Strings and necklaces: on learning and browsing medical image segmentations
Ghebreab, S.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)

Download date: 27 Dec 2018
Chapter 2

Review of Methods for the Interactive Segmentation of Medical Images

In recent years considerable attention has been devoted to develop automatic methods for image segmentation, which is a prerequisite for many multidimensional image analysis tasks. Despite this, fully automatic segmentation often fails, producing incorrect results. Most current computer-based segmentation techniques therefore still need significant user input to specify a region of interest, to initialize or control the segmentation process, or to perform subsequent adjustment of results. In this chapter, we depart from the view that, as user intervention is inevitable, interaction has to be approached in a structured manner and integral to computation. Following [89] we review interactive segmentation in literature with the goal of determining promising approaches to interactive segmentation. We first characterize, review and examine human-computer interaction in terms of user input, interpretation of user input and purpose thereof. We then focus on deformable models, as they turn out to be a promising platform for interactive segmentation. We characterize, review and examine some well-known methods in terms of their boundary model, objective function and optimization. We conclude this chapter with research questions that need to be addressed to develop generic interactive segmentation methods that optimize user input.
2.1 Introduction

Accurate image segmentation is one of the key problems in computer vision. The segmentation problem is to divide an image into parts that have a one to one correspondence with objects in the real world. The segmentation problem arises wherever there is a desire for high-level reasoning about the image. In content-based image retrieval, for example, images containing objects may be fetched on the basis of a boundary sketch provided by the user. In order to accomplish this task, objects in images need to be segmented and results matched with the sketch. In computer-aided diagnosis, segmentation is required for such tasks as visualization, registration and measurement of anatomical structures. To measure the deformation of the spine, for example, vertebral structures in CT or MRI images need to be outlined. Shape features are then computed from the resulting delineations.

A large number of image segmentation methods is found in literature. Where the purpose of all methods is identification of objects, different approaches are taken to solve the problem. Image segmentation methods can roughly be classified into methods based on boundary detection and methods based on similarity. The concept of segmenting an image by boundary detection is based on abrupt changes in gray-level values in the image, due to the fact that the two sides of the boundary have a different physical basis in the real world. Areas of interest within this category are detection of isolated points and detection of lines and edges in an image. The alternative approach is based on region growing. This concept of segmenting an image is based on similarity of the gray-level values in the image under the assumption that the gray values of an entity in the real world must have some property in common. Region growing techniques can be used to extract a single region from the image or to solve the full segmentation problem. Both approaches have received much attention to improve upon their shortcomings.

The large number of segmentation methods to date reflects the complexity of the image segmentation problem. Especially in the field of medical imaging, a wide variety of segmentation methods have been proposed to solve the segmentation problem. This is mainly due to the fact that objects in medical images are complex in shape and image appearance, and vary considerably across subjects. Complexity in appearance also refers to the fact that distinct objects may have very similar gray-level appearance in an image due to characteristic of the image modality, while the same objects can have different gray-level appearance in different regions of the image due to distortions caused by the imaging equipment. These properties of digital medical images have been a roadblock for the emergence of entirely automatic segmentation methods.

This explains why in most segmentation methods the user still plays a decisive role. The user's responsibility may be the initialization of the segmentation process, intervening during the process or adjusting the result. The minimal contribution of the user in medical image processing is judging the end result. However, despite the fact that human-computer interaction is required in almost all segmentation methods the emphasis in literature lies on description of the computational part, deprecating human-computer interaction. Few methods can be found which squarely describe interaction and computation. The question that arises is: how to integrate interaction
and computation in such a manner as to produce accurate, repeatable and efficient image segmentation? These requirements are of particular importance for the segmentation of medical images [89].

In this chapter segmentation methods in literature are examined. In section 2 relevant interactive segmentation methods in literature will be reviewed and compared from a human-computer interaction perspective. In section 3, the emphasis lies on deformable model methods. A review will be given of promising deformable models found in literature. We conclude this chapter with section 4 were research questions are posed to be addressed in the remainder of this thesis.

2.2 Interactive Segmentation

An attempt to a better understanding of human-computer interaction, particularly in the segmentation of medical images, is found in [89]. In the reference, a comprehensive assessment of interaction in contemporary segmentation methods is given. A general scheme is provided for interactive segmentation method consisting of a user, a computational part, an interactive part, and a user interface. The focus is on the computational part, which is responsible for generating segmentation results on the basis of parameters and the interactive part which mediates between the user and the computational method.

As in the reference, we consider human-computer interaction as a process wherein the human operator conveys his/her knowledge to the computational method (see figure 2.1). The human operator easily recognizes objects in the image where the computational method sometimes fails to do so. In this context, the role of the human operator should be to guide the computational method. In this context the question is posed of: how to minimize and ease interaction for the purpose of steering image segmentation?

2.2.1 Characterization of Interactive Segmentation Methods

To characterize human-computer interaction, the interactive part of segmentation methods is captured in the following aspects: the type of user-input, the interpretation of user input and the purpose of user input. We follow [89] closely.

Type of User-Input

The type of user input considers the form in which the user conveys his/her knowledge to the computational method. As discussed in [89] three types of user input are common in interactive segmentation methods.

In the first type the user provides the computational method with parameter values. This is done in, for instance [88] where weights indicating relative importance of object properties are set and in [110] where the user provides a threshold. This type of interaction requires knowledge of the functioning of the computational part, with interaction potentially becoming inefficient.
Figure 2.1: The interactive part of the human-computer interaction process in image segmentation. Main aspects are: type of user input, interpretation of user input and purpose of user input.

*Visual input* by indicating positions in the image grid is another type of interaction input. The positioning of seeds in the image, e.g. in [110], the indication of regions corresponding with objects of interest, e.g. in [77], and the repositioning of graphical models in the image, e.g. in [88], are different forms of visual input. This type of input is simple and efficient for the user if it concerns roughly indicating points, lines or regions. Otherwise it is cumbersome and time-consuming.

Interaction to select from a *pre-defined menu* where entries refer to parameter values for the computational method is also an re-occurring type of interaction [89]. For instance, the user may be allowed to select a graphical model from a set of pre-defined ones as in e.g. [54]. This type of input is simple for the user and the most efficient type of interaction. The drawback is that often a sophisticated computational method is required to translate user input to a language understood by the computational method.

<table>
<thead>
<tr>
<th>Input Type</th>
<th>Accuracy</th>
<th>Repeatability</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter values</td>
<td>/+</td>
<td>/+</td>
<td>-</td>
</tr>
<tr>
<td>visual input</td>
<td>+</td>
<td>-</td>
<td>/+</td>
</tr>
<tr>
<td>menu selection</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 2.1: Recapitulation of user input type in the human-computer interaction process in image segmentation.

To achieve accurate, repeatable and efficient segmentation, it is natural to select visual input on the image grid over parameter setting and menu-based input for several reasons. In the first place, visual input enables the user to convey his/her knowledge in the same domain in which the image is defined, allowing the input of information on the same level of abstraction as the image information. This is expected to support and facilitate efficient user input. In the second place, visual input allows user input close to the object of interest, increasing the accuracy of interaction. Lastly, visual
2.2. Interactive Segmentation

input may be realized by a superficial glance at the screen and a few mouse clicks. Point-and-click type of user input leads to efficient interaction. Table 2.1 summarizes types of user input.

**Interpretation of User Input**

The interpretation of user input considers how the data provided by the user is translated to a form suitable for the computational method. Two approaches have been identified in [89].

In its most simple form user input is directly used by the computational method. An example of a method which directly interprets user input is found in [94]. The drawback, in this case, is that the user is required to know the functioning of the computational method to a certain extent. Moreover, due to the direct impact on the computational method repeatability is difficult to reach.

The alternative is to use user input indirectly. A mapping function is then required to transform user input into low-level parameters recognized by the computational method. For instance, rough indications of positions in the image grid may be used as a starting point for the search of object regions and boundaries as in e.g. [39], [81], [38], [77]). Or the user provides an initial delineation which is subsequently refined, e.g. in [88], [54].

<table>
<thead>
<tr>
<th>Interpretation of User Input</th>
<th>Accuracy</th>
<th>Repeatability</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>direct interpretation</td>
<td>-/+</td>
<td>-</td>
<td>-/+</td>
</tr>
<tr>
<td>indirect interpretation</td>
<td>-/+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Table 2.2:** Recapitulation of interpretation of user-input in the human-computer interaction process in image segmentation.

For medical image segmentation indirect interpretation of user input is preferred for accurate, repeatable and efficient segmentation for the following reasons. In the first place, indirect interpretation enables the input of information on a level of abstraction which is consistent with that of the user’s. High-level user input is expected to support the user in performing interaction. In the second place, indirect interpretation of user leads to repeatable results. This is due to the fact that user input is processed ultimately by the computational method, which is less subjective than the user. Finally, indirect interpretation enables learning from user input, making reduction of future user input possible. Table 2.2 summarizes types of user input interpretation.

**Purpose of User Input**

The purpose of interaction considers what it is the user tries to achieve with his/her input. Several interactive scenarios have been distinguished in [89].

The role of the user may be to judge whether the computational method has produced an acceptable result. This role is present in all segmentation methods since
the user is always in a position to reject and re-actuate a segmentation process. This is particularly relevant in medical applications where accuracy is important.

A user may also correct the result of the computational method manually as is done in [94]. Manual correction may guarantee user satisfaction, but leads to subjective results because the user has the last say in the process. Manual correction does not favor repeatability.

Setting parameters for the computational method is the most occurring approach to interactive segmentation. It is found in various forms. For example, in [88], the user provides spatial parameters to bring a graphical model closer to the true boundary, while in [110] the user provides thresholds for a better correspondence of an image region with a real world object. Parameter setting produces repeatable results and allows learning based on user-input.

Another purpose of interaction is to compose results by combining primitive results produced by the computational method, e.g. in [105]. The accuracy of such methods may not obey the precision needed due to the fact that results depend on limited set primitives.

The user may also act with the purpose of building a segmentation process. For instance, in [67] the user determines the type and sequence of low-level image processing operations to produce results. The drawback with this kind of interaction is that the user is required to have knowledge of low-level image processing operations, even in cases where visual languages aid the user in the construction of a segmentation process.

<table>
<thead>
<tr>
<th>Purpose of User Input</th>
<th>Accuracy</th>
<th>Repeatability</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>judge</td>
<td>+</td>
<td>-</td>
<td>-/+</td>
</tr>
<tr>
<td>correct</td>
<td>+</td>
<td>-</td>
<td>-/+</td>
</tr>
<tr>
<td>parameter setting</td>
<td>+</td>
<td>+</td>
<td>-/+</td>
</tr>
<tr>
<td>compose</td>
<td>-/+</td>
<td>+</td>
<td>-/+</td>
</tr>
<tr>
<td>build</td>
<td>-/+</td>
<td>+</td>
<td>-/+</td>
</tr>
</tbody>
</table>

Table 2.3: Recapitulation of purpose of interaction in the human-computer interaction process in image segmentation.

We prefer to conceive of parameter setting as the best mechanism to convey intention in user interaction. In the first place, parameter setting puts the user in control of the entire segmentation process, guaranteeing accurate results, while at the same time the computational method produces results efficiently. In the second place, parameter setting leads to repeatable results, because the computational method ultimately produces the results. Finally, parameter setting is efficient because it enables individual treatment of parameters. This is important because interaction is generally confined to parts of objects which form an obstacle, requiring only specific parameters to be corrected. Table 2.3 summarizes purpose of human-computer interaction.
2.2. Interactive Segmentation

2.2.2 Review of Interactive Segmentation Methods

A variety of interactive segmentation methods in literature combine the knowledge of the human operator with the strength of a computational method into one segmentation process. A limited number lets the user provide visual input which is translated to set parameters of the computational method. In the following we review some of these methods.

Interactive Boundary Detection Methods

The piecewise deformable model presented in [88] is a generic and locally controllable deformable model. The idea behind this method is to treat image segmentation as an optimization problem, where an objective function is used to reward certain boundaries in the image. The piecewise deformable model consists of a combination of boundary models and deformable splines, each with a particular segmentation task. A boundary model segment encodes expected shape and image features locally. A spline segment describes a boundary piece in the image in terms of shape and gray-level appearance. An objective function which measures the difference between sampled and expected features is optimized by deforming the spline segments in the image. The optimum is expected when the spline segments lie on the boundary of interest. The spline segments are initialized in the image by the user. The user also helps spline segments over local obstacles. This has only local consequence to ensure that correctly positioned spline segments in the image remain unaffected. The piecewise deformable model adopts human-computer interaction to initialize the model and to steer it by dragging with the mouse any of the spline pieces to better locations, from which the deformation continues.

In [54], a shape-based model is presented that uses prior knowledge of an object’s shape to guide the search for boundaries. The shape-based model is defined by a long axis, which forms the midpoint of a collection of radial contours. A radial contour stores constraints describing the relative positions of discrete points on both sides of a boundary, creating an uncertainty interval. During an automated search-and-propagate process, boundaries are searched along the initial radial contour with a one-dimensional boundary detector, the radial lengths are computed and the uncertainty intervals are reduced to zero. The first computed radial length is propagated to adjacent slices to find complete surfaces. Given a shape model the user specifies at least a single radial length by selecting two points in the image which usually correspond to extremal points of the shape. The user may select more landmark points in the image to enhance the initialization. The shape-based model adopts human-computer interaction for bootstrapping by pointing and clicking landmark points in the image and to correct errors made during the search-and-propagate process by dragging contours in image slices.

In [39], two paradigms, referred to as live wire and live lane, are introduced for image segmentation. In these approaches pixels are considered as vertices of a graph where oriented boundaries are its arcs. To each oriented boundary a set of features is assigned which is converted to a single cost value. In both approaches, the problem of finding the best boundary segment between two points specified on the true boundary
is then translated to finding the minimum cost-path between the two points. In live wire, the user first selects an initial point on the boundary. For any subsequent point indicated by the cursor, an optimal path from the initial point to the current point is found and displayed in real time as if the user has a live wire at hand which is moved by moving the cursor. If the cursor goes close to the boundary, the live wire snaps onto the boundary and follows it. If the live wire describes a boundary segment appropriately, the user deposits the cursor to fix the live wire and provide a new starting point. In live lane, the user selects only the initial point. Subsequent points are selected automatically as the cursor is moved within a lane surrounding the boundary whose width changes as a function of the speed and acceleration of cursor motion. Live-wire segments are generated and displayed in real time between successive points. The user gets the feeling that the live wire snaps onto the boundary as he/she roughly mark in the vicinity of the boundary. In short, the live-wire and live-lane adopt human-computer interaction to actively, but approximately, track with the mouse boundaries in the image which minimize a cost function.

Mortensen et al. [81] present an interactive tool called *intelligent scissors*. Intelligent scissors allow objects to be extracted using simple gesture motions with a mouse. When the gestured mouse position comes in proximity to an object boundary, a live-wire boundary snaps to it, and wraps around the object of interest. Live-wire boundary detection formulates boundary detection as an optimal path search in a weighted graph. Optimal graph searching provides mathematically piece-wise optimal boundaries while greatly reducing sensitivity to local noise or other intervening structures. Robustness is further enhanced with on-the-fly training which causes the boundary to adhere to the specific type of boundary currently being followed, rather than simply the strongest edge in the neighborhood. Boundary cooling automatically freezes unchanging segments and automates input of additional seed points, allowing the user to be much more free with the gesture path, thereby increasing the efficiency and finesse with which boundaries can be extracted. The intelligent scissor adopts human-computer interaction to track boundaries in the image by simple *gesture motions* with the mouse.

<table>
<thead>
<tr>
<th>Segmentation Method</th>
<th>Object model</th>
<th>Interaction target</th>
<th>Interaction type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piecewise model</td>
<td>splines</td>
<td>object boundary</td>
<td>dragging</td>
</tr>
<tr>
<td>Shape-based model</td>
<td>radial contours</td>
<td>object boundary</td>
<td>pointing and clicking</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dragging</td>
</tr>
<tr>
<td>Live-wire/live-lane</td>
<td>polylines</td>
<td>object boundary</td>
<td>tracking</td>
</tr>
<tr>
<td>Intelligent scissors</td>
<td>polylines</td>
<td>object boundary</td>
<td>gesture motions</td>
</tr>
</tbody>
</table>

**Table 2.4**: Recapitulation of interaction in boundary-based image segmentation methods.

Table 2.4 summarizes interaction in boundary-based image segmentation methods.
Interactive Region Growing Methods

Sivewright et al. [110] present an interactive seeded region growing method that assists the clinical user in outlining relevant volumes. Starting from a seed representing an initial interior, statistical tests are performed for adjacent image positions in a manner equivalent to specifying an upper and lower value in thresholding methods. Image positions that pass the test are added to the interior of the region. This process is repeated until regions are constructed consisting of connected areas or volumes having similar pixel intensity values. Region growth is extended into adjacent slices to building up a three-dimensional volume. Direct 3D volume growth from a seed voxel is also possible. Results are returned as a set of interior and exterior regions boundaries. Threshold parameters may be determined automatically from a sample region or may be set by interaction. Region or volume growing is initiated by pointing and clicking with the mouse in the image. The seeded region growing method adopts human-computer interaction to point and click single points in the image and to grow homogeneous regions from there.

The watershed method described in [80] comes from the framework of mathematical morphology and may also be conceived of as a region growing method. In this method an image is seen as a topographical landscape by interpreting the strength of each edge as an altitude. The watershed transformation is the flooding process of this landscape from a number of sources. The water that falls on this landscape will follow the steepest slope until it reaches a minimum. The catchment basins that are formed in this manner represent regions of similar intensity in the image. The watershed lines, dams in the flooding interpretation, that divide adjacent basins are edges that separate the regions. In practice, many irrelevant basins are constructed leading to significant over-segmentation. If prior to the watershed transformation it is known which minima correspond to the desired object and which to the background, markers can be used to pierce only those points in the topographical surface that will lead to a significantly better segmentation. Markers are set by clicking with the mouse at the image locations corresponding to the object and background. Hence, the watershed method adopts human-computer interaction to decide on existence and location of markers by pointing and clicking in the image and to form connected regions from these within boundary conditions.

The active paintbrush method [77] computes primitive regions by a watershed like procedure applied on the gradient magnitude of the image intensity. The watershed boundaries coincide with boundaries in the image. To achieve fewer and larger primitive regions neighboring regions are merged based on intensity similarity, which is formulated as a global optimization problem using the minimum description length principle. This results in a hierarchical description of the image as a tree of merged segmentation primitives where boundaries of increasing granularity are preserved. The computational method then searches primitive region tree on the basis of user input and presents the user the outline of the largest region in the tree that satisfies these constraints. Selecting the regions from the hierarchical region tree that make up the object of interest is done by interaction. The interaction consists of the user dragging the mouse pointer over the object in the screen, whereby all regions hit by
the mouse are selected and displayed in the image. Alternatively, the user may specify a region of interest around the object to be segmented or indicate points in the image that do or do not belong to the object of interest. The active paintbrush adopts human-computer interaction to indicate image locations by sweeping the mouse over it and to group small entities corresponding to the locations into complete objects by merging them.

In the *magic crayon tool* described in [6] an image is presented by a hierarchy which is built using multi-scale and differential geometric methods. The nodes in the hierarchy represent small, primitive regions in the image. The regions are constructed using a ridge flow method, which identifies ridges in the image on the basis of local maxima in the intensity graph. A region of points is assigned to each ridge, each point in this region being the initial value of a flow line on the intensity graph that terminates at the ridge. Each node in the hierarchy represents primitive regions of the image on a selected scale. Small regions at one scale are merged with other regions at a larger scale, i.e. small regions are linked to a parent node at the next level in the hierarchy. The aim is to select a set of subtrees within the hierarchy so that the subtrees corresponding with primitive regions correctly define the object of interest. In conclusion, the magic crayon tool adopts human-computer interaction to indicate, by sweeping with the mouse, subtrees within a hierarchy representing the image and to merge these subtrees into complete objects.

<table>
<thead>
<tr>
<th>Segmentation Method</th>
<th>Object Model</th>
<th>Interaction Target</th>
<th>Interaction Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region growing</td>
<td>-</td>
<td>point single object</td>
<td>point and click</td>
</tr>
<tr>
<td>Watershed method</td>
<td>-</td>
<td>points on multiple objects</td>
<td>point and click</td>
</tr>
<tr>
<td>Active paintbrush</td>
<td>-</td>
<td>object region</td>
<td>sweeping</td>
</tr>
<tr>
<td>Magic crayon</td>
<td>-</td>
<td>graph area</td>
<td>sweeping</td>
</tr>
</tbody>
</table>

Table 2.5: Recapitulation of interaction in region-based image segmentation methods.

Table 2.5 summarizes interaction in region-based image segmentation methods.

### 2.2.3 Examination of Interactive Segmentation

The reviewed methods adopt human computer interaction to guide the computational method in the form of letting the user provide visual input which is translated to parameters suitable for the computational method. In accordance with our aim of minimizing and confining user input, next, the question is addressed of *how to use interaction in such a manner that the role of the user is consolidated to delivering visual input by a few mouse clicks to guarantee accurate, repeatable and efficient segmentation?* We approach this question in the context of the reviewed methods and examine which of them is most suitable for interactive segmentation.

**Visual Input**

Visual input is targeted at the *object boundary* or at the *object region*. 
In the first case, no object model is available and input is performed on the basis of the visual evidence in the image only, as is the case in the live wire and live lane methods [39] and the intelligent scissors [81]. Alternatively input is performed on the basis of the correspondence between the visual evidence and the visual model depicting a possible boundary solution. This is the case in the piecewise deformable model [88] and the shape-based model [54] where respectively splines and radial contours model possible object boundaries throughout the segmentation process.

In the second case, no object model is present and input is performed on the basis of the visual evidence only. The input is targeted at points within one region such as in the seeded region growing method [110] or within multiple regions such as in the watershed method [80]. The input may also be targeted at whole regions such as in the active paintbrush [77].

We choose for visual input on the basis of the correspondence between the visual evidence and a visual model. A visual model, preferably of the entire boundary, provides the user continuously with visual feedback of plausible solutions in the form of an object model defined by spatial parameters. The user only needs to recognize the correspondence between the boundary in the image and the object model at a glance and provide visual information with a few mouse clicks where he/she thinks necessary. The type of user input where an object model is already at hand and only needs to be related with the object boundary in the image is efficient, in particular when it concerns the segmentation of large data sets.

**Indirect Interpretation of User Input**

The purpose of indirect interpretation of user-input is either high-level user input or reduction of future user intervention.

*High-level human-computer interaction* hides technical aspects of the computational method from the user. This is the case with the live wire and live lane [39], the intelligent scissors [81], the magic crayon [6] and the active paintbrush [77], where rough indications in the image suffice for region growing or boundary detection. In the piecewise deformable model [88] and shape-based model [54] higher level user-input is realized in the form of bootstrapping an a prior model of the boundary in the image or dragging parts of it to new locations to refine the result.

In the second place, indirect interpretation of user-input also allows *learning from user-input* by progressively renewing the knowledge contained in the method. In the live wire [39] and intelligent scissors [81], the image properties of stable boundary parts are used to dynamically adjust parameters of the computational method.

To achieve consolidated user input where visual input is delivered by a few mouse clicks, it is natural to support both high-level human-computer interaction and learning on the basis of user input. High-level human-computer makes visual input possible, while learning on the basis of user input reduces user intervention and increases repeatability.
Parameter Setting

Setting parameters is done with the purpose of initialization or steering.

_Initialization_ is the process of bootstrapping a segmentation process. It is done by positioning an initial model in the image as done with the piecewise deformable model [88] and shape-based model [54]. Alternatively it is done by placing seeds and markers in the image as done in seeded region growing [110], the watershed method [80], the live wire and live lane [39] and intelligent scissors [81].

_Steering_ is the process of guiding the computational method through the segmentation process. It is done either by active participation from the part of the user, as done in the live wire and live lane [39], intelligent scissors [81], the active paintbrush [77] and the magic crayon [38]. It may also be done by passively guiding the computational method as done in [88] where the user occasionally drags spline segments to correct locations and in [54] where the user edits contours to correct errors in the shape-based model [54].

Initialization by bootstrapping an initial model close to the true boundary supports consolidated user input where visual input is delivered by a few mouse clicks. This significantly increases efficiency and accuracy. In addition, the combination of having a plausible solution in the form of a model and a user who at a glance recognizes the corresponding boundary in the image reduces steering to a few and simple mouse clicks in the image. In this type of human-computer interaction the user remains in control of the entire segmentation process in the sense that he/she can always intervene in case of unsatisfactory progress of the process he/she initialized.

2.2.4 Discussion

The deformable model approach seems most promising for the purpose of interactive image segmentation for the following reasons. Deformable models combine interaction and computation elegantly. They support visual input by the user in the form of spatial parameters in the image grid, both to accurately initialize the complete boundary model close to the true boundary and to locally steer the computational method to produce better results efficiently. The user continuously receives visual feedback in the form of curves or surfaces defined by the spatial parameters and remains in control of the process in the sense that he/she can always intervene in case of unsatisfactory results. Hence, of the reviewed papers, the _piecewise deformable model_ [88] is most suited for development of interactive segmentation methods.

Moreover, most of the reviewed segmentation methods are mainly directed towards the segmentation of two-dimensional images, where interaction is a depreciated but well developed concept. When dealing with three-dimensional images many concepts from two-dimensional interactive segmentation can be directly inherited, however, some aspects require a completely different approach. For example, while the rough indication of object regions in two-dimensions is straightforward and may lead to accurate interaction, in three dimensional images this task is far from trivial and efficient, reducing accuracy and possibly downplaying methods that adopt this type of user-input. Volume segmentation with help of three-dimensional deformable models
has however been studied [33] more intensively providing a relatively stable platform for the development of three-dimensional interactive segmentation methods.

In short, we conclude that the piecewise deformable model in particular and deformable models in general have properties suitable for interactive segmentation strategies to solve multi-dimensional segmentation problems. We take over the idea behind deformable models and employ them as our main platform for further investigation.

2.3 Deformable Model Methods

Deformable models are a top-down approach to image segmentation, needed for segmentation problems that are difficult, if not impossible, to tackle using classical approaches. They challenge the widely held view that vision tasks are bottom-up processes; features are extracted from an image and higher level processes group or interpolate to find plausible object boundaries. Deformable models start with an a priori boundary model of what an object boundary should look like and refine the model on the basis of image and shape features extracted from a new image to find a mathematically optimal boundary. As a result, they are more robust than non-model based methods, which make little use of a priori knowledge.

The role of the computational method is to find the optimal boundary in the image in an automatic fashion. In our interactive segmentation intention, the question that arises is: how to use the computational method in combination with human-computer interaction in such a way as to produce accurate and reproducible results in an efficient manner? We will address this question in the context of deformable models.

2.3.1 Characterization of Deformable Model Methods

The essence of most deformable model methods can be captured in terms of the following aspects. What prior boundary information is specified? How this is represented geometrically or otherwise? How boundaries in the image are qualified in terms of image and shape features? How the optimization deforms the model from an initial configuration to find an optimal boundary in the image? The main components of the computational part of deformable models (see figure 2.2) correspond with these aspects: the boundary model, the objective function and the optimization. Much of the current research in deformable models is focussed on generalizing boundary models to cope with more than one segmentation problem, reformulating objective functions to obtain more manageable feature landscapes, and overcoming convergence and stability problems encountered during the optimization process.

The deformable model which has attracted most attention is the snake described in [61]. Snakes represent a special case of the general multi-dimensional deformable model theory [79]. In the remainder of this section we discuss the formulation of snakes in its simplest form and use it as the reference model for all papers to be reviewed in the remainder of this chapter.

The boundary model of the snake specifies the expected shape of the object and
Chapter 2. Review of Methods for the Interactive Segmentation of Medical Images

The size and location of the $D$-dimensional image $I(x), I: x \in \mathbb{R}^D \rightarrow \mathbb{R}$. For the two-dimensional case it is represented geometrically by the contour

$$v(u) = (x(u), y(u))$$

where $x$ and $y$ are coordinates in the image and $u \in [0..1]$ is in the parameter domain.

To attract the snake to object boundaries a local image objective is defined whose local minima coincide with intensity discontinuities. The snake is attracted to edges by means of the information derived from the image gradient $\nabla(G_\sigma * I)(x)$, where $\nabla$ is the gradient operator and $G_\sigma * I$ denotes the image convolved with a Gaussian smoothing filter whose characteristic width $\sigma$ controls the spatial extent. The local image objective is defined from the gray-level data as follows

$$E_g(v(u)) = -|\nabla(G * I(v(u)))|^2.$$  \hspace{1cm} (2.2)

The shape of potential objects is described on the basis of features extracted from the contour $v(u)$ itself. To guarantee physically feasible shapes a local shape objective is defined, which keeps this contour from stretching or contracting along its length and another one to keep it from bending. Stretching is measured in terms of the first derivative of the snake’s shape while bending is measured in terms of the second derivative. Assuming arc-length parameterization the local shape objective is defined as

$$E_s(v(u)) = |\frac{\partial v}{\partial u}|^2 + |\frac{\partial^2 v}{\partial u^2}|^2.$$  \hspace{1cm} (2.3)

The global objective function qualifies a sampled boundary in the image on the basis of the object properties expressed in terms of the above image and shape information. The fit quality is the weighted combination of the local image objective value and the local shape objective value, integrated along the entire contour, with the application dependent weight $\omega_g$ and $\omega_s$ indicating the relative importance of each feature type

$$Q(v) = \int_u (\omega_s E_s(v(u)) + \omega_g E_g(v(u)))du.$$  \hspace{1cm} (2.4)
An initial position for the boundary model is specified to bootstrap the search for the optimal boundary. Initialization is done either by a higher-level mechanism, or by other processing. In most cases this is the user. The user also has the opportunity to control the snake behavior by means of an external energy (not defined here), that allows points of attraction and repulsion or other user-defined interactions. Initialization leads to a preliminary result \( \mathbf{v}_b(u) \) which needs to be refined to obtain the final outcome \( \mathbf{v}_f(u) \). The result after refinement is defined such that

\[
\mathbf{v}_f = \arg\max_{\mathbf{v}_b} Q(\mathbf{v}_b)
\]

(2.5)

Optimization aims at finding the optimal boundary in the image, starting from \( \mathbf{v}_b(u) \). The best snake position is defined as the solution of a variational problem, requiring the minimization of the objective function. In accordance with the calculus of variation, the model which minimizes equation 2.4 must satisfy the Euler-Lagrange equation [61]

\[
-\frac{\partial}{\partial u} \left( \frac{\partial \mathbf{v}}{\partial u} \right) + \frac{\partial^2 \mathbf{v}}{\partial u^2} + \nabla E_g(\mathbf{v}(u,t)) = 0
\]

(2.6)

The corresponding Euler equations, which give the necessary conditions for this minimizer, comprise a force balance equation. By introducing a temporal parameter \( t \), the force balance equation can be made dynamic. When the dynamic equation reaches its steady state, a solution to the static problem has been found. The resulting contour is then assumed to be the boundary of the target object.

Analysis of snakes has revealed some common drawbacks. In general the problem is that the two terms 2.2 and 2.3 are defined for the entire boundary. As a consequence the objective function makes the snake always move towards objects with an entirely homogeneous boundary where the intensity changes are highest and the shape is smooth, even if the object of interest does not comply to that description. Many reformulations have been devised in an effort to tackle this and other problems.

### 2.3.2 Review of Deformable Model Methods

A review of some of deformable model methods is found in [59]. In the reference, a classification is made of deformable model methods in literature on the basis of the boundary representation adopted by the methods. A similar approach is found in [48] where deformable models are reviewed from computer graphics point of view. A more general review is found in [79]. McInerney and Terzopoulos discuss several aspects of deformable models for medical images. A comprehensive assessment is given without providing a particular classification of deformable models.

In the following sections, we will order several deformable model methods in literature on the basis of their important contributions.

**Boundary Model**

Yuille [131] proposes a method for detecting and describing features of objects using deformable templates. The object of interest is described by a detailed parameterized
template. An objective function is defined which links edges, peaks, and valleys in the image intensity to corresponding properties of the template. The template then interacts dynamically with the image by altering its parameter values to minimize the energy function, thereby deforming itself to find the best fit. The final parameter values are used as descriptors for the object. The template needs to be hand-crafted completely for other segmentation problems. In the reference, the outer boundary of the eye is defined by two parabolic structures and so on until a q-dimensional feature vector is derived to capture the characteristics of the eye. The objective function is a function of this parameter vector. The priori boundary information in this method is specified in terms of the geometric primitives, the mutual relations between these and the gray-level appearance of the boundary. The deformable template explicitly models prior information about boundary shapes and their respective gray-level appearance with help of geometric primitives.

Duncan et al. [114] propose a more general shape-based deformable model. They apply flexible constraints in the form of a probabilistic parametric deformable model, to the problem of segmenting natural 2-D objects whose diversity and irregularity of shape make them poorly represented in terms of fixed features or form. Probability distributions on the parameters of the representation bias the model to a particular overall shape while allowing for deformations. Boundary finding is formulated as an optimization problem using a maximum a posteriori objective function. Object boundaries are represented geometrically by the continuous function

\[ v^*(u) = \inf_{k=1}^{\infty} p_k \phi_k(u), p_k = \int_a^b v(u) \phi_k(u) du \]  \hspace{1cm} (2.7)

where \( p \) are the coefficients which are projections of the function onto the \( k \) Fourier basis functions \( \phi \) and \((a, b)\) is the interval on which the shape is defined. The Fourier parameterization is chosen because of the geometric interpretation in terms of ellipses and because it is invariant to rotation, scale and translation. The priori information in this method is expressed in the contour representation: the truncation of the sum limits the number of parameter and smoothes the contour. The number of basis functions is a tradeoff between the desired accuracy, conciseness and degree of smoothing. The parametric deformable model captures prior information about global object shape with help of Fourier descriptors, requiring objects to be smooth in order to be represented and detected by the model.

Terzopoulos introduces deformable superquadrics in [120] which are a class of dynamic models that can deform both locally and globally, this way combining global properties of a conventional superellipsoid with the local degrees of freedom of a spline. The model's global deformational degrees of freedom capture gross shape features, while local deformation parameters reconstruct the details of complex shapes that the global abstraction misses. The equations of motion which govern the behavior of deformable superquadrics make them responsive to externally applied forces. They fit models to visual data by transforming the data into forces and simulating the equations of motion through time \( t \) to adjust the translational, rotational, and deformational degrees of freedom of the model. Object boundaries are represented
where $c(u, t)$ is the origin of an inertial frame $\phi$, $R$ is the rotation matrix which gives the orientation of $\phi$ and $p(u, t)$ denotes the position of a point on the model. $p(u, t)$ is expressed as the sum of a reference shape $s(u, t)$ and a displacement function $d(u, t)$: $p = s + d$. Global boundary shape is specified in terms of $s$ which is a superquadric ellipsoid, while $d$ specifies local boundary shape. The main contribution of the reference is the simultaneous modeling of prior information about global and local object shape with help of superquadrics and splines.

In [90], Olstad and Torp propose a grammatical encoding to represent boundary shape and the associated signatures in the underlying images. The variability encountered in an object's shape is addressed with the energy minimization procedure which is embedded in the grammatical framework. They propose an algorithmic solution that combines a non-deterministic version of the Knuth-Morris-Pratt algorithm for string matching [102] with a time-delayed discrete dynamic programming algorithm for energy minimization. Object boundaries are represented by polygonal shapes, existing of straight lines connected at a $N$ ordered vertices $v_n$, $n = 1, ..., N$ with corresponding terminals $\alpha_n$ with which prior boundary information is associated. The encoding of a priori boundary information in a grammatical model taking the form of $\alpha_n = l*cs*$. This expression generates sentences like $ll*llcss*ss$ where the number of $l$ terminals and $s$ terminals are arbitrary. $l$ penalizes all deviations from a straight line, $c$ checks that the point is a corner with a negative bend and $s$ is a smoothness model with a hard constraint on the convexity. The boundary model captures prior information about object shape in an explicit way using grammatical encodings, making it generic enough to express a wide variety of shapes.

Cootes et al. [70] describe a compact parameterized model of facial appearance which takes into account many sources of variability. The model represents both shape and gray-level appearance, and is created by performing a statistical analysis over a training set of face images. A robust multi resolution search algorithm is used to fit the model to faces in new images. This allows the main facial features to be located, and a set of shape, and gray-level appearance parameters to be recovered. Object boundaries are represented geometrically by the point distribution model with cardinality $N$

$$v = \{v_1, ..., v_N\}. \tag{2.9}$$

The point distributions in a training set are used to construct a model of the boundary denoted by $\bar{v} + \bar{P}b$, where $P$ is a matrix of unit eigenvectors of the covariance of deviations, and $b$ is a vector of eigenvector weights. By modifying $b$, new instances of the shape model are created. The elements of $b$ determine the deviation of the model from the prior mean boundary defined by points $\bar{v}$. Having found the shape using the point distribution model, the face is deformed into a normalized frame, in which a model of the intensities of the shape-free face is used to interpret the image. The deformation of the face is done on the basis of landmarks similar to the
thin-plate spline technique of Bookstein [9]. The statistical model learns information about objects from a training set, allowing to consider natural variations in shape and gray-level appearance when looking for objects in new images.

The piecewise deformable model presented in [88] also models shape and gray-level appearance. As described in the previous section the piecewise deformable model is a combination of pieces of splines representing boundary segments in the image. Prior boundary information is specified in terms of basic characteristics of the contour and the expected value of local image and shape feature values. An objective function which measures the difference between sampled and expected features is optimized by deforming these spline segments in the image. The optimum is expected when the spline segments is on the boundary of interest. Object boundaries are represented geometrically by B-splines

$$
\mathbf{v}^*(u) = \left[ \sum_{n=1}^{N} v_n^1 B_n(u), \sum_{n=1}^{N} v_n^2 B_n(u) \right]
$$

where $N$ is the number of control points, $(v_n^1, v_n^2)$ is the two-dimensional position of the $n$th control point, and $B_n(u)$ is the weight defined by the value of the corresponding weight function at path parameter $u$. B-splines can provide a very compact representation for curves, requiring less control points, than for polygonal representation for comparable accuracy. Apart from this the use of B-splines also allows for better representations of highly curved shapes. The piecewise model captures boundary information about local shape as well as gray-level appearance piecewise with help of splines, allowing to deal with heterogeneous boundaries.

<table>
<thead>
<tr>
<th>Boundary model</th>
<th>Boundary representation</th>
<th>Prior boundary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deformable template</td>
<td>parabolic</td>
<td>global shape and gray-level</td>
</tr>
<tr>
<td>Parametric deformable model</td>
<td>elliptic</td>
<td>global shape</td>
</tr>
<tr>
<td>Deformable superquadric</td>
<td>quadric</td>
<td>global and local shape</td>
</tr>
<tr>
<td>Grammatical model</td>
<td>polygon</td>
<td>local shape</td>
</tr>
<tr>
<td>Point distribution model</td>
<td>point set</td>
<td>global shape and gray-level</td>
</tr>
<tr>
<td>Piecewise deformable model</td>
<td>spline</td>
<td>local shape and gray-level</td>
</tr>
</tbody>
</table>

Table 2.6: Recapitulation of boundary representation and a priori information adopted by deformable model methods.

Table 2.6 summarizes the characteristics of the examined boundary models.

**Objective Function**

In [129], segmentation is achieved by tuning the parameters of the geometrical model in such a way that the boundary template locates and describes the object in the image in an optimal way, taking into account directional information. That is, in contrast to other methods, in the reference the optimality of the solution is based on an objective function that matches directional image information with directional shape
information. This is expressed by an image objective function based on directional gradient information derived from Gaussian smoothed derivatives of the image defined as

$$E^*_g(v(u)) = \nabla I(v(u)) \cdot v'_\perp(u) \quad (2.11)$$

where the dot product measures the correspondence of the direction of the normal of the contour with the direction of the image gradient is. The shape objective is similar to equation 2.3. It is based on the contour curvature, but made independent of spatial scale by multiplying the curvature with the length of the contour. The proposed objective function locates an object boundary even in the case of a conflicting object positioned close to the object of interest.

Cohen [17] presents an image objective function that significantly increases the capture range range of the snake method. The local image objective is an increasing continuous function of the distance to the closest point in the set of already detected contour points. Using this potential field, the contour can detect features from further away. Apart from this, weak image evidence will only be ignored if there is a better edge in the vicinity. The local image objective is defined as

$$E^*_g(v(u)) = D(I(v(u))) \quad (2.12)$$

where \(D\) denotes the Euclidean or Chamfer distance map. The objective function does not contain a shape term since the contour is represented by a B-spline which has an inherent regularization part. Hence, the main contribution of the reference is that the proposed objective function increases the capture range of snakes, thereby making it more robust to local minima and initialization.

Xu et al. [130] present a new image objective function which they call gradient vector flow. It is computed as a diffusion of the gradient vectors of gray-level or binary edge map derived from the image. It differs fundamentally from the traditional image objective in that it cannot be written as the negative gradient of a potential function, and the corresponding snake is formulated directly from a force balance condition rather than a variational formulation. To obtain the corresponding dynamic snake equation the image term in equation 2.2 is replaced by

$$-\nabla E^*_g(v(u)) = r(x) \quad (2.13)$$

where \(r : \mathbb{R}^n \rightarrow \mathbb{R}^n\) is the gradient vector flow field that minimizes \(\int_{\mathbb{R}^n} |\nabla r|^2 + |\nabla f|^2 |r - \nabla f|^2 dx\) where \(\nabla f\) is an edge map computed in the traditional manner. The shape objective is identical to equation 2.3. This formulation retains the nice properties of the above mentioned distance based-based image feature, while coping well with concavities in the object boundary. The proposed objective function increases the capture range of snakes and attracts snakes to boundary concavities, reducing the sensitivity to the initial configuration and local minima in the energy landscape.

In [107], Schnabel et al. present a hierarchical multi-scale shape descriptor based on snakes. In contrast to the original snake method the proposed method considers object boundaries at various levels of detail using multi-scale differential invariants.
Starting at a coarse scale for close initialization, image features are extracted and the snake is optimized at decreasing scales. Additionally, distance information is used to attract the snake from large distance. The local image objective function is represented as a function of scale $\sigma$

$$E^*_b(v(u), \sigma) = |\nabla I(v(u), \sigma)|^2 + D(I(v(u)), \sigma). \quad (2.14)$$

The authors also introduce a new shape objective. They propose to minimize the deviation of the isophote image intensity curvature from the contour curvature, instead of minimizing the curvature along the contour. The shape objective is also represented as a function of scale

$$E^*_s(v(u)) = \left| \frac{\partial^2 v}{\partial u^2} \right|^2 - C(v(u), \sigma) \quad (2.15)$$

where $C$ denotes the isophote curvature of the image intensity at scale $\sigma$. In conclusion, the proposed objective is designed to attract the snake to multi-resolution object boundaries with points of high curvature.

Staib et al. \[113\] propose a gradient-based deformable model finding approach that integrates region information. Their approach uses Green's theorem to derive the boundary of a homogeneous region-classified area in the image and integrates this with gray-level gradient-based deformable model to combine the perceptual notions of edge information with homogeneous region information. The image objective function is formulated as

$$E^*_a(u) = |\nabla I(v(u), \sigma)|^2 + \int_A \Delta I(v(u)) dA \quad (2.16)$$

where $\Delta I$ denotes the image indicating homogeneous regions and $A$ is the area bounded by the contour $v(u)$. The shape objective is expressed as multi-variate Gaussian prior obtained from previous outlines, which is used to constrain the optimization within the a posteriori framework. The proposed objective function makes deformable models more robust to noise and poor initialization.

<table>
<thead>
<tr>
<th>Segmentation method</th>
<th>Boundary properties</th>
<th>Boundary configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piecewise DM</td>
<td>curvature + gradient + laplacian + ..</td>
<td>heterogenous</td>
</tr>
<tr>
<td>DM in gradient vector field</td>
<td>curvature + gradient vector field</td>
<td>homogeneous</td>
</tr>
<tr>
<td>Balloon</td>
<td>curvature + distance map</td>
<td>homogeneous</td>
</tr>
<tr>
<td>GFV snake</td>
<td>curvature + gradient vector flow</td>
<td>homogeneous</td>
</tr>
<tr>
<td>Scale-space DM</td>
<td>isoph. curv. + gradient scale space</td>
<td>homogeneous</td>
</tr>
<tr>
<td>Boundary and region DM</td>
<td>curvature + gradient magnitude/region</td>
<td>homogeneous</td>
</tr>
</tbody>
</table>

**Table 2.7:** Recapitulation of objective functions in terms of boundary properties and configurations as adopted by deformable model methods.

Table 2.7 summarizes the conclusions with respect to the definition of objective functions as proposed in the reviewed papers.
Optimization

Amir et al. [1] pointed out several problems with the variational approach to energy minimization. In their work, they showed that the convergence properties of the original snake can not be predicted within the variational framework because variational approaches do not guarantee global optimality of the solution and because they require estimates of higher order derivatives of the image data, which tend to be unstable with noisy data. They also noted that the variational approaches only allow constraints which are additive and differentiable, making it impossible to embed strict hard constraints into snake’s energy functional. They proposed a new formulation for the snake using a dynamic programming method. The snake is discretized to a set of contour points \( v = (v_1...v_n) \) and the objective function is minimized by combining solutions to sub-problems. With the image term omitted for simplicity this reduces to finding

\[
S_n(v_{n+1}, v_n) = \min_{v_{n-1}} S_{n-1}(v_n, v_{n-1}) + \alpha(|v_n - v_{n-1}|) + \beta|v_{n+1} - 2v_n + v_{n-1}|^2. \tag{2.17}
\]

where \( S_n \) is the optimal value function which is obtained by performing a minimization over the discrete point \( v_n \). A discrete dynamic programming algorithm solves sub-problems just once and then saves its answer in a table, thereby avoiding the work of recomputing the answer every time the sub-sub-problem is encountered. Convergence of the energy minimization is guaranteed without ensuring that the global minimum is eventually found. Hence, the contribution of the proposed optimization approach is improvement of the optimality, numerical stability and convergence of snakes, while at the same time allowing hard constraints to be satisfied.

Williams and Shah’s greedy algorithm for active contours [128] performs an efficient local neighborhood search which is more speedy and less memory consuming than the variational approach and dynamic programming. The greedy algorithms performs a search where update of contour elements is explicit and allows the simple inclusion of hard constraints into the model. A neighborhood is defined around each element. For each element in turn the overall energy change caused by a move to each candidate position is calculated. The position that minimizes the resultant contour energy is chosen as the new position of that element. As each move is chosen so that the overall energy of the contour cannot increase, convergence is assured. The snake is discretized to a set of contour points \( v = (v_1...v_n) \) and is evolved according to

\[
v_n = \alpha(d - |v_n - v_{n-1}|) + \beta(|v_{n-1} - 2v_n + v_{n-1}|^2) + \gamma(E_g) \tag{2.18}
\]

where \( d \) is the average distance between point which is updated after each iteration. The objective function is computed for \( v_n \) and each of its neighbors. The location having the smallest value is chosen as the new position for \( v_n \). \( v_{n-1} \) has already been moved to its new position during the current iteration, while the location of \( v_{n+1} \) has not yet been moved. In the next iteration \( v_{n+1} \) is optimized in the same manner with new \( d \) and new weights \( \alpha, \beta, \gamma \). This process continues until a local optimum is found for all points. The greedy optimization algorithm is primarily designed to increase the speed up convergence of snakes while allowing hard constraints to be satisfied.
Cohen introduces the balloon method in [16]. He presents a model of deformation which solves some of the problems encountered with the snake method. To obtain more stable results, the definition of the external forces derived from the gradient of the image is modified to avoid instability due to discretization of the evolution problem to obtain more stable results. A new image objective is also introduced that changes the evolving behavior of the contour in that it pushes the contour to edges like a balloon. The contour passes over edges and is stopped only if the edge is strong, thereby avoiding local minima. Contour finding is performed only in a given area. To enforce this behavior an additional energy term is added that is a negative scalar multiple of the area inside the contour. The equation of motion becomes

\[
\nabla E_f(u) = k_1 n(u) - k \frac{\nabla E_g}{||\nabla E_g||}
\]

where \( n \) is the normal vector to the contour at point \( v(u) \) and \( k \) is the amplitude of this force. Changing the sign of \( k \) or the orientation of the curve causes the contour to deflate instead of inflate. In this manner the contour expands and is attracted and stopped by edges. If the edge is too weak the contour passes due to the pressure force. The proposed optimization approach improves the convergence of snakes even in case of poor initialization.

In [51] a dual active contour is presented. In contrast to other methods which approach image boundaries from one side, the dual active contour uses two contours: one contour expands from inside the target feature, the other contracts from the outside. The two contours are interlinked to provide a balanced technique with an ability to reject weak local energy minima. The internal energy of the contour is reformulated to be scale invariant, and allows a relative assessment to reject poor local minima. The dual active contour allows any additional local shape information to be integrated within the minimization process. This shape models local control over the contour’s equilibrium. The problem of determining parameters is simplified by reducing the parameters to a single regularization parameter which is consistent with the original paradigm. The snake is discretized to a set of contour points \( v = (v_1...v_n) \). Evolution of this contour reduces to

\[
v^{t+1}_n = v^t_n + \frac{1}{2}(\lambda \frac{e_n}{h} + (1 - \lambda)F_n) + g(t) \frac{u_n - v^t_n}{|u_n - v^t_n|}
\]

where \( F_n \) is the force derived from the image objective, \( u \) is the other contour and \( g(t) \) is the strength of the adaptive driving force which is modified by the minimization algorithm. Once both contours have found equilibrium, elements of the two are compared to see if the same minimum energy point has been reached. When this is not the case, the contour with large energy is perturbed until its energy further decreases and finds equilibrium once again. The proposed optimization strategy relieves the user from the problem of initialization by reducing the search space to an area bounded by two contours. Apart from this, the local minima are also dealt with.

In [84] a snake-based approach is proposed which allows a user to specify only the distant end points of the curve he/she wishes to delineate without having to supply
an almost complete polygonal approximation. This is achieved by using the image information around the end points to provide boundary conditions and by introducing an optimization schedule that allows a snake to take image information into account first only near its extremities and then, progressively, toward its center. In effect, the snakes are clamped onto the image contour in a manner reminiscent of a ziplock being closed. Neueschwander et al. reformulate $K \cdot V = F_\text{v}$, which is equation 2.2 after discretization, in order to perform the optimization gradually starting from the head and tail of the contour. The equation which governs the contour evolution is formulated as

$$ (K^* + \gamma^{[t]} I) \cdot V^{*[t]} = \gamma^{[t]} V^{*[t-1]} + 1|_{\gamma^{[t]} V^{*[t-1]}} + F^{*}_{\gamma^{[t]} V^{*[t-1]}} $$

(2.21)

where $\gamma^{[t]}$ is a viscosity term, the superscript $[t]$ denotes the iteration step, $V^*$ stands either for $x$ or $y$ and $F^{*}_{\gamma^{[t]} V^{*[t-1]}}$ is the driving image force. The snake can be used to alleviate the often repetitive task practitioners face when segmenting images by eliminating the need to sketch a feature of interest in its entirety, that is, to perform a painstaking, almost complete, manual segmentation. The ziplock optimization approach simplifies the initialization process and yields better convergence properties.

<table>
<thead>
<tr>
<th>Segmentation method</th>
<th>Initial estimate</th>
<th>Optimization strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic active contour</td>
<td>contour far from boundary</td>
<td>global</td>
</tr>
<tr>
<td>Greedy active contour</td>
<td>contour near boundary</td>
<td>local</td>
</tr>
<tr>
<td>Balloon</td>
<td>contour inside/outside boundary</td>
<td>oriented</td>
</tr>
<tr>
<td>Dual active contour</td>
<td>two contours closing boundary</td>
<td>confined</td>
</tr>
<tr>
<td>Ziplock snake</td>
<td>points on boundary</td>
<td>constrained</td>
</tr>
</tbody>
</table>

Table 2.8: Recapitulation of optimization in terms of initial estimate and optimization strategy as adopted by deformable model methods.

Table 2.8 recapitulates the conclusions with respect to the optimization strategies adopted by the reviewed methods.

### 2.3.3 Examination of Deformable Model Methods

In this section we briefly examine the deformable model approach, primarily on the basis of the reviewed methods.

**Boundary Model**

The boundary model is concerned with expressing qualities of boundaries as observed in the image. Two aspects are considered: the type of a priori information that is encoded onto the boundary model and the type of geometrical representation for the boundary model. When aiming at employing all available information a boundary model is chosen on the basis of its capacity to capture (priori) boundary information about shape, gray-level appearance and the variations therein.
Prior boundary information can range from very general formulations, such as "smooth object" to very specific where detailed information about the expected shape and gray-level appearance of the boundary is captured for every boundary point.

One type of boundary model specifies information about local shape of an object, its size and location in the image. This is the case in the snake model [61] where the boundary model is not biased towards a specific shape. Rather it tries to adhere to a geometric constraint that accidently may describe the shape of the object of interest. The generality of such constraints is appealing. However, it also means that complex shapes are hard to deal with as they cannot not be expressed geometrically or analytically.

Another type of boundary model captures global shape information about an object boundary. This is the case in the parametrically deformable model [114] where the prior information is encoded in the boundary representation. The boundary model can not take any form, rather is restricted to a certain shape range. The shape parameters, in this case, define a specific geometric structure. They bias the boundary model towards this structure. This guarantees almost always a feasible shape as the outcome of the segmentation. That does not imply the shape is located at the right place for each specific image, nor does it imply the shape is a proper answer to the data configuration. In addition, it also reduces the generality of the method to the extent that many complex objects cannot be approached by this approach.

Another frequently recurring type of boundary model captures local shape and image information. This is the case in the piecewise deformable model [88]. Besides information about the local shape of the object the boundary model also specifies expected gray-level appearance along the boundary. Boundary information is captured piecewise so as to capture the variety of information along the contour of heterogeneous boundaries. As a consequence the range of objects it can handle is large. The drawback is that often a detailed configuration of the expected boundary needs to be constructed manually prior to segmentation. Many parameters need to be estimated making the boundary model instable.

Finally several boundary models capture global shape and gray-level appearance. The deformable eye template of Yuille [131] and the facial shape and appearance model of Cootes et al. [70] are an example of this. In the eye template a geometric primitive represents the shape of a boundary. Each such primitive is associated with an expected gray-level appearance. The face model takes a similar approach but specifies global shape and grey-level appearance separately. This is desirable as such global models allow interaction between parts of the model that are far away from each other. However when considering interactive segmentation, this may also be viewed as a drawback as local adjustments by the user will always lead to modifications at parts where this is not wanted.

Boundary representation deals with the geometric description of boundaries. We follow [59] in making a distinction between free-form boundary models and parametric boundary models.

Free form boundary models can represent any arbitrary shape as long as some general regularization constraint such as continuity or smoothness are satisfied. The snake method and the spline-based piecewise deformable model have a free form
2.3. Deformable Model Methods

Boundary model. The boundary model can take any form and is usually hard to bias towards specific non-circular geometrical structures.

Parametric deformable models are capable of encoding specific characteristic shape and shape variation. The shape can either be characterized by a geometrical or analytic formula, templates [131], or using a prototype and variation modes such as in the Fourier model in [114] and the point distribution model in [70]. Prototype-based models are generally more flexible since prior probability distribution is used to constrain the model to vary within a set of allowed shapes.

We opt for specifying both shape and gray-level information in the boundary model because this is more informative and hence leads to more accurate and robust results. Apart from this, we suggest that boundary models should have a geometrical representation which allows free forms that are confined to feasible forms by shape and image criteria encoded in the objective function. This permits to handle a wide range of structures. In addition, it facilitates local steering by the user. The piecewise deformable model [88], the grammatical model [90] and appearance model of Cootes et al. [70] are most appropriate to this end.

Objective Function

The objective function is concerned with providing a quality to boundaries in the image. Two aspects are considered: boundary properties and boundary configuration.

As concerns boundary properties, the quality is determined on the basis of general properties such as “highest image gradient points” or on the basis of specific boundary information such as “specific image gradient values with their variations”. In both cases boundary properties are expressed in terms of image and shape features.

Few variations are found in literature with regard to shape features. Shape is commonly based on bending information, such as (isophote) curvature, which is a popular descriptor due to its invariance to rotation and translation. An emerging class of deformable models, the statistical active shape [22] and appearance models [19], evaluates shape on the basis of prior probability distributions of boundary point coordinates. In this case boundary information in a training set is used to learn boundary properties (and their variation).

A large variety of features is found in literature to capture the image boundary properties. They are broadly divided into three types. In the first, boundaries in the image are defined on the basis of edge information, including distance information [17] and orientation information [130]. The most common type being used is differential image information, popular because of the invariance properties. The disadvantage is that image information is used only associated with the boundary of an object, whereas the interior may contain relevant information as well.

In the second type, objects in the image are defined on the basis of region information, generally specified on the basis of homogeneity or texture. The advantage is that the region of interest does not need to be detected directly from the initial one: as long as a good model of the data comprising the region as a whole is available, the region can grow over arbitrarily large distance to find the desired object region. The disadvantage is the lack of built-in constraints on connectivity of the extracted region
and the smoothness of the boundary.

The third occurring type combines edge information with region information forming hybrid information. Edge information is either integrated with region information, as is the case in [113], or edge information and region information are used competitively, as is the case in [132]. The integration of various image cues of different origin leads to more robust results. Hence, they are preferred over single cue image information.

We consider learning shape and images features, their values and variations therein as a promising approach. Learning from a training set relieves the user from setting feature values and model parameters to specify an object. More importantly, feature values and model parameter settings are more accurate and reproducible and are based on natural observations rather than on user predictions.

Boundary configuration considers the arrangement of properties along the boundary. There are generally two types of configurations. Homogeneous boundaries in the image are defined using the same image and shape features. Although this is alluring from computational point of view, the homogeneity assumption is expected to fail for many segmentation applications, simply because most objects exhibit local inhomogeneities. Despite the broad range of features in literature, the majority of the deformable models do not take into account in their objective function the diversity of shape and gray-level appearance along boundaries.

The alternative is to consider object boundaries as heterogeneous. In this case, a repertoire of features along an object boundary is used to qualify it. Deformable models that scrutinize multiple features at each point along the boundary are expected to outperform those that omit to do so. They can be considered as a superset of homogeneous methods. One of the few reported methods which take into account heterogeneous boundaries are the piecewise deformable model [88] and the active contour with grammatical model [90]. The active shape and appearance model inherently take into account the inhomogeneities along object boundaries in the learning phase.

In conclusion, considering the increasing number of digital images, and the complexity of objects in medical images, it is natural to learn multiple image and shape features along object boundaries and to qualify an object boundary heterogeneously. We view the piecewise deformable model and the active shape and appearance model as most appropriate to this end.

Optimization

Optimization is concerned with actively finding the most optimal boundary in the image. Optimization departs from an initial estimate, usually provided by a higher-level mechanism or the user, assigning initialization an important role in optimization because it influences the final outcome of optimization greatly. Types of optimization strategies found in literature can be described according to their search area.

Global optimization tries to find the most optimal boundary in the entire image. An example of such an optimization is the dynamic programming in [1]. This type of optimization increases the accuracy of segmentation results because the deformable model is less likely to be trapped in a local minima. However, it is not efficient because
it performs an exhaustive search.

*Local optimization* tries to find the most optimal boundary in the neighborhood of the initial model. The greedy algorithm [128] performs such a local search. This type of optimization is efficient but requires a good initialization to prevent being stuck into local minima. Initialization is an inherent difficulty in deformable models as the typically non-convex objective functions might have many local minima that mislead the model. In our interactive setting, we leave this problem to the user, making local a optimization strategy an appealing approach from computational efficiency point of view.

*Oriented optimization* tries to find the most optimal boundary on one side of the deformable model. The balloon [16] performs such an oriented optimization. An initialization of the initial model is required within the area bounded by the boundary of the object of interest or outside this area or else the balloon is not able to find the object boundary. Also, if part of the boundary has weak edge properties, this type of optimization can push the contour over that part towards a stronger edge in the vicinity. The disadvantage is that situation are difficult to handle where, parts of a model need to be inflated while other places need to be deflated.

*Confined optimization* tries to find the most optimal boundary in a confined area on both sides of the deformable model. This type of optimization is found in the dual snake [51]. Starting from two contours, one outside the boundary of interest and one inside the boundary of interest, the two contours are deformed towards each other to find the optimal boundary, capable of overcoming local minima in the energy landscape. The drawback here is that correspondence between the two contours has to be established and maintained during optimization.

*Constrained optimization* tries to find the most optimal boundary in the image by gradually optimizing specific points on the model. The ziplock [84] snake performs a constrained optimization. It first optimizes the outer points of an open contour and continues inward with the optimization of the other points under specific constraints, hence it is very efficient. This type of optimization generally requires a higher-level mechanism to decide which part of the model to optimize first. A disadvantage may be that wrongly optimized initial points negatively affect the remainder of the optimization.

We conclude that in order to achieve accurate results in an efficient manner, it is natural to use a local optimization strategy. The greedy algorithm is a positive example of such a strategy, requiring only an initialization close to the true boundary. In case prior information is present about the location, shape and appearance of the object boundary, it is obvious to select a local constrained optimization strategy which decreases even more the search space by taking into account this information. In this case, optimization like in the ziplock snake [84] is most appropriate.

### 2.3.4 Discussion

To summarize, we strive to qualify objects on the basis of boundary information rather than on the basis of area information. This is expected to facilitate human-computer interaction. Object boundaries are considered to be inhomogeneous and
therefore require definition of multiple features. We choose to represent objects by local shape as well as image information. This combination provides a more detailed boundary description, and hence allows more accurate and controllable segmentation. Furthermore, in light of the interactive setting we endorse and for efficiency purpose we opt for a local optimization strategy. Of the reviewed method, the piecewise deformable model [88] and the active shape and appearance model, seem most suitable for interactive segmentation. While the piecewise deformable offers nice properties that we can take over in designing the interactive part of segmentation, the concept of learning features in the active shape and appearance models is very appealing for the computational part.

2.4 Research Questions

In this chapter we have reviewed a number of deformable model methods in literature, taking the point-and-click type of human computer interaction with visual object models as a basis for further investigation. We have examined deformable models in terms of their boundary model, their objective function and their optimization. This examination has given us a better insight in the benefits and drawbacks of deformable models for interactive image segmentation. It has made clear that while offering a promising platform for combining computation with interaction in an elegant way, deformable models lack the capacity to handle objects that are fractured, occluded, convoluted or inhomogeneous otherwise.

As medical images often contain anatomical structures with boundaries that are locally missing, vague, overlapping or abnormal and hence that violate the homogeneity assumption under which many deformable models operate, the above considerations rationalize the following two research questions. In the first place: how can deformable models be exploited to capture objects that require the definition of multiple boundary features? If deformable models are capable of capturing such objects: how can they be used in combination with user interaction to exploit local boundary inhomogeneities in favor of optimizing user input? These questions form the underlying motivation for the research described in this thesis. Inspired by the appealing properties of piecewise deformable models and active shape models we will conduct further research on the topic of inhomogeneous variational deformable models.