathcal{C})) \left| \frac{\partial \mathcal{C}}{\partial p} \right| \, dp. \tag{2.14}$$

By letting $|\partial \mathcal{C}/\partial p| \, dp = ds$, where ds is the elementary Euclidean distance, they expressed the Euclidean curve length by

$$L(\mathcal{C}) = \int_0^1 \left| \frac{\partial \mathcal{C}}{\partial p} \right| \, dp = \int_0^{L(\mathcal{C})} ds, \tag{2.15}$$

and rewrote (2.14) as

$$E^{\text{GAC}}(\mathcal{C}) = \int_0^{L(\mathcal{C})} g(|I_0(\mathcal{C})|) \, ds. \tag{2.16}$$

[Caselles et al., 1995] shown that minimizing this new functional is equivalent to minimizing the original energy, with the additional advantage of being independent of the contour parameterization since it does not directly involve the parametric variable p. Note that equation (2.16) is also valid in 3D, provided ds is interpreted as the elementary surface.

- Due to its parametric formulation, the contour has a rigid topology, in the sense that it cannot split into several contours, nor can several contours merge. This is a substantial limitation in applications where an unknown number of objects are to be detected in a single image. This problem is not solved here, however the authors proposed a level set formulation of this model, as we shall see in section 2.3.

The geodesic model is more robust than the original snake model, but the problem of reaching distant edges is not addressed, since the balloon term of [Cohen, 1991] cannot be introduced within this explicit formalism.

The *Gradient Vector Flow* models

The main drawback of the previously mentioned balloon model clearly arise from the fact that there is no link between the inflation force and the image data. A preferable solution would pull the contour automatically from any initial position in the right direction. This is the key idea behind the *Gradient Vector Flow* models originating from [Xu and Prince, 1997] and developed in [Xu and Prince, 1998],[Xu and Prince, 2000] and [Ray and Acton, 2005]. The principle is to propagate the gradient information of the target over the whole image domain, producing a vector field that would attract the contour from any initial position (see figure 2.6).

| (a) | (b) |
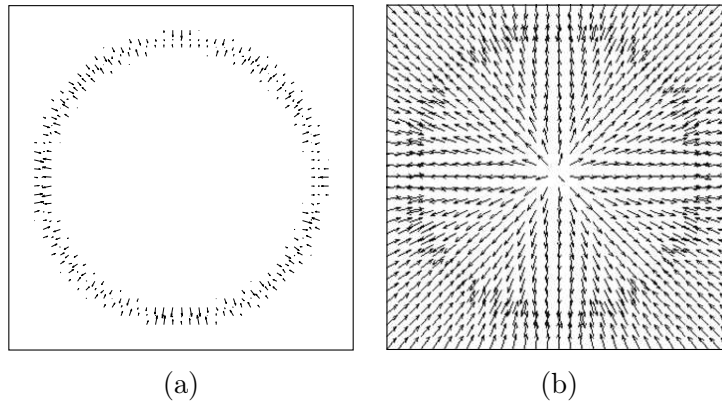
Figure 2.6: Comparison of a classical gradient map (a) and the gradient vector flow (b). The gradient vector flow is obtained by propagating the gradient information iteratively over the entire image domain, so as to attract a contour initialized far away.

The gradient vector flow V is a vector field of same dimensionality as I_0, and is defined as the equilibrium solution of the vector diffusion equation

$$V_t = g(|\nabla I_0|)\nabla^2 V - h(|\nabla I_0|)(V - \nabla I_0), \tag{2.17}$$

with the initial condition $V_0 = \nabla I_0$. g is a monotonically non-increasing function weighting the smoothness of the vector field, while h is a monotonically non-decreasing function weighting the data attachment term (i.e., how close the field should resemble to the initial gradient map ∇I_0). Among the many candidates for g and h, a good choice is to set $h = 1 - g$, while g can be given, e.g., by

$$g(|\nabla I_0|) = exp\left[-(\mu|\nabla I_0|)^2\right] \tag{2.18}$$

The resulting vector field will be equivalent to ∇I_0 on edges, and smoother as one moves away from edges. The scalar parameter μ weights the balance between smoothness and resemblance to the original gradient.

This model is more powerful and robust than the snake and balloon models, yielding a great interest for biomedical applications, e.g., [Xu et al., 1999], [Ray et al., 2002], [Tang and Acton, 2004] and [Tang et al., 2006]. The model performs similarly on the examples given in Figs. 2.4 and 2.5, nonetheless [Ray and Acton, 2005] pointed out that errors may persist for special cases of initialization, as illustrated on the top row of figure 2.7. In such cases, they modified the model and proposed the *enhanced gradient vector flow* (EGVF) where the gradient is propagated only outside the initial contour, as illustrated on the bottom row of figure 2.7. This version tailors the flow specifically to the initial contour, allowing a more generic initialization, but the contour must necessarily be initialized inside the target, otherwise no gradient is propagated.
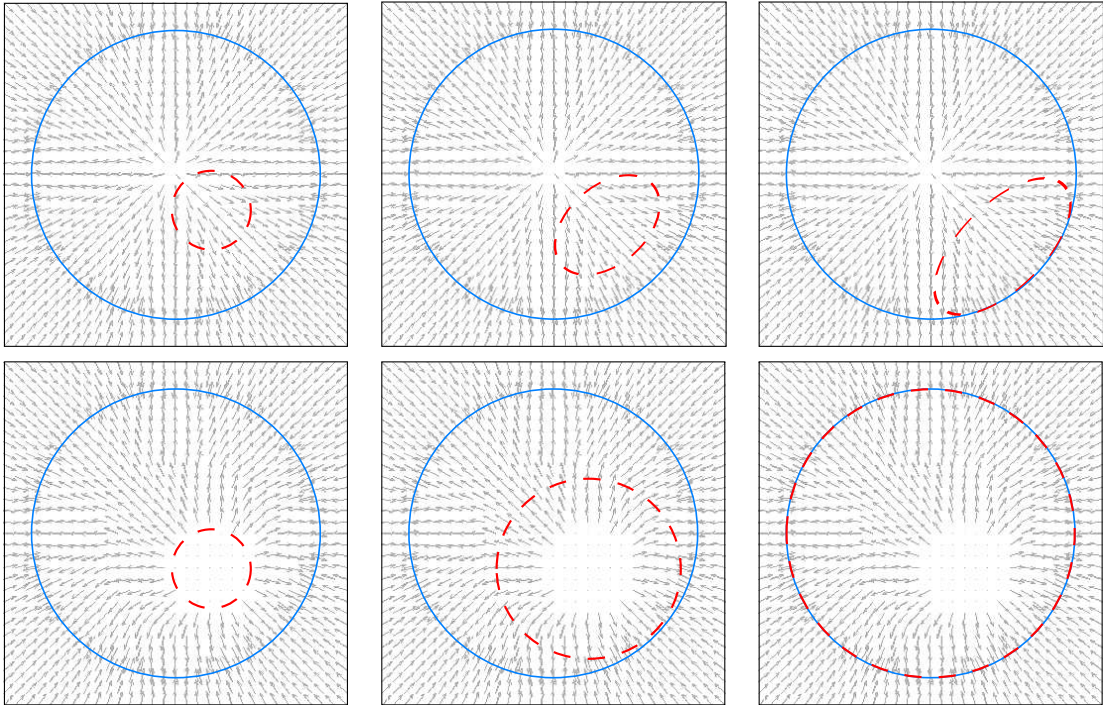
Figure 2.7: Comparison between the GVF (top row) and the EGVF (bottom row) for a particular initial contour. Since the GVF does not depend on the initial contour, its deformation is erroneous as long as it does not contain the flow source (here the circle center). The EGVF propagates gradient everywhere but inside the initial contour, yielding better behavior.

2.2.3 Region-based models

The efficiency of edge-based models relies on the assumption that the targets have a boundary defined by high image gradient. Whereas this is commonly the case for 2D images, this assumption is not true anymore for 3D images acquired in fluorescence microscopy, where the data is significantly blurred along the depth direction, as illustrated in figure 1.2. Therefore, the gradient magnitude is very low along the depth axis comparatively to the others, such that edge-based methods would fail to detect the target correctly, as

illustrated in Figure 2.8 (top row). To solve this problem, another part of the literature relies on region information instead of edge information, as we shall see below.

Region information has been mostly employed in the fields of image restoration and denoising, where one often wishes to simplify an image by smoothing it everywhere but on edges, producing a piecewise smooth restored image. The target regions can then be easily extracted from the restored image, potentially yielding a similar result than with an edge-based approach, implying a duality between region and edge information. Hence, the problem is solved globally (at the image scale) rather than locally (at the pixel scale), yielding more robustness to the method.

Most of the literature on region-based segmentation expresses the problem as a statistical classification problem. Some methods have been adapted to the formalism of deformable models, but they are computationally expensive. We rather look for a method that does not involve probability laws, and that solves a simpler problem with the least prior knowledge as possible. In this context, the only remaining candidate that is compatible with a contour-based representation is the Mumford-Shah functional [Mumford and Shah, 1989].

The Mumford-Shah functional

The Mumford-Shah (or *cartoon*) model attempts to simplify an image I_0 into a piecewise-smooth approximation (referred to as *cartoon* image) $\hat{I}$ by minimizing the following multidimensional energy functional:

$$E^{MS}(\hat{I}, K) = \int_{\Omega} |\hat{I} - I_0|^2 d\omega + \nu \int_{\Omega - K} |\nabla \hat{I}|^2 d\omega + \mu \mathcal{H}^{N-1}(K), \qquad (2.19)$$

where $\hat{I}$ is the final *cartoon* image restored from I_0, K is the set of discontinuities in $\hat{I}$, and ν and μ are non-negative weighting parameters. The first term states that $\hat{I}$ should stay similar to I_0, while the second term tends to smooth $\hat{I}$. Finally, the latter term ensures that the set K of boundaries between the regions is minimal. Concretely speaking, this is done by minimizing the $(N-1)$-dimensional Hausdorff measure of K.

A particular case of this functional is obtained by making a constant assumption on each restored region R_i of $\hat{I}$. Under this assumption, the second term in equation (2.19) vanishes, and [Mumford and Shah, 1989] showed that the constant value of each region R_i is actually equal to the average value c_i of I_0 on that region. Hence, equation (2.19) can be rewritten

$$E^{MSc}(c_{1..n}, K) = \sum_{i=1}^{n} \left[\int_{R_i} |c_i - I_0|^2 d\omega \right] + \mu \mathcal{H}^{N-1}(K), \qquad (2.20)$$

where n is the number of regions in $\hat{I}$. This equation is known as the minimal partition problem.

Simplified Diffusion Snakes

To our knowledge, the first implementation of the Mumford-Shah functional under the parametric formalism is the 2D *simplified diffusion snakes* model of [Cremers et al., 2002]. The author embeds the functional in a probabilistic framework in order to incorporate shape prior knowledge on the target to cope with occlusions. Here we will focus only on the image-based energy of the model. The discontinuity set K is replaced by a closed

spline C, and the last term in equation (2.20) is expressed as the L_2 norm of C, yielding

$$E^{SDS}(c_{1..n}, \mathcal{C}) = \sum_i^n \left[\int_{R_i} |c_i - I_0|^2 d\omega \right] + \mu \int_0^1 \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 dp. \tag{2.21}$$

Since the model uses a single closed curve to detect the target, the image is divided into two regions (the target and the background components), reducing equation (2.21) to two phases:

$$E^{SDS2}(c_1, c_2, \mathcal{C}) = \int_{R_1} |c_1 - I_0|^2 d\omega + \int_{R_2} |c_2 - I_0|^2 d\omega + \mu \int_0^1 \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 dp \tag{2.22}$$

where R_1 and R_2 indicate the image regions inside and outside C, or vice-versa, as defined by the user. The corresponding evolution equation is given by

$$\frac{\partial \mathcal{C}}{\partial t} = \left[(c_2 - I_0)^2 - (c_1 - I_0)^2 \right] \overrightarrow{\mathcal{N}} + \mu \int_0^1 \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 dp \tag{2.23}$$

where $\overrightarrow{N}$ being the unit vector normal to C pointing outward. The bottom row of figure 2.8 shows the result of this model on the fuzzy spot image that edge-based model fail to detect as the contour passes through the object boundary and keeps shrinking to a point.

| (a) | (b) | (c) | (d) |
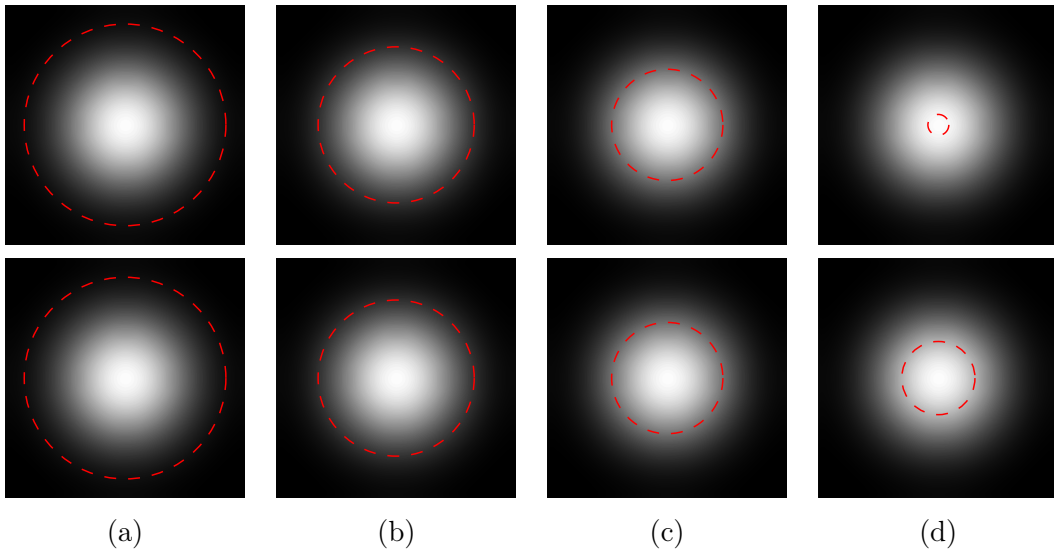
Figure 2.8: Comparison of edge-based model (top row) and region-based model (bottom row) on an image with weak gradients. Initial contour (a), contour evolution (b,c,d). Without edge information, the contour passes "through" the object boundary without stopping.

This model is particularly well adapted to our segmentation context, since biological entities have a relatively constant intensity, so has the background. However, several remarks should be noted:

- Firstly, it should be noted that, in contrast to edge-based models, the energy now has to be minimized with respect to C but also with respect to $\hat{I}$, e.g. here c_1 and c_2. Therefore, for every iteration, c_1 and c_2 must be evaluated with respect to C, then C

must be deformed with respect to c_1 and c_2. While the latter operation is present in edge-based models, the former increases the computational time cost of the method since a full sweep through the entire image domain must be done at every iteration to compute R_1 and R_2, and subsequently c_1 and c_2. The complexity now reaches $O(M^N)$ for an image of size M in each dimension. A solution is to perform the update of c_1 and c_2 less frequently than C, which will yield similar results provided the initialization is sufficiently good.

- Secondly, the model is not well suited for multiple object tracking. Let us assume that two distinct contours are tracking two targets that move into contact during a video sequence. For edge-based models, the targets will exhibit a common edge at the contact point. Hence, the contours are likely to stop at the correct location. With the current model however, the energy functional (2.21) states that all regions having similar intensity are considered as in the same phase. Therefore, as soon as the two targets touch, they appear as one single connected component, and the two contours will absorb each other as to minimize the energy, resulting in a doubly wrong segmentation, as illustrated in figure 2.9.

Coupled Parametric Active Contours

To avoid contours to absorb each other, they must have mutual knowledge of their location. This can be done by linking multiple contours through the energy functional. This is the base idea of the *coupled parametric active contours* of [Zimmer and Olivo-Marin, 2005]. The new model is somewhat similar to the simplified diffusion snakes above, although the regularization term is replaced by that of the original snakes model. Assuming n targets are being tracked by n splines, with a final phase R_0 representing the background with average intensity c_0, they define the energy functional as

$$
\begin{aligned}
E^{\text{ZO}}(\mathcal{C}_{1..n}, c_0, c_{1..n}) &= \sum_{i=1}^{n} \int_0^1 w_1 \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 + w_2 \left| \frac{\partial^2 \mathcal{C}}{\partial p^2} \right|^2 dp + \lambda_1 \int_{R_i} |c_i - I_0|^2 d\omega \\
&+ \lambda_2 \int_{R_0} |c_0 - I_0|^2 d\omega \\
&+ \gamma \sum_{i=1}^{n} \sum_{j=i+1}^{n} \int_{R_i \cap R_j} d\omega.
\end{aligned}
\tag{2.24}
$$

The latter term (weighted by γ) expresses an overlap penalty between each pair of contours, and is equal to the image area contained in each intersection $R_i \cap R_j$. This energy functional is the parametric counterpart of the level set formulation published earlier in [Zhang et al., 2004]. All contours are now linked by pairs through the energy functional, and several targets can be tracked simultaneously, even if they belong to the same phase. The evolution equation for each contour is given by

$$
\frac{\partial \mathcal{C}_i}{\partial t} = w_1 \frac{\partial^2 \mathcal{C}_i}{\partial p^2} - w_2 \frac{\partial^4 \mathcal{C}_i}{\partial p^4} - \left[\lambda_1 (c_1 - I_0)^2 - \lambda_2 (c_2 - I_0)^2 + \gamma \sum_{j \neq i} \xi_j(\mathcal{C}_i) \right] \left| \frac{\partial \mathcal{C}_i}{\partial p} \right| \overrightarrow{N_i}, \tag{2.25}
$$

where ξ is an indicator function such that $\xi_j = 1$ for control points of $\mathcal{C}_i$ located inside $\mathcal{C}_j$ and 0 otherwise, and $\mathcal{N}_i$ is the normal vector pointing outside $\mathcal{C}_i$.

frame $T = 0$ frame $T = 1$ frame $T = 2$
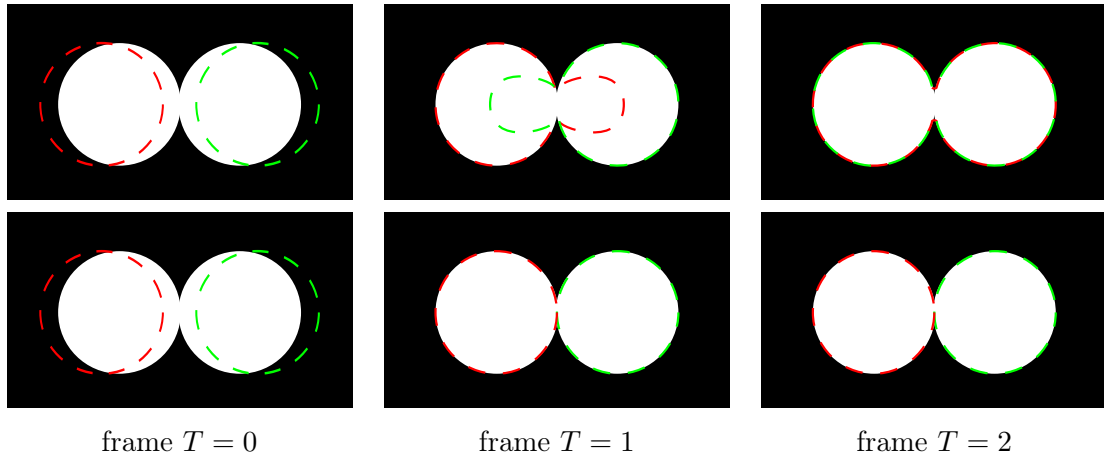
Figure 2.9: Impact of the coupling term for multi-object tracking. (Top row) Without coupling, the contours have no mutual knowledge, therefore if the two objects belong to the same phase, the contours penetrate each other. (Bottom row) Coupled contours aim at minimizing the overlap between them and hence stop the deformation at the collision point.

The drawback of the coupling term is that it links contours by pairs without involving the underlying image data. As a result, contours will indeed stop when they collide, but there is no guarantee that the frontier between colliding contours lies on the actual frontier between the real objects. figure 2.9 illustrates an ideal scenario where the two objects move toward each other at the same speed and in a straight direction, therefore the object portion that each contour must track is identical. In practice, more complex scenarii take place, where the objects may move with different speeds and directions. Since contours do not practically distinguish objects of the same phase, they may absorb a portion of the wrong object before collision occurs, yielding erroneous results (cf. figure 2.10). We shall propose in chapter 4 an extension of this model to 3D images, while proposing a solution to reduce this phenomenon.

(a)   (b)
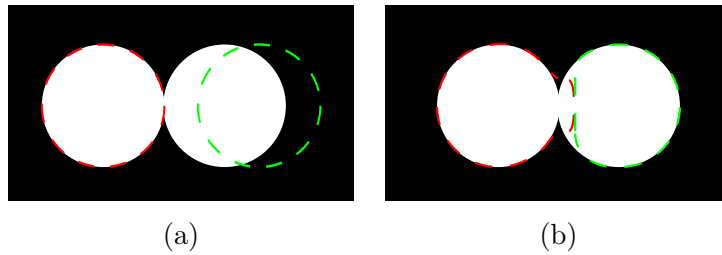
Figure 2.10: Limitation of the coupling term in equation (2.25). Since the coupling term does not involve image data, initial contours (a) may mislocalize the frontier between objects moving at different speed, yielding a wrong result (b).

2.2.4   3D discrete models

From a theoretical point of view, all the models presented above can be straightforwardly extended to 3D. However, the majority relies on a spline formulation, and their practical extension to 3D spline surfaces induces a high complexity and time consumption for many tasks including re-parametrization, manipulation and polynomial fitting.

Early attempts to perform 3D segmentation consisted in stitching consecutive 2D contours extracted from each slice of the data [Fuch et al., 1977], [Ayache et al., 1989], and [Lin et al., 1989]. These approaches have several drawbacks. The first is that a matching operation must ensure the link between two contours belonging to two consecutive image slices. In case several objects are to be reconstructed, ambiguity may yield a wrong matching between the contours. The second drawback is that the final contour dos not "feel" real, in the sense that the reconstructed surface appears discontinuous on each slice throughout the final volume. Finally, it would be profitable to utilize the full volumetric information to improve the performance of the method, especially if consecutive slices are close enough from each other, allowing more precise measurements through image interpolation.

The first 3D approach was proposed in [Terzopoulos, 1987], who made symmetry assumptions on the model in order to simplify the heavy calculations. This approach restricts the method to deal with targets that are homeomorphic to a symmetric object such as a sphere. The model was extended to super quadratics in [Terzopoulos and Metaxas, 1991], slightly augmenting the shape deformation possibilities, although not sufficiently enough for highly deformable shapes. Then again, none of these models are able to handle topological changes in a fast and efficient way.

Discrete models gain all their interest and power in this particular area, since they provide fast computation, efficient visualization (especially in 3D), and allow the introduction of geometrical rules to handle topological changes. The literature on discrete surface representation being quite large, an exhaustive review falls beyond the scope of this thesis. We will rather focus below on the major contributions and applications of deformable mesh models to object segmentation and reconstruction, which mostly relate to medical imaging.

Geometrically Deformed Models (GDM)

One of the first discrete deformable contour is the *Geometrically Deformed Model* proposed by [Miller, 1990][Miller et al., 1991]. This model expresses a 2D curve as a polygon and a 3D surface as a triangular mesh. The external energy drives each vertex toward the target boundary based on intensity difference to a given threshold value. This implies that the target is relatively well-defined by an arbitrary threshold value, and that the sign of the intensity difference between the target and the background is known. The prior energy pulls each vertex in the outer normal direction to the contour, so as to inflate the model. This suggests that the model is initialized inside the target. Finally, the internal energy expresses how the local configuration of each vertex and its neighbors satisfies the initial model topology. In practice, this term tends to homogenize the distance between the vertex and its neighbors by maintaining local edge lengths within an interval computed from the initial local average edge length of the model. In order to increase the mesh resolution when more detail is needed, Miller *et al.* introduced a re-sampling method that subdivides each face into four new faces, as illustrated on figure 2.11.

The major drawback of the GDM framework is that they aim at reconstructing an isosurface of the target, which requires clean data with no intensity variations and high signal-to-noise ratio. Also, the GDM performs local topology handling, but the problem of contour splitting and merging is not addressed. This limitation has been addressed in [Duan and Qin, 2001], where the authors have implemented collision detection in a similar
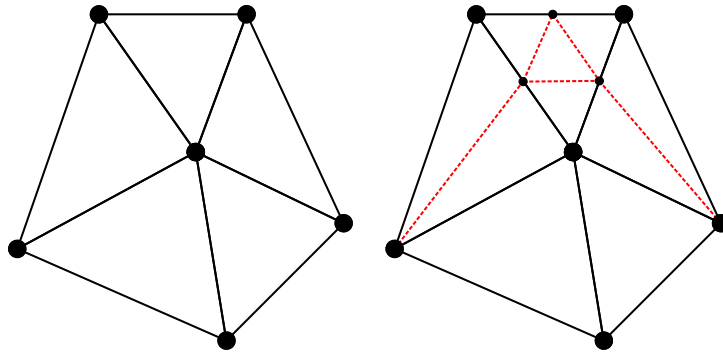
Figure 2.11: Re-sampling schemes in the GDM framework. The re-sampled face is replaced by four new faces, and neighbor faces are adjusted to respect the triangular topology.

way to Lachaud's model in order to detect potential mesh splitting or merging. This new GDM framework has been employed for general reconstruction of objects with arbitrary topology, and applied to volumetric point cloud data [Duan and Qin, 2004].

[Zhukov et al., 2002] also employed a triangular mesh model similar to the GDM for MRI heart segmentation, but they replaced the local re-sampling scheme by a dynamic meshing algorithm presented by [Kobbelt et al., 2000]. This algorithm maintains an evenly sampled mesh during the deformation process by minimizing the valence of each vertex (i.e., the number of neighbors).

Simplex Meshes

Delingette proposed in [Delingette, 1994] and [Delingette, 1999] a very powerful and geometrically sound 3D framework called *simplex mesh*. The surface is represented by a 2-simplex manifold, i.e., a set of interconnected vertices where each vertex has exactly 3 neighbors. It can be noted that this representation is the dual of a triangular mesh with a 6-connectivity neighborhood, as shown on figure 2.12.

Thanks to efficient Eulerian operators and algebraic topology, the model incorporates

(a)   (b)
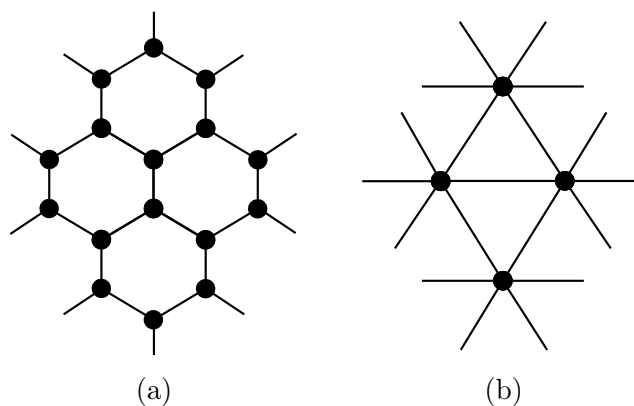
Figure 2.12: Example of a 2-simplex mesh (a), and its dual 2-manifold mesh (b).

both local and global re-sampling methods, introducing the possibility to perform surface splitting and merging. While local re-sampling is performed automatically by subdividing problematic faces and edges (cf. figure 2.13), the drawback is that global topology changes

are only possible through user intervention. The model also proposes an adaptive resolution scheme where more points are concentrated in regions of high curvature, so as to refine the mesh detail in those regions. Nevertheless, this refinement operation is not parallel to the main deformation, and must be performed in a post-processing step, increasing substantially the overall computation time.
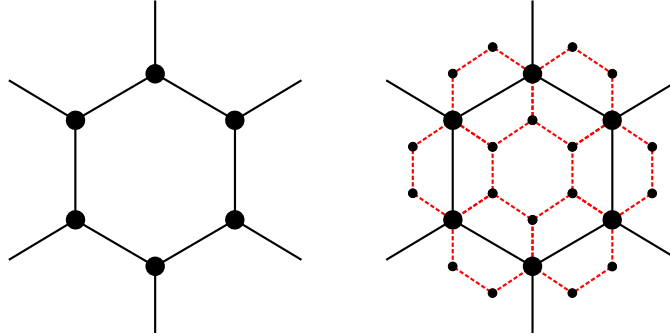
Figure 2.13: Example of surface refinement on a simplex mesh.

Simplex meshes using gradient information together with a shape constraint have been applied to medical image segmentation in [Montagnat and Delingette, 1997] and [Montagnat et al., 2003]. They have also been utilized for computer-based surgery simulation [Cotin et al., 1999], where the driving forces come from user-interaction with real-time realistic physical feedback simulation. More recently, [Giachetti and Zanetti, 2004] and [Bornik et al., 2005] present applications of this model for tubular structures representation in ultrasound imaging.

δ-snakes

Following the work of Delingette, Lachaud *et al.* proposed in [Lachaud and Bainville, 1994], [Lachaud and Montanvert, 1995] and [Lachaud, 1998] a 3D deformable mesh model able to handle global topological changes in a fully automated manner. This model uses a triangular mesh representation and is called δ-snakes, where δ describes the global resolution of the mesh. This parameter is used to compute an edge length interval similar to that of [Miller et al., 1991], suggesting a homogeneous mesh resolution. Local face re-sampling is somewhat similar to the GDM, although the process focuses on problematic edges rather than faces. This yields a slightly different behavior, as illustrated in Figs. 2.14 and 2.15.

In order to speed up computations, the model is embedded in a multi-resolution framework. Therefore, Lachaud separates local and global re-sampling schemes. Local re-sampling is performed on edges instead of faces, as shown in figure 2.14-top, in order to keep all edge lengths within the interval defined around δ. A global subdivision scheme is employed to refine all the mesh faces at the same time, as shown in figure 2.15-bottom, thereby reducing δ by an arbitrary factor (here $1/\sqrt{3}$). Global re-sampling allows to evolve a low-resolution mesh, i.e., with few vertices, in a very fast way. The mesh is then globally refined before it is deformed again to increase the precision when close to the solution.

The δ-snakes have mostly been applied to isosurface reconstruction in medical imaging [Lachaud and Montanvert, 1999], where gradient information has been incorporated to improve the isosurface reconstruction. The same model has also been applied to biological images for object visualization [Benassarou et al., 2002] and tracking [Bittar et al., 2003],
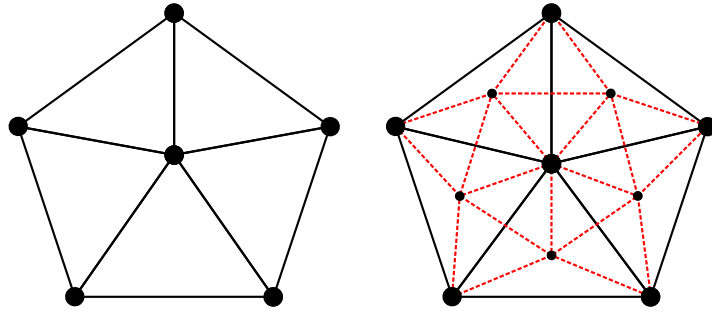
Figure 2.14: δ-snake global refinement scheme. A new vertex is created in the barycenter of each old face, and is connected to the three vertices of the old face as well as the new vertices created in the neighbor faces.
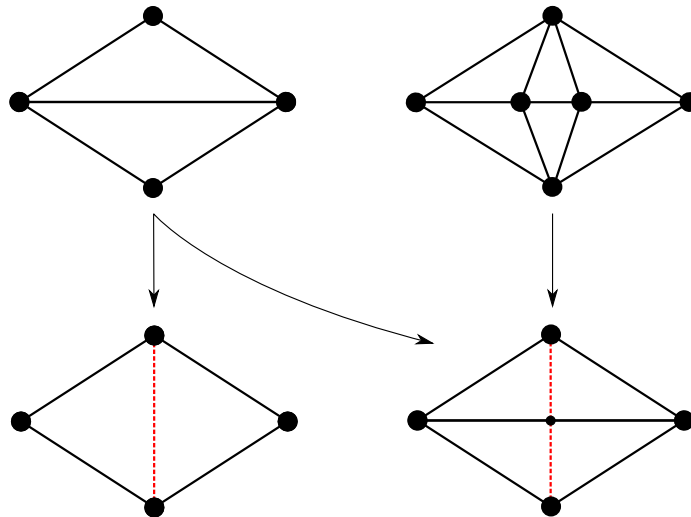
Figure 2.15: δ-snake local re-sampling schemes. If an edge gets too long (top left), it is either split to create 2 new faces or inverted if the resulting edge length is within the distance interval. If an edge gets too short (top right), the two corresponding faces are deleted and the vertices are merged.

however the isosurface extraction is much more problematic in microscopy images than in medical images and requires very good image qualities, as we mentioned before for the GDM framework.

[Zagrodsky et al., 2005] recently proposed a triangular mesh model to track the deformation of the left ventricle in a video sequence. The model has similar characteristics as the δ-snakes, however the topology is rigid since the target is known and initialized by a precise homeomorphic model. The deformation is driven by gradient vector flow information and constrained by the shape of the initial mesh to improve robustness to image noise.

T-snakes: Topologically adaptable snakes

This model was proposed in [McInerney and Terzopoulos, 1995b] for 2D images and extended to 3D in [McInerney and Terzopoulos, 1995a], taking the name of *T-surfaces*. This discrete approach is slightly different from other models, in the way that the vertices are

displaced along a grid partitioning the image space instead of the image space itself. This decomposition allows to represent any closed contour in a non-ambiguous manner, as opposed to non-simplicial decompositions leading to topological ambiguities, e.g., the marching squares in 2D and the marching cubes in 3D [Lorensen and Cline, 1987] (see figure 2.16). Typical simplicial methods are the Coxeter triangulation [Coxeter, 1963] in 2D, and the marching tetrahedra algorithm [Payne and Toga, 1990][Gueziec and Hummel, 1995] in 3D.

non-simplicial decomposition   simplicial decomposition
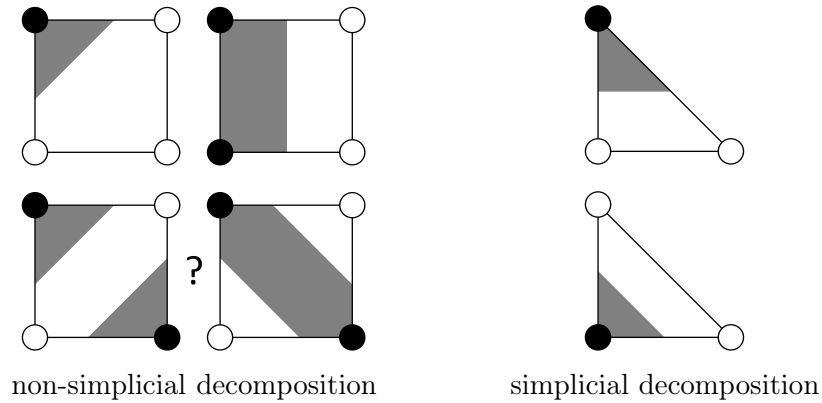
Figure 2.16: Contour approximation using a non-simplicial (square) and a simplicial (triangle) decomposition. In the non-simplicial case, a diagonal layout of the interior points yields a topological ambiguity. This situation cannot occur in the simplicial case. Black dots are inside the contour, and white dots are outside. Note that all other combinations can be obtained by symmetry or duality from these examples.

The T-snake model employs a Coxeter-type triangulation, later renamed *Affine Cell Image Decomposition* or *ACID* framework in [McInerney and Terzopoulos, 2000]. This triangulation is obtained in 2D by performing a first decomposition of the image into square-shaped cells, followed by a uniform subdivision of each cell into two triangular faces. The analogous process in 3D decomposes the image into cube-shaped cells, subsequently divided into six tetrahedra. The nodes of the snake are then obtained by computing all intersections of the initial contour with the grid, as shown on figure 2.17.
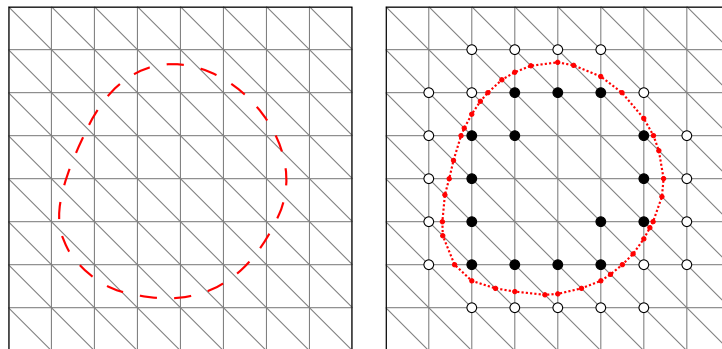
Figure 2.17: Simplicial decomposition and corresponding approximation of a contour. The control points of the snake (in red) evolve on the simplicial grid.

The advantage of evolving the nodes on a grid with non-ambiguous contour approximation immediately allows the detection of contour self-intersection, splitting and merging, simply by monitoring the state of consecutive grid nodes. If the grid topology is no longer

respected, the contours are re-parametrized without ambiguity, as shown on figure 2.18.
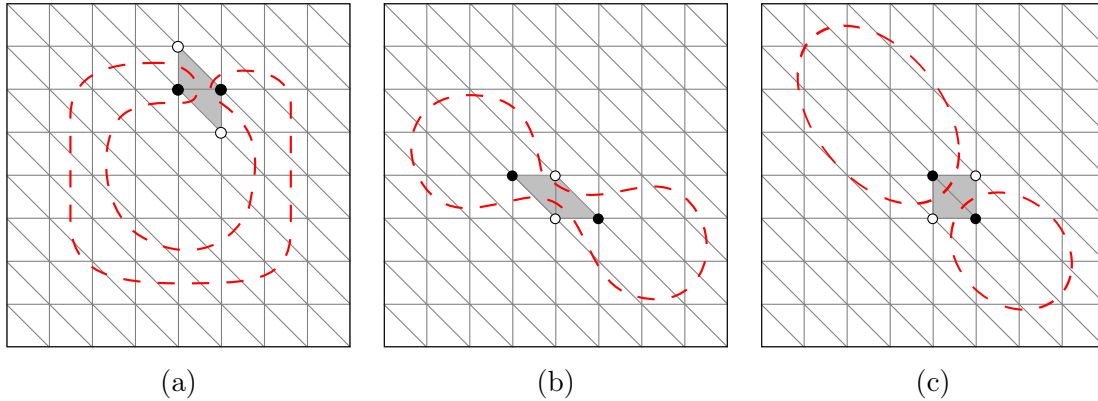
Figure 2.18: Detection of topological breakdowns in the ACID framework. (a) self-intersection, (b) splitting, (c) merging.

The main drawback of the method is its dependence to the simplicial grid resolution. Since the resolution is fixed, topology is checked in every simplex of the grid, yielding a high computational cost for fine grids (especially in 3D). T-snakes and T-surfaces have been mostly employed for medical image segmentation, and more particularly for MRI brain images [McInerney and Terzopoulos, 1997] and [McInerney and Terzopoulos, 1999]. They have been later replaced in [Park et al., 2001] by a triangular mesh model similar to Lachaud's. In [Giraldi et al., 2000], a dual approach is suggested where a first T-snake is initialized inside the target while another one is initialized outside, in order to avoid local energy minima.

2.3 *Level set* or *implicit* models

2.3.1 From fluid dynamics to images

Level set models appeared in image analysis later than snakes, but they have an older origin. The idea of level sets was originally motivated by the work of [Sethian, 1985], who encountered some of the limitations mentioned in the previous section while studying physical phenomena such as crystal growth and flame propagation.

Let us give an example to outline these limitations. A sheet of paper is being burnt by a flame at two separate locations under the sheet, and we wish to monitor the evolution of the flames and their propagation on the sheet (see figure 2.19). Now let us assume that a parametric curve is assigned to track each burnt region on the sheet. As the paper progressively burns, the outline of the burnt regions come closer to each other. When close enough, the paper lying between the fronts burns out, and the two flame fronts become one. The first problem when employing a parametric curve is that one can only monitor the shape of the flame front, not the flame itself (being of one dimension higher than the image). The second problem is linked with the topological issues mentioned earlier. Indeed, as the two burnt regions merge, the two flames also merge into a unique fluid entity.

This situation, as many others, cannot be featured by parametric curves since they are not able to merge. This yields a great limitation for studying fluid dynamics.
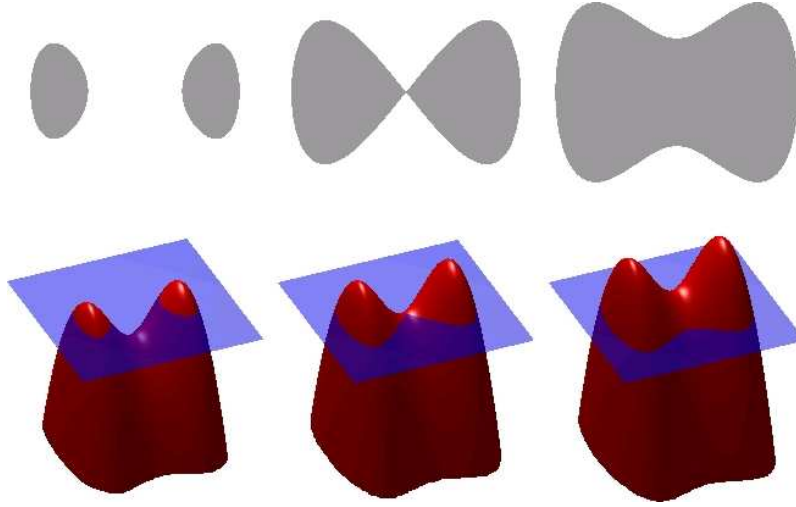
Figure 2.19: Principle of level sets in 2D. Instead of propagating one curve per region and face incorrect topology as the regions merge, the fronts are embedded into a single time-varying surface (called level set function) that intersects the image plane at the regions boundaries. This intersection is called the zero level set.

The solution proposed by [Osher and Sethian, 1988] is to track the fronts *implicitly* by representing mathematically the flames themselves. Assuming that the two flames are in fact a unique gas fluid entity somewhere under the paper sheet, the two evolving fronts actually correspond to the unique intersection set between the fluid entity and the paper surface. The flame entity can thus be modeled by an unknown time-varying Lipschitz function φ (also called level set function) bounded by the domain of consideration $\Omega \subset \mathbb{R}^N$:

$$\varphi : \Omega \times \mathbb{R} \longmapsto \mathbb{R} \tag{2.26}$$

where N is the dimension of the front domain (the paper sheet in our example). By convention, it is commonly imposed that

$$\forall (x_1, \cdots, x_N) \in \Omega \subset \mathbb{R}^N, \forall t \begin{cases} \varphi(x_1, \cdots, x_N, t) > 0 & \text{outside the front} \\ \varphi(x_1, \cdots, x_N, t) < 0 & \text{inside the front} \\ \varphi(x_1, \cdots, x_N, t) = 0 & \text{on the front.} \end{cases} \tag{2.27}$$

Therefore, at any time of the flame propagation, the front(s) may freely evolve, eventually merging (and respectively splitting) without extra calculations (hence the *implicit* naming), and can be obtained at any time t_i of the evolution by retrieving the zero-level of φ

$$\tilde{\varphi}(t_i) = (x_1, \cdots, x_N) \in \Omega \subset \mathbb{R}^N, \varphi(x_1, \cdots, x_N, t_i) = 0. \tag{2.28}$$

Level set based methods are based on the principle and theory of front propagation under curvature-dependent speed. This type of movement is commonly found in various problems involving the evolution of incompressible fluid dynamics, and has been studied in earlier work by [Dervieux and Thomasset, 1981]. Later, a relation between the propagation of level set curves and that of wave surfaces in some geometric optic models has been demonstrated in [Barles, 1985]. The propagation principle can be easily verified with our flame example: assume that two regions are being burnt, one has a finger shape, while

the other is flat. It is clear that the finger-shaped region will burn faster than the flat-shaped region, since the finger-shaped region is almost completely surrounded by a high-temperature region, characterized on the finger extremity by a high curvature of the flame front.

The curvature-dependent propagation of the level set curves is originally expressed by an equation of Eikonal type

$$\frac{\partial \varphi}{\partial t} = F(\kappa)|\nabla \varphi|, \tag{2.29}$$

where F is a speed function and κ the curvature of φ. This equation belongs to the family of Hamilton-Jacobi equations, since φ does not directly appear in the equation, only its derivatives.

The analogy with image segmentation becomes straightforward. Instead of defining one (or more) initial curves on the image, a single level set function is assigned over the whole image domain, with initial conditions in respect to the chosen convention. The contour or front disappears from the energy functional, since it is implicitly embedded in the level set function, and the regularization terms of the parametric formulation expressing the first derivatives of the front are replaced by the curvature of φ. figure 2.20 show an example of level set front evolving exclusively according to its curvature.
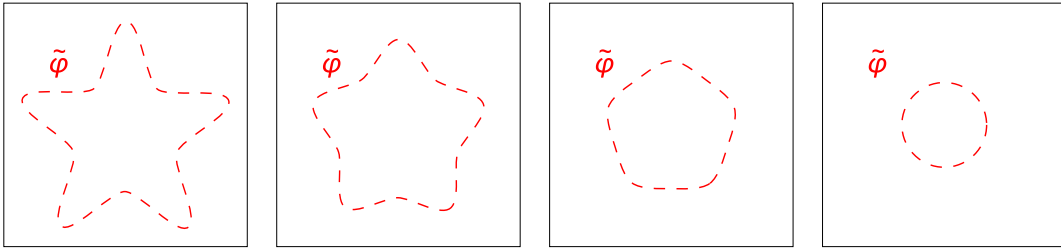
Figure 2.20: Evolution of a star-shaped level set front under the sole influence of the curvature. The star progressively deforms to a circle and keeps shrinking until it disappears from the image.

Thanks to their flexible formulation, level set methods have become popular in many other areas of image processing, such as optical flow computation [Kumar et al., 1996], denoising [Aubert and Vese, 1997], [Whitaker and Xue, 2001], [Ogier, 2005], compression [Froment, 1999], classification [Samson et al., 2000] and inpainting [Du et al., 2006]. Other references and applications of level set methods can be found in [Osher and Paragios, 2003], [Aubert and Kornprobst, 2001] and [Osher and Fedkiw, 2003].

2.3.2 Optimizations

Signed-distance initialization

Following the above conventions, φ is often initialized as the signed distance function to the embedded front, as illustrated in figure 2.21. The first advantage is that the two representation are equivalent, i.e., to each front corresponds a unique distance function. The second advantage is that a signed distance function yields $|\nabla \varphi| = 1$, which greatly simplifies the resolution of the system.

It was however shown in [Barles, 1985] and [Gomes and Faugeras, 1999] that the solution to equation (2.29) is not a signed distance function, and that the resolution of the
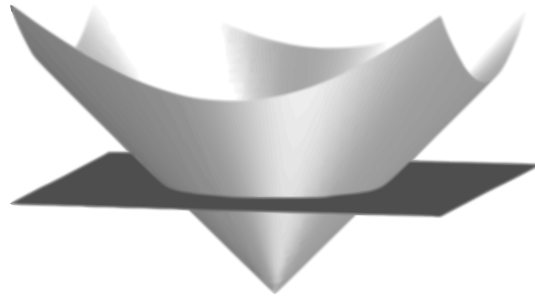
Figure 2.21: 2D example of a level set initialized with a signed distance function following the convention (2.27). The level set cuts the image plane on the initial contour, i.e., $\tilde{\varphi}$.

problem may considerably change the shape of φ, implying $|\nabla\varphi| \neq 1$. This behavior may induce errors in the numerical approximation of κ. Therefore, it is preferable to replace φ by a signed distance function (without changing the location of $\tilde{\varphi}$) every time the level set undergoes important shape changes during its evolution. This procedure is known as *reinitialization*, and may be performed in different ways. A common PDE-based approach employed for instance in [Zhao et al., 1996],[Peng et al., 1999],[Tsai and Osher, 2003] is to evolve the level set until a steady state according to the following equation:

$$\frac{\partial\varphi}{\partial t} + \text{sign}(\tilde{\varphi})(|\nabla\varphi| - 1) = 0. \tag{2.30}$$

The resolution of this equation converges to a distance function, and is usually performed numerically through *fast marching* methods [Sethian, 1999]. Another option is to perform the reinitialization directly in the discrete domain using distance transforms [Borgefors, 1986][Cuisenaire, 1999][Svensson and Borgefors, 2002].

[Gomes and Faugeras, 1999] proposed an alternative framework that maintains the level set as a distance function during the evolution in order to keep simple calculations and avoid the need for reinitialization. The idea is to derive the speed function only from $\tilde{\varphi}$, and apply this speed to all level lines of φ, which therefore remain at all times at the same distance from each other. The resulting equation is however not of Hamilton-Jacobi type, thus different numerical strategies apply. Also, this rigidity has the drawback that the level set can not segment an object that is not connected to the initial contour in the sense of the image term. Indeed, if an object is isolated in a region where φ is positive, its detection would require φ to change sign over this region, which is not possible due to the translational nature of the movement (cf. figure 2.22).
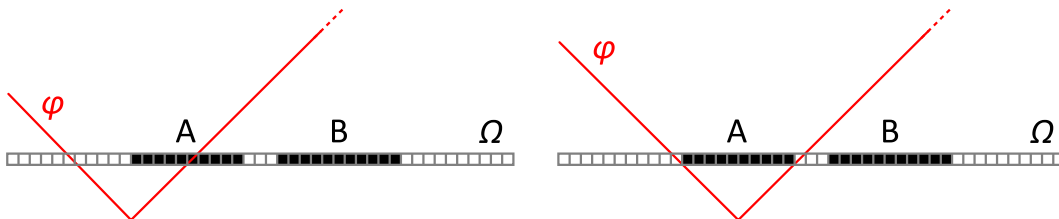
Figure 2.22: Evolution of a level set constrained to a distance function. φ translates from its initial position (left) toward object A. After convergence (right), $\tilde{\varphi}$ rests on A, but cannot access B since it requires a deformation forbidden by the constraint.

The Narrow-band approach

In the original formulation, level set methods have a higher computational complexity than parametric models. Indeed, for each iteration, the level set structure has to be updated over its entire definition domain (the image). The advantage of a global update is that features far away from the initial contour can be seamlessly detected. Nonetheless, this update has a heavy complexity of $O(M^N)$ for an image of size M in each dimension. Comparatively, for parametric methods, only the control points are updated, yielding a complexity of $O(P)$, where P is the number of control points, usually of the order of M. Note that for spline models, the fitting of the polynomial function on the control points must be added to the overall computation time.

[Adalsteinsson and Sethian, 1995] and [Malladi et al., 1995] proposed to reduce the high complexity of the level set method by updated the level set structure only within a narrow band around $\tilde{\varphi}$, i.e., surrounding the moving front (cf. figure 2.23). This scheme reduces the complexity down to about the same order as the parametric case. However, the narrow band needs to be recomputed as soon as the front reaches the band edge (also called land mine). There are several ways to perform this update, e.g. distance transforms [Cuisenaire, 1999] or fast marching methods [Sethian, 1999], for a cost of $O(M \log M)$.

<div align="center">(a)   (b)</div>
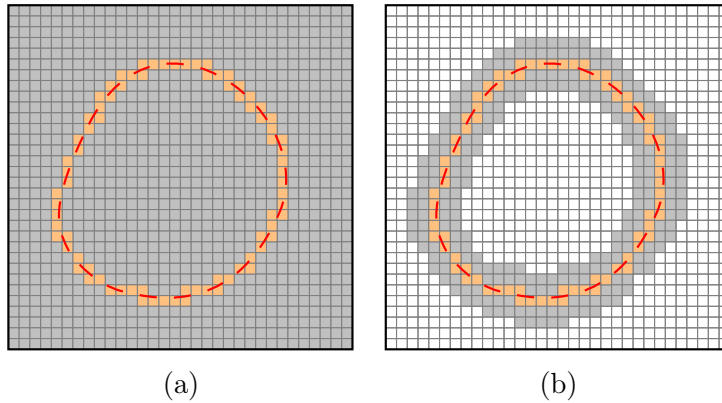
Figure 2.23: Principle of the Narrow-band optimization. In the original level set method (a), φ is updated over the whole image domain (gray pixels). In the narrow-band method (b), φ is updated only within a band located around $\tilde{\varphi}$, yielding a lower complexity. If $\tilde{\varphi}$ hits the edge of this narrow band, a new narrow band of same width is computed.

The principle of the narrow band relies on the assumption that the front progressively moves from its initial to its final location in the image. Therefore, distant objects that are located out of reach of the narrow band cannot be detected, since the level set is never updated in that area. This suggests that a relatively good initialization is performed, the quality of which depending on the narrow band width.

It should also be noted that the narrow band width should be well adapted to the problem in hand. Indeed, a small narrow band width may require frequent re-computation if the contour moves a lot, increasing considerably the computation time. Conversely, if the width is chosen too big compared to the estimated movement of the contour, the speed gain factor may not be as optimal as one could expect.

Parallel implementation using splitting schemes

Let us consider the case where the level set is maintained as a signed distance function at all times of its deformation. The curvature term is thus reduced to a simpler form since $|\nabla \varphi| = 1$. This reduced form is analogous to the well-known parabolic *heat* equation, which reads in the 1D case

$$\frac{\partial \Phi}{\partial t} = \frac{\partial^2 \Phi}{\partial x^2}, \tag{2.31}$$

where Φ is the density of the diffusing material (the diffusion coefficient is assumed constant and equal to 1). This functional has been extensively studied in the field of variational image filtering and denoising (see [Perona and Malik, 1990], [Nitzberg and Shiota, 1992], [Catte et al., 1992], [Alvarez et al., 1992] and [Weickert, 1996]). As many other equations in variational problems, the resolution of this equation in the discrete domain may be subject to stability restrictions, depending on the discretization scheme employed. It is commonly agreed that three main time stepping schemes dominate the literature.

The easiest and most common approach is the *forward Euler* scheme, also called *explicit* scheme, and computes the new value of Φ only from its value at the previous iteration. This approach can be written as

$$\Phi^{t+1} = (I - \tau \mathcal{A})\Phi^t,$$

where I is the identity matrix and $\mathcal{A}$ is the matrix notation of the differential operator discretizing the space derivatives of the original equation, usually expressed using second-order central differences. This scheme is first order accurate in time, and the Courant-Friedrichs-Levy (CFL)

The second approach is conversely called *backward Euler* or *fully implicit*, and discretizes equation (2.31) as

$$(I - \tau \mathcal{A})\Phi^{t+1} = \Phi^t. \tag{2.32}$$

This scheme is also first-order accurate in time, but becomes unconditionally stable, i.e., the CFL condition does not apply. The drawback is that the resolution of this system requires the inversion of the matrix $I - \tau \mathcal{A}$, which may be computationally expensive. In practice, since $\mathcal{A}$ is usually expressed using central space differences, the matrix is tri-diagonal and can be inverted efficiently using the Thomas algorithm.

The third approach is a hybrid of the two previous, and is commonly known as the *Crank-Nicolson* scheme. The discretization gives

$$(I - \frac{\tau}{2}\mathcal{A})\Phi^{t+1} = (I - \frac{\tau}{2}\mathcal{A})\Phi^t.$$

This last scheme is also unconditionally stable and has the advantage of being second-order accurate in time, but is only employed in very specific applications where high accuracy is needed. In practice, the backward Euler scheme is stable enough for the current problem.

The explicit scheme is the most popular since it is naturally multi-dimensional. In the two other cases, since the matrix to invert is no longer tri-diagonal, the inversion process becomes very delicate. A solution is to split the computations into one-dimensional equations, and merge the intermediate results to obtain the final multi-dimensional solution. The most popular splitting schemes are the fractional step (FS), the alternating direction implicit (ADI) and the additive operator splitting (AOS). The most interesting of these

schemes is the AOS, and was introduced in [Weickert, 1997] and [Weickert et al., 1998]. It is based on the backward Euler time-stepping approach and extends equation (2.32) to

$$\frac{1}{N} \sum_{d=1}^{N} \left(I - N\tau \mathcal{A}_d \right) \Phi^{t+1} = \Phi^t, \tag{2.33}$$

where d iterates on each space dimension. The advantage of the AOS scheme compared to the others is that each dimension is treated in total independence, yielding rotational invariance and allowing parallel computations.

The analogy with level set methods is quite straightforward, and the curvature term can be approximated with great accuracy for any time step. Nonetheless, as we mentioned earlier, the level set has to be maintained as a distance function at all times, otherwise space operators in the curvature term can no longer be separated according to each dimension.

2.3.3 Edge-based models

As one can imagine, most of the level set based approaches for segmentation are "simple" reformulation of the models described in the previous section. The use of the level set formulation has mainly allowed the possibility of segmenting an unknown number of objects using a single initial contour. This section will thus simply hold a simple overview of the reformulated models, and state several applicative contexts in which they have shown better performance than their parametric counterparts.

The first model transposing the level set theory to image segmentation has been introduced in [Caselles et al., 1993], and is derived from the balloon model [Cohen, 1991]. Starting from the general equation (2.29), the external energy terms of the balloon model are incorporated in the speed function F, yielding

$$\frac{\partial \varphi}{\partial t} = g(I_0) \left(\nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} + k \right) |\nabla \varphi| \tag{2.34}$$

where g is an edge detector function similar to that defined in equation (2.12) and k is constant equivalent to the balloon force in equation (2.13). 3D extensions of this model have been proposed for ultrasound and MRI images in [Whitaker and Chen, 1994]. A narrow-band implementation of the model was later proposed in [Malladi et al., 1995] to speed up the computations, and was later employed for cell segmentation in confocal microscopy in [Sarti et al., 2000].

Similarly to the parametric case, an extension of this model to geodesic metrics has been proposed in [Caselles et al., 1995], where the level set formulation justifies and generalizes the parametric version of the geodesic active contours. Note that a similar model has been proposed simultaneously in [Kichenassamy et al., 1995]. The evolution equation now reads

$$\frac{\partial \varphi}{\partial t} = g(I_0) \left(\nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} + k \right) |\nabla \varphi| + \nabla g(I_0) \cdot \nabla \varphi. \tag{2.35}$$

A 3D extension of this model has been discussed in [Kichenassamy et al., 1995] and was further developed for MRI image segmentation in [Caselles et al., 1997] and [Yezzi et al., 1997].

Another application of this model has been proposed in [Kimmel et al., 1996] to compute sub-pixel accurate geodesic distance transforms on surfaces as well as shape from shading.

A fully optimized version of this model was proposed in [Goldenberg et al., 1999] for segmentation and tracking in color movies. This model incorporates narrow band, fast marching for reinitialization and AOS scheme to speed up computations. This optimized version was extended to 3D in [Goldenberg et al., 2002] for cortex segmentation in MRI images. A similar model without AOS was parallely proposed in [Paragios and Deriche, 2000] for object tracking in 2D movies.

The first incorporation of the gradient vector flow information in a level set framework appeared in [Paragios et al., 2001] for 2D images. This model also incorporates the narrow band and AOS enhancements, and was later applied to 3D cardiac MRI images in [Paragios, 2002], where statistical-based region information is incorporated to enhance the segmentation result. In [Paragios et al., 2004], both image gradient and gradient vector flow are combined to increase the contour precision.

2.3.4 Region-based models

Geodesic Active Regions

This model is an extension of the geodesic active contour model, and was proposed for texture segmentation in [Paragios and Deriche, 1998]. The method relies on a probabilistic framework that unifies boundary-based and region-based information under the same energy formulation, similarly to the work published earlier in [Chakraborty et al., 1996] and [Zhu and Yuille, 1996]. This model assumes that probability density functions have been initially defined to describe the target textures to detect, and joints information from multiple frames of a video sequence in order to increase the maximization of these probabilities. In [Paragios and Deriche, 1999], a similar method was proposed for tracking in video sequences, where the probability densities tend to differentiate moving regions from non-moving regions in a scene, thereby isolating the target objects, e.g., cars moving on a highway.

The framework was reformulated and generalized in [Paragios and Deriche, 2002], in order to be suitable to any type of joint region-boundary segmentation task. The authors also address the problem of multiple simultaneous segmentation by evolving multiple level sets in the image. In order to avoid pixels to be classified into more than one objective region, the level sets are coupled through the energy functional such that their interior regions never overlap, as suggested in [Zhao et al., 1996] and [Samson et al., 2000]. The evolution equation for each level set reads

$$\frac{\partial \varphi_i}{\partial t} = |\nabla \varphi| \left(w_1 \left[b_i \kappa_i + \nabla b_i \cdot \frac{\nabla \varphi_i}{|\nabla \varphi_i|} \right] + w_2 (r_i - r_{O_i}) + w_3 \sum_{j=1}^{n} H(r_i, \varphi_j) \right),$$

where w_1, w_2 and w_3 are positive weights, b_i expresses the boundary detector, r_i describes the current interior region of φ_i, r_{O_i} describes the characteristics of the target region, and H is an indicator function that counts the number of multiply-classified pixels, i.e., belonging to the region r_i as well as any other region r_j segmented by φ_j.

This framework is very generic, and has mostly been applied to shape-constrained segmentation and tracking in 3D medical images [Rousson et al., 2004], with emphasis on cardiac ventricle segmentation, registration and tracking [Paragios, 2002][Paragios, 2003].

As we mentioned earlier, probabilistic model are very efficient when sufficient information is available to define a precise and robust probability density function corresponding to the target region. This is the main reason why probabilistic models are very well suited to deal with prior shape knowledge that describe the target region efficiently. Unfortunately, we do not benefit of such information in our context. Moreover, from a different point a view, it can be argued that probabilistic models are not generic, since each probability function is adapted to a very specific type of target. We rather employ a simpler framework that only focuses on region homogeneity, allowing to detect a broader range of objects and shapes, while avoiding the task of manually defining a probability density function, which can have unwanted consequences on the final result.

The Chan-Vese-Mumford-Shah model

The level set formulation of the Mumford-Shah segmentation problem appeared earlier than the parametric formulation with the work of [Chan and Vese, 1999] who gave it the name of "active contours without edges" model [Chan and Vese, 2001b],[Chan and Vese, 2001a]. The method employs the two-phase piecewise constant Mumford-Shah functional, i.e., a restriction of equation (2.20) with $n = 2$. The length of the contour (expressed through the Hausdorff measure) is adapted to the level set functional as follows [Chang et al., 1994]:

$$
\begin{aligned}
Length(\tilde{\varphi}) &= \int_\Omega |H(\nabla\varphi)|d\omega \\
&= \int_\Omega \delta(\varphi)|\nabla\varphi|d\omega
\end{aligned}
\tag{2.36}
$$

where H and δ are the Heaviside and Dirac operators, respectively. The new energy functional becomes

$$
E^{CV}(c_1, c_2, \varphi) = \int_{\varphi>0} |c_1 - I_0|^2 d\omega + \int_{\varphi<0} |c_2 - I_0|^2 d\omega + \mu \int_\Omega \delta(\varphi)|\nabla\varphi|d\omega.
\tag{2.37}
$$

Note that the Euler-Lagrange equation associated with the minimization of equation (2.37) can only be obtained if it is continuously defined, therefore the authors employ a regularized function δ_ϵ that tends toward the Dirac operator as ϵ tends toward 0. The development of the Euler-Lagrange is quite long, also we refer the reader to [Zhao et al., 1996] for more details. The evolution equation finally reads

$$
\frac{\partial\varphi}{\partial t} = \delta_\epsilon(\varphi)\left(|c_2 - I_0|^2 - |c_1 - I_0|^2 + \mu\nabla\cdot\frac{\nabla\varphi}{|\nabla\varphi|}\right).
\tag{2.38}
$$

This model has received great interest from the image processing community and is still now a well-known reference region-based segmentation method.

In order to detect more than two phases, the authors proposed a multiphase variant of their model in [Chan and Vese, 2001b] based on the principle of the *four color map theorem* [Saaty and Kainen, 1986]. They show that n phases can be detected using only $\log_2(n)$ level sets. However, if objects of same phase come into contact, the corresponding level set merges them, losing their identity.

Coupled geometric level sets

[Zhang et al., 2004] were the first to propose an extension of the Chan-Vese model for real multi-object tracking, where objects can potentially have the same intensity, and thus belong to the same phase. To do so, they assign one level set per object, and introduce a coupling term that minimizes the overlap between pairs of contours. This overlapping term is derived from a similar term introduced for image partitioning and classification in [Samson et al., 1999]. The energy functional reads

$$
\begin{aligned}
E^{ZZO}(\varphi_{1..n}, c_0, c_{1..n}) \;\;=\;\; & \int_\Omega \left[\sum_{i=1}^{n} \; \mu\delta(\varphi_i)|\nabla\varphi_i| + \lambda_1 H(\varphi_i)|c_i - I_0|^2 \right. \\
& + \frac{\lambda_2}{n} \prod_{j=1}^{n} (1 - H(\varphi_j))|c_0 - I_0|^2 \\
& \left. + \gamma \sum_{j=i+1}^{n} H(\varphi_i)H(\varphi_j) \right] d\omega.
\end{aligned}
\tag{2.39}
$$

and the evolution equation for each level set function is given by

$$
\frac{\partial \varphi_i}{\partial t} = \delta_\epsilon(\varphi_i) \left(\mu\nabla \cdot \frac{\nabla\varphi_i}{|\nabla\varphi_i|} + \lambda_2 \prod_{j\neq i} H(\varphi_i)|c_0 - I_0|^2 - \lambda_1|c_i - I_0|^2 - \gamma \sum_{j\neq i} H(\varphi_j) \right),
\tag{2.40}
$$

where the curvature term $\kappa = \nabla \cdot (\nabla\varphi/|\nabla\varphi|)$ creates a force that drives the front along its local curvature. Without the other energy terms, this force would cause the front to shrink into a point and disappear.

Above we have described the parametric and level set formulations of the Mumford-Shah functional, which is, in our opinion, the best candidate for our segmentation and tracking context. In the following chapter, we shall thus present our main contributions to these models, i.e., the adaptation of these two frameworks to three-dimensional segmentation, tracking and visualization in biological fluorescence microscopy.

3

The Active Surfaces framework

In this chapter we present a segmentation and tracking method based on implicit deformable models called *active surfaces*. The method proposes several extensions and improvements to the coupled geometric active contours model of [Zhang et al., 2004]. This work has been published in [Dufour et al., 2005] and [Zimmer et al., 2006], and was supervised by Dr. Jean-Christophe Olivo-Marin and Dr. Christophe Zimmer from the Quantitative Image Analysis unit, Institut Pasteur, Paris.

Contents

3.1 Proposed model

Our model relies on the "coupled geometric level sets" model, proposed for segmentation and tracking in 2D microscopy in [Zhang et al., 2004]. The energy functional reads

$$
E^{ZZO}(\varphi_{1..n}, c_0, c_{1..n}) \;=\; \int_\Omega \left[\sum_{i=1}^n \mu\delta(\varphi_i)|\nabla\varphi_i| + \lambda_1 H(\varphi_i)|c_i - I_0|^2 \right.
$$
$$
+ \frac{\lambda_2}{n} \prod_{j=1}^n (1 - H(\varphi_j))|c_0 - I_0|^2
$$
$$
\left. + \gamma \sum_{j=i+1}^n H(\varphi_i)H(\varphi_j) \right] d\omega.
$$

and the corresponding evolution equation for each level set is given by

$$
\frac{\partial\varphi_i}{\partial t} = \delta_\epsilon(\varphi_i)\left(\mu\nabla\cdot\frac{\nabla\varphi}{|\nabla\varphi|} + \lambda_2 \prod_{j\neq i} H(\varphi_i)|c_0 - I_0|^2 - \lambda_1|c_i - I_0|^2 - \gamma \sum_{j\neq i} H(\varphi_j) \right). \quad (3.1)
$$

We shall now review below the extensions and improvements that have been incorporated to this model.

3.1.1 3D implementation

The first of our contribution is to extend the model to deal with 3D images. The extension does not raise significant theoretical difficulties, thanks to the multi-dimensional nature of the level set formalism. Nonetheless, the practical implementation of the 3D evolution equation may raise technical difficulties and calculation errors. More particularly, we focus below on the discretization of the space derivatives, for which we have chosen finite difference schemes, well adapted to regular-spaced grids like, e.g., digital images. Let us first review below the principle of finite difference schemes.

Finite difference schemes

Finite-difference approximations originate from the Taylor's series expansion

$$
f(x + h) = f(x) + \sum_{p=1}^{n-1} f^{(p)}(x)\frac{h^p}{p!} + O(h^n), \quad (3.2)
$$

where n is the expansion order and $f^{(p)}$ expresses the p-th derivative of f. The second order expansion of f reduces Eq. 3.2 down to

$$
f(x + h) = f(x) + hf'(x) + O(h^2)
$$

which can be rewritten

$$
f'(x) = \frac{f(x + h) - f(x)}{h} + O(h) \approx \frac{f(x + h) - f(x)}{h}.
$$

This approximation is known as the *forward* difference approximation, and abbreviated $D_x^+ f$. Similarly, the *backward* difference approximation, noted $D_x^- f$ is obtained by expanding $f(x - h)$, giving

$$f'(x) \approx \frac{f(x) - f(x - h)}{h}. \tag{3.3}$$

Finally, the *central* difference approximation, noted $D_x^c f$ is obtained by subtracting the 4th order expansions of $f(x + h)$ and $f(x - h)$, which comes down to

$$f(x + h) - f(x - h) = 2hf'(x) + O(h^3),$$

i.e.,

$$f'(x) = \frac{f(x + h) - f(x - h)}{2h} + O(h^2) \approx \frac{f(x + h) - f(x - h)}{2h}. \tag{3.4}$$

Note that the forward and backward schemes yield an approximation error of order h (also called *first-order accurate*), while the central scheme is *second-order accurate*, with an error of order h^2.

Approximating the space derivatives

The only term involving space derivatives in Eq. 3.1 is the curvature of the level set. In 2D, this curvature is given by

$$\kappa = \nabla . \frac{\nabla \varphi}{|\nabla \varphi|} = \frac{\varphi''_{xx} \varphi'^2_y - 2\varphi'_y \varphi'_x \varphi''_{xy} + \varphi''_{yy} \varphi'^2_x}{(\varphi'^2_x + \varphi'^2_y)^{3/2}}. \tag{3.5}$$

In 3D, there are an infinity of ways to compute this curvature, depending on the orientation chosen. The two most common candidates are the mean and Gaussian curvatures. We employ the mean curvature, which is simpler to compute and is the closest to the 2D curvature:

$$\kappa_{mean} = \frac{\left\{ \begin{array}{c} (\varphi''_{yy} + \varphi''_{zz})\varphi'^2_x + (\varphi''_{xx} + \varphi''_{zz})\varphi'^2_y + (\varphi''_{xx} + \varphi''_{yy})\varphi'^2_z \\ -2\varphi'_x \varphi'_y \varphi''_{xy} - 2\varphi'_x \varphi'_z \varphi''_{xz} - 2\varphi'_y \varphi'_z \varphi''_{yz} \end{array} \right\}}{(\varphi'^2_x + \varphi'^2_y + \varphi'^2_z)^{3/2}}. \tag{3.6}$$

The above partial derivatives are then discretized as follows

$$\begin{array}{rclcl} \varphi'_x & = & D_x^c \varphi & & \\ \varphi''_{xx} & = & D_x^+ D_x^- \varphi & = & D_x^- D_x^+ \varphi \\ \varphi''_{xy} & = & D_x^c D_y^c \varphi & = & D_y^c D_x^c \varphi, \end{array} \tag{3.7}$$

all other derivatives are then defined by analogy.

3.1.2 Automatic initialization

A notorious disadvantage of implicit methods is their considerable computational complexity compared to explicit methods [Delingette and Montagnat, 2000], [Terzopoulos, 2003], since the level set function has the same dimension as the image, i.e., one dimension more than the object boundary. This drawback is particularly limiting when analyzing 3D sequences, which can reach several gigabytes. Although pre-processing can be done offline,

reducing computation time is still highly desirable. Therefore, we seek to initialize the active surfaces as close as possible to the actual cell boundaries, in order to limit the number of iterations required for convergence. For tracking already identified objects, this is achieved by initializing with the segmented surface from the previous time point as already described.

For the first frame, however, no prior knowledge is available about the objects positions. To compute an initialization surface for the first frame, we choose to minimize the original Chan-Vese-Mumford-Shah functional (2.37) that will detect all non-touching objects with a single level set. However, for further computational speed-up, we have adopted the fast optimization scheme proposed in [Song, 2003], which is a non-PDE technique to minimize (2.37) that considers each voxel separately and changes the sign of φ whenever the change induces an energy decrease. The discrete approximation of equation (2.37) is given in the 3D case by

$$
E(H(\varphi), c_1, c_2) \quad = \quad \sum_{(i,j,k) \in I} |I_{i,j,k} - c_1|^2 + |I_{i,j,k} - c_2|^2
$$
$$
+ \mu \sqrt{[H(\varphi_{i\pm1, j\pm1, k\pm1}) - H(\varphi_{i,j,k})]^2}, \qquad (3.8)
$$

where μ weights a term that roughly approximates the curvature term of (2.37). Starting from any initial partition, this method generally achieves a fast and approximate segmentation of the image within a small number of iterations. The drawback of this technique is that it is much less robust to noise than the original PDE version. Also, the rough approximation of the curvature term induces significant artifacts in the segmentation result (see figure 3.1-(b)).

(a)   (b)   (c)
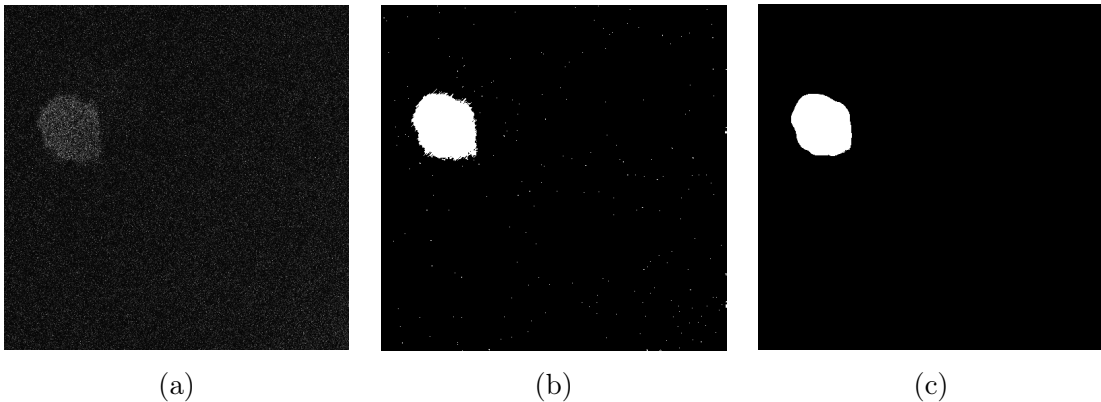
Figure 3.1: Segmentation result on a real image. (a) Slice of a 3D image of size $512 \times 512 \times 24$ voxels. (b) Slice of the result image after minimization of equation (3.8). Convergence was reached in 14 seconds with only 20 iterations, regardless of the initialization. (c) Slice of the final result after minimizing the 3D version of equation (2.37) with a narrow band of 3 voxels wide. Convergence was reached in 35 seconds with 30 iterations.

Once the optimized algorithm has converged, a level set φ is initialized by a signed distance map is computed from the obtained partition. To do so, we employ a chamfer algorithm (see details in function `SignedDistance`). Then, the Chan-Vese-Mumford-Shah model is applied, which will refine the initial result within a small number of iterations (see figure 3.1-(c)). Finally, all connected components are extracted from $H(\varphi)$ in order to initialize all the level sets of our model.

In a tracking context, a useful option is to hold $c_{0...n}$ constant during the evolution of the level sets using final values from the segmentation of the previous stack. This is justified as long as the mean object intensity does not change strongly from one frame to the next. The advantage of is that computation time is saved, since $c_{0...n}$ do not need to be re-evaluated at each iteration. Moreover, the evolution of active surfaces is then significantly less sensitive to initialization. As a result, good segmentations can be obtained even if the overlap between the initial surface and the object is small, as long as it is not empty. With this option, the tracking method can, thus, handle displacements as large as the size of the object. Finally, to take into account progressive changes in objects intensity, such as fluorescence decrease due to photo-bleaching, the mean values can be updated once after segmentation of each frame.

3.1.3 Narrow-band optimization

The narrow-band implementation of the level set framework allows to decrease the complexity in 3D from $O(M^3)$ down to $O(2w \cdot card(\tilde{\varphi}))$ per level set per iteration, where w is the width of the narrow band and $card(\tilde{\varphi})$ the number of voxels forming the active surface. Each level set is thus updated only at voxels lying within a distance up to w voxels from $\tilde{\varphi}$.

For further speed up, instead of computing the narrow band explicitly, we keep a copy of the signed distance map used to initialize the level sets. Hence, φ is updated only at voxels which have a value between $-w$ and w in the distance map. Also, to detect if the contour touches the edge of the narrow band, we simply check if φ changes sign at a voxel with value $\pm w$ in the map. When this occurs, a full distance map is recomputed for the level set, which allows us to reinitialize φ and replace the old map with the new one as the new narrow band indicator.

The narrow band width w should be chosen small enough to sweep through a few number of image voxels, but on the other hand, big enough to avoid too frequent updates of the distance map, which is not a cost-free process.

3.1.4 Volume conservation constraint for tracking

The coupling term $\gamma \sum_{j \neq i} H(\varphi_j)$ avoids contours to overlap in case objects touch each other through time. However, since it does not involve image data, this constraint does not favor any particular location for the boundary between touching objects. Hence, when objects of similar intensities touch, the image content is often insufficient to outline the boundary between cells, even for the human eye. In such cases, segmentation results will generally be incorrect, because the image dependent terms in Eq. 3.1 are unable to guide the surface toward the correct location. Such behavior can be avoided by introducing shape prior knowledge on the objects, e.g. the round-shaped cell constraint proposed for leukocytes tracking by [Ray et al., 2002]. To maintain a relatively generic framework, we do not wish to impose shape constraints. However, for applications where it is known that the objects volume does not change over time, we introduce a new term in the energy functional that maintains the volume inside φ close to the reference volume computed from the first frame. By expressing the volume of each level set as $V(i) = \int_{\Omega} H(\varphi_i)$, and denoting by $V_0(i)$ the reference volume (obtained form the first frame), the energy functional to minimize for all

frames, but the first, becomes

$$
\begin{aligned}
E^{DZO}(\varphi_{1..n}, c_0, c_{1..n}) \;=\; & \int_\Omega \Bigg[\; \sum_{i=1}^{n} \mu \delta(\varphi_i) |\nabla \varphi_i| + \lambda_1 H(\varphi_i) |c_i - I_0|^2 \\
& + \frac{\lambda_2}{n} \prod_{j=1}^{n} (1 - H(\varphi_j)) |c_0 - I_0|^2 \\
& + \gamma \sum_{j=i+1}^{n} H(\varphi_i) H(\varphi_j) \\
& + \eta \left(V(i) - V_0(i) \right)^2 \Big] d\omega,
\end{aligned}
\tag{3.9}
$$

yielding the new evolution equation

$$
\begin{aligned}
\frac{\partial \varphi_i}{\partial t} \;=\; \delta_\epsilon(\varphi_i) \; & \left(\mu \nabla . \frac{\nabla \varphi_i}{|\nabla \varphi_i|} + \lambda_2 \prod_{j \neq i} H(\varphi_i) |c_0 - I_0|^2 - \lambda_1 |c_i - I_0|^2 \right. \\
& \left. - \gamma \sum_{j \neq i} H(\varphi_j) - 2\eta(V(i) - V_0(i)) \right).
\end{aligned}
\tag{3.10}
$$

The first advantage of the volume conservation constraint is to improve the overall robustness of the model against noise, especially when photo bleaching occurs (see figure 3.2). Note that this new functional allows to enable or disable volume conservation easily through the weighting parameter η.

3.2 Discussion

The performance of the proposed model will be evaluated in chapter 5. However, from a conceptual point of view, we can already state several comments and remarks on the model properties.

- **Memory and time consumption**
  In the first pass of the algorithm, we exploit the ability of level sets to split and merge freely in order to extract automatically the number objects in the scene. Then, a single level set is set to track each object, and an energy term ensures that the level sets do not overlap each other, thereby forbidding their natural topological freedom. Knowing the complexity and computational cost of implicit models, it becomes questionable whether they are the best choice for a multi-object context, where parametric models could probably perform faster for a lower computational load.

- **Tracking**
  The proposed tracking procedure still has limited capabilities. Indeed, the nature of the image-based term in the energy functional suggests that each level set φ_i aims at segmenting the image region with average intensity c_i, generally higher than the background intensity c_0. Hence, the interior region of φ_i will shrink where the image intensity is close to c_0, and conversely grow where the image intensity is close to c_i. Hence, in order for the model to keep track of the objects, a significant overlap

should exist between two successive positions of the moving object. Fortunately, the objects we focus on have a low speed compared to the minimal time interval between two acquired frames. Also, this interval is usually adjusted to the movement nature of the objects.

- **3D visualization**
  Independently from the model itself, the implicit representation of the contour yields a serious visualization issue. Since the segmented objects are represented by the negative region of the level sets, their representation as a 3D object is not immediate. The conversion from raster to 3D objects requires a reconstruction algorithm such as the "Marching Cubes" [Lorensen and Cline, 1987] or the "Marching Tetrahedra" [Gueziec and Hummel, 1995]. The former triangulates the image volume using a cube-shape decomposition, while the latter uses a tetrahedral decomposition. Other methods based on Delaunay triangulation and Voronoi diagrams are efficient to reconstruct a 3D shape from cloud point data, however they are not adapted to a raster-type structure, where the number of points to triangulate is equal to the number of voxels forming the object surface.

  The efficiency and speed of reconstruction algorithms is determined by the size of the grid used to decompose the image. If the grid size is chosen large, fast reconstruction can be performed, but the obtained surface yields many approximation errors since it misses all surface particularities that lie within the decomposing grid. To avoid approximation errors, the grid size should correspond to the voxel size in order to reconstruct precisely every single element of the object surface. Unfortunately, this choice has a dramatic impact on the computation time, while imitating the grid structure of the image, as shown on figures 3.3, 3.4 and 3.5.

- **Shape description**
  From a quantitative analysis point of view, the most appealing property of deformable models is their high level of interpretation of each object being segmented or tracked. This facilitates the use of efficient descriptors for shape recognition, classification and monitoring [Clarysse et al., 1997]. Since our model relies on a level set formulation of the contour, the final data on which shape descriptors will be measured is generally a binary image of same size as the original. As a result, the computation of classical descriptors such as moments and Fourier descriptors [Reeves et al., 1988] can be complex and time-consuming, as the complexity increases with the number of voxels forming the object (or its contour). Moreover, it can be argued that local shape information, e.g., surface normal or curvature, can only be approximated using such voxel-based representation. Shape information is usually easier to compute on mesh models, with no or little approximation error. Nevertheless, we cannot rely on the meshes obtained by reconstruction, since they imitate the grid nature of the 3D image, and would yield the same approximation errors.

3.3 Algorithms & Pseudo-codes

In this section we include pseudo-codes of the various functions and algorithms employed in our method.

Function `SignedDistance`(B). Computes a signed distance map from a binary partition.

Input : B: binary partition with 1s inside object(s) and 0s in the background

1 $\mathsf{a} \leftarrow 1, \mathsf{b} \leftarrow \sqrt{2}, \mathsf{c} \leftarrow \sqrt{3}$

2 $\mathsf{map}_{in} \leftarrow \mathsf{map}_{out} \leftarrow$ empty image of same size as B

3 **foreach** *voxel* p *in* B **do** **if** $B(\mathsf{p}) = 1$ **then** $\mathsf{map}_{in}(\mathsf{p}) = \infty$ **else** $\mathsf{map}_{out}(\mathsf{p}) = \infty$

4 **foreach** map *in* ($\mathsf{map}_{in}, \mathsf{map}_{out}$) **do**

5   **foreach** *voxel* p *in* map *using a forward sweep* **do**

6   calculate $\mathrm{Min}(\mathsf{map}(\mathsf{p}), \mathsf{map}(\mathsf{p}) + \mathsf{a})$ for forward 6-connectivity neighbors

7   calculate $\mathrm{Min}(\mathsf{map}(\mathsf{p}), \mathsf{map}(\mathsf{p}) + \mathsf{b})$ for forward 12-connectivity neighbors

8   calculate $\mathrm{Min}(\mathsf{map}(\mathsf{p}), \mathsf{map}(\mathsf{p}) + \mathsf{c})$ for forward 26-connectivity neighbors

9   **end**

10   **foreach** *voxel* p *in* map *using a backward sweep* **do**

11   calculate $\mathrm{Min}(\mathsf{map}(\mathsf{p}), \mathsf{map}(\mathsf{p}) + \mathsf{a})$ for backward 6-connectivity neighbors

12   calculate $\mathrm{Min}(\mathsf{map}(\mathsf{p}), \mathsf{map}(\mathsf{p}) + \mathsf{b})$ for backward 12-connectivity neighbors

13   calculate $\mathrm{Min}(\mathsf{map}(\mathsf{p}), \mathsf{map}(\mathsf{p}) + \mathsf{c})$ for backward 26-connectivity neighbors

14   **end**

15 **end**

16 **return** $0.5 \times \mathsf{map}_{out} - 0.5 \times \mathsf{map}_{in}$

Function `OptimizedLS`(I,B). Minimizes the discrete approximation of the Chan-Vese functional, cf. equation (3.8).

Input : I: image, B: binary partition with 1 outside the object(s) and -1 inside.

1 $c_1 \leftarrow \mathtt{Mean}(I, B \leq 0), c_2 \leftarrow \mathtt{Mean}(I, B > 0)$

2 Compute E according to equation (3.8)

3 **repeat**

4   **foreach** *voxel* p *in* I **do**

5   Change the sign of $B(\mathsf{p})$

6   Compute updated values c_1', c_2' and E'

7   **if** $E' < E$ **then** replace c_1, c_2 and E by c_1', c_2' and E'

8   **end**

9 **until** B *does not change*

10 **return** B
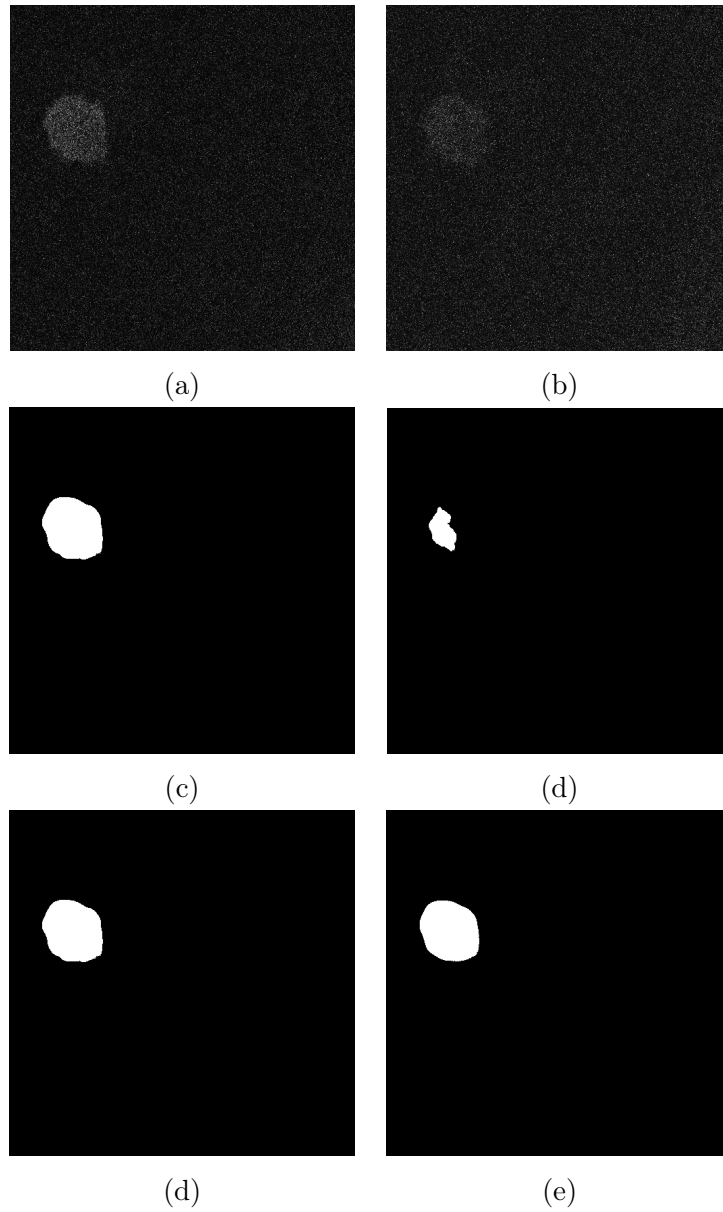
Figure 3.2: Importance of the volume conservation constraint in presence of photo-bleaching. (a) Slice of the first sequence frame. (b) Slice of the next frame, acquired 20 seconds later. (c,d) Segmentation results on (a) and (b) without conservation constraint. (d,e) Results with conservation constraint.

(a)

(b)

(c)

(d)
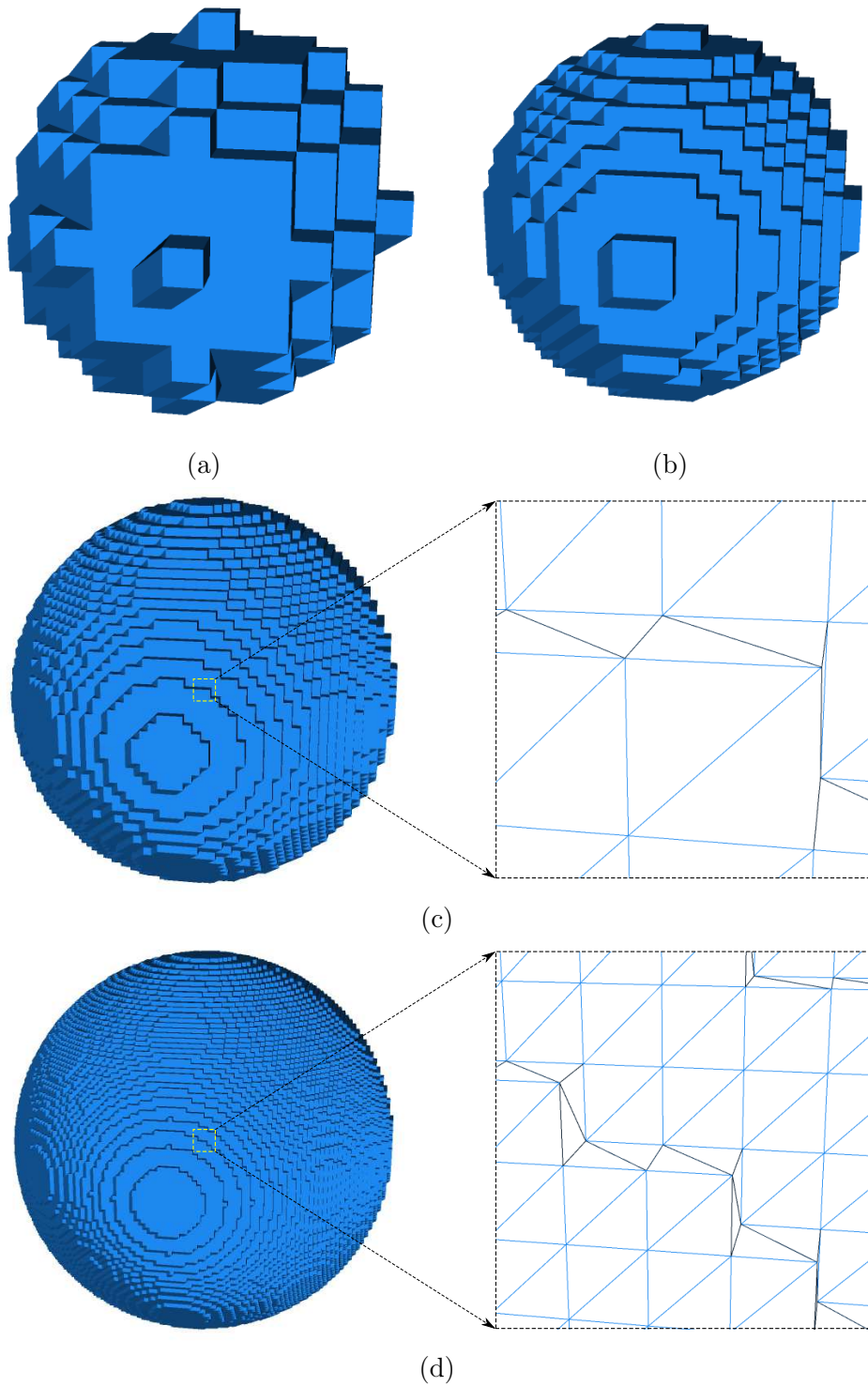
Figure 3.3: 3D reconstructions of a segmented sphere using the Marching Tetrahedra algorithm at various grid sizes. The image size is $200 \times 200 \times 200$ voxels with isotropic resolution, and the sphere radius is 50 voxels. (a) grid size: 10 voxels, time: 0.1 sec. (b) grid size: 5 voxels, time: 1 sec. (c) grid size: 2 voxels, time: 20 sec. (d) grid size: 1 voxel, time: 4 min.
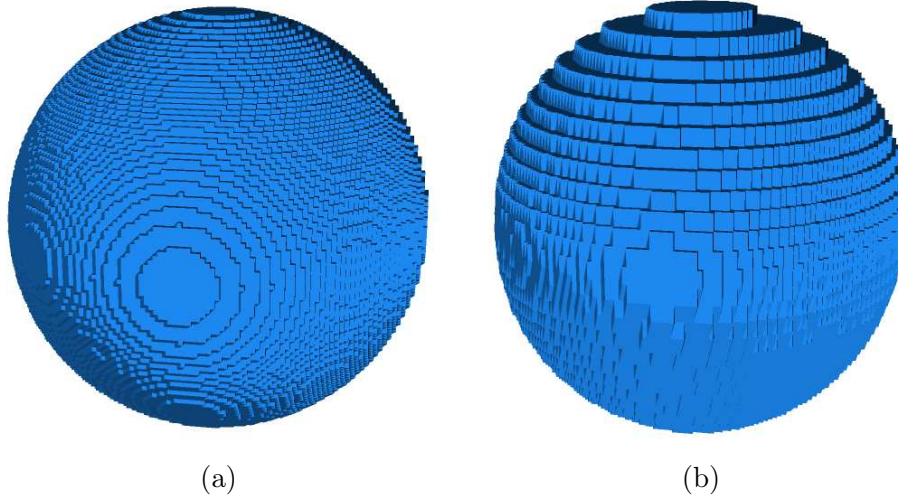
Figure 3.4: Impact of the depth resolution of an image on the segmentation result. (a) image size: $200 \times 200 \times 200$ voxels, resolution: $0.1 \times 0.1 \times 0.1 \mu m$ per voxel. (b) image size: $200 \times 200 \times 50$ voxels, resolution: $0.1 \times 0.1 \times 0.4 \mu m$ per voxel.

Function `ChanVeseLS`(I,B,w). Minimizes the Chan-Vese functional, cf. equation (2.37).

Input : I: image, B: binary partition, w: narrow band width

1 $\varphi \leftarrow$ map $\leftarrow$ `SignedDistance`(B)
2 $c_1 \leftarrow$ `Mean`($I, \varphi \leq 0$), $c_2 \leftarrow$ `Mean`($I, \varphi > 0$)
3 updateBand $\leftarrow$ false
4 Compute E according to equation (2.37)

5 **repeat**
6   **foreach** *voxel* p *in I such that* $|$map(p)$| \leq w$ **do**
7   Update φ(p) according to equation (2.38)
8   **if** map(p) $= w$ **and** φ(p) *changed sign* **then** updateBand $\leftarrow$ true
9   **end**
10   Update c_1 and c_2 from I and φ
11   Update E according to equation (2.37)
12   **if** updateBand **then**
13         $B \leftarrow 1 -$ `Heaviside`(φ)
14   map $\leftarrow$ `SignedDistance`(B)
15   updateBand $\leftarrow$ false
16   **end**
17 **until** E *has converged*

18 **return** φ

(a)

(b)

(c)

(d)
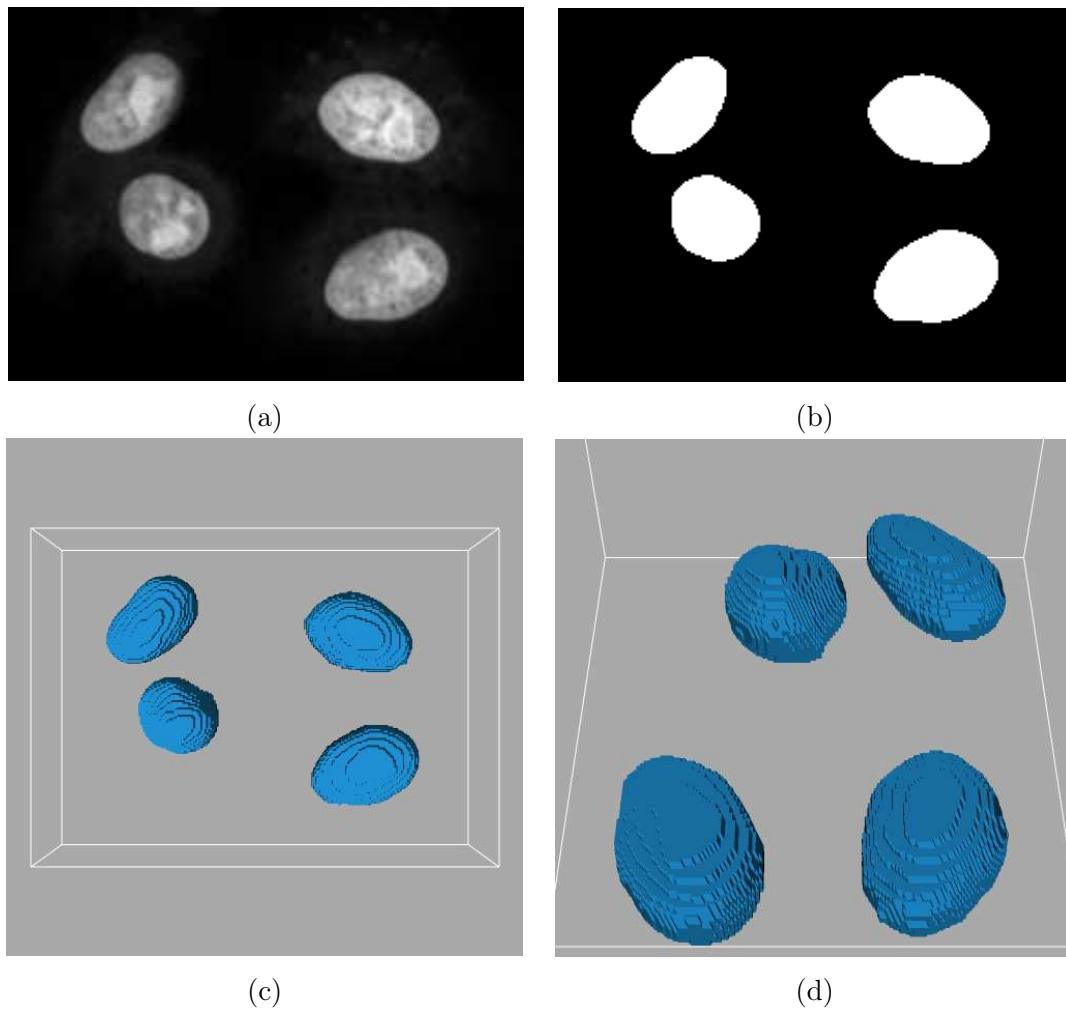
Figure 3.5: Segmentation result and reconstruction of 4 objects. (a) Slice of the original image of size $245 \times 177 \times 45$ voxels and resolution $0.327 \times 0.327 \times 0.75\mu m$ per voxel. (b) Slice of the segmentation result (total time: 30 sec.). (c,d) Two views of a 3D reconstruction using the Marching Tetrahedra with smallest grid size (reconstruction time: 160 sec.).

Function `MultipleLevelSetEvolution`(I,$B[]$,$c_{obj}[]$,c_{bgd},w,*updateMeans*). Minimizes the proposed functional, cf. equation (3.9).

Input: I: image, $B[]$: binary partition array, $c_{obj}[]$: objects mean intensities, c_{bgd}: background mean intensity, w: narrow band width, *updateMeans*: boolean indicating if the means should be updated at very iteration.

1 **for** *i=1* **to** n **do**
2 $\varphi[i] \leftarrow$ map$[i] \leftarrow$ `SignedDistance`($B[i]$)
3   updateBand $[i] \leftarrow$ false
4 **end**
5 Compute E according to equation (3.9)

6 **repeat**
7   **for** *i=1* **to** n **do**
8   **foreach** *voxel* p *in* I *such that* $|$map$[i]($p$)| \leq w$ **do**
9   Update $\varphi[i]($p$)$ according to equation (3.10)
10   **if** map$[i]($p$) = w$ **and** $\varphi[i]($p$)$ *changed sign* **then** updateBand $[i] \leftarrow$ true
11   **end**
12   **end**
13   **if** *updateMeans* **then** Update $c_{obj}[]$ and c_{bgd} from I and $\varphi[]$
14   Update E according to equation (3.9)
15   **for** *i=1* **to** n **do**
16   **if** updateBand **then**
17 $B[i] \leftarrow 1 -$ `Threshold`($\varphi[i]$)
18   map$[i] \leftarrow$ `SignedDistance`($B[i]$)
19   updateBand $[i] \leftarrow$ false
20   **end**
21   **end**
22 **until** E *has converged*
23 **return** $(\varphi[], c_{obj}[], c_{bgd})$

Algorithm 5: Main process of the Active Surfaces framework

Input: S: image sequence with T frames, B: initial partition, w: narrow band width

1 $I \leftarrow$ S $[1]$
2 $B \leftarrow$ `OptimizedLS`(I, B)
3 $\varphi \leftarrow$ `ChanVeseLS`(I, B)
4 Extract all connected components of $\varphi_{\tilde{init}}$ into a binary mask array $B[]$
5 **for** *i=1* **to** n **do** $c_{obj}[i] \leftarrow$ `Mean`($I, B[i] = 1$)
6 $c_{bgd} \leftarrow$ `Mean`($I, \bigcap_{i=1}^{n} B[i] = 0$)
7 $(\varphi[], c_{obj}[], c_{bgd}) \leftarrow$ `MultiLevelSetEvolution`($I, B[], c_{obj}[], c_{bgd}, w$, true)
8 **foreach** *t=2* **to** T **do**
9 $I \leftarrow$ S $[t]$
10 $(\varphi[], c_{obj}[], c_{bgd}) \leftarrow$ `MultiLevelSetEvolution`($I, B[], c_{obj}[], c_{bgd}, w$, false)
11 **end**

4

The Active Meshes framework

In this chapter we present a discrete parametric counterpart to the active surfaces, based on deformable meshes. We emphasize on the advantages of the discrete representation for lower computational cost and real-time visualization purposes. This work has been published in [Dufour et al., 2006], [Dufour et al., 2007a] and [Dufour et al., 2007b], and was supervised by Pr. Nicole Vincent from the Intelligent Perception Systems (SIP-CRIP5) team, Paris Descartes university, and by Dr. Auguste Genovesio from the Image Mining group, Institut Pasteur Korea.

Contents

4.1 Introduction

4.1.1 Motivations

At the end of the previous chapter, we discussed the limitations of the active surfaces, among which the most limiting factors toward practical applications are computational cost and poor visualization. The first factor is due to the fact that a distinct level set is assigned to each object in the image. The second factor arises from the fact that the level set function φ is discretized on a grid identical to that of the original image, and thus requires the use of a reconstruction algorithm in order to visualize the contour, inducing additional time consumption and visual approximation errors.

Since multi-object segmentation and tracking do not take advantage of the topological flexibility of level sets, we wondered whether these tasks could be accomplished in a faster and memory-efficient way using explicit models, eventually allowing easier and more precise visualization. To this end, we have focused more particularly on triangular mesh models, since they adopt the same data structure as computer graphic cards (implying fast manipulation and cost-free visualization), and can also handle self re-parameterization and topological changes in an automated manner. Moreover, practical implementation of mesh models has been greatly simplified over the past few years, thanks to the evolution of modern programming languages and graphic libraries. Therefore, we propose in this chapter to adapt the whole framework to mesh models, yielding the name of *active meshes*.

4.1.2 Parametric formulation of the problem

Let us first rewrite the energy functional (3.10) with a parametric formulation, i.e., by applying the contributions of the active surfaces method to the *multiple coupled active contours* model published in [Zimmer and Olivo-Marin, 2005]:

$$
\begin{aligned}
E^{AM}(\mathcal{S}_{1..n}, c_0, c_{1..n}) \;=\; & \sum_{i=1}^{n} \Bigg[\int_0^1 w_1 \left|\frac{\partial \mathcal{S}_i}{\partial p}\right|^2 + w_2 \left|\frac{\partial \mathcal{S}_i}{\partial q}\right|^2 + w_3 \left|\frac{\partial^2 \mathcal{S}_i}{\partial p^2}\right|^2 \\
& + w_4 \left|\frac{\partial^2 \mathcal{S}_i}{\partial q^2}\right|^2 + w_5 \left|\frac{\partial^2 \mathcal{S}_i}{\partial p \partial q}\right|^2 dp\, dq \\
& + \lambda_1 \int_{R_i} |c_i - I_0|^2 d\omega + \eta (V(i) - V_0(i))^2 \Bigg] \\
& + \lambda_2 \int_{\bigcap_i R_i} |c_0 - I_0|^2 d\omega + \gamma \int_{R_0} d\omega.
\end{aligned}
\tag{4.1}
$$

It should be noted that the following equation does not totally coincide with the Mumford-Shah functional. Indeed, although the image term is similar, the regularization term minimizing the length of the boundary set between the two regions has been expressed using the internal terms of the original snake model. As pointed by [Caselles et al., 1995], it is clear that both terms minimize the contour curvature, and since the contour is closed, its overall length. In our context, we assume that the control points sample the surface regularly enough, allowing to work only with the first regularization term, which will greatly simplify the complexity per iteration. Therefore, we set $w_3 = w_4 = w_5 = 0$ and the

evolution equation for each surface reduces to

$$\frac{\partial \mathcal{S}_i}{\partial t} = w_1 \frac{\partial^2 \mathcal{S}_i}{\partial p^2} + w_2 \frac{\partial^2 \mathcal{S}_i}{\partial q^2} - \left| \frac{\partial^2 \mathcal{S}_i}{\partial p \partial q} \right| \overrightarrow{F_i}, \quad \text{with}$$

$$\overrightarrow{F_i} = \left[\lambda_1 (c_i - I_0)^2 - \lambda_2 (c_0 - I_0)^2 + \gamma \sum_{j \neq i} \xi_j(\mathcal{S}_i) + \eta(V(i) - V_0(i)) \right] \overrightarrow{N_i}, \quad (4.2)$$

where we recall that w_i, λ_i, γ and ν are nonnegative weights and ξ is an indicator function such that $\xi_j = 1$ for points of $\mathcal{S}_i$ located inside $\mathcal{S}_j$ and 0 otherwise.

As explained earlier, this theoretical model is complex to solve numerically due to the two-dimensional nature of the parametric domain. Indeed, the parametric surface must be discretized through a uniform grid in both dimensions (following p and q), and regularly re-parametrized as the control points evolve in the image domain. To reduce the complexity of the problem, the spatial derivatives in equation (4.2) should be replaced by a simpler (but equivalent) alternative. We shall see below how this can be achieved using deformable meshes.

4.2 Proposed model

Our active mesh model is similar the triangular mesh models proposed in [Miller, 1990] and in [Lachaud, 1998], in which we have made several modifications and improvements according to our context. The definition and properties of an active mesh are given below. We will then focus on the practical implementation of the energy minimization process using Active Meshes.

4.2.1 Active Mesh: definition and properties

Definition

An active mesh $\mathcal{M}$ is a discrete surface defined by a list of three-dimensional vertices $\mathrm{v}_i = (x_i, y_i, z_i) \in \Omega \subset \mathbb{R}^3$ forming a closed set of counter-clockwise oriented triangles (called *faces*), such that the mesh boundary represents at all times the contour of a volumetric object. The tetrahedron is the simplest example of a triangular mesh. This definition is more restrictive than the original parametric representation where open surfaces are allowed, but is sufficient for object segmentation and tracking. Moreover, this will allow efficient topology manipulation. The deformation of the mesh is driven by that of its vertices, under the influence of various forces computed, in our case, from the minimization of the energy functional (4.1). Since the vertices evolve in a real-space coordinate system, image values are computed using trilinear interpolation of the neighboring image voxels. Finally, from an implementation point of view, the data structure is tailored to fit the pipeline of the graphics adapter, which will allow us to employ hardware optimized functions to speed up computations and rendering.

Local topology handling

To avoid excessive complexity in the manifold structure, local re-sampling rules are defined to ensure that all connected mesh vertices remain within an arbitrary distance interval $[d_{min}, d_{max}]$ from each other. During the mesh deformation, if two vertices v_i and v_j

come too close to each other $(dist(\mathtt{V}_i, \mathtt{V}_j) < d_{min})$, they are merged together, i.e. they are deleted and replaced by a new vertex (arbitrarily placed in their center) that will be linked to all neighbor vertices of $\mathtt{V}_i$ and $\mathtt{V}_j$ as depicted on figure 4.1-(a). Conversely, if $\mathtt{V}_i$ and $\mathtt{V}_j$ move too far away from each other $(dist(\mathtt{V}_i, \mathtt{V}_j) > d_{max})$, a new vertex is created in their center, thereby cutting the two faces sharing the edge $\mathtt{V}_i\mathtt{V}_j$ into four new faces as shown on figure 4.1-(b).

(a)   (b)
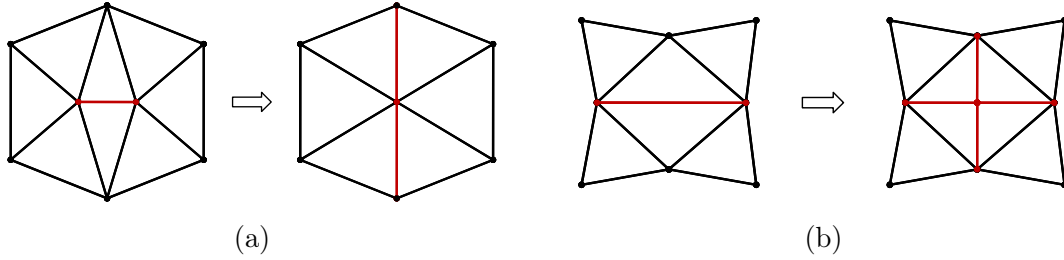
Figure 4.1: Local mesh re-sampling rules. (a) Vertices are merged if they move close to each other. (b) Edge and faces are split if the vertices move far away from each other.

Global topology handling

In addition to handling local mesh topology, monitoring the distance between all vertices (connected or not) allows to detect and handle important topology breaks such as mesh splitting, vanishing, merging, and collision detection. Contrary to level set methods, here these tasks must be performed explicitly, which may result in expensive calculations, their optimization is thus a crucial aspect of the problem. In our context, we focus on the splitting aspect, and we forbid mesh merging through collision detection (we refer the reader to [Lachaud, 1998] for implementation details of mesh merging). Mesh splitting is detailed below, while collision detection will be discussed in section 4.2.2.

A typical example of mesh splitting is when a mesh is pulled into a bone shape because each extremity is attracted by a different object. As the mesh evolves, the extremities move away from each other, and the center slowly shrinks to form a tube-like structure, as illustrated in figure 4.2. As soon as the length of one of the edges (e.g. AB) falls below d_{min}, the local topology check-up requests a merging operation. However, C is a common neighbor of A and B and $\widehat{ABC}$ is not a mesh face. Merging A and B would thus create a topological inconsistency, therefore the mesh must be split first. This is done by by cutting the mesh at $\widehat{ABC}$, and by closing the wholes in the new meshes by two oriented faces $\widehat{BAC}$ and $\widehat{A'B'C'}$.

A particular case of mesh splitting occurs when one (or both) side(s) of the cutting triangle contains only one vertex. Here there is not need to cut the mesh, the vertex is simply removed, and the hole is filled with the cutting triangle. If there is only one vertex on both sides, the whole mesh is deleted.

Multi-resolution approach

Although the data structure of a discrete mesh is lighter than a 3D level set structure, the complexity is still dependent on the number of mesh vertices, which grows rapidly
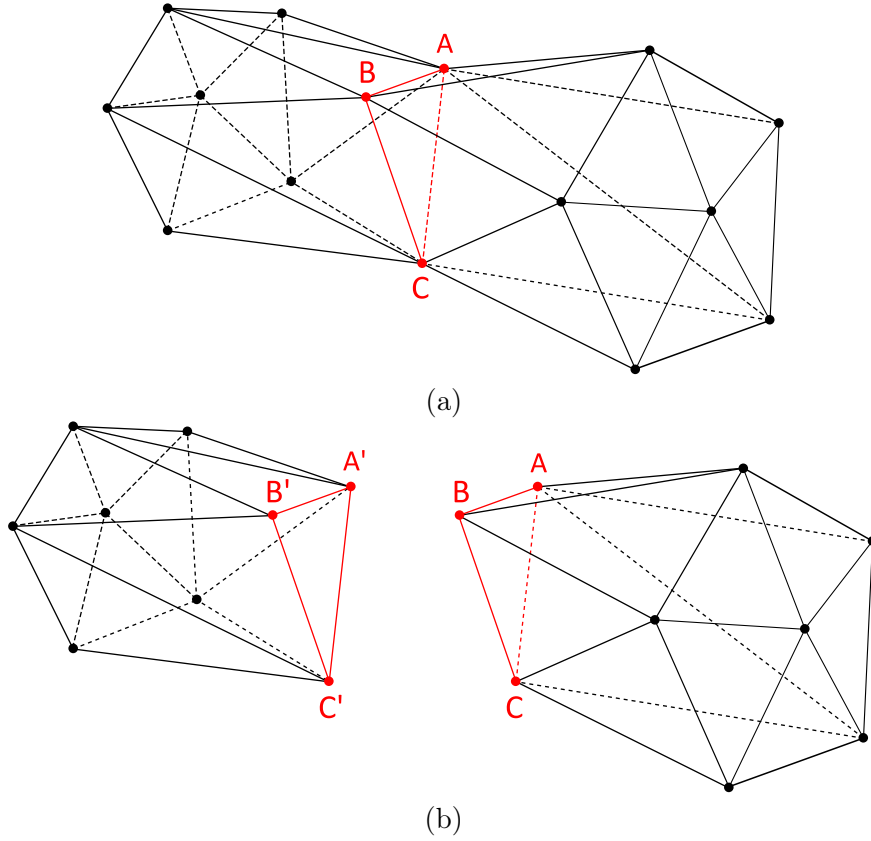
Figure 4.2: Active mesh before splitting (a) and after splitting (b). Edge AB should be merged, but they have a common neighbor C such that $\widehat{ABC}$ is not a mesh face. Therefore we use $\widehat{ABC}$ to cut the mesh.

with the mesh size and the Level Of Detail (LOD). It is thus interesting to focus on a multi-resolution strategy to speed up computations. The idea is to give high initial values to d_{min} and d_{max}, and reduce these values when the LOD should increase. This can be done either on-the-fly during the deformation (which implies to set a decrease speed for d_{min} and d_{max}), or more simply to refine the mesh globally at key steps of the evolution, let's say, after it reaches a steady-state for the current LOD. This solution allows us to employ subdivision algorithms used in computer-assisted design. Nevertheless, most of these algorithms are not suitable, since some schemes incorporate a surface interpolation process (e.g. the *Butterfly* scheme [Dyn et al., 1990]), while other schemes even modify the surface parametrization by reorganizing the existing vertices (e.g. the *Loop* scheme [Loop, 1987]). The interpolation process produces arbitrarily smooth surfaces, which is an undesirable effect in our case, since each newly created point is linked to the image through the energy functional and thus contributes actively to the mesh deformation. Therefore, we perform a subdivision similar to the *Butterfly* scheme without interpolation, which splits each face into 4 new faces (as shown on figure 4.3). This algorithm is fast and has the advantage of dividing each edge length by a factor 2 exactly, as opposed to $\sqrt{3}$ in [Lachaud, 1998]. Therefore, for every subdivision performed, d_{min} and d_{max} are also divided by 2, and the mesh preserves identical sampling properties from a LOD to another.

(a)   (b)   (c)
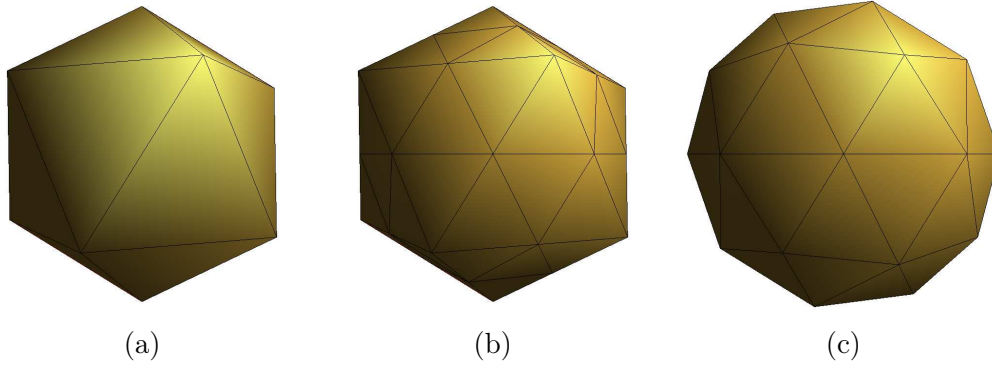
Figure 4.3: Global subdivision schemes for triangular meshes. (a) Original mesh (regular icosahedron). (b) 4-split scheme. (c) Butterfly scheme.

Morphological descriptors

Another appealing property of meshes is that morphological measurements can be directly computed from the mesh structure in a faster and more precise manner than with voxel-type structures like level sets. Indeed, efficient algorithms have been proposed to compute multiple shape features using a sum over the mesh surface instead of the whole image data (see for instance [Zhang and Chen, 2001] and the applications of Green's theorem).

A typical example is the mesh volume, which can be efficiently computed as follows: given an oriented face $\widehat{ABC}$, one calculates the cross product as $\overrightarrow{p} = (p_x, p_y, p_z) = \overrightarrow{AB} \wedge \overrightarrow{AC}$. The elementary mesh surface is then given by $ds = \|\overrightarrow{p}\|/2$, and the corresponding elementary volume by $dv = s \cdot p_x (A_x + B_x + C_x)/3$. The global mesh surface and volume are then obtained by computing and summing these values for over all mesh faces.

Finally, it should be pointed out that a discrete mesh representation offers the possibility to monitor a movement in a more efficient way than with level sets. Indeed, after segmentation of the objects on the first frame, it can be decided to disable local topology handling (under the assumption that the objects do not undergo strong deformations during the remainder of the sequence). The successive positions of each mesh vertex can then be stored in order to perform a time-lapse analysis of the deformation and, hence, of the shape descriptors themselves [Clarysse et al., 1997].

4.2.2 Mumford-Shah based multiple coupled Active Meshes

Let us now describe how the evolution equation (4.2) can be adapted to the mesh formalism. Each term must first be translated into a 3D vector. The final force that is applied to each vertex should correspond to the weighted sum of these vectors. The evolution equation can thus be rewritten into the following form:

$$\begin{cases} \overrightarrow{F} &= \tau \overrightarrow{F_D} + f_{V_0}(\overrightarrow{F_D}), \qquad \text{with} \\ \overrightarrow{F_D} &= \mu \overrightarrow{F_{int}} + \lambda \overrightarrow{F_I} + \gamma \overrightarrow{F_c}, \end{cases} \qquad (4.3)$$

where τ is the time step, $\overrightarrow{F_{int}}$ expresses the internal force that minimizes the mesh surface, $\overrightarrow{F_I}$ is the image attachment force, $\overrightarrow{F_c}$ is the coupling force that prevents mesh overlap-

ping through collision detection, and f_{V_0} is a function expressing the volume conservation constraint. We describe these terms below.

- **Internal term**

  The internal energy term of our model corresponds to an elasticity term, which aims at smoothing the local curvature of a contour by pulling each control point toward the mass center of its direct neighbors. We express this for a triangular mesh through the following force:

$$\overrightarrow{F_{int}}(v_i) = \frac{1}{card(\mathcal{N}_i)} \sum_{V_j \in \mathcal{N}_i} \overrightarrow{v_i v_j}, \tag{4.4}$$

where $\mathcal{N}_i$ is the list of direct neighbors of v_i. We remind that this force smooths the mesh curvature in a very local fashion, since only the direct neighbors of each vertex are taken into account (cf. figure 4.4). This implies that the deformations of the mesh are not too important from an iteration to the other, meaning that the regularization weight must not be too strong compared to the image force.
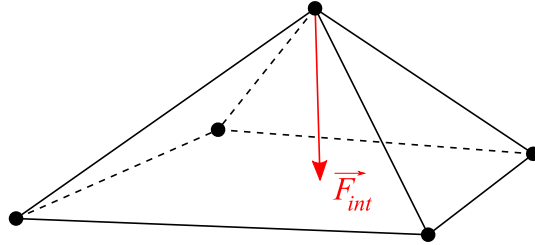
Figure 4.4: Illustration of the internal force of an active mesh.

For applications where stronger regularization is necessary, more neighborhood levels may be taken into account. A second order smoothing force would thus be written:

$$\overrightarrow{F_{int}}(v_i) = \frac{1}{card(\mathcal{N}_i)} \sum_{V_j \in \mathcal{N}_i} \left[\overrightarrow{v_i v_j} + \frac{1}{2 card(\mathcal{N}_j)} \sum_{V_k \in \mathcal{N}_j} \overrightarrow{v_j v_k} \right],$$

and so on for higher orders.

- **Image term**

  The image force is almost identical to that of equation (4.2), and can be written

$$\overrightarrow{F_I}(v_i) = \left(|c_{bgd} - I(v_i)|^2 + |c_{obj} - I(v_i)|^2 \right) \overrightarrow{N}(v_i),$$

where $I(v_i)$ is obtained by trilinear interpolation of the image voxels with closest integer coordinates to v_i. The normal vector $\overrightarrow{N}(v_i)$ can be computed for each vertex using the neighborhood information, but the process may be expensive. Fortunately, mesh normals are permanently computed by the graphics adapter for rendering purposes. The operation is hardware optimized and thus cost-free compared to the rest of the computations.

A more complex operation is to update the background mean intensity c_{bgd} and that of each object c_{obj} at certain steps of the evolution (e.g., after each deformation iteration, or after convergence on a given LOD). To update these variables, we must

(a)

(b)
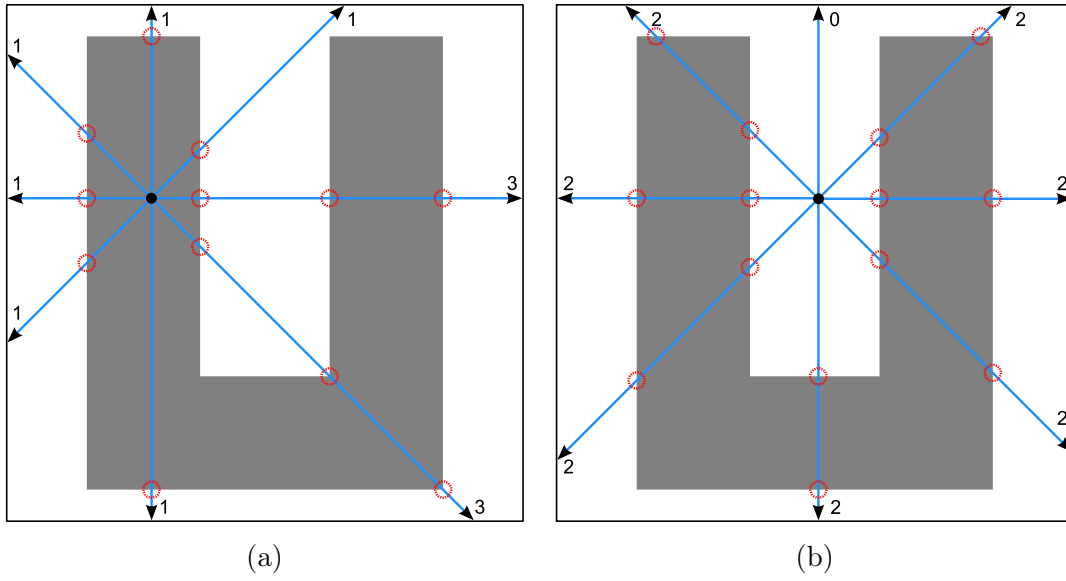
Figure 4.5: Illustration of the odd parity test derived from the Jordan curve theorem. (a) Any ray traced from an interior point crosses the polygon an odd number of times. (b) Any ray traced from an exterior point crosses the polygon an even number of times.

determine the interior and exterior regions of each mesh. In the level set case, this operation is straightforward, since the sign of φ indicates whether the current pixel lies inside or outside the contour. Here it is much more complex to determine whether a voxel is inside or outside the surface of a mesh. A solution is to use the Jordan curve theorem, stating that a point is inside a polygon if any ray traced from this point crosses the polygon surface an odd number of times (cf. figure 4.5). Hence, the interior and exterior of the mesh can be obtained by tracing such a ray for every voxel of the image (cf. figure 4.6-(a)). We call this the *Point-to-Mesh rasterization* scheme (PMR). Here again, we may benefit from the fact that ray-tracing is one of the most common tools in computer graphics, and is thus also hardware optimized by the graphics adapter. Nevertheless, repeating the operation for every image voxel yields unacceptable computational costs.

In our implementation, we have optimized this technique such that it performs only one ray tracing per image line, reducing the number of intersection tests by a factor equal to one image dimension. This new scheme, that we call the *Line-to-Mesh rasterization* scheme (LMR), sweeps through voxels lying on an edge slice of the image, and traces a ray from each voxel through the image space. For instance, one may sweep through the first slice of a stack, and trace a ray for each voxel along the depth direction. To perform a minimal number of tests, we choose to trace the rays along the largest image dimension, which is usually the image width. The LMR scheme uses an intersection function provided by the graphics library that returns a list of intersection distances that we can sort and use to extract the interior voxels efficiently (cf. figure 4.6-(b)). Further details of this rasterization algorithm are given in function `Rasterize(`$I, \mathcal{M}[]$`)` (see section 3.3).

We compare in table 4.1 the memory and time costs to compute c_{bgd} and c_{obj} on an image of size $256 \times 256 \times 256$ voxels for the Point-to-Mesh and our Line-to-Mesh schemes. The equivalent cost for the level set method is given as reference.

(a)   (b)
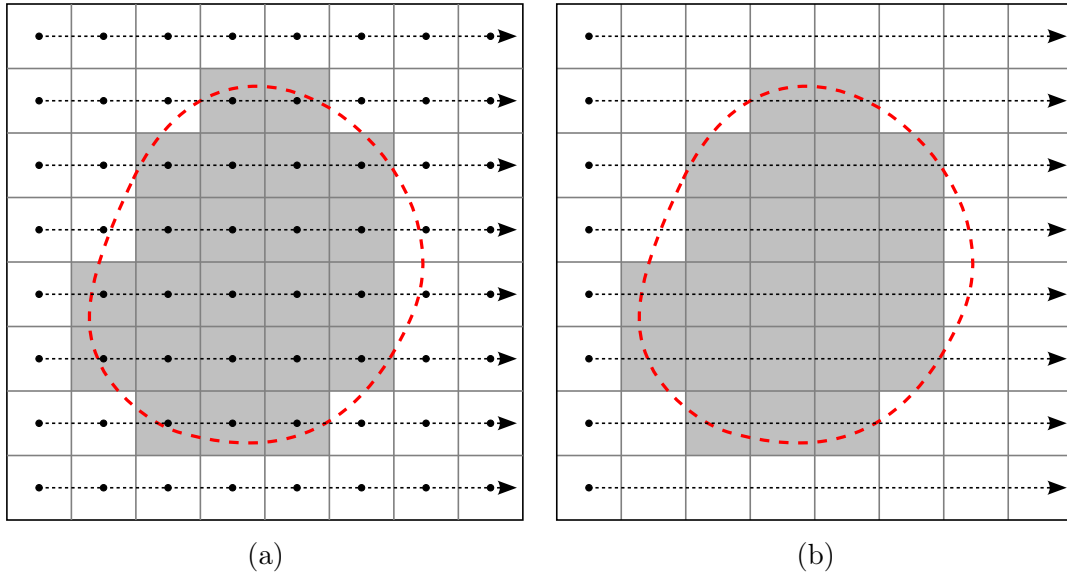
Figure 4.6: Mesh-to-Raster conversion methods. (a) In a Point-to-Mesh scheme, a ray is traced in each voxel to determine if it lies inside or outside the mesh. (b) In our Line-to-Mesh scheme, a ray is traced in the first voxel of each line. The interior and exterior regions are determined by the intersection coordinates returned by the test.

Method	Data structure	Memory load	Time cost (sec.)
Level set	2^{24} elements	64 MBytes	0.61
PM / LM	94 vertices	7 KBytes	87 / 0.62
PM / LM	366 vertices	27 KBytes	180 / 1.3
PM / LM	1455 vertices	108 KBytes	630 / 2.8

Table 4.1: Comparison of the level set, the Point-to-Mesh (PM) and Line-to-Mesh (LM) schemes for computing the interior and exterior of a contour on an image of size $256 \times 256 \times 256$ voxels.

- **Coupling term**

  The coupling term prevents meshes from overlapping, or in other terms, prevent vertices from a mesh to penetrate other meshes. This process is commonly called collision detection, and is a very important task in computer graphics, especially in physical modeling and video games. Unfortunately, the literature is mostly dedicated to collision between rigid objects, and the few algorithms proposed for deformable objects are computationally expensive, especially if one needs to compute the amount of penetration in order to produce an adequate feedback force (we refer the reader to [Lin and Manocha, 2004] for a comprehensive review of the field). Therefore, we have taken interesting ideas from both worlds and designed our own collision detection algorithm. The process consists of a progressive collision detection and can be seen as a multi-resolution collision detection approach. Given two meshes, a first test checks whether the bounding spheres intersect. If they do, a second step extracts for each mesh a list of vertices located within the bounding sphere of the other mesh. Finally, if these lists are not empty, each vertex of the list is tested for intersection with the other mesh, in order to apply a precise feedback force. This algorithm performs progressive collision handling, and is extremely fast since the number of vertex-mesh

intersection tests is much lower than in a brute-force approach. Moreover, the two most expensive operations (bounding sphere computation and ray-mesh intersection) are hardware-optimized by the graphics adapter. Further implementation details are given in algorithm 6 (see section 3.3).

- **Volume conservation constraint**
  The volume conservation is a global constraint that should maintain a relatively constant mesh volume throughout its evolution. One can determine whether the mesh grows or shrinks for a given vertex if its displacement is known. Since the mesh is a closed object, it will grow if the displacement of a vertex follows the direction of the mesh outer normal at this vertex, and shrink if it follows the opposite direction. we express the volume conservation for each vertex as a function of all the other forces described above. This function is defined as:

$$f_{V_0}(\overrightarrow{F_D}(\mathrm{v}_i)) = \eta \frac{\overrightarrow{F_D}(\mathrm{v}_i)}{\|\overrightarrow{F_D}(\mathrm{v}_i)\|} \left(\overrightarrow{N}(\mathrm{v}_i) \cdot \overrightarrow{F_D}(\mathrm{v}_i) \right) (V_0 - V),$$

where η is a positive weight, V_0 the reference volume of the mesh and V its volume at the current iteration. The term $\overrightarrow{N} \cdot \overrightarrow{F_D}$ computes the amount of growing or shrinking by projecting the displacement force $\overrightarrow{F_D}(\mathrm{v}_i)$ on the normal vector $\overrightarrow{N}(\mathrm{v}_i)$. Growing or shrinking is then determined by the sign of the dot product. The term $V_0 - V$ measures the difference between the reference volume and the current volume. The combinations of signs and forces are given in table 4.2. The scalar result is then multiplied by the normalized displacement force in order to balance $\overrightarrow{F_D}$ in (4.3).

$\overrightarrow{N}(\mathrm{v}_i) \cdot \overrightarrow{F_D}$	$V_0 - V$	$f_{V_0}(\overrightarrow{F_D}) \cdot \overrightarrow{F_D}$
+	+	+
+	-	-
-	+	-
-	-	+

Table 4.2: Direction of the volume conservation force depending on the displacement force of each vertex and the difference between the reference and current mesh volumes.

4.2.3 Initialization

In the active surfaces method, the initialization was straightforward, since the mask resulting from the pre-processing step is in the same format as a level set structure. Here however, the binary mask must be converted into a mesh representation. This is precisely the purpose of 3D reconstruction techniques such as Voronoi diagrams, Delaunay triangulation and marching algorithms. Among those possibilities, we have chosen the Marching Tetrahedra for three main reasons. Firstly, the output must necessarily be a triangular mesh in order to be seamlessly compatible with our framework. Secondly, marching algorithms possess a scale parameter that specifies the reconstruction resolution. Therefore, contrary to other methods, very fast low-detail reconstruction can be achieved, which perfectly fits our multi-resolution approach. And finally, marching tetrahedra are more robust than marching cubes regarding ambiguous spatial configurations. An example is given in figure 4.7.

(a)   (b)

(c)
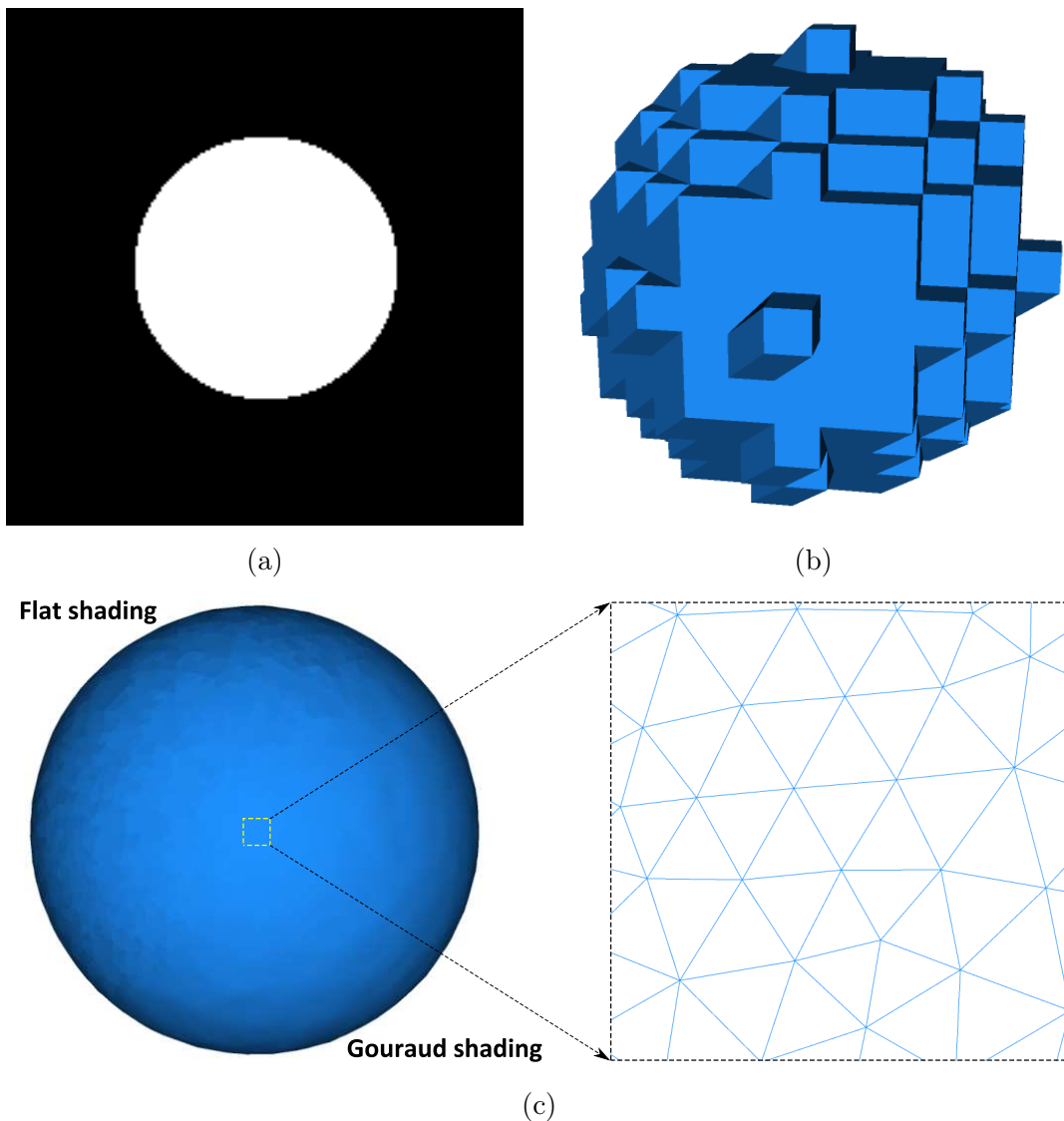
Figure 4.7: Segmentation of a 3D sphere using an active mesh. (a) Slice of the input mask. (b) Initialization obtained by a coarse reconstruction of the input mask (reconstruction time: 0.5 sec.). (c) Final result after evolution, including three global refinements in order to reach the image resolution (total evolution time: 3.5 sec.). The right frame shows a magnified view of the surface, showing the benefit of trilinear interpolation on the smoothness of the mesh.

4.2.4 Comparison with the Active Surfaces

Both methods rely on the same base energy functional. Therefore it is clear that they will yield very similar results in terms of segmentation and tracking results. For the same reason, it is also obvious that both models suffer from the same requirement that objects positions overlap from a frame to another. This issue yet needs to be solved.

Multi-object segmentation is treated in a simpler way than with active surfaces. As soon as a mesh splits, two new meshes are created automatically, and the evolution carries on normally. A similar procedure could be done with the level set formulation, but it would require to extract the number of connected components inside each level set after

each iteration, which is a very expensive process.

Active meshes are light in memory. Level set methods hold a structure that is of the size of the image, and the active surfaces require one level set structure per object being tracked. Here, the only big structure in memory is the image. Each mesh holds a very small space in memory since it is formed of a small set (compared to the image size) of 3D vertices and their direct neighborhood information. This allows to work with bigger images, and track more objects. This gain in memory however has the drawback that interior and exterior regions of each mesh are more complex to compute, due to the local nature of mesh models. Although we have developed an optimized rasterization scheme, the computation time is still more important than for the level set framework, where interior and exterior regions are maintained through the sign of the level set function.

Another advantage of mesh models over 3D level set models, regardless of their applications, is their visualization efficiency. Indeed, meshes perfectly fit the pipeline of current graphics adapters, allowing fast rendering on screen, and optimization of many calculations of the model. We have taken advantage of these properties to create a user-interface allowing to visualize the meshes deformations in real-time, during the energy minimization process (see figure 4.8).

These advantages have a great impact on the biological applications that can be dealt with. Indeed, real-time visualization is nowadays a common request from biologists who wish to understand (and eventually tweak) the analysis process, while producing screen-captures and movies to illustrate their work. On the other hand, low memory consumption and fast computation time are key requirements for fast and automated analysis tasks, especially in the context of High-Content Screening (HCS) applications.

Application to real biological problems will be presented in chapter 5.7. Before that, we will perform in the next chapter a quantitative evaluation of both methods on simulated and trial data sets.

4.3   Algorithms & Pseudo-codes
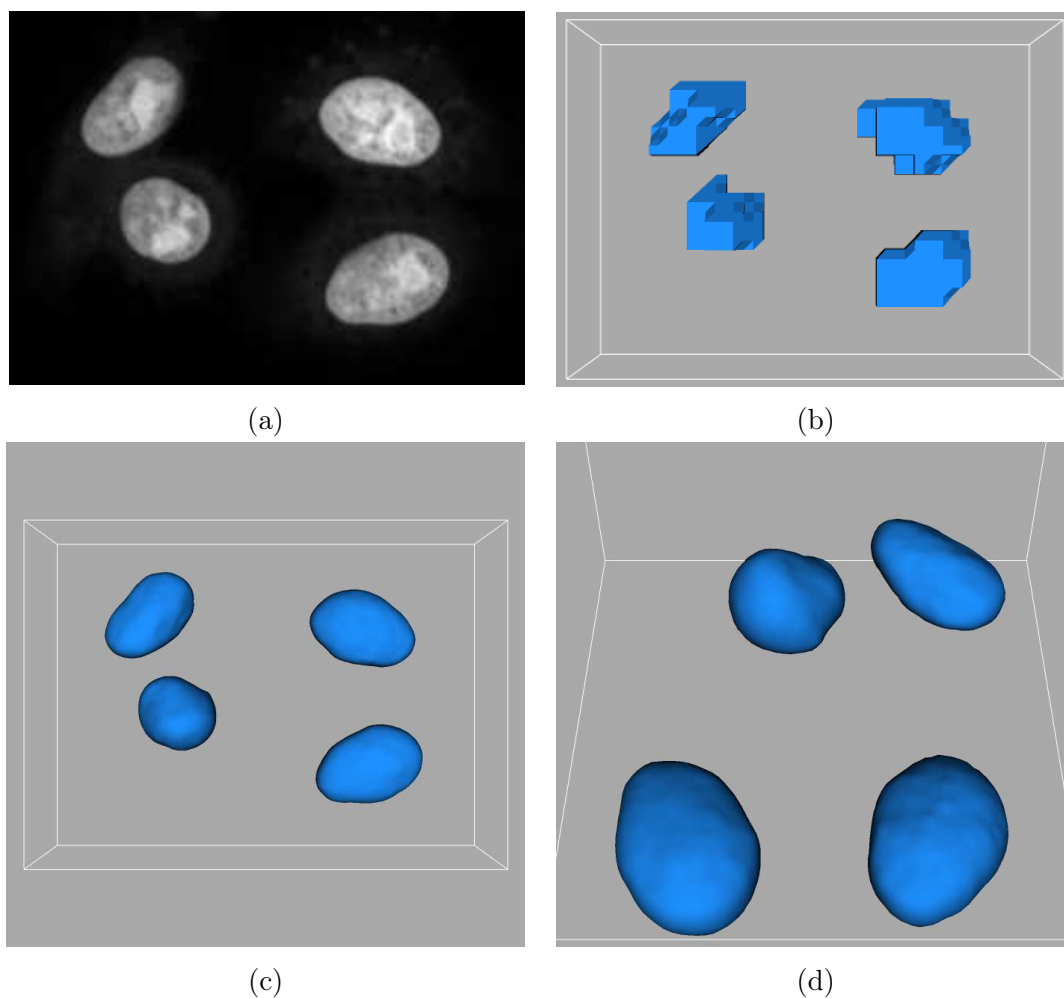
Figure 4.8: Segmentation result on the 4 objects of figure 3.5. (a) Slice of the original image of size $245 \times 177 \times 45$ voxels and resolution $0.327 \times 0.327 \times 0.75\mu m$ per voxel. (b) Coarse reconstruction of the binary mask (reconstruction time: 1.1 sec.). (c,d) Two views of the segmentation result after evolution and three refinements (evolution time: 8.5 sec.).

Procedure `CollisionDetection`($\mathcal{M}_1$, $\mathcal{M}_2$). Performs a fast progressive collision detection and applies feedback forces to the colliding vertices.

Input: $\mathcal{M}_1$: a mesh, $\mathcal{M}_2$: another mesh

1 $c_1 \leftarrow$ `Center`($\mathcal{M}_1$), $c_2 \leftarrow$ `Center`($\mathcal{M}_2$)
2 $r_1 \leftarrow$ `Radius`($\mathcal{M}_1$), $r_2 \leftarrow$ `Radius`($\mathcal{M}_2$)
3 **if** `Distance`(c_1, c_2) $> r_1 + r_2$ **then exit**

4 List all vertices v in $\mathcal{M}_1$ such that `Distance`(v, c_2) $< r_2$
5 If the list is empty, **exit**

6 **foreach** *vertex* v *found* **do**
7   Trace a ray R from v to c_1
8   **if** R *intersects* $\mathcal{M}_2$ *in face* F **then**
9   Store a feedback force directed by R with length `Distance`(v, F)
10   **end**
11 **end**
12 Exchange $\mathcal{M}_1$ and $\mathcal{M}_2$ and repeat steps 4 to 11
13 Apply all feedback forces to their respective vertices

Function `Rasterize`(I,$\mathcal{M}$[]). Rasterizes the meshes against the image I to compute the mean intensities of each object and of the background.

Input: I: an image, $\mathcal{M}$[]: a mesh array.

1 Initialize a list L_{out} of background voxel coordinates with all image voxels
2 Initialize n empty lists L_{in} of interior voxel coordinates

3 **foreach** *mesh* $\mathcal{M}_i$ *in* $\mathcal{M}$[] **do**
4   **for** $pZ = 1$ **to** `Depth`(I) **do**
5   **for** $pY = 1$ **to** `Height`(I) **do**
6   Create a ray R with origin $(1, pY, pZ)$ and direction $(1, 0, 0)$
7   `CrossDist`[] $\leftarrow$ `Intersect`($R, \mathcal{M}$)
8   **if** `CrossDist` *is empty* **then loop**
9   **foreach** *distance interval* [`CrossDist`[i], `CrossDist`[$i+1$]], *with* i *odd* **do**
10   **for** $pX =$ `CrossDist`[i] **to** `CrossDist`[$i+1$] **do**
11   Add (pX,pY,pZ) to L_{in}[i]
12   Remove (pX,pY,pZ) from L_{out}
13   **end**
14   **end**
15   **end**
16   **end**
17   Compute the mean object intensity c_{obj}[i] from I using L_{in}[i]
18 **end**
19 Compute the mean background intensity c_{bgd} from I using L_{out}

20 **return** c_{obj}[], c_{bgd}

Function MultipleMeshEvolution(I,$\mathcal{M}[]$,$c_{obj}[]$,c_{bgd}). Minimizes the proposed functional (3.9).

Input: I: an image, $\mathcal{M}[]$: a mesh array, $c_{obj}[]$: the objects mean intensities, c_{bgd}: the background mean intensity, L: level of detail of the meshes.

Compute E according to equation 4.1 ;
repeat
    foreach *mesh $\mathcal{M}$ in $\mathcal{M}[]$* **do**
        foreach *vertex* V*in $\mathcal{M}$* **do** Move Vaccording to equation 4.3 ;
        foreach *edge* (V_1, V_2) *in $\mathcal{M}$* **do**
            if Length$(V_1, V_2) < d_{min}(L)$ **then** Merge (V_1,V_2) and break ;
            if Length$(V_1, V_2) > d_{max}(L)$ **then** Split (V_1,V_2) and break ;
        end
        if *edges have been merged or split* **then** return to step 5 ;
        else if *$\mathcal{M}$ has been split* **then**
            $(c_{obj}[], c_{bgd}) \leftarrow$ Rasterize$(I, \mathcal{M}[])$;
            return to step 1 ;
        end
    end
    foreach *mesh pair $(\mathcal{M}_i, \mathcal{M}_j)$ in $\mathcal{M}[]$* **do** CollisionDetection ;
until *E has converged* ;

Algorithm 9: Main process of the Active Meshes framework

Input: S: image sequence with T frames, LOD: level of detail of the meshes

$I \leftarrow$ S$[1]$;
$B \leftarrow$ Kmeans$(I, 2)$;
$c_{bgd} \leftarrow$ Mean$(I, B = 0)$;
Extract all connected components of B into a binary mask array $B[]$ of size n;
for *i=1* **to** n **do**
    $c_{obj}[i] \leftarrow$ Mean$(I, B[i] = 1)$;
    $\mathcal{M}[i] \leftarrow$ MarchingTetrahedra$(B[i], LOD)$;
end
repeat
    $\mathcal{M}[] \leftarrow$ MultipleMeshEvolution$(I, \mathcal{M}[], c_{obj}[], c_{bgd})$;
    $(c_{obj}[], c_{bgd}) \leftarrow$ Rasterize$(I, \mathcal{M}[])$;
    Perform a global refinement the meshes ;
until LOD *is coherent with the image resolution* ;
foreach *t=2* **to** T **do**
    $I \leftarrow$ S$[$t$]$;
    $\mathcal{M}[] \leftarrow$ MultipleMeshEvolution$(I, \mathcal{M}[], c_{obj}[], c_{bgd}, $false$)$;
end

5

Experiments and results

The aim of this chapter is to propose a quantitative performance evaluation of the methods presented in the previous chapters. Since ground truth is very complex to produce in biological imaging, the first part of the evaluation will be performed on simulated data, in which we have attempted to reproduce most of the known disturbances due to the image acquisition process. To do so, we start by describing the different imaging systems and protocols available in the institute, and select the imaging system that we wish to reproduce. In a second step, similar evaluation will be performed on images of fluorescent beads, for which real information is available (size, shape, etc.). Experiments on real biological applications will be discussed in the final section of this chapter.

Contents

5.1 Evaluate what and how ?

There are several approaches to evaluate the performance of a given algorithm in image analysis. The most comprehensive reviews of image segmentation algorithms have been published in [Zhang, 1996], [Zhang, 2001] and recently in [Zhang, 2006]. Evaluation approaches can be grouped into two categories. On one side, unsupervised approaches evaluate a method based on its mathematical and practical properties (e.g., relevance to the problem, complexity, speed, convergence, stability). On the other side, supervised approaches evaluate the quality of the segmentation result against various criteria, either based on user-defined "goodness" measures, or based on discrepancy to ground-truth data.

In biomedical imaging, discrepancy approaches are the most popular, but their reliability mostly depends on the quality of the ground-truth. In the ideal case, precise ground-truth would be based on precise and exhaustive knowledge of the objects or structures being segmented as well as the various characteristics of the imaging process. A well-known example is the *BrainWeb* database [Collins et al., 1998], which holds a theoretical model of the human brain accompanied by a complete set of noise simulation processes imitating the different modalities or magnetic resonance imaging. In such context, segmentation quality can be evaluated in a very precise way. Unfortunately, precise theoretical models are not always available for the problem in hand due to insufficient information on the objects. To cope with this limitation, artificial ground-truth can be obtained by manual segmentation performed by experts on a set of trial images. This task is tedious, especially for large 3D image databases, and prone to reproducibility issues and variability between experts.

In the absence of a precise ground-truth, we must explore various validation approaches in order to estimate the performance of the proposed methods. In a first step, we shall evaluate the performance of both contributions on simulated data that tend to imitate real imaging conditions. This first evaluation will confirm that the segmentation quality is globally equivalent for both methods, allowing us to restrict further evaluation to one of these methods. The second part of the validation process is performed on fluorescent beads for which ground-truth is given by their diameter. The final part of the evaluation be performed on real biological objects, and more particularly on 3D images of nuclei having a relatively low shape variability.

5.2 Experimental setup and imaging protocols

Trial and real biological experiments have been conducted on three different imaging systems. Let us describe their characteristics in table 5.1.

5.3 Evaluation on synthetic data

Efficient and realistic simulation of fluorescence microscopy data is very complex topic due to the great variety of optical systems, fluorescent markers, and imaging conditions. Yet, the most reliable approach is to adapt the simulation specifically to the imaging systems and protocols that will be used for the real experiments. In this section, we will aim at simulating images that imitate the imaging conditions of the Opera system (see table 5.1).

Device name	LSM 5 *Live*, Zeiss (Germany)
Type	Laser-light confocal line-scanning microscope
Lens	Oil-immersed plan apochromat 63× (refraction index: 1.515)
Optical magnification	1×
Numerical Aperture	1.4
Optimal resolution	$0.1 \times 0.1 \times 0.1 \mu m$
Typical slice size	512×512
Description	The LSM 5 *Live* microscope is one of the fastest commercial microscope for the acquisition of 3D images, thanks to a line-shape detector able to acquire 512 pixels at once. The drawback is the asymmetrical nature of the PSF, inducing atypical disturbances in the images, especially for small objects.

Device name	Opera screening platform, Evotec (Germany)
Type	Laser-light confocal spinning Nipkow-disk microscope
Lens	Water-immersed plan apochromat 40× (refraction index: 1.33)
Optical magnification	1×
Numerical Aperture	0.9
Optimal resolution	$0.327 \times 0.327 \times 0.4 \mu m$
Typical slice size	688×520
Description	The Opera platform is a motorized microscope designed for High-Content Screening (HCS) applications. The system is able to image an entire well-plate in a fully automated manner, allowing to produce robust statistics if coupled with a robust and automated image processing algorithm. The main drawback is the poor spatial resolution, limiting the range of potential applications.

Device name	Eclipse 90i, Nikon (Japan)
Type	Halogen-lamp widefield microscope
Lens	Oil-immersed plan apochromat 100× (refraction index: 1.515)
Optical magnification	2×
Numerical Aperture	1.3
Optimal resolution	$0.03 \times 0.03 \times 0.1 \mu m$
Typical slice size	1280×1024
Description	This traditional widefield microscope can reach very high spatial resolutions and detect lower levels of light than confocal microscopes. Unfortunately, widefield microscopes suffer from a stronger influence of the microscope PSF, therefore images *must* be deconvolved before they can be processed, which is not always necessary for confocal microscopes.

Table 5.1: Description of the imaging devices used throughout the experiments.

5.3.1 Segmentation evaluation

The first step of our evaluation process focuses on the segmentation aspect of the proposed models. The objects we intend to detect are homeomorphic to an ellipse, but size varies from an application to another. In order to account for various imaging conditions, we have generated three sets of 100 images containing random-sized ellipses with different noise rates (cf. figure 5.1).

Data simulation

Many efforts have been carried out to develop a generic simulation method, but each practical application usually requires a very specific simulation method to account for both biological and imaging conditions. Therefore, we propose below a noise simulation process that imitates the major characteristics of the optical system of a typical fluorescence microscope. The pseudo-code for the simulation process is given in the algorithm 10.

Algorithm 10: Synthetic data simulation process for segmentation evaluation.

Input: W,H,D: image dimensions, N: number of objects to generate, Rmin: minimum sphere radius, Rmax: maximum sphere radius, σ: standard deviation of the Gaussian noise.

1 Create an empty image of size $W \times H \times D$
2 Generate N non-touching ellipses of radius between Rmin and Rmax voxels
3 Store the current data as the ground-truth image
4 Set the intensity values for the background and the objects
5 Convolve the stack with the microscope PSF model
6 Convert the image to its Poisson distribution
7 Add Gaussian noise with zero mean and standard deviation σ
8 Downscale the depth resolution to about $D/3$

- Step 1: The original volume reflects the real world, therefore the resolution should be identical in all dimensions. Following the Opera™™ protocol, we have used a spatial resolution of $0.327 \times 0.327 \times 0.327 \mu m$ per voxel.

- Step 2: Ellipses are generated at random locations in the image, under the constraints that they must not touch each other, otherwise clusters will be considered as a unique object during the pre-processing step. Also, we prevent ellipses from lying on the image border, since shape descriptors have no meaning for partially visible objects. Rmin and Rmax are set to 10 and 50 voxels respectively, in order to evaluate the robustness against size variability.

- Step 4: The background intensity value should represent the auto-fluorescence of the medium. By setting a constant value, we assume a homogeneous fluorescent background. In case more realistic simulation is needed, one may reproduce real backgrounds estimated from real images, as done for instance in [Boulanger et al., 2006]. To simulate challenging conditions, we have set the values to 10 units for the background and 20 for the objects (in practice the ratio is generally much lower than 50%).

- Step 5: The microscope PSF is a major source of disturbance in fluorescence microscopy. Its effects are mostly visible on small spot-like objects, therefore precise modelization and deconvolution is a topic of major interest in biological imaging research [Voort and Strasters, 1995], [van Kempen, 1999], [Dusch et al., 2007]. While this task is complex and strongly dependent on the optical system, it is a common consensus to represent the PSF using a Gaussian approximation [Zhang et al., 2007]. The formulas not only depend on the optical system but also on the emission and excitation wavelengths of the fluorescent molecules. We have chosen to simulate the DRAQ5 fluorescent dye (nuclear marker), since it generates the biggest PSF disturbance, with a standard deviation of $0.115 \mu m$ in the XY plane and $0.519 \mu m$ in the Z plane.

- Step 6: The number of photons acquired by the camera detector is subject to quantum noise. We simulate this phenomenon by applying a Poisson noise process depending on each voxel intensity.

- Step 7: The additive Gaussian noise is used to represent all the other disturbances induces by the optical system that have not been explicitly modelized in the previous steps. By changing the standard deviation of this distribution, we will change the signal-to-noise ratio (SNR) of the final simulated images. Here we have generated three sets of data, one with high SNR using $\sigma = 5$ voxels, one with medium SNR with $\sigma = 10$ voxels, and one with low SNR using $\sigma = 15$ voxels.

- Step 8: The downscaling process imitates the incoherent resolution between the XY plane axis and the depth axis of most microscopes. Although the Opera™ system can produce depth resolutions as low as $0.4 \mu m$, we choose to downscale the images to $1 \mu m$ per voxel, which is approximately three times the XY plane resolution. This choice is frequent for real images since the final images require smaller storage space without losing too much information.

Results

We have measured the segmentation quality using a discrepancy measure, i.e., a measure of difference between the result and the original binary object in terms of misclassified voxels. However, instead of simply calculating the number of misclassified voxels, it is important to give a certain weight to this error [Ciofolo, 2005]. In this sense, misclassified voxels located very close to the real object boundary are assumed less problematic than those located far away from the object, suggesting a wrong segmentation. Therefore, we give three discrepancy measures: one absolute error corresponding to the exact proportion of misclassified voxels, one with an error tolerance of one voxel and one with an error tolerance of two voxels. Evaluation of the active surfaces and active meshes are given in tables 5.2 and 5.3 respectively, with a sample illustration in figure 5.2.

These measures were obtained using an optimal set of model parameters that was determined empirically. The tables above do not show noticeable differences between the models. Nonetheless, one can notice that the active mesh framework performs better in the current context. The difference between the approaches is the computation time which is much lower with active meshes. Moreover, all 3D snapshots presented in this section have been obtained in a straightforward manner, whereas the Marching Tetrahedra algorithm
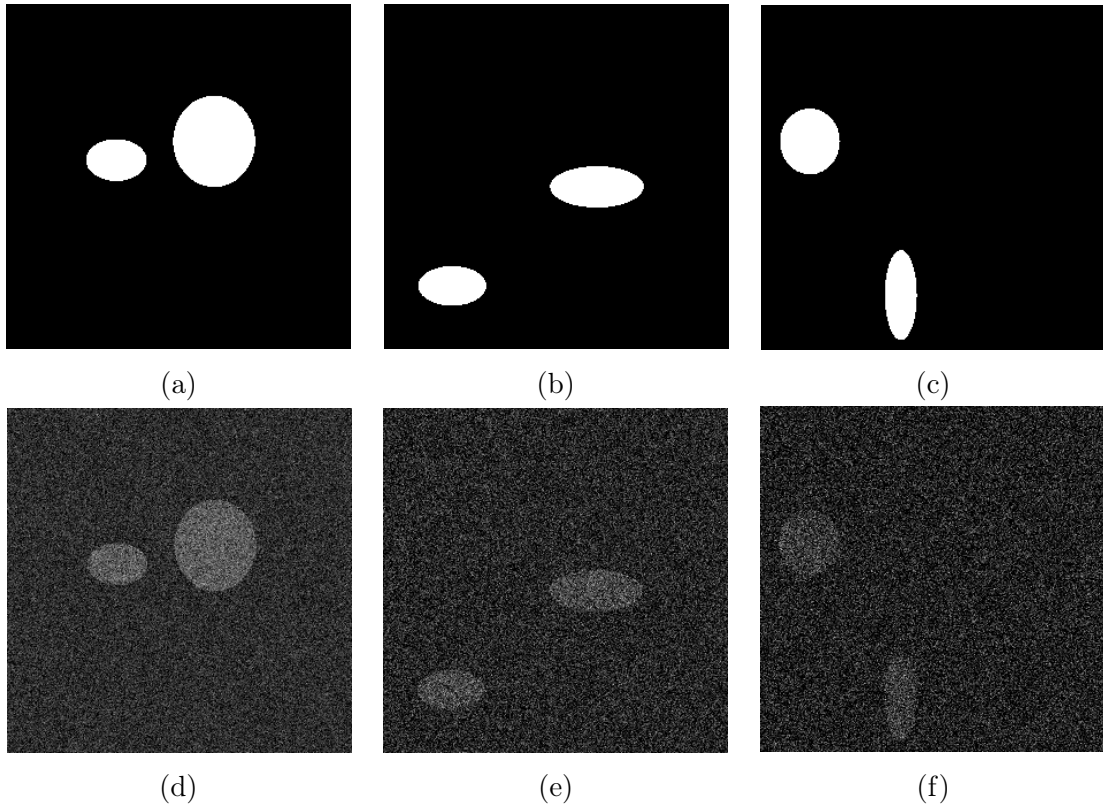
Figure 5.1: Samples of the three data sets with different signal-to-noise ratios, generated using algorithm 10. (a,b,c) Slices of three original images. (d) Noisy image slice with high SNR ($\sigma = 5$). (e) Noisy image slice with medium SNR ($\sigma = 10$). (f) Noisy image slice with low SNR ($\sigma = 15$).

Image noise	misclassified voxels			time
	absolute error	> 1 voxel away	> 2 voxels away	(seconds)
low ($\sigma = 5$)	4.6%	0.1%	0.03%	52
medium ($\sigma = 10$)	11.5%	0.8%	0.2%	78
high ($\sigma = 15$)	12.1%	1.9%	1.3%	125

Table 5.2: Segmentation evaluation of the active surface model on the synthetic data sets. Values for each set are averaged over 100 images. One can see that the segmentation error always lies within a small distance to the real object contour. Computation time includes initialization and evolution, but not the 3D reconstruction.

was applied on the results of the active surfaces to produce approximately similar images. As illustrated in figure 5.2, one can clearly see the digitized effect of the image grid on the reconstructed surfaces, compared to the smooth mesh surfaces obtained with same lighting and shading conditions. It should finally be noted that computation times mentioned in the tables above do not include the 3D reconstruction step. If both times were added, the active mesh model would appear almost 10 times faster than its level set counterpart for slightly better results.

Since the active mesh model performs equivalently to the active surface model, with the advantage of providing easy visual feedback, we will focus more particularly on the evaluation of this method throughout the remainder of this chapter.
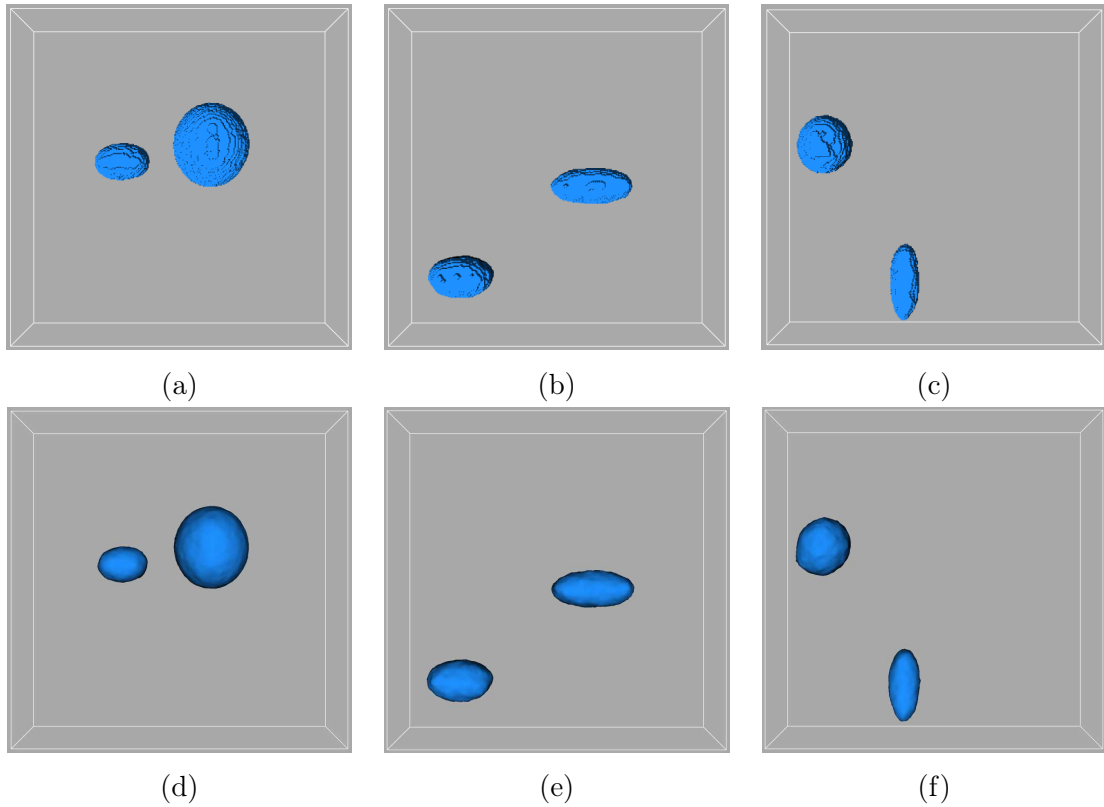
Figure 5.2: Segmentation results on synthetic data using active surfaces (top row) and active meshes (bottom row). (a,d) Result on a high SNR sample (reconstruction time for the active surfaces: 73 sec.). (b,e) Result on a medium SNR sample (reconstruction time for the active surfaces: 31 sec.). (c,f) Result on a low SNR sample (reconstruction time for the active surfaces: 37 sec.).

Image noise	misclassified voxels			time
	absolute error	> 1 voxel away	> 2 voxels away	(seconds)
low ($\sigma = 5$)	3.5%	0.01%	0%	10
medium ($\sigma = 10$)	8.2%	0.3%	0.04%	13
high ($\sigma = 15$)	12.4%	1.5%	0.4%	15

Table 5.3: Segmentation evaluation of the active mesh model on the synthetic data sets. Values for each set are averaged over 100 images. As for the active surfaces, segmentation error always lies within a small distance to the real object contour. Computation time includes initialization, evolution and real-time visualization.

5.3.2 Tracking

Tracking evaluation is performed in two steps. First, we measure the efficiency of the contour coupling term, by verifying that touching objects are not confused by the model. In a second step, we will show how the volume conservation constraint can further improve these preliminary tracking results. For this experiment we have generated 100 binary sequences of 3 frames each, where two sphere-shaped objects are separated in the first frame, touch in the second and separate in the third. In order to simulate realistic contact, the objects overlap by a small amount of voxels in the second frame. Note that we have not

employed noisy data here as we do not want to modify the evaluation with an additional disturbance factor that has already been studied in the previous section.

Tracking results are presented in table 5.4. This table describes three different cases. The first column presents results with neither coupling nor volume conservation. In the second column, only coupling is involved. An illustration is given in figure 5.3. Finally, the last column presents results using volume conservation on top of the coupling, as illustrated in figure 5.5.

	no coupling		coupling alone		full model	
	id.	error	id.	error	id.	error
T=1 (before contact)	yes	0.15%	yes	0.15%	yes	0.03%
T=2 (during contact)	no	81%	yes	3.55%	yes	0.05%
T=3 (after contact)	no	99%	yes	0.21%	yes	0.04%

Table 5.4: Tracking performance results on 100 contact situations using an uncoupled model, a coupled model without volume conservation, and our full model (cf. figures 5.3 and 5.5). The "id." column indicates whether the objects are successfully distinguished by the model. The "error" column is the absolute misclassification error, as defined in the previous section.

| frame $T = 0$ | frame $T = 1$ | frame $T = 2$ |
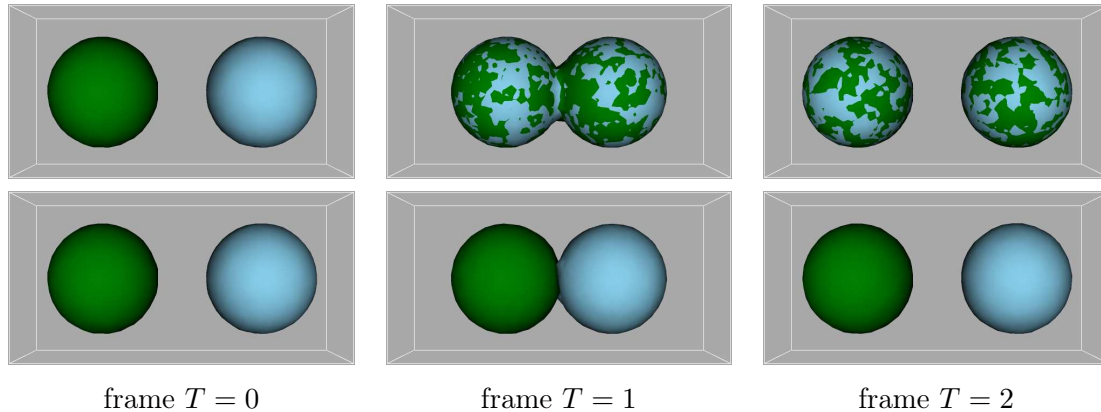
Figure 5.3: Impact of the coupling term on touching objects. Top row shows results with uncoupled meshes, while bottom row shows uncoupled meshes. On the first frame, object identification in unambiguous. On the second frame, uncoupled meshes inter-penetrate and mutually absorb each other, whereas coupled meshes stop when they come into contact. On the last frame, uncoupled meshes split to track the migrating objects, resulting in both erroneous segmentation and identification, while coupled meshes track their respective object correctly.

5.4 Shape descriptors

The next steps in our experiments do not rely on precise ground-truth. Therefore, we shall evaluate the quality and robustness of the methods based on the following set of shape descriptors: *Surface, Volume, LongAxis, Roughness, RadiusCV, HullDiff*. While *Surface* and *Volume* are quite self-explanatory, the other criteria are less obvious. We detail below.

- The *LongAxis* measure is the longest distance between two mesh vertices, eventually serving as an object elongation indicator.

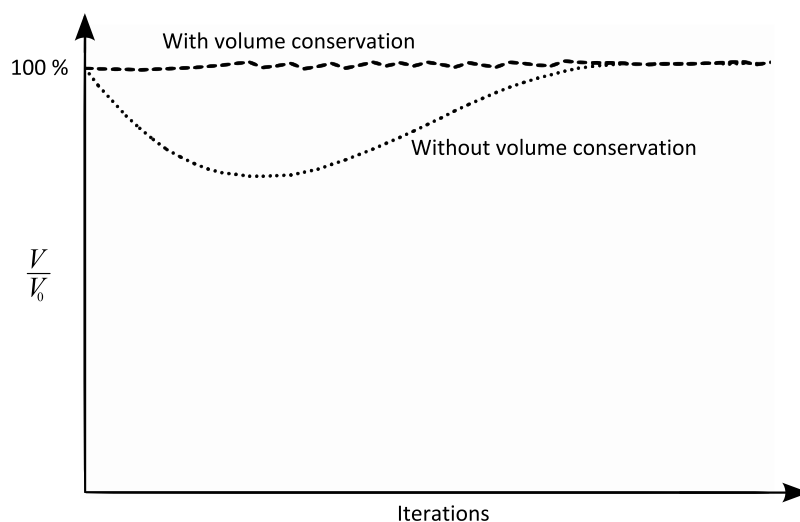Figure 5.4: Evolution of the mesh volume during its deformation with and without volume conservation. Due to the regularization term of the energy functional, mesh shrinking is favored over mesh growing, therefore the volume of unconstrained mesh reduces in the first iterations. With volume conservation, shrinking and growing are simultaneous, yielding a constant volume during the deformation.

<div align="center">frame $T = 0$ frame $T = 1$ frame $T = 2$</div>
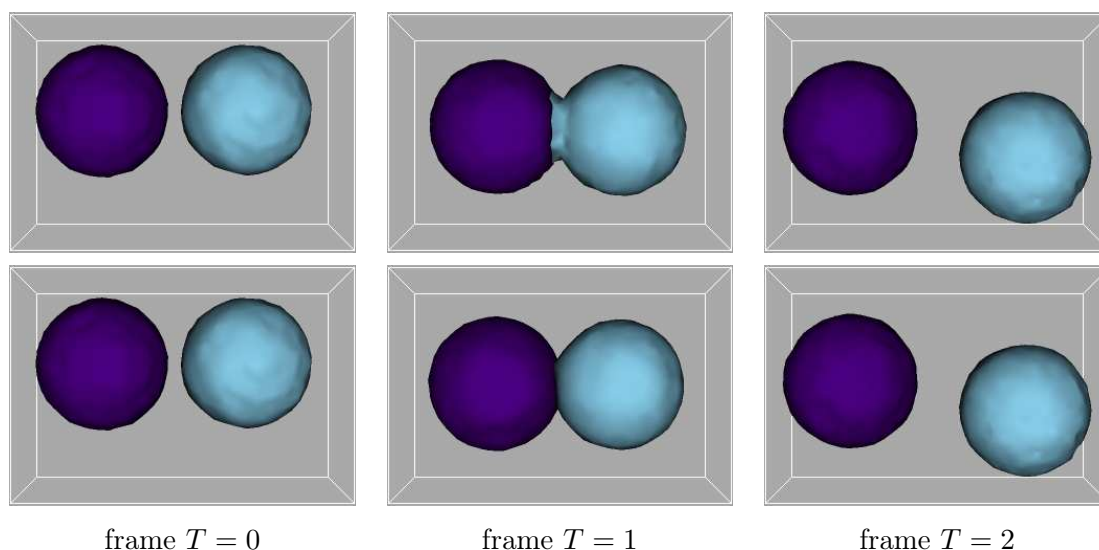
Figure 5.5: Impact of the volume conservation constraint when tracking objects with different speed. Without volume conservation (top row), the right-hand mesh segments a portion of the other object. With volume conservation (bottom row) meshes keep their original volume, as measured on the first frame, yielding correct results.

- The *Roughness* is a measure outlining the local vibrations of the surface membrane. This measure should be low for convex objects and higher when the surface exhibits local concavities. To compute this value, we start by defining a local curvature measure for each mesh vertex v as the dot product between the outer normal $\overrightarrow{N_V}$ (of unit length) and the barycentric normal $\overrightarrow{B_V}$ linking v to the center of its neighbor vertices in the mesh (see figure 5.6). If the vectors have opposite directions (i.e. negative dot product), the surface is locally convex. If the vectors have same directions (i.e. positive dot product), the surface exhibits a local concavity at the given vertex.

Finally, the roughness measure is defined as the standard deviation of all the local curvature values. Reference value is 0 for a sphere.

$$\overrightarrow{N_v} \cdot \overrightarrow{B_v} < 0 \qquad\qquad \overrightarrow{N_v} \cdot \overrightarrow{B_v} > 0$$
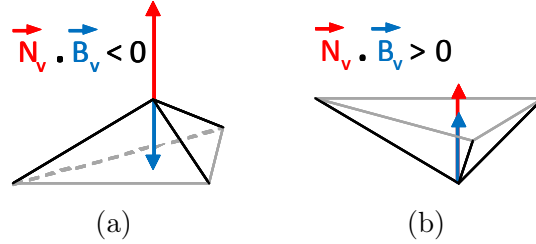
(a)   (b)

Figure 5.6: Description of the roughness measure at a given vertex v. (a) In case of a local convexity, the outer normal (red) and barycentric normal (blue) have a negative dot product. (b) In case of a local concavity, the dot product is positive.

- The *RadiusCV* measure describes how different the object shape is from a sphere. This measure is obtained for each mesh by computing the standard deviation of the distances between the mass center and each vertex, normalized by the mean radius (definition of the coefficient of variation). Reference value is 0 for a sphere.

- The *HullDiff* measure is the difference percentage between the volume of the object and that of its convex hull (i.e. the smallest convex surface that can contain it). This measure will be useful to discriminate bean-shaped objects for instance. Reference value is 0 for a convex object.

Note that these descriptors may not only be used to validate the methods performance but also constitute a first basis to shape description in a real biological context, if they prove to be independent and discriminant.

5.5 Fluorescent beads experiment

5.5.1 Protocol

A screening experiment was conducted on a well-plate containing $10\mu m$ fluorescent beads located randomly in the wells (96-Whatman without skirt, Evotec, Germany). Images were acquired on an automated Nipkow-disk confocal microscope (Opera™, Evotec, Germany) equipped with a water-immersed plan apochromat 40x lens of numerical aperture 0.9 (Olympus, Japan).

5.5.2 Results

Expected values and average measures over 100 beads are given in table 5.5. Although all measures are close from the expected values, detected objects seem generally bigger than the real objects (e.g. the *LongAxis* measure is 14% higher). This is due to the blurring effect of the microscope PSF along the Z axis. It is known that this disturbance is inversely dependent on the object size, i.e., small objects are more affected than big objects. Although we did not have bigger beads to test on, we verified that the LongAxis measure climbed up to 50% higher than expected on beads of diamater $5\mu m$, and up to 90% on beads of diamater $2.5\mu m$, suggesting the strong need for deconvolution before processing

the images. Fortunately, the objects we focus on have a diameter of about twice the order of the biggest beads, therefore the PSF effect should be neglectable.

	LongAxis	Surface	Volume	Roughness	RadiusCV	HullDiff
Expected	10	314.1	523.5	0	0	0
Measured	11.4	326.9	546.2	0.02	0.09	0.002
Coef. Var.	0.016	0.230	0.015	0.058	0.032	0.165

Table 5.5: Evaluation of shape descriptors on 10 μm fluorescent beads. Measured values are averaged over 100 beads. Coefficients of variation below 1 indicate low-variance populations.

5.6 Trial screening experiment

5.6.1 Protocol

A first experiment was conducted on the HEK-293 cell line (Human Embryonic Kidney), and a second experiment on the Hela cell line (Henrietta Lack). All cells were grown on 96-well optical bottom plates, black (Greiner) under same culture conditions (DMEM with 10% FBS). Nuclei were labeled using DNA-specific DRAQ5 fluorescent dye (Biostatus, UK) following the instructions of the manufacturer. Images were then acquired at room temperature using $633nm$ excitation wavelength with $650nm$ long pass emission filter.

Z-stacks of HEK-293 nuclei were obtained on the LSM 5 *Live* microscope. 22 Z-stacks were acquired, with size $512 \times 512 \times 60$ voxels and spatial resolution $0.28 \times 0.28 \times 0.5$ μm.

For the Hela nuclei, images were acquired on the Opera™ microscope. The experiment was conducted on 20 wells of a 96-well plate, with one Z-stack per well, yielding 20 Z-stacks of size $688 \times 520 \times 31$ voxels and spatial resolution $0.327 \times 0.327 \times 0.75$ μm.

5.6.2 Results

Figures 5.7 and 5.8 present results for the HEK-293 and Hela cells experiments respectively. Left images show a maximum intensity projection (MIP) of one of the Z-stacks. Middle images show a snapshot of the 3D scene taken right after initialization. One can clearly see that cells touching the image edge have been automatically removed, and that the coarse 3D reconstruction using the Marching Tetrahedra are fast and efficient estimates of the nuclei surfaces. Right images show a similar snapshot at the end of the segmentation.

Dispersion

Culture conditions being homogeneous, we can assume that the morphology of the imaged nuclei is relatively constant over each population. This allows to compute another quality measure of the model by evaluating the dispersion of our shape descriptors. This is done by computing their coefficient of variation (CV) for each population, i.e. the standard deviation-to-mean ratio. Results are shown in table 5.6. All coefficients are below 1, implying stable measures, nonetheless, some measures have a higher value than others. For instance, the *HullDiff* measure has a CV around 0.5 for both populations, therefore care should be taken in its interpretation in a shape comparison context. Same remark applies to the *Volume* measure in the HEK-293 case.

	LongAxis	Surface	Volume	Roughness	RadiusCV	HullDiff
HEK nuclei	0.145	0.280	0.425	0.108	0.281	0.544
Hela nuclei	0.135	0.196	0.278	0.128	0.264	0.455

Table 5.6: Coefficient of variation of each shape descriptor on each population.

Robustness

Finally, we determine the robustness of our criteria by computing two correlation measures: the classical correlation and the Hoeffding measure of dependence D [Hoeffding, 1948]. Results are given in Figures 5.9 (HEK cell line) and 5.10 (Hela cell line). Figures were obtained using the multiple correlation analysis tool of JMP software (SAS Institute, 1994). The strong correlation between the *Volume* and *Surface* measures, as well as with the *LongAxis* measure, coincides with the fact that these three measures are closely linked for any convex object. Another interesting observation is the relation between the *HullDiff* and the *Roughness* measures. This is due to the fact that a surface concavity at a given point creates a volume gap with the convex hull at that point. However, due to its local nature, the *Roughness* measure is not suited to detect large but smooth concavities such as for bean-shaped objects, for which *HullDiff* is much more efficient.

5.7   Application to the study of the *Leishmania* parasite

This work is currently performed in collaboration with the Systems Biology of Pathogens group at Institut Pasteur Korea.

Leishmaniasis is a neglected disease affecting over 10 million people in more than 80 countries. There are currently no vaccines against this disease, and available treatments frequently show considerable toxicity. We are currently focusing on one of the causes of this disease, namely the *leishmania* parasite.

An important aspect that can play a role in the emergence of new antigenic and adhesive variants of the *Leishmania* parasite is how genes regulating the adhesion properties of the parasite are organized within the nucleus. Little is known about how genes are organized in the nucleus of Leishmania parasites. We wish to investigate the spatial organization of *Leishmania major* telomeres in the promastigote stage. In order to perform colocalization of the telomeric clusters inside the parasite nucleus, we are working with 2-channel fluorescence images acquired on the Nikon and the Zeiss system (cf. figure 5.11).

Preliminary results are shown in figure 5.12. The telomeric cluster detection algorithms can be found in [Thomann et al., 2002] for the Nikon system and in [Dusch et al., 2007] for the Zeiss system. The interesting property of the leishmania parasite is that it contains two sub-cellular structures with DNA: the nucleus and the kinetoplast, which is an important part of the motility mechanism of the parasite. With DNA-specific marking, both structures appear on the images, allowing us to measure the position of the telomeric clusters relatively to the nuclear membrane while using the kinetoplast as a reference point to improve spatial distribution statistics.

Figure 5.7: Segmentation of a HEK-293 cell nuclei Z-stack (size $512 \times 512 \times 60$). Top: maximum intensity projection of the original stack. Middle: snapshot after initialization (coarse 3D reconstruction). Bottom: snapshot after segmentation. Note that cells lying on the image edge have been automatically removed since shape description has no meaning for partially visible objects.
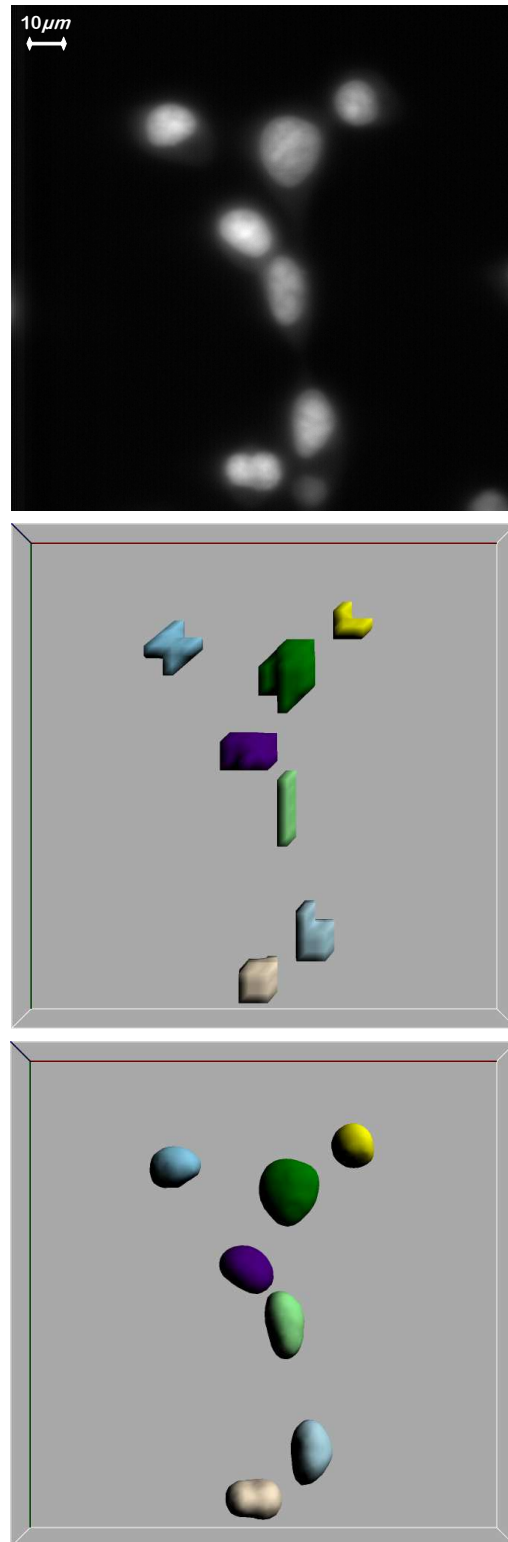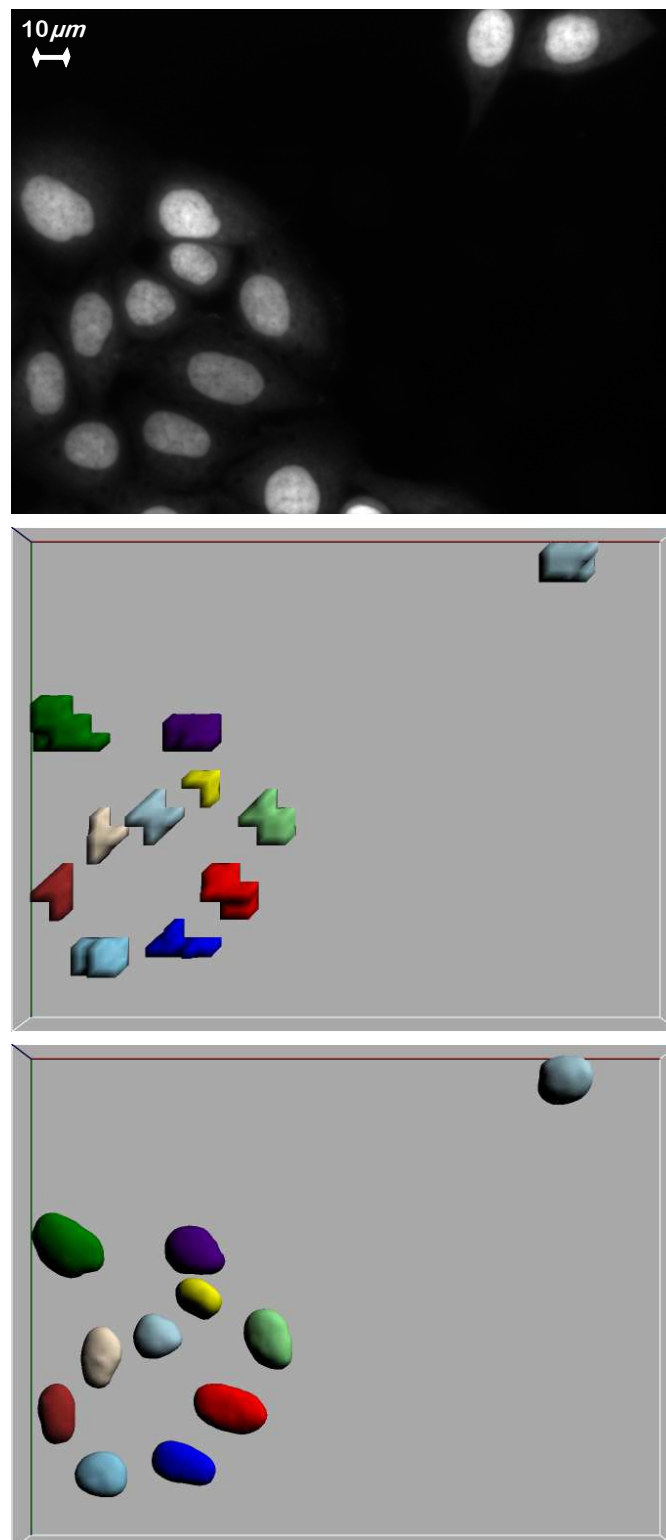
Figure 5.8: Segmentation of a Hela cell nuclei Z-stack (size $688 \times 520 \times 31$). Top: maximum intensity projection of the original stack. Middle: snapshot after initialization (coarse 3D reconstruction). Bottom: snapshot after segmentation.
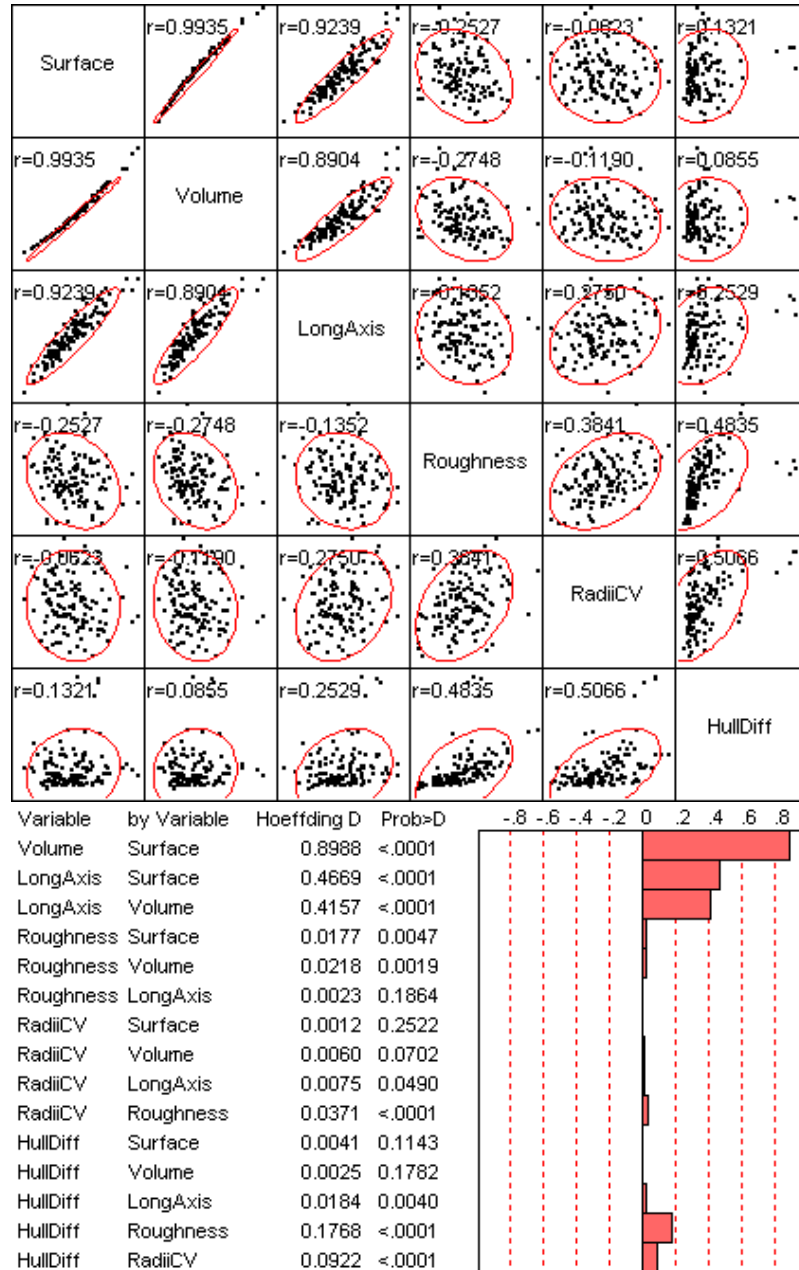
Figure 5.9: Shape statistics on the HEK-293 cell line. Correlation (top) and Hoeffding's D (bottom) measures are given for the criteria presented in section 5.4. D values range from -0.5 to 1, 1 indicating complete dependence. Red ellipses cover 90% of the population. Statistics obtained with the JMP software (SAS institute, 1994).
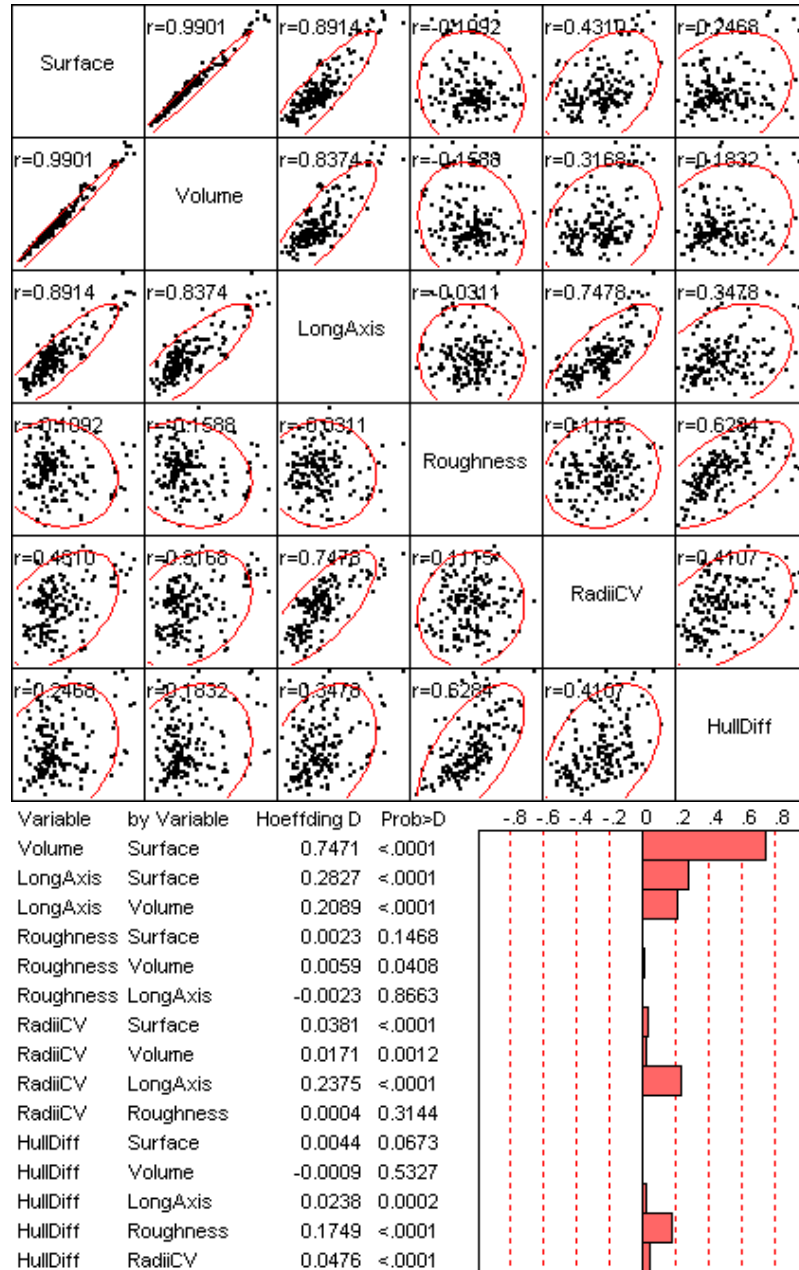
Figure 5.10: Shape statistics on the Hela cell line. Correlation (top) and Hoeffding's D (bottom) measures are given for the criteria presented in section 5.4. D values range from -0.5 and 1, 1 indicating complete dependence. Red ellipses cover 90% of the population. Statistics obtained with the JMP software (SAS institute, 1994).
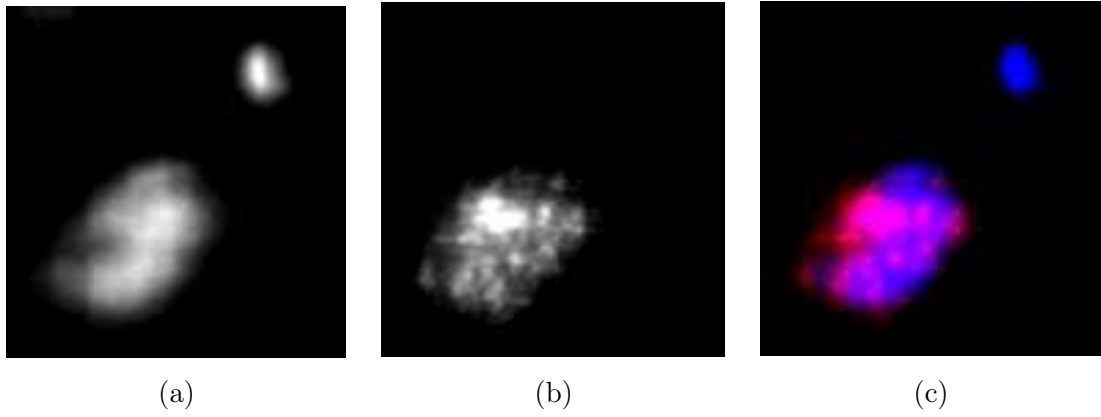
Figure 5.11: 2-channel image slice of a leishmania parasite acquired on the Nikon system. (a) DNA marking (nucleus + kinetoplast). (b) Telomeres. (c) merged channels. Note that since a widefield microscope is used, images are deconvoluted before the analysis.
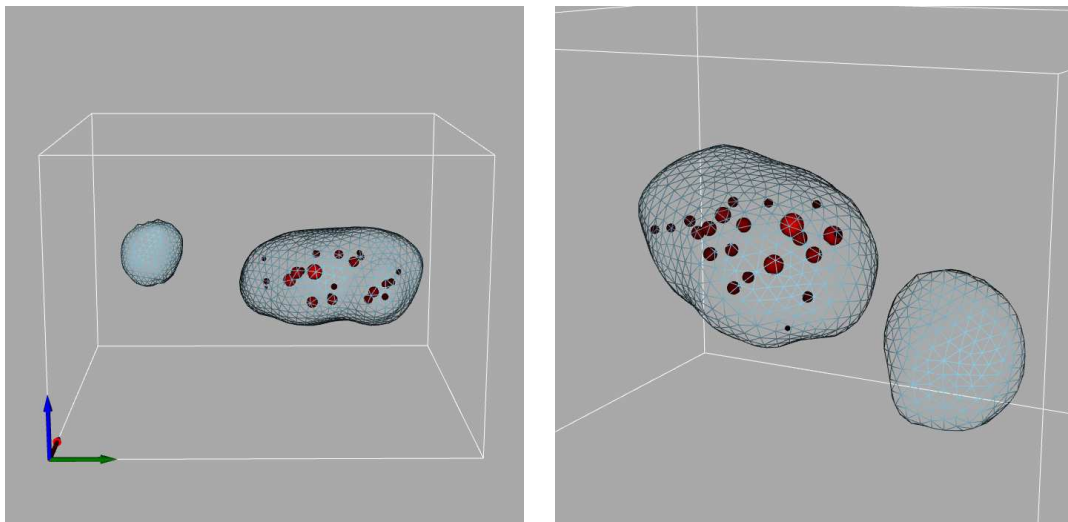
Figure 5.12: Segmentation of a leishmania parasite. Nucleus and kinetoplast are shown in blue, while red spots corresponds to detected telomere clusters.

6

Conclusion

This chapter concludes this thesis by first emphasizing on the main contributions of this work and how these contributions may help better understanding of images in biological applications. Then, further extensions and research directions are discussed, potentially opening the path to new challenges and future applications of images analysis in biological studies.

Contents

Throughout this work, we have focused upon the problems of multi-object segmentation, tracking and visualization in the context of 3D biological fluorescence microscopy images. Among the segmentation and tracking approaches available in the literature, we have concentrated our research efforts on energy-minimizing deformabe contour models. Indeed, they are believed to be part of the best techniques for such tasks thanks to their flexibility and high level interpretation of the entities. They commonly come in two different mathematical formulations, namely parametric and level set approaches. Parametric approaches are light in memory and perform fast, but usually rely on local information to deform the contour. Also, contour splitting and merging can be labor-intensive tasks if not well taken care of. Level set approaches are notably heavy in memory and slower, but can easily incorporate both local and global information to handle the deformation. They also allow contour splitting and merging implicitly.

The particularity of microscopy images is the atypical nature of the image formation process, compared to images coming from other acquisition systems (scanners, digital cameras, etc.). In the case of 3D fluorescence microscopy, the optical system induces a significant blur effect to the images, particularly pronounced along the depth axis. For this reason, traditional edge-finding techniques, relying on strong image gradients, usually fail to detect object boundaries correctly in 3D. We therefore had to turn to region-based approaches, among which the Mumford-Shah functional is a popular solution that fits well the formalism of deformable contours. Both models assign a single contour to each target object, and a coupling term is minimized, so as to avoid the fusion of two contours in case target objects touch one another through time, yielding impressive results. In this thesis, we have proposed an extension of both approaches to 3D fluorescence microscopy.

6.1 Contributions

Thanks to the multi-dimensional nature of level set methods, we have first proposed a level set-based extension to 3D images called *active surfaces*, as opposed to two-dimensional *active contours*. Since 3D implementation dramatically increases computational time, we proposed to speed-up the computations in different ways.

- Firstly, we perfomed an automated initialization of the model using a fast discrete version of the algorithm giving a good initial guess on the objects location. The initialization being very close from the desired solution, we could reduce the risk of the model being trapped in local energy minima and also reduce the number of iterations needed by the model to converge.

- Secondly, we implemented the narrow-band approach to update the level set structure only in a small region around the evolving contour. This choice is paricularly justified by our initialization scheme, and changes the algorithm complexity per iteration from the order of $O(M^3)$ to the order of $O(M^2)$.

While testing the method on simulated data, two limitations, unaddressed in the 2D models, came to light. Firstly, the coupling term avoiding contours to overlap did not relate to the image information, therefore the frontier location between touching contours was arbitrary, yielding erroneous results for objects moving at different speeds. Secondly, it appeared that for some applications the important consumption of fluorophore induced a decrease of the objects intensity through time (phenomenon known as photo-bleaching), causing the model to lose the objects when the intensity becomes arbitrarily low.

We proposed to solve these two limitations by introducing a volume conservation constraint, ensuring that contours maintain a constant volume during the evolution of the model. This reference volume is automatically computed from the segmentation results on the first sequence frame.

With the growing trend for cell-based assay development and screening applications, recent advances in automated imaging systems allowed researchers to start thinking about 3D High-Content Screening to enhance biological understanding of cellular and subcellular conformation. To this end, the model we proposed fitted the theoretical requirements, but memory and time consumption were still too important to handle the collossal amount of data coming from automated microscopes. Moreover, such applications require a visual feedback to allow the user either to adjust the model parameters or to store results for further off-line interpretation. Therefore, we started wondering whether our active surfaces could be optimized or even replaced by a faster alternative while keeping similar results.

After investigating the various parametric models available, our attention was quickly driven toward 3D deformable mesh models, which are discrete equivalents of parametric surface models. Meshes provided many appealing properties for our problematic, however, they had never been employed in a region-based segmentation and tracking context. The second part of this thesis thus focuses on the re-formulation of the active surfaces under a deformable mesh framework that we called *active meshes*, which can also be interpreted as an extension to 3D of the coupled parametric active contours of Zimmer and Olivo-Marin.

With constant focus on the goals and requirements of the biological context, we have developed a deformable mesh framework joining fast performance and real-time visual rendering of the meshes during their deformation. This framework would allow automatic surface re-parametrization and division, which is the major limiting factor of classical parametric models. During the deformation process, vertices are automatically added and removed to maintain homogeneous sampling of the surface. During this local resampling process, mesh splitting can be efficiently detected and performed whenever removing a vertex creates an incoherence in the manifold structure. Also, our model features a multi-resolution framework, where mehes can be globally refined at certain key steps of their deformation, so as to fit low-scale details in a fast and progressive coarse-to-fine manner.

Then, we focused on the integration of the Mumford-Shah functional to this framework, which raised significant difficulties. Indeed, minimizing the Mumford-Shah functional requires to update variables that express the mean image intensities inside and outside the meshes. Whereas this operation was straightfoward with level set models, mesh models have no immediate knowledge about region information. The meshes must first be transformed into a image-based representation (process called rasterization) in order to compute these variables. While this is a straightforward operation in 2D, basic 3D implementation is computationally expensive. Therefore, we proposed an optimized version of the parity test derived from the Jordan curve theorem, reducing the rasterization time from a few minutes to about a second.

We also expressed the coupling term preventing overlap by designing a fast collision detection algorithm specifically adapted to deformable meshes, performing simple and global intersection tests in order to reduce the area in which local collisions may actually occur.

Finally, the volume conservation constraint was applied by reducing the deformation amount of vertices in case the deformation induces an change in mesh volume away from

its reference value.

Our evaluation procedure started by a quantitative comparison of the proposed models on synthetic data, in terms of segmentation error, memory load and computational time cost. Segmentation results confirmed our assumption that both models would exhibit very similar results, since they rely on the same energy functional. However, active meshes turned out to be much more efficient in memory and time costs, with computation times from 5 to 10 times faster than active surfaces, and almost 20 times faster if one includes a 3D reconstruction step on top of the segmentation process.

In order to apply the Active Meshes method to real screening experiments, we had to integrate the method as a plug-in of the image processing software developed by the Image Mining group at Institut Pasteur Korea. This software provides tools to access image databases in a seemless manner, while providing a result database that can be employed to store the results for further use. A snapshot of the plugin is presented in figure 6.1.
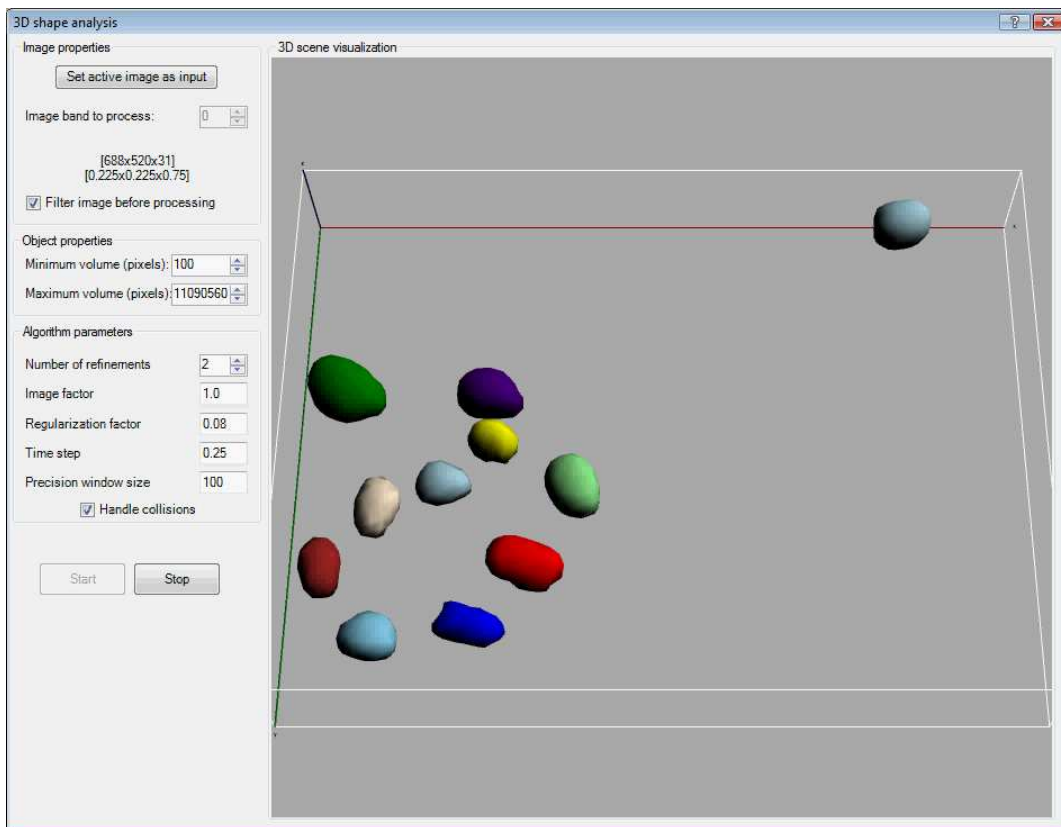

Figure 6.1: Snapshot of the Active Meshes plugin in the Image Mining software.

6.2 Future works

The main motivation of this work has been oriented toward practical applications through fully automated biological image analysis. Several points could be further investigated to push this automation a step further:

- Deformable models are known for being quite sensitive to the parameters, which are usually determined empirically. Those parameters are mostly dependent on the type of objects being observed and the image formation process. We believe that these parameters could be automatically estimated directly from the biological and imaging protocols. For instance, one could specify the type and concentration of cells and fluorochromes, as well as the optical characteristics and imaging setup of the microscope (this is already the case for some deconvolution softwares). This would remove the parameter estimation process, while allowing biologists to employ the software independently, without too much knowledge about the underlying algorithm.

- In order to perform correct segmentation, both models rely on the assumption that the objects do not touch one another in the image (or the initial frame of a sequence). Unfortunately, in many biological applications, cells are usually packed in clusters, therefore they can only be analyzed as a whole. While many cluster separation methods are available, the delicate task in biological applications is to automatically distinguish unique objects from clusters due to the potential size and shape variability of the observed objects. A solution can be found in a tracking context in case one or more objects in a cluster separate through time. Whenever splitting is detected, one can perform a backward tracking process to distinguish the objects in the previous frames.

The active mesh model we developed has been tested on trial screening experiments, for which we have measured the robustness of simple shape descriptors. With encouraging preliminary results, we are currently preparing a real screening campain aiming at studying morphology variability of cell nuclei under the influence of various molecules. With the large size of current chemical compound libraries, we expect that our small set of shape descriptors will not be sufficient to differentiate and classify the large array of nuclear phenotypes that will be observed. Therefore, one of the main directions of this work will be to study broader types of shape measures such as Zernike moments and Fourier descriptors. Hopefully, with sufficiently diverse descriptors, we will be able to detect specific phenotype variations, leading to the potential discovery of new drugs for a wide range of infectious diseases.

Bibliography

[Adalsteinsson and Sethian, 1995] Adalsteinsson, D. and Sethian, J. (1995). A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277.

[Alvarez et al., 1992] Alvarez, L., Lions, P.-L., and Morel, J.-M. (1992). Image selective smoothing and edge detection by nonlinear diffusion 2. *SIAM Journal of Numerical Analysis*, 29(3):845–866.

[Amos and White, 2003] Amos, W. and White, J. (2003). How the confical laser scanning microscope entered biological research. *Biology of the Cell*, 95:335–342.

[Aubert and Kornprobst, 2001] Aubert, G. and Kornprobst, P. (2001). *Mathematical problems in image processing*. Springer.

[Aubert and Vese, 1997] Aubert, G. and Vese, L. (1997). A variational method in image recovery. *SIAM Journal of Numerical Analysis*, 34(5):1948–1979.

[Ayache et al., 1989] Ayache, N., Boissonnat, J., Brunet, E., Cohen, L., Chieze, J., Geiger, B., Monga, O., Rocchisani, J., and Sander, P. (1989). Building highly structured volume representations in 3D medical images. In *Proceedings of the 3rd International Symposium on Computer Aided Radiology*.

[Barles, 1985] Barles, G. (1985). Remarks on a flame propagation model. Technical Report 464, INRIA.

[Benassarou et al., 2002] Benassarou, A., Freitas-Caires, J., and Bittar, E. (2002). An integrated framework to analyze and visualize the evolution of multiple topology changing objects in 4D image datasets. In Greiner, G., Niemann, H., Ertl, T., Girod, B., and Seidel, H.-P., editors, *Proceedings of the 7th workshop on vision, modeling and visualization*, pages 147–154. Erlangen.

[Beucher, 1990] Beucher, S. (1990). *Segmentation d'images et morphologie mathematique*. PhD thesis, Ecole des Mines, Paris.

[Bittar et al., 2003] Bittar, E., Benassarou, A., Lucas, L., Elias, E., Tchelidze, P., Ploton, D., and O'Donohue, M. (2003). How to see inside living cells in 4D. In Hamza, M., editor, *Proceedings of the third IASTED conference Visualization, Imaging and Image Processing*, pages 238–245, Benalmadena, Spain. ACTA Press.

[Borgefors, 1986] Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371.

[Bornik et al., 2005] Bornik, A., Reitinger, B., and Beichel, R. (2005). Reconstruction and representation of tubular structures using simplex meshes. In *Bilderverarbeitung fur die Medizin*, Informatik aktuell. Springer.

[Bossart et al., 1997] Bossart, P.-L., David, D., Dinten, J.-M., and Chassery, J.-M. (1997). Detection of regular boundaries in noisy and textured images: a multiscale active contour approach. *Traitement du signal*, 17:209–225.

[Boulanger et al., 2006] Boulanger, J., Kervrann, C., and Bouthemy, P. (2006). Estimation of dynamic background for fluorescence video-microscopy. In *Proceeding of IEEE International Conference on Image Processing*, Atlanta, USA.

[Burr, 1981] Burr, D. (1981). Elastic matching of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):708–713.

[Canny, 1986] Canny, J. (1986). A computational appraoch to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714.

[Caselles et al., 1993] Caselles, V., Catte, F., Coll, B., and Dibos, F. (1993). A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31.

[Caselles et al., 1995] Caselles, V., Kimmel, R., and Sapiro, G. (1995). Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision*, pages 694–699, Cambridge, USA.

[Caselles et al., 1997] Caselles, V., Kimmel, R., Sapiro, G., and Sbert, C. (1997). Minimal surfaces based object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):394–398.

[Catte et al., 1992] Catte, F., Lions, P.-L., Morel, J.-M., and Coll, T. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal of Numerical Analysis*, 29(1):182–193.

[Chakraborty et al., 1996] Chakraborty, A., Staib, L., and Duncan, J. (1996). Deformable boundary finding in medical images by intergrating gradient and region information. *IEEE Transactions on Medical Imaging*, 15(6):859–870.

[Chan and Vese, 1999] Chan, T. and Vese, L. (1999). An active contour model without edges. In *Scale Spaces theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*. Springer.

[Chan and Vese, 2001a] Chan, T. and Vese, L. (2001a). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.

[Chan and Vese, 2001b] Chan, T. and Vese, L. (2001b). A level set algorithm for minimizing the Mumford-Shah functional in image processing. In *Proceedings of the IEEE workshop on Variational and Level Set Methods in computer vision*, pages 161–168, Los Alamitos, USA. IEEE Computer Society.

[Chang et al., 1994] Chang, Y., Hou, T., Merriman, B., and Osher, S. (1994). A level set formulation of eulerian interface capturing methods for incompressible fluid flows. Technical report, Computational and Applied Mathematics University of California Los Angeles.

[Ciofolo, 2005] Ciofolo, C. (2005). *Segmentation des formes guidee par des modeles en neuro-imagerie*. PhD thesis, Universite Rennes I.

[Cipolla and Blake, 1990] Cipolla, R. and Blake, A. (1990). The dynamic analysis of apparent contours. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 616–623, Osaka, Japan.

[Clarysse et al., 1997] Clarysse, P., Friboulet, D., and Magnin, I. (1997). Tracking geometrical descriptors on 3-D deformable surfaces: application to the left-ventricular surface of the heart. *IEEE Transactions on Medical Imaging*, 16(4):392–404.

[Claxton et al.,] Claxton, N., Fellers, T., and Davidson, M. Laser scanning confocal microscopy. Online review article.

[Cohen, 1991] Cohen, L. (1991). On active contours and balloons. In *CVGIP : Image Understanding*.

[Cohen and Cohen, 1993] Cohen, L. and Cohen, I. (1993). Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147.

[Collins et al., 1998] Collins, D., Zijdenbos, A., Kollokian, V., Sled, J., Kabani, N., Holmes, C., and Evans, A. (1998). Design and consutrcution of a realistic digital brain phantom. *IEEE Transactions on Medical Imaging*, 17(3):463–468.

[Cotin et al., 1999] Cotin, S., Delingette, H., and Ayache, N. (1999). Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5:62–73.

[Coxeter, 1963] Coxeter, H. (1963). *Regular polytopes*. The Macmillan Co.

[Cremers et al., 2002] Cremers, D., Tischhauser, F., Weickert, J., and Schnorr, C. (2002). Diffusion snakes: introducing statistical shape knowledge into the Mumford-Shah functional. *International Journal on Computer Vision*, 50:295–313.

[Cuisenaire, 1999] Cuisenaire, O. (1999). *Distance tranformations: fast algorithms and applications to medical image processing*. PhD thesis, Universite catholique de Louvain.

[de Boor, 2001] de Boor, C. (2001). *A practical guide to splines*. Applied Mathematics Science. Springer-Verlag, revised edition.

[Delingette, 1994] Delingette, H. (1994). *Modelisation, deformation et reconnaissance d'objets tridimensionnels a l'aide de maillages simplexes*. PhD thesis, Ecole Centrale de Paris.

[Delingette, 1999] Delingette, H. (1999). General object reconstruction based on simplex meshes. *International Journal on Computer Vision*, 32:111–146.

[Delingette and Montagnat, 2000] Delingette, H. and Montagnat, J. (2000). Topology and shape constraints on parametric active contours. Technical Report 3880, INRIA.

[Dervieux and Thomasset, 1981] Dervieux, A. and Thomasset, F. (1981). Sur l'approximation d'ecoulements multifluides incompressibles visqueux par des elements finis triangulaires de degre un. Technical Report 67, Institut National de Recherche en Informatique et en Automatique.

[Dinten and Trouve, 1992] Dinten, J.-M. and Trouve, A. (1992). A deformable model approach for the determination of transition strips on radiographic images. In *Proceedings of the 11th International Conference on Pattern Recognition*, pages 355–358.

[Dorval et al., 2003] Dorval, T., Laurent, C., and Laurent, N. (2003). A friendly object selection tool combining salient features detection and potential-based active contour. In *Proceedings of Content Based Multimedia Indexing*.

[Du et al., 2006] Du, X., Cho, D., and Bui, T. (2006). Image inpainting and segmentation using hierarchical level set method. In *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision*.

[Duan and Qin, 2001] Duan, Y. and Qin, H. (2001). Extracting boundary surface of arbitrary topology from volumetric datasets. In *Volume Graphics*, pages 237–248. Springer.

[Duan and Qin, 2004] Duan, Y. and Qin, H. (2004). A subdivision-based deformable model for surface reconstruction of unknown topology. *Graphical Models*, 66:181–202.

[Dubrovin et al., 1991] Dubrovin, B., Fomenko, A., Novikov, S., and Burns, R. (1991). *Modern geometry - methods and applications*, volume 1: The Geometry of Surfaces, Transformation Groups, and Fields of *Graduate Texts in Mathematics*. Springer.

[Dufour et al., 2007a] Dufour, A., Lee, J., Vincent, N., Grailhe, R., and Genovesio, A. (2007a). 3D automated nuclear morphometric analysis using Active Meshes. In Rajapakse, J., Schmidt, B., and Volkert, G., editors, *Pattern Recognition in Bioinformatics*, volume 4774 of *Lecture Notes in Bioinformatics*, pages 356–367. Springer.

[Dufour et al., 2005] Dufour, A., Shinin, V., Tajbaksh, S., Guillen, N., Olivo-Marin, J., and Zimmer, C. (2005). Segmenting and tracking fluorescent cells in dynamic 3d microscopy with coupled active surfaces. *IEEE Transactions on Image Processing*, 14(9):1396–1410.

[Dufour et al., 2006] Dufour, A., Vincent, N., and Genovesio, A. (2006). 3D Mumford-Shah based active mesh. In Martinez-Trinidad, J., Ochoa, J., and Kittler, J., editors, *Progress in Pattern Recognition, Image Analysis and Applications*, volume 4225 of *Lecture Notes in Computer Science*, pages 208–217. Springer.

[Dufour et al., 2007b] Dufour, A., Vincent, N., and Genovesio, A. (2007b). 3D multi-object segmentation, tracking and visualization in fluorescence microscopy using Active Meshes. In *2nd International Workshop on Pattern Recognition in Bioinformatics*, Singapore. Short paper.

[Dusch et al., 2007] Dusch, E., Vincent, N., and Genovesio, A. (2007). 3D fluorescent spot detection in line-scanning confocal microscopy. In *IEEE International Conference on Image Processing*, San Antonio, Texas, USA.

[Dyn et al., 1990] Dyn, N., Levin, D., and Gregory, J. (1990). A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9:160–169.

[Egger and Petran, 1967] Egger, M. and Petran, M. (1967). New reflected-light microscope for viewing unstained brain and ganglion cells. *Science*, 157:305–307.

[Farin, 1992] Farin, G. (1992). *Courbes et surface pour la CGAO*. Masson.

[Froment, 1999] Froment, J. (1999). A compact and multiscale image model based on level sets. In *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 152–163.

[Fuch et al., 1977] Fuch, H., Kedem, Z., and Uselton, S. (1977). Optimal surface reconstruction using planar contours. *Communications of the ACM*, 20(10):693–702.

[Geman et al., 1990] Geman, D., Geman, S., Graffigne, C., and Dong, P. (1990). Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:609–628.

[Giachetti and Zanetti, 2004] Giachetti, A. and Zanetti, G. (2004). Reconstruction of complex vascular structures from CT data. In *Proceedings of the 14th International Conference on Mechanics in Medicine and Biology*, Bologna, Italy.

[Giraldi et al., 2000] Giraldi, G., Goncalves, L., and Oliveira, A. (2000). Dual Topologically Adaptable Snakes. In *(JCIS) Joint Conference on Information Sciences*, pages 103–106.

[Glasbey, 1993] Glasbey, C. (1993). An analysis of histogram-based thresholding algorithms. *Computer Vision, Graphics and Image Processing*, 55:532–537.

[Goldenberg et al., 1999] Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M. (1999). Fast geodesic active contours. In *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 34–45. Springer.

[Goldenberg et al., 2002] Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M. (2002). Cortex segmentation: a fast variational geometric approach. *IEEE Transactions on Medical Imaging*, 21(2):1544–1551.

[Gomes and Faugeras, 1999] Gomes, J. and Faugeras, O. (1999). Reconciling distance functions and level sets. Technical Report 3666, INRIA.

[Gueziec and Hummel, 1995] Gueziec, A. and Hummel, R. (1995). Exploiting triangulated surface extraction using tetrahedral decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):328–342.

[Haralick and Shapiro, 1985] Haralick, R. and Shapiro, L. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132.

[Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detection. In *The Fourth Alley Vision Conference*, pages 147–151.

[Hausdorff, 1978] Hausdorff, F. (1978). *Set theory*. Chelsea Publishing Company.

[Hoeffding, 1948] Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325.

[Huebner et al., 2001] Huebner, K., Dewhirst, D., Smith, D., and Byrom, T. (2001). *The finite element method*. J. Wiley & Sons, New York. 4th edition.

[Kass et al., 1988] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes : active contour models. *International Journal of Computer Vision*, 1:321–331.

[Kichenassamy et al., 1995] Kichenassamy, S., Kumar, A., Olver, P., and Yezzi, A. (1995). Gradient flows and geometric active contour models. In *Proceedings of the International Conference on Computer Vision*, pages 810–815.

[Kimmel et al., 1996] Kimmel, R., Kiryati, N., and Bruckstein, A. (1996). Sub-pixel distance maps and weighted distance transforms. *Mathematical Imaging and Vision*, 1:223–233.

[Kobbelt et al., 2000] Kobbelt, L., Bareuther, T., and Seidel, H. (2000). Multiresolution shape deformations for meshes with dynamic vertex connectivity. In *Proceedings of Eurographics*, volume 19, page C249.

[Kumar et al., 1996] Kumar, A., Tannenbaum, A., and Balas, G. (1996). Optical flow: a curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):598–610.

[Lachaud, 1998] Lachaud, J. (1998). *Extraction de surfaces à partir d'images tridimensionnelles : approche discrète et approche par modèle déformable*. PhD thesis, Universite Joseph-Fourier, Grenoble I.

[Lachaud and Montanvert, 1995] Lachaud, J. and Montanvert, A. (1995). Volumic segmentation using hierarchical representation and triangulated surface. Technical report, Institut Albert Bonniot, UMR 5525, laboratoire TMIC-IMAG, equipe INFODIS,.

[Lachaud and Montanvert, 1999] Lachaud, J. and Montanvert, A. (1999). Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. *Medical Image Analysis*, 3(2):187–207.

[Lachaud and Bainville, 1994] Lachaud, J.-O. and Bainville, E. (1994). A discrete adaptive model following topological modifications of volumes. In *Proceedings of the 4th Conference on Discrete Geometry for Computer Imagery*, pages 183–194.

[Leman and Getzenberg, 2002] Leman, E. and Getzenberg, R. (2002). Nuclear matrix protein as biomarkers in prostate cancer. *Journal of Cell Biochemistry*, 86(2):213–223.

[Lin and Manocha, 2004] Lin, M. and Manocha, D. (2004). Collision detection and proximity queries. In Goodman, J. and O'Rourke, J., editors, *Handbook of Discrete and Computation Geometry*. Chapman & Hall/CRC. Second edition.

[Lin et al., 1989] Lin, W., Chen, S., and Chen, C. (1989). A new surface interpolation technique for reconstructing 3D objects from serial corss-sections. *Computer Vision, Graphics, and Image Processing*, 48(1):124–143.

[Loop, 1987] Loop, C. (1987). *Smooth subdivion surfaces based on triangles*. PhD thesis, University of Utah.

[Lorensen and Cline, 1987] Lorensen, W. and Cline, H. (1987). Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169.

[Malladi et al., 1995] Malladi, R., Sethian, J. A., and Vemuri, B. C. (1995). Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175.

[McInerney and Terzopoulos, 1995a] McInerney, T. and Terzopoulos, D. (1995a). A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83.

[McInerney and Terzopoulos, 1995b] McInerney, T. and Terzopoulos, D. (1995b). Medical image segmentation using topologically adaptable snakes. In Ayache, N., editor, *(CVRMed) Computer Vision, Virtual Reality and Robotics in Medicine, Nice, France*, volume 905 of *Lecture Notes in Computer Science*, pages 92–101. Springer-Verlag.

[McInerney and Terzopoulos, 1997] McInerney, T. and Terzopoulos, D. (1997). Medical image segmentation using topologically adaptable surfaces. In Troccaz, J., Grimson, W. E. L., and Moesges, R., editors, *(CVRMed-MRCAS) Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine - Medial Robotics and Computer-Assisted Surgery*, volume 1205 of *Lecture Notes in Computer Science*, pages 23–32, Grenoble, France. Springer-Verlag.

[McInerney and Terzopoulos, 1999] McInerney, T. and Terzopoulos, D. (1999). Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Transactions on Medical Imaging*, 18(10):840–850.

[McInerney and Terzopoulos, 2000] McInerney, T. and Terzopoulos, D. (2000). T-snakes: topology adaptive snakes. *Medical Image Analysis*, 4:73–91.

[Miller, 1990] Miller, J. (1990). On GDM's: geometrically deformed models for the extraction of closed shapes from form volume data. Master's thesis, Rensselaer Polytechnic Institute, Troy, New York.

[Miller et al., 1991] Miller, J., Breen, D., Lorensen, W., O'Bara, R., and Wozny, M. (1991). Geometrically deformed models: a method for extracting closed geometric models form volume data. In *Proceedings of SIGGRAPH*.

[Montagnat and Delingette, 1997] Montagnat, J. and Delingette, H. (1997). Volumetric medical images segmentation using shape constrained deformable models. In Troccaz, J., Grimson, W. E. L., and Moesges, R., editors, *(CVRMed-MRCAS) Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine - Medial Robotics and Computer-Assisted Surgery*, volume 1205 of *Lecture Notes in Computer Science*, pages 13–22, Grenoble, France. Springer-Verlag.

[Montagnat et al., 2003] Montagnat, J., Sermesant, M., Delingette, H., Malandain, G., and Ayache, N. (2003). Anisotropic filtering for model-based segmentation of 4D cylindrical echocardiographic images. *Pattern Recognition Letters*, 24:815–828.

[Mumford and Shah, 1989] Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure App. Math.*, 42:577–684.

[Nitzberg and Shiota, 1992] Nitzberg, M. and Shiota, T. (1992). Nonlinear image filtering with edge and corner enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):826–833.

[Ogier, 2005] Ogier, A. (2005). *Méthodes de restauration en imagerie médicale tridimensionnelle.* PhD thesis, Universite Rennes I.

[Osher and Fedkiw, 2003] Osher, S. and Fedkiw, R. (2003). *Level set methods and dynamic implicit surfaces.* Springer.

[Osher and Paragios, 2003] Osher, S. and Paragios, N. (2003). *Geometric level set methods in imaging, vision and graphics.* Springer.

[Osher and Sethian, 1988] Osher, S. and Sethian, J. (1988). Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49.

[Paragios, 2002] Paragios, N. (2002). A variational approach for the segmentation of the left ventricle in caridiac image analysis. *International Journal on Computer Vision*, 50(3):345–362.

[Paragios, 2003] Paragios, N. (2003). A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE Transactions on Medical Imaging*, 22:773–776.

[Paragios and Deriche, 1998] Paragios, N. and Deriche, R. (1998). Geodesic active regions for texture segmentation. Technical Report 3440, INRIA.

[Paragios and Deriche, 1999] Paragios, N. and Deriche, R. (1999). Geodesic active regions for tracking. In *Proceedings of the Internation Conference on Computer Vision*, pages 688–694, Corfu, Greece. IEEE Computer Society.

[Paragios and Deriche, 2000] Paragios, N. and Deriche, R. (2000). Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280.

[Paragios and Deriche, 2002] Paragios, N. and Deriche, R. (2002). Geodesic active regions: a new framework to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13:249–268.

[Paragios et al., 2001] Paragios, N., Mellina-Gottardo, O., and Ramesh, V. (2001). Gradient vector flow fast geodesic active contours. In *Proceedings of the International Conference on Computer Vision*, pages 67–75.

[Paragios et al., 2004] Paragios, N., Mellina-Gottardo, O., and Ramesh, V. (2004). Gradient vector flow fast geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):402–407.

[Park et al., 2001] Park, J., McInerney, T., Terzopoulos, D., and Kim, M. (2001). A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images. *Computers & Graphics*, 25:421–440.

[Payne and Toga, 1990] Payne, B. and Toga, A. (1990). Surface mapping brain function on 3D models. *IEEE Computer Graphics and Applications*, 10(3):33–41.

[Peng et al., 1999] Peng, D., Merriman, B., Osher, S., Zhao, H., and Kang, M. (1999). A pde-based fast local level set method. *Journal of Computational Physics*, 155:410–438.

[Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

[Poggio et al., 1985] Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317(26):314–319.

[Ray and Acton, 2005] Ray, N. and Acton, S. (2005). Data acceptance for automated leukocyte tracking through segmentation of spatiotemporal images. *IEEE Transactions on Biomedical Engineering*, 52(10):1702–1712.

[Ray et al., 2002] Ray, N., Acton, S., and Ley, K. (2002). Tracking leukocytes in vivo with shape and size constrained active contours. *IEEE Transactions on Medical Imaging*, 21(10):1222–1235.

[Reeves et al., 1988] Reeves, A., Prokop, R., Andrews, S., and Kuhl, F. (1988). Three-dimensional shape analysis using moments and Fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):937–943.

[Rosenfeld and Thurston, 1971] Rosenfeld, A. and Thurston, M. (1971). Edge and curve detection for visual scene analysis. *IEEE Transactions on Computer*, 20(5):562–569.

[Rousselle, 2003] Rousselle, J.-J. (2003). *Les contours actifs, une methode de segmentation : application a l'imagerie medicale*. PhD thesis, Universite Francois Rabelais, Tours.

[Rousson et al., 2004] Rousson, M., Paragios, N., and Deriche, R. (2004). Implicit active shape models for 3d segmentation in MR imaging. In *Proceedings of the 7th International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 3216 of *Lecture Notes in Computer Science*, Saint-Malo, France.

[Saaty and Kainen, 1986] Saaty, T. and Kainen, P. (1986). *The four-color problem: assaults and conquest*. Dover Publications.

[Samson et al., 1999] Samson, C., Balnc-Feraud, L., Aubert, G., and Zerubia, J. (1999). A level set model for image classification. In *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 306–317. Springer.

[Samson et al., 2000] Samson, C., Blanc-Feraud, L., Aubert, G., and Zerubia, J. (2000). A variational model for image classification and restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:460–472.

[Sarti et al., 2000] Sarti, A., de Solorzano, C. O., Lockett, S., and Malladi, R. (2000). A geometric model for 3D confocal image analysis. *IEEE Transactions on Biomedical Engineering*, 47(12):1600–1609.

[Sethian, 1985] Sethian, J. (1985). Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499.

[Sethian, 1999] Sethian, J. A. (1999). *Level set methods and fast marching methods*. Cambridge University Press, 2nd edition edition.

[Sheppard and Mao, 1988] Sheppard, C. J. R. and Mao, X. Q. (1988). Confocal microscopes with slit apertures. *Journal of Modern Optics*, 35(7):1169–1185.

[Song, 2003] Song, B. (2003). *Topics in variational PDE image segmentation, inpainting and denoising*. PhD thesis, Department of Mathematics University of California, Los Angeles.

[Staib and Duncan, 1992] Staib, L. and Duncan, J. (1992). Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075.

[Svensson and Borgefors, 2002] Svensson, S. and Borgefors, G. (2002). Digital distance transforms in 3D images using information from neighbourhoods up to $5{\times}5{\times}5$. *Computer Vision and Image Understanding*, 88:24–53.

[Szeliski and Terzopoulos, 1991] Szeliski, R. and Terzopoulos, D. (1991). Physically based and probabilistic models for computer vision. In *Proceedings of the SPIE Conference on Geometric Methods in Computer Vision*, volume 1570, pages 140–152.

[Tang and Acton, 2004] Tang, J. and Acton, S. (2004). Vessel boundary tracking for intravital microscopy via multiscale gradient vector flow snakes. *IEEE Transactions on Biomedical Engineering*, 51(2):316–324.

[Tang et al., 2006] Tang, J., Millington, S., Acton, S., Crandall, J., and Hurwitz, S. (2006). Surface extraction and thickness measurement of the articular cartilage from MR images using directional gradient vector flow snakes. *IEEE Transactions on Biomedical Engineering*, 53(5):896–907.

[Terzopoulos, 1986] Terzopoulos, D. (1986). Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424.

[Terzopoulos, 1987] Terzopoulos, D. (1987). On matching deformable models to images. In *Topical meeting on machine vision*, volume 12 of *Technical Digest Series*, pages 160–163. Optical Society of America.

[Terzopoulos, 2003] Terzopoulos, D. (2003). *Geometric level set methods in imaging, vision, and graphics*, chapter Deformable Models: Classic, Topology-Adaptive and Generalized Formulations, pages 21–40. Springer.

[Terzopoulos and Metaxas, 1991] Terzopoulos, D. and Metaxas, D. (1991). Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714.

[Terzopoulos and Szeliski, 1992] Terzopoulos, D. and Szeliski, R. (1992). Tracking with Kalman snakes. In Blake, A. and Yuille, A., editors, *Active Vision*, pages 3–20, Cambridge. MIT Press.

[Thomann et al., 2002] Thomann, D., Rines, D., Sorger, P., and Danuser, G. (2002). Automatic fluorescent tag detection in 3D with super-resolution: application to the analysis of chromosome movement. *Journal of Microscopy*, 208(1):49–64.

[Tsai and Osher, 2003] Tsai, R. and Osher, S. (2003). Level Set Methods and their Applications in Image Science. Technical report, Department of Mathematics Princeton University.

[van Kempen, 1999] van Kempen, G. M. (1999). *Image Restoration in Fluorescence Microscopy*. PhD thesis, Advanced School for Computing and Imaging.

[Vonesch et al., 2006] Vonesch, C., Aguet, F., Vonesch, J.-L., and Unser, M. (2006). The colored revolution of bioimaging. *IEEE Signal Processing Magazine*, 23(3):20–31.

[Voort and Strasters, 1995] Voort, H. V. D. and Strasters, K. (1995). Restoration of confocal images for quantitative image analysis. *Journal of Microscopy*, 178(2):165–181.

[Weickert, 1996] Weickert, J. (1996). *Anisotropic diffusion in image processing*. PhD thesis, Keiserlautern University.

[Weickert, 1997] Weickert, J. (1997). Recursive separable schemes for nonlinear diffusion filters. In *Scale-Space Theories in Computer Vision*, volume 1252 of *Lecture Notes in Computer Science*, pages 260–271. Springer.

[Weickert et al., 1998] Weickert, J., Romeny, B., and Viergever, M. (1998). Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410.

[Whitaker and Chen, 1994] Whitaker, R. and Chen, D. (1994). Volumetric deformable models: active blobs. In Robb, R., editor, *Proceedings of the 1st Conference on Visualization in Biomedical Computing*, pages 122–134.

[Whitaker and Xue, 2001] Whitaker, R. and Xue, X. (2001). Variable-conductance, level-set curvature for image denoising. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 142–145, Thessaloniki, Greece. IEEE.

[Xu et al., 1999] Xu, C., Pham, D., Rettman, M., Yu, D., and Prince, J. (1999). Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18(6):467–480.

[Xu and Prince, 1997] Xu, C. and Prince, J. (1997). Gradient vector flow: a new external force for snakes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

[Xu and Prince, 1998] Xu, C. and Prince, J. (1998). Generalized gradient vector flow external forces for active contours. *Signal Processing*, 71(2):131–139. not printed.

[Xu and Prince, 2000] Xu, C. and Prince, J. (2000). Gradient vector flow deformable models. In Bankman, I., editor, *Handbook of Medical Imaging*. Academic Press. not printed.

[Yezzi et al., 1997] Yezzi, A., Kichenassamy, S., Kumar, A., Olver, P., and Tannenbaum, A. (1997). A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209.

[Yuille et al., 1989] Yuille, A., Hallinan, P., and Cohen, D. (1989). Feature extraction from faces using deformable templates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 104–109.

[Zagrodsky et al., 2005] Zagrodsky, V., Walimbe, V., Castro-Pareja, C., Qin, J., Song, J.-M., and Shekhar, R. (2005). Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models. *IEEE Transactions on Medical Imaging*, 24(9):1089–1099.

[Zhang et al., 2007] Zhang, B., Zerubia, J., and Olivo-Marin, J.-C. (2007). Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics*, 46:1819–1829.

[Zhang et al., 2004] Zhang, B., Zimmer, C., and Olivo-Marin, J.-C. (2004). Tracking fluorescent cells with coupled geometric active contours. In *International Symposium on Biomedical Imaging*, pages 476–479, Arlington.

[Zhang and Chen, 2001] Zhang, C. and Chen, T. (2001). Efficient feature extraction for 2D/3D objects in mesh representation. In *International Conference on Image Processing*, pages 935–938, Thessaloniki.

[Zhang, 1996] Zhang, Y. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346.

[Zhang, 2001] Zhang, Y. (2001). A review of recent evaluation methods for image segmentation. In *Proceedings of the 6th International Symposium on Signal Processing and its Applications*.

[Zhang, 2006] Zhang, Y. (2006). *Advances in image and video segmentation*, chapter XX: A summary of recent progresses for segmentation evaluation, pages 422–439. IRM Press.

[Zhao et al., 1996] Zhao, H., Chan, T., Merriman, B., and Osher, S. (1996). A xariational level set approach to multiphase motion. *Journal of Computational Physics*, 127(0167):179–195.

[Zhu and Yuille, 1996] Zhu, S. and Yuille, A. (1996). Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900.

[Zhukov et al., 2002] Zhukov, L., Bao, Z., Guskov, I., Wood, J., and Breen, D. (2002). Dynamic deformable models for 3D MRI heart segmentation. In *SPIE Medical Imaging*, volume 4684, pages 1398–1405.

[Zimmer and Olivo-Marin, 2005] Zimmer, C. and Olivo-Marin, J.-C. (2005). Coupled Parametric Active Contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1838–1842.

[Zimmer et al., 2006] Zimmer, C., Zhang, B., Dufour, A., Thebaud, A., Berlemont, S., Meas-Yedid, V., and Olivo-Marin, J.-C. (2006). On the digital trail of mobile cells. *Signal Processing Magazine*, 23(3):54–62.