SEGMENTATION OF THE COLON IN MAGNETIC RESONANCE IMAGES

NICOLAS PAYET

NATIONAL UNIVERSITY OF SINGAPORE

2009

SEGMENTATION OF THE COLON IN MAGNETIC RESONANCE IMAGES

NICOLAS PAYET
(B.Eng., Supélec)

A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgements

Contents

Summary

The use of magnetic resonance (MR) images for virtual colonoscopy is a relatively new method for the prevention of colorectal cancer. Unlike computed tomography, magnetic resonance technology does not use ionizing radiations and offer a good contrast for soft tissues. However, the processing of MR images is a very challenging issue due to noise and inhomogeneities.

The first step in processing MR images for virtual colonoscopy is to segment the colon in order to construct a model. Since this model must be as close to the reality as possible, the colon must be segmented with great precision. In this work, we compared two different methods for the segmentation of the colon in 2D MR images.

The first method is based on thresholding algorithms. We first determine the threshold by applying a Bayes classification rule on the histograms. A region growing algorithm is then used to remove non-colonic pixels. We show that due to the presence of significant amount of noise, a good preprocessing algorithm is also needed for the thresholding algorithms to perform well. Therefore, we use anisotropic diffusion as a preprocessing algorithm for noise reduction. We develop a strategy to choose the optimal parameters for this algorithm.

The second method uses deformable models or snakes. Snakes are dynamic contours that move through the images according to internal and external forces. We use gradient vector flow (GVF) as external forces. The snakes are implemented using B-splines. Such snakes are referred to as B-snakes. A control points insertion algorithm and a stopping condition are also implemented to give more flexibility to the snake.

A quantitative evaluation of the results is made based on 30 images from two different datasets. We use Jaccard's measure and obtain an average performance rate of 94%

for both thresholding methods and B-snakes. A qualitative evaluation made on 235 images from the two same datasets shows that segmented regions obtained with B-snakes have a more regular aspect than those obtained with thresholding algorithms.

Finally we show the possibility of a 3D reconstruction of the colon from 2D images segmented with B-snakes on a series of 40 images.

List of Tables

List of Figures

1.   Introduction

1.1   Motivation

With an estimated 677 000 deaths in 2007, colon and rectal cancers (usually referred to as colorectal cancer) are the third most common cancers in the world [4]. In the United States, around 150 000 new cases are expected to occur in 2008 [5]. In Singapore, colorectal cancer is the second most common cancer with approximatively 1000 new cases every year [6].

The colon, or large intestine, is the last part of the digestive system. Its function is to absorb water from the remaining indigestible food matter. The colon begins at the cecum which receives undigested matter from the small intestine. The cecum is followed by the ascending colon, the transverse colon, the descending colon, the sigmoid colon. It ends with the rectum where feces is stored before being ejected through the anus (Fig-

Figure 1.1: Colon [1]

(a) Colorectal polyp [9]   (b) Colorectal polyp with a long stalk [10]
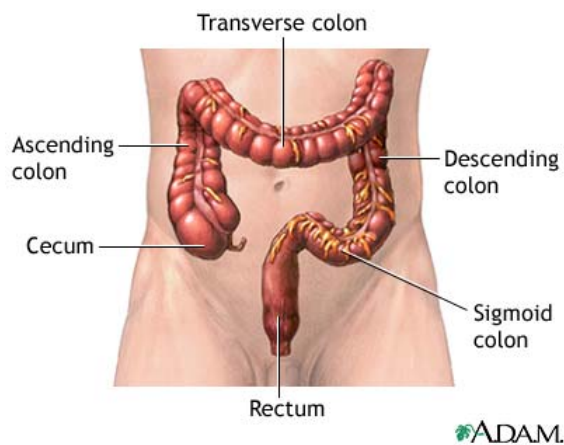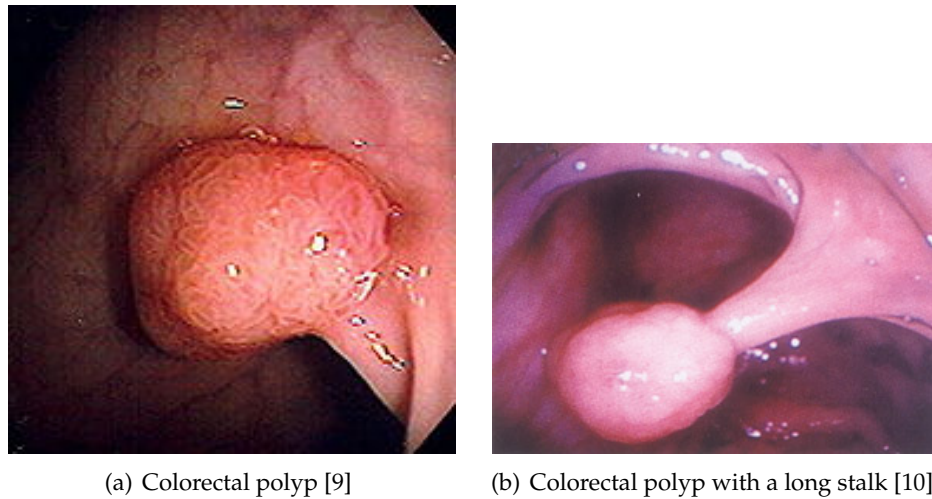
Figure 1.2: Examples of colorectal polyps
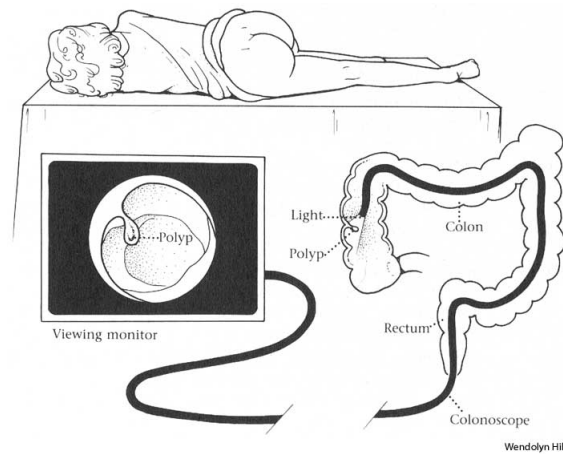
Figure 1.3: Optical colonoscopy [2]

ure 1.1). For more information about the colon, the reader can refer to [7] and [8].

The main form of colorectal cancer is due to the presence of adenomatous polyps in the colon (Figure 1.2). Those polyps, originally benign, may develop into cancer. A cancerous polyp may go through the surface of the colon, and then spreads in the whole body due to the presence of many lymph nodes around the colon. Thus, the early detection and the removal of adenomatous polyps reduces the risk of colorectal cancer [11] [12].

Statistics show that people over the age of 60 are more likely to develop a colorectal cancer. The disease can also have a genetic origin and families with this genetic abnormality present a higher risk [13].

The oldest and most common colonoscopy is the conventional colonoscopy or optical colonoscopy, which consists of inserting a endoscope in the colon of the patient

(a) CT scanner [17]　　　　　　　　(b) MR scanner [18]
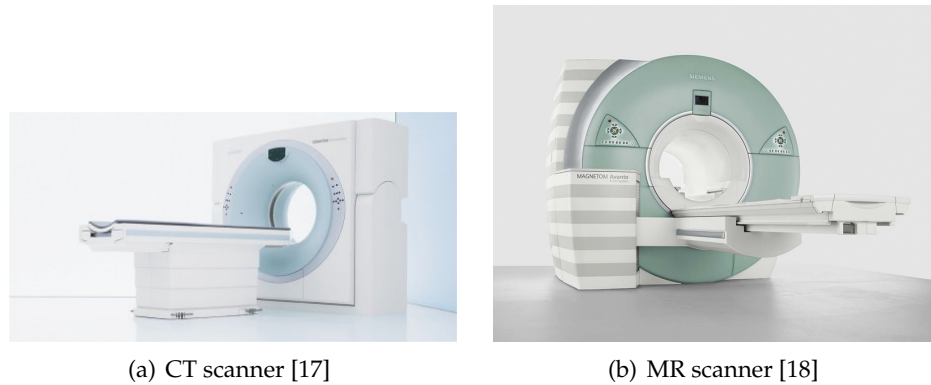
Figure 1.4: Scanners

(Figure 1.3). The two main advantages of conventional colonoscopy are that the resolution of images is high and that it is possible to remove the polyps during the procedure. However, this is a very invasive examination and it is sometimes impossible to complete a whole examination because of patient's discomfort, or because of colon obstruction. Moreover, the rate of missed polyps can be quite high, depending on the experience of the gastroenterologist [14].

To avoid these problems, more recent methods tend to be as uninvasive as possible. One of the first non-invasive methods is called double contrast barium enema (DCBE). Details of this method can be found in [15]. The main drawback of this method is its low sensitivity [16] and the high-level skills required from the radiologist [15].

Currently, the state-of-the-art method in non-invasive colonoscopy is CT colonography [19]. This method consists of acquiring a stack of 2D cross-sectional CT images of the abdomen with a CT scanner (Figure 1.4). Those images can be observed slice by slice by a radiologist in order to detect the polyps. However, such observations are time consuming and require an experienced radiologist. For this reason, the method often requires the images to be processed in order to make them easier to interpret. A software which offers the possibility to reconstruct a 3D model of the colon and to fly through this model is already available commercially [20].

A more recent method is MR colonography. The principle is the same as CT colonography except that the images are acquired with an MR scanner (Figure 1.4(b)).

It has been shown that MR and CT colonography are much better tolerated by patients than conventional colonoscopy [21]. As CT colonography is more cost effective than MR colonography [22], the research has focused more on CT colonography than

(a) CT image of the abdomen   (b) MR image of the abdomen
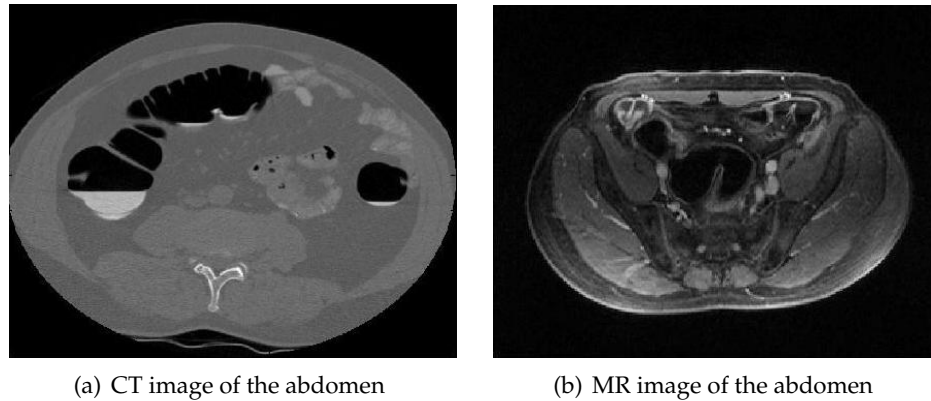
Figure 1.5: Comparison between CT and MR images

on MR colonography. However, as CT colonography becomes more and more popular, questions are raised about the influence of radiation exposure, particularly for people who need to be examined frequently [23]. Since no radiation is involved in MR colonography, radiologists are willing to use this method rather than CT colonography [24]. In the present work, we focus on the processing of images in MR colonography.

1.2 Aim of the Thesis

One of the most important steps in the processing of the images is segmentation. This step consists of isolating the colon from the rest of the image to focus on the region of interest. A good segmentation procedure must fulfill at least two important requirements :

- It must be as accurate as possible. A common requirement for a colonography is the ability to detect polyps as small as 5 mm or at least 1 cm. Therefore, the segmentation process should be able to represent even the fine details of the colon.

- It must be as automatic as possible. Ideally, the radiologist should interact minimally with the software . Practically, it is still difficult to achieve this goal and the radiologist must often give some information to the system (the location of a seed point in the colon for example).

The segmentation of MR images is often challenging; they are very noisy and the contrast between air and tissues is weak. Unlike CT images, which have little noise and a good air/tissue contrast, simple segmentation methods do not perform well (Figure 1.5)

The aim of the present work is to evaluate two different segmentation methods on MR images of the colon :

- The first method comprises two steps. A preprocessing step aims to reduce the noise in the image while enhancing the edges. In this work, we use an anisotropic diffusion algorithm, which has been shown to be a efficient preprocessing step for our application [25]. Then, the segmentation in itself is a combination of thresholding and region growing similar to the one used for CT images [26].

- The second method is based on deformable models or snakes. An efficient initialization method is developed in order to limit the interaction between the user and the software.

1.3 Contributions of the thesis

The contributions of the thesis are summarized here:

- The anisotropic diffusion algorithm, used as a preprocessing step in the segmentation of the colon, is improved by introducing variable parameters. This improvement makes the algorithm less sensitive to noise.

- A procedure to optimize the parameters of the anisotropic diffusion is developed.

- We compare two different approaches to the segmentation problem. One approach is based on pixel classification and thresholding. The other approach is based on contour deformation and snake model.

- A comprehensive implementation scheme is presented for both the finite difference snake model and the B-snake model.

- The feasibility of colon segmentation with B-snakes is demonstrated. Arguments are proposed to explain the superiority of snakes over traditional thresholding methods.

1.4 Organization of the thesis

The outline of the thesis is as follows :

- **Chapter 2**. We present a literature review firstly on the status of colonography (CT and MR) and then on the different methods for the segmentation of noisy images.

- **Chapter 3**. We present a method based on global thresholding, region growing and adaptive local thresholding for segmenting the colon in MR images.

- **Chapter 4**. Anisotropic diffusion is presented as a preprocessing method to improve the results of segmentation. An optimization of the parameters of anisotropic diffusion is proposed.

- **Chapter 5**. After a general introduction on deformable models, we present a traditional implementation of snakes. The limitations of this model are shown, leading to the implementation of a more sophisticated deformable model called B-snake with a different deformation force, the gradient vector flow (GVF). We describe an algorithm to insert control points in order to improve the flexibility of the model.

- **Chapter 6**. The two methods are applied to a series of 2D MR images of the abdomen.

- **Chapter 7**. Concluding remarks and future perspectives are presented.

2. Literature Review

2.1 CT Colonography

CT colonography was described for the first time in 1994 by Vining *et al.*[27]. The first feasibility studies showed promising results. In a unblinded study on 10 patients[1], Hara showed a detection rate of 100% for polyps bigger than 1 cm [28].

Those good results motivated researchers to propose several blind studies. Until 2003, the conclusions of those studies were not satisfactory. Most of the time, the detection rates found by those studies were too low (66% for polyps between 6 mm and 9 mm and 75% for polyps larger than 10 mm in [29], 47.2% and 75.2% in [30] and 56% and 61% in [31]). Other studies obtained much better detection rates (82% and 91%) but the patients had a high risk for colorectal polyps and were therefore not representative.

In 2003, Pickhardt *et al.* [32] published the result of a blind study on an asymptomatic[2] population of 1233 patients. Their results were outstanding with a detection rate of 88.7% for polyps between 6 mm and 9 mm, and 93.8% for polyps larger than 10 mm. Although their results raised some controversies [33], they confirmed the potential of CT colonography as a mass screening method.

Moreover, CT colonography offers the possibility to detect extracolonic lesions, which is obviously impossible with conventional colonoscopy [34].

---

[1]The patients underwent a conventional colonocopy before the CT colonography and the observers were aware of the results of this colonoscopy.

[2]The patients had a normal risk of colorectal polyps, thus being representative of a real population.

2.2 MR Colonography

MR colonography has an even more recent history than CT colonography. It was first described in 1997 by Luboldt in [35], when the technology of MR scanners allowed the possibility of acquiring images of the abdomen in one single breath hold. The first preliminary studies were essentially based on visual assessment [36] and demonstrated the feasibility of MR colonography.

Compared to CT colonography, the literature about MR colonography is relatively poor. We can mention the work of Hartmann *et al.* [37] who obtained an impressive detection rate of 84.2% for polyps between 6 mm and 9 mm and 100% for polyps larger than 10 mm. However, those results are to be taken with precaution because of the small size of the study group (92 people) and because the patients of this study group had a high risk for colorectal polyps. Florie *et al.* [38] had more modest results in a study that focused on limiting the bowel preparation for the comfort of the patient. They found a sensitivity of 75% for polyps larger than 10 mm. A large scale study is still needed to assess the possibilities of MR colonography.

Currently, the main drawbacks of MR colonography compared to CT colonography are:

- the quality of MR images compared to CT images. MR images are more noisy and the contrast between colonic air and tissue is lower than in CT images [39],

- the cost of MR examinations [22].

However, the main reasons that justify the research in MR colonography are:

- Absence of radiation. The exposure to radiations is a main concern in CT colonography [40]. Efforts have been made to reduce the radiation dose without affecting the image quality [41] and reports have been published to show that the risk associated with radiation exposure is not significant [23]. However, in some countries like Germany, the regulations in terms of radiation are so strict that they justify the research effort for MR colonography [40].

- Better contrast in soft tissue. Although the contrast colonic air / soft tissue is lower than in CT images, MR images offer a better contrast between soft tissues of different nature than CT images [39] [40]. This property could be used to observe

the thickness and nature of the colon wall and thus to detect polyps that are not necessarily visible just by looking at the shape of the colon wall (in the case of flat polyps for example). No investigation has been made in that direction so far.

2.3 Colon segmentation

Segmentation is a crucial step in virtual colonoscopy. The literature about colon segmentation in CT images is abundant. Originally, the segmentation methods were mainly based on the thresholding/region growing algorithm. The principle of this algorithm is quite simple: from a seed point located in the colon, we grow a region according to the values of the neighbour pixels. If the value of a neighbour pixel is within a certain range, the pixel is added to the region. This method is described in [42] and [43].

However the results of this algorithm are not always satisfactory. Indeed, the algorithm is vulnerable to fluctuations of intensity in the image. This problem is even more true for MR images. Another drawback of the thresholding/region growing algorithm is that the contour generated is usually quite jagged.

Therefore, many efforts have been made to find more sophisticated methods that could improve the segmentation of the colon. We describe some of the most significant methods in the recent literature.

Van Uitert *et al.* [44] develop an interesting segmentation procedure based on level-set and thresholding/region growing. They first detect the inner wall of the colon with a classical thresholding/region growing algorithm. Since the contrast between colonic air and colonic wall is high enough in CT images, they consider this method as acceptable for the inner wall. However, the contrast between colonic wall and gray tissues around is very low. Therefore, a more sophisticated method is required to segment the outer colonic wall. The authors implemented a level-set algorithm to segment the outer wall. The principle of the level-set algorithm is to consider a level-set function $\Phi(x, y, t)$ and to define the contour Γ as the set of points where $\Phi(x, y, t) = 0$.

$$\Gamma(t) = \{(x, y)/\Phi(x, y, t) = 0\} \tag{2.1}$$

The main advantage of level-set algorithm is its ability to handle topological changes in the contour very easily (merging or splitting of the contour). The results obtained by

Van Uitert *et al.* are quite impressive. However the detection of the outer wall requires a certain homogeneity in the gray tissues surrounding the colon. Therefore, it seems difficult to apply the same method for MR images where many details of the gray tissues are visible.

Another remarkable method is the one developed by Franaszek *et al.* [45]. They first organize the alternance of colonic air pockets and tagged fecal residues pockets in a pocket tree. Then, they combine several algorithms including thresholding/region growing, fuzzy-connectedness, and level-set. Thresholding/region growing is used as an initialization step. Then, fuzzy-connectedness improves the results of the initialization. Fuzzy-connectedness can be seen an improved region growing. From a seed point p_0, a strength path is calculated for each pixel :

$$f(p, p_0) = f_0 \exp\left(-\frac{\Delta^2(p, p_0)}{2\sigma_2}\right) \tag{2.2}$$

with

$$\Delta(p, p_0) = \frac{I_p + I_{p_0}}{2} - \mu \tag{2.3}$$

μ and σ are the mean and variance of pixel intensity calculated in the area found by region growing. f_0 stands for the maximum strength. A pixel p is added to the region if $f(p, p_0) \geq T_{fuzz}$, T_{fuzz} being a predefined threshold. The final contour is obtained by using a level-set algorithm.

Franaszek's algorithm combines many different segmentation algorithms and is particularly efficient for segmentation of colon with tagged fecal residues. In our application, it does not seem that the accumulation of segmentation methods can improve the results significantly.

Finally, Wyatt *et al.* [46] used a particular deformable model called a geometric deformable model to segment the colon in 3D CT images. They consider a surface

$$X(u, v, t) \longrightarrow \mathbb{R}^3 \tag{2.4}$$

This surface evolves according to the equation :

$$\frac{\partial X}{\partial t} = (\Phi H - \Delta \Phi.\vec{U}).\vec{U} \qquad (2.5)$$

where $\vec{U}$ is the surface normal, H is the mean curvature and Φ is a stopping function. In their paper, the authors focus on finding the optimal stopping function. They show the superiority of their method compared to thresholding/region growing.

We have mentioned just a few examples of the most popular methods employed for colon segmentation. Other references can be found for each of those methods, but we will not go further since our main interest is in the segmentation in MR images.

2.4 Segmentation in MR images

Compared to CT colonography, only a few papers have focused on the segmentation process in MR colonography. In the first paper on MR colonography [35], the authors suggest that the colon was segmented with a thresholding/region growing method.

Le Manour [25] proposes to improve the classical thresholding/region growing algorithm in two ways:

- A preprocessing step is implemented to reduce noise and enhance edges. The author uses an anisotropic diffusion algorithm for this purpose

- The results of the thresholding/region growing algorithm are refined by using adaptive thresholding. This algorithm uses a threshold map instead of a single threshold value for the whole image, thus being insensitive to the fluctuations of intensity in the image.

This segmentation procedure shows promising results but is very dependent from the quality of the preprocessing step.

The low level of interest in segmentation methods in MR colonography is quite surprising. Indeed, segmentation in MR images in general is much more challenging than segmentation in CT images (due to the noise and the inhomogeneity of the images).

Segmentation of other organs in MR images has been extensively studied, particularly for the brain and the heart. In the following paragraphs, we give a few examples of those studies.

Admasu *et al.* [47] use a method based on fuzzy-connectedness and artificial neural networks (ANN) to segment sclerosis lesions from brain MR images. Fuzzy-connectedness is used to detect all the parts of the brain considered as normal (white matter, gray matter and cerebrospinal fluid). The remaining parts serve as inputs of a ANN with a single hidden layer. The ANN's purpose is to identify the lesions from other objects among those remaining parts. This method could be applied to detect polyps in the colon, but an important work must be done to identify the possible candidates. This task is much more difficult than identify the candidates for brain sclerosis lesions since there is a very weak contrast between the polyps and the colonic wall.

Chenoune [48] observed the deformation of the left ventricle of the heart in MR images with a level-set algorithm. The particularity of the method is to be implemented in a 2D+t space. The time is indeed considered as an additional dimension to observe the deformations of the left ventricle.

Last but not least, parametric deformable models or snakes have also been widely used in MR segmentation. A snake in 2D is represented by a parametric curve :

$$[0,1] \quad \longrightarrow \quad \mathbb{R}^2 \tag{2.6}$$

$$s \quad \longmapsto \quad v(s) = (x(s), y(s)) \tag{2.7}$$

This curve is deformed by internal forces that keep the curve smooth and external forces that attract the curve to the edges in the image. Deformable models have been widely used in segmentation of the heart in MR images for their strong robustness against noise. Gupta *et al.* [49] and Ranganath [50] showed promising results with a basic model of deformable models. More recently, Terzopoulos *et al.* [51] unified more than 20 years of work on deformable images. They propose a unique finite element model that can describe most of the implementations of snakes that have been used over the past years. They applied their model on many different kinds of images, including:

- CT images of the lungs

- MR images of the brain

- MR images of the liver

- MR images of the legs to detect the growth plates

- Mammograms.

They show that snakes are a very robust model that can adjust to many different kinds of applications.

2.5 Conclusion

Some important points were raised in this literature review:

- Virtual colonoscopy is a promising technology which still needs to be developed.

- Segmentation of the colon in CT images has been extensively studied.

- Those major methods have also been applied to the segmentation of different organs in MR images, mainly the heart and the brain. However, very few efforts have been made to improve the segmentation of the colon in MR images.

- Deformable models appear as an efficient and robust method for segmentation in MR images.

Those arguments motivate us to investigate a new method for segmenting MR images of the colon and to compare it with existing methods.

3. Thresholding methods

3.1 Introduction

The first and most intuitive approach for a segmentation problem is to consider it as a classification problem. The image can be considered as a set of pixels q, each pixel being defined by its position $(x(q), y(q))$ and its intensity $I(q)$. Since we desire to isolate the colon in the image, we decide to assign a label or a class to each pixel. A pixel q will be assigned the label ω_1 if it is located inside the colon and ω_2 otherwise. Therefore, the image will be divided into two sets of pixels. C will be the set of pixels inside the colon and $\overline{C}$ the complementary set.

This chapter presents a method to achieve this classification. This method consists of 3 algorithms:

- Global thresholding,

- Region growing,

- Adaptive thresholding.

The two first algorithms are presented in the first part. Adaptive thresholding is developed in the second part.

3.2 Global thresholding and region growing

The first observation that can be made about MR images of the abdomen is that the intensity of pixels located inside the colon is lower than the intensity of gray tissues

surrounding the colon. Global thresholding exploits directly this property. The main idea is to find an intensity value T, called the threshold, for which

$$\forall q \in \omega_1 \quad , \quad I(q) \leq T \tag{3.1}$$

$$\forall q \in \omega_2 \quad , \quad I(q) > T \tag{3.2}$$

The challenge is to find an optimal value for T. Many methods can be found in the literature. A good summary of the most efficient methods has been made by Sezgin *et al.* [52].

In our work, we use a histogram based algorithm. In Figure 3.1 we represent some histograms of pixel intensities in 2D MR images of the abdomen. Those histograms have been normalized so that the total area under the curve is equal to 1. The background of the images is not taken into consideration. We observe two local maxima. The first one in the low intensities corresponds to colonic air. The second one in higher intensities corresponds to gray tissues. We therefore decide to use a Bayesian rule to find the optimal threshold.

Let z be a random variable representing the pixel intensity. The normalized histogram can be considered as a good approximation of the probability density function (pdf) of pixel intensity. We denote $p(z)$ the values of this function. We also denote $p(z/\omega_1)$ and $p(z/\omega_2)$ the pdf of z given ω_1 and ω_2 respectively. Given the shape of the histograms, we can assume that $p(z/\omega_1)$ and $p(z/\omega_2)$ are Gaussian distributions with means m_1 and m_2 and variances σ_1 and σ_2 respectively. The Bayesian rule defines the optimal threshold as:

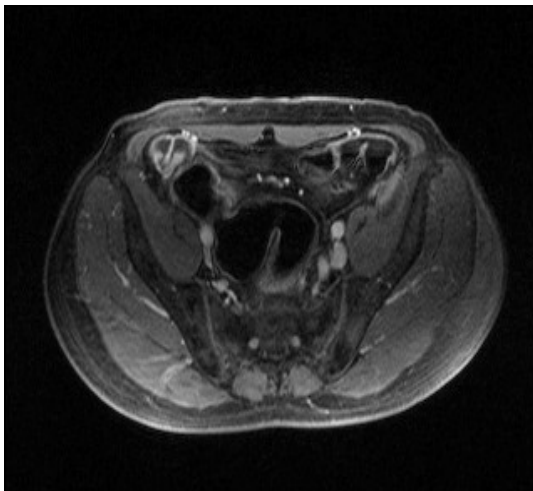$$P(\omega_1)p(T/\omega_1) \quad = \quad P(\omega_2)p(T/\omega_2) \tag{3.3}$$

with

$$p(z/\omega_i) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(z-m_i)^2}{2\sigma_i^2}\right) \text{ for } i = 1, 2 \tag{3.4}$$
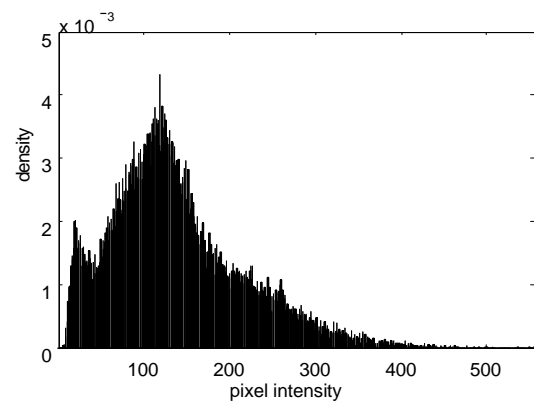
(a) Image 1
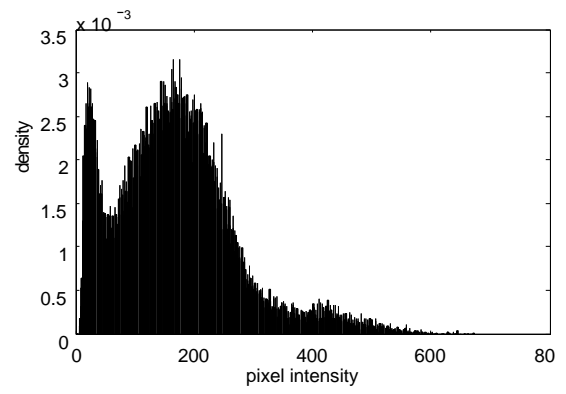
(b) Histogram image 1

(c) Image 2

(d) Histogram Image 2

Figure 3.1: MR images and their corresponding histograms

and where $P(\omega_1)$ and $P(\omega_2)$ are the probability of occurrence of pixels from class ω_1 and ω_2 respectively. From Equations 3.3 and 3.4 we obtain

$$aT^2 + bT + c = 0 \tag{3.5}$$

with

$$a = \sigma_1^2 - \sigma_2^2 \tag{3.6}$$

$$b = 2(m_1\sigma_2^2 - m_2\sigma_1^2) \tag{3.7}$$

$$c = m_2^2\sigma_1^2 - m_1^2\sigma_2^2 + 2\sigma_1^2\sigma_2^2 \ln \frac{\sigma_2 P(\omega_1)}{\sigma_1 P(\omega_2)} \tag{3.8}$$

To solve Equation 3.5 , we need to find the values of $P(\omega_1)$, $P(\omega_2)$, m_1, m_2, σ_1 and σ_2. $P(\omega_1)$ and $P(\omega_2)$ are empirically determined by observing the ratio between pixels inside the colon and pixels outside the colon. For all images, we choose the values $P(\omega_1) = 0.3$ and $P(\omega_1) = 0.7$ which are the values suggested by Yeo [26].

To determine m_1, m_2, σ_1 and σ_2, we require from the user to select one region inside the colon and one region in the gray tissues. We estimate the mean and the variance in each of those regions to determine our parameters[1].

Once the parameters are estimated and the threshold T is found, we classify the pixels of the image according to the rule defined in Equations 3.1 and 3.2. The result is a binary images as we can see in Figures 3.2(a) and 3.2(b). In those images, the pixels with the label ω_1 are represented in white whereas the pixels with the label ω_2 are represented in black.

We notice that not only the pixels inside the colon have an intensity smaller than T. The background of the image and some organs also verify this condition. Thus, further processing is required. We use a region growing algorithm. The user is required to give one seed point in each intra colonic region. Therefore, those seed points have the label ω_1. Each of them initializes a growing region which grows according to the following rule: all the pixels that are 8-connected to the growing region and have the label ω_1 are added to the region. The algorithm proceeds iteratively until the regions remain stable. The results are shown in Figures 3.2(c) and 3.2(d).

We observe that the regions obtained are very noisy and have many holes. This

---

[1]the variance is approximated by an unbiased estimator
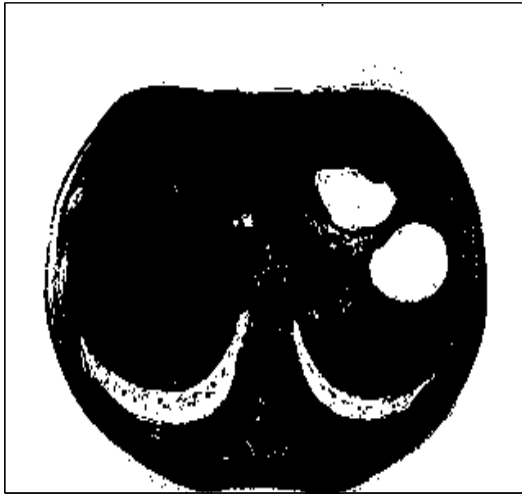
(a) Image 1 after global thresholding
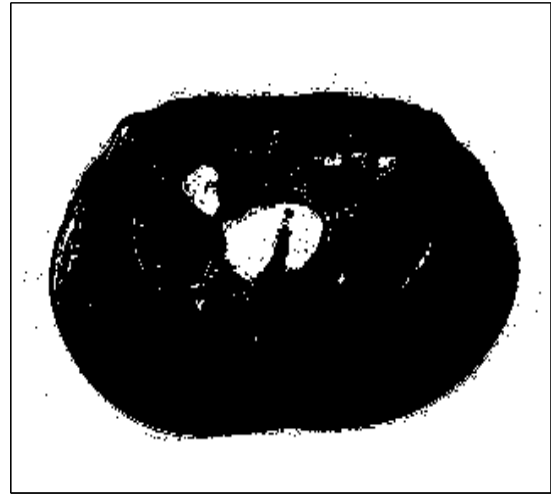
(b) Image 1 after global thresholding
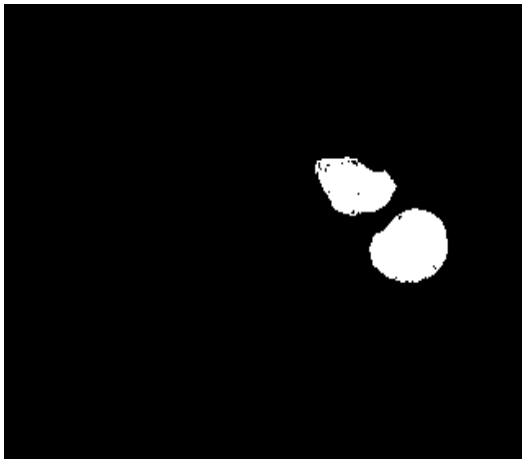
(c) Image 1 after region growing
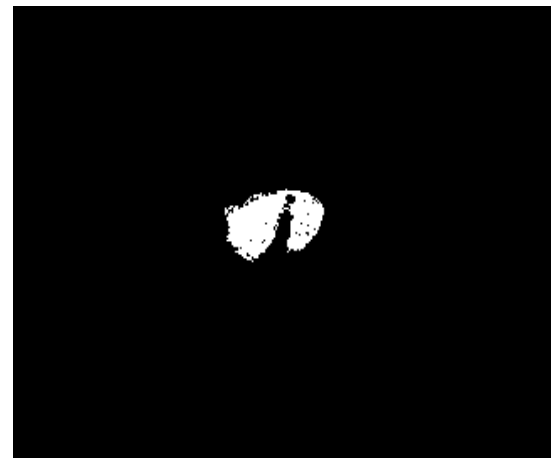
(d) Image 2 after region growing

Figure 3.2: Binary images obtained after global thresholding and region growing
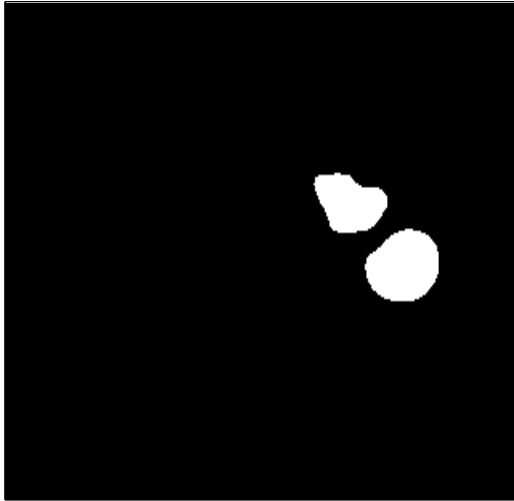
(a) Image 1 after preprocessing with
Gaussian blurring
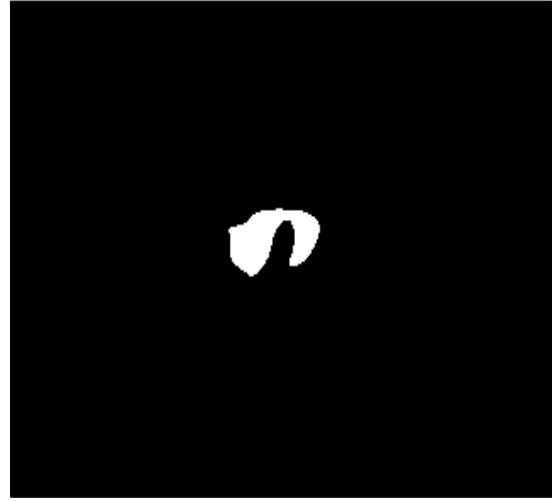
(b) Image 2 after preprocessing with
Gaussian blurring
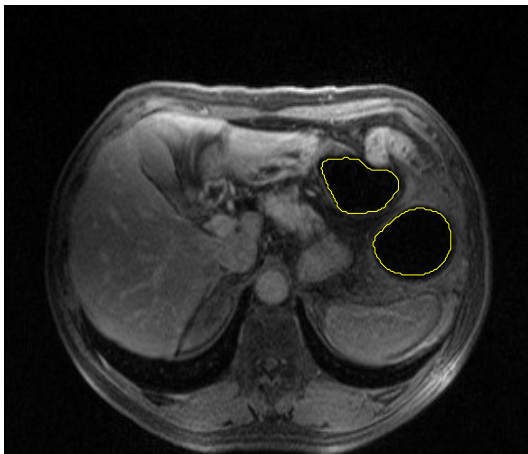
(c) Contours of segmented regions on
image 1
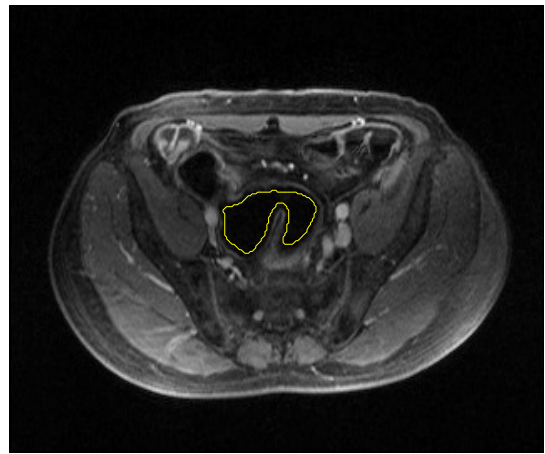
(d) Contours of segmented region on
image 2

Figure 3.3: Influence of a preprocessing with Gaussian blurring

is mainly due to the presence of noise in the image. One idea to solve this problem is to apply a Gaussian filter to the image before the thresholding algorithm. Gaussian filtering can then be considered as a preprocessing step to improve the segmentation procedure. In Figures 3.3(a) and 3.3(b), we represent the regions that we obtained using a Gaussian blurring preprocessing. We obsevre that they are more homogeneous than the segmented regions we obtain in Figure 3.2.

At this point, we would like to evaluate the accuracy of the segmented regions by comparing the regions with the original image. We define the contour of the segmented regions B as the set of pixels q satisfying:

- $q \in C$, and

- there is at least one pixel q' such as $q' \in N_4(p)^2$ and $q' \in \overline{C}$.

Figures 3.3(c) and 3.3(d) represent the contour of the segmented regions superimposed on the image. In these figures, we see that the drawback of preprocessing the image with Gaussian blurring is that the final result of the segmentation algorithm does not accurately include the whole intra colonic region, but a smaller region inside. This is mainly due to the fact that Gaussian blurring not only removes the noise but also blurs the edges. Thus, our segmentation algorithm only allows us to obtain an approximate location of the edges.

Global thresholding and region growing are therefore limited in our application. In the next paragraph, we describe adaptive local thresholding as a way to refine the results of those two algorithms.

3.3 Adaptive local thresholding

Instead of applying the same threshold to all pixels of the image, adaptive local thresholding algorithm adjusts locally the threshold for each pixel according to its neighbourhood. One of the first implementations of adaptive local thresholding can be found in [53]. In this paper, the authors divide the image in small windows. They use a method similar to the one we used previously to determine an optimal threshold for every window. Then, they interpolate the values of those thresholds to assign a threshold for each

---

[2] $N_4(p)$ stands for the set of pixels that are 4-connected to q

pixel in the image. A binary image is then reconstructed according to those thresholds.

This method is not particularly suitable for our application because the region of interest[3] is small compared to the size of the image. Then, we would spend much effort segmenting regions we are not interested in.

Therefore, we develop an algorithm that uses the results from Section 3.2. We suppose that we have already computed segmented regions as in Figure 3.3. We consider one contour B of those regions. For each pixel $q \in B$, we consider $W(q)$, a square window centered in q (we choose a square window with a side of 20 pixels). If we compute the histograms of those windows, we notice that they are similar to the histograms of the entire image. They can therefore be interpreted as the sum of two Gaussian distributions (Figure 3.4 and 3.5). We decide to use a Bayesian rule to compute a new threshold value $T(q)$ for each pixel $q \in B$. We define :

$$
\begin{aligned}
n_1 &= \operatorname{card}\{q \in C \cap W(q)\} & (3.9) \\
n_2 &= \operatorname{card}\{q \in \overline{C} \cap W(q)\} & (3.10)
\end{aligned}
$$

where card stands for the cardinal of the set.

We compute m_1, m_2, σ_1 and σ_2 as the means and variances of intensity among the pixels in $C \cap W(q)$ and $\overline{C} \cap W(q)$ respectively. We also approximate the probability of occurrence of ω_1 and ω_2 in $W(q)$ as:

$$
\begin{aligned}
P(\omega_1) &= \frac{n_1}{n_1 + n_2} & (3.11) \\
P(\omega_2) &= \frac{n_2}{n_1 + n_2} & (3.12)
\end{aligned}
$$

Using Equations 3.5 to 3.8, we calculate a new threshold for the pixel q, $T(q)$. Then, if the intensity of pixel q, $I(q)$ is such as $I(q) > T(q)$, it means that global thresholding misclassified this pixel according to the local properties of the image around this pixel. Therefore, we update B:

$$
B \longleftarrow B \cup q \qquad (3.13)
$$

This process is repeated until B remains stable.

---

[3]the colon and the regions surrounding the colon
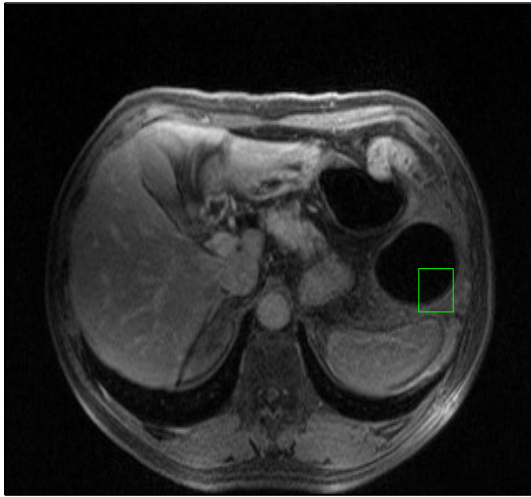
(a) Window on image 1
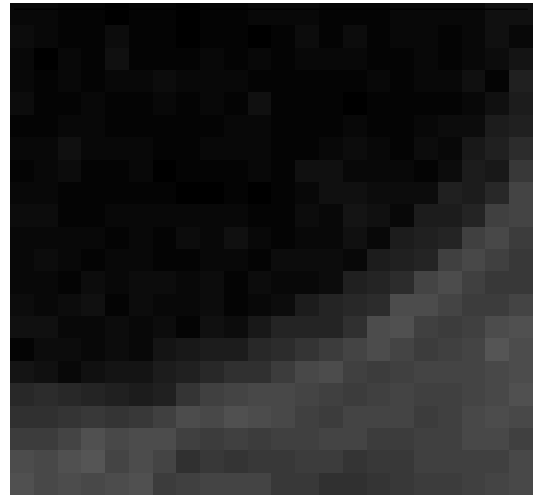
(b) Zoom on the window
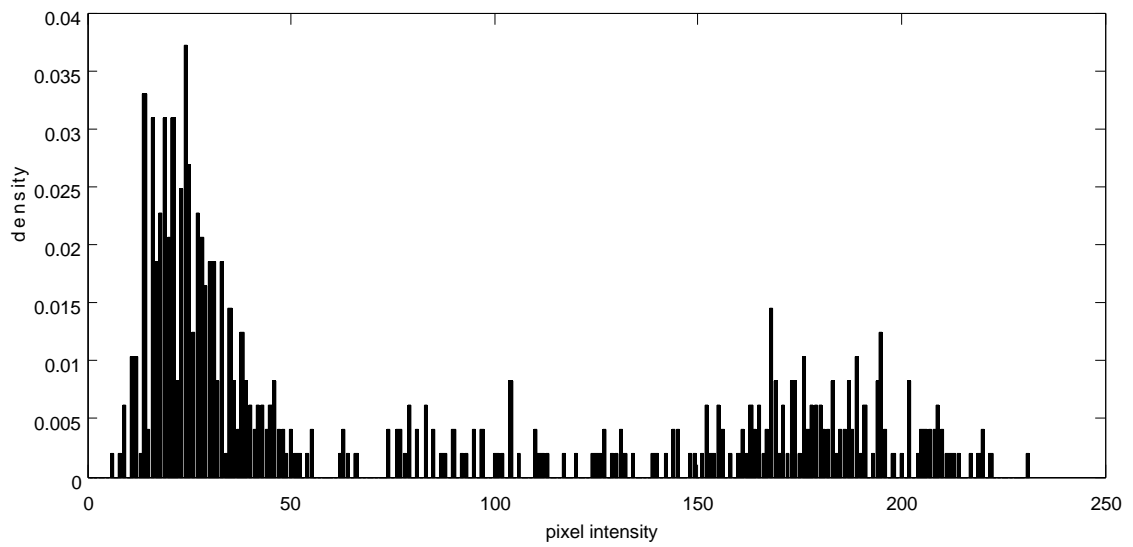
(c) Histogram of the window

Figure 3.4: Histogram of a small window located near the colon wall in image 1
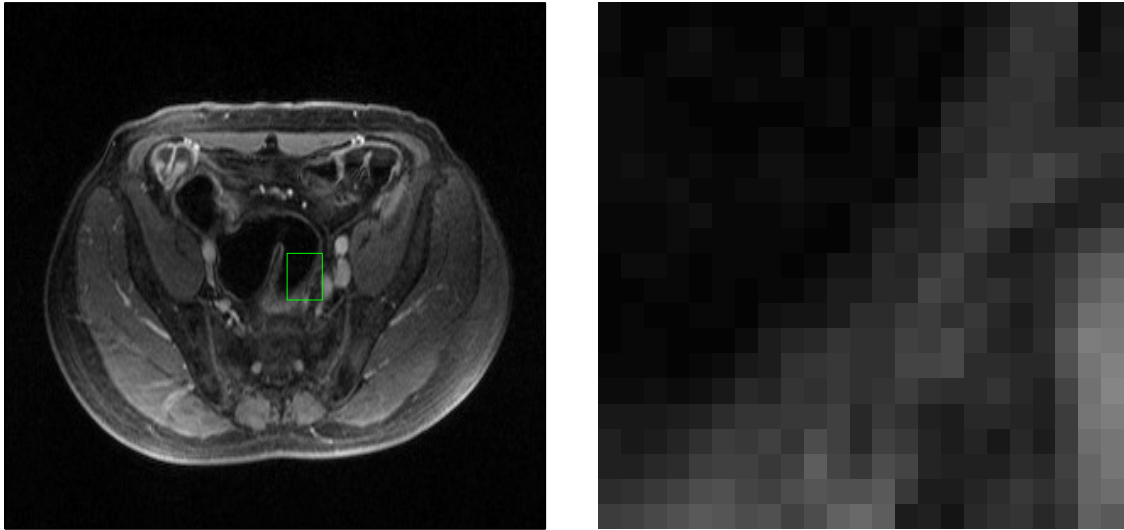
(a) Window on image 2

(b) Zoom on the window
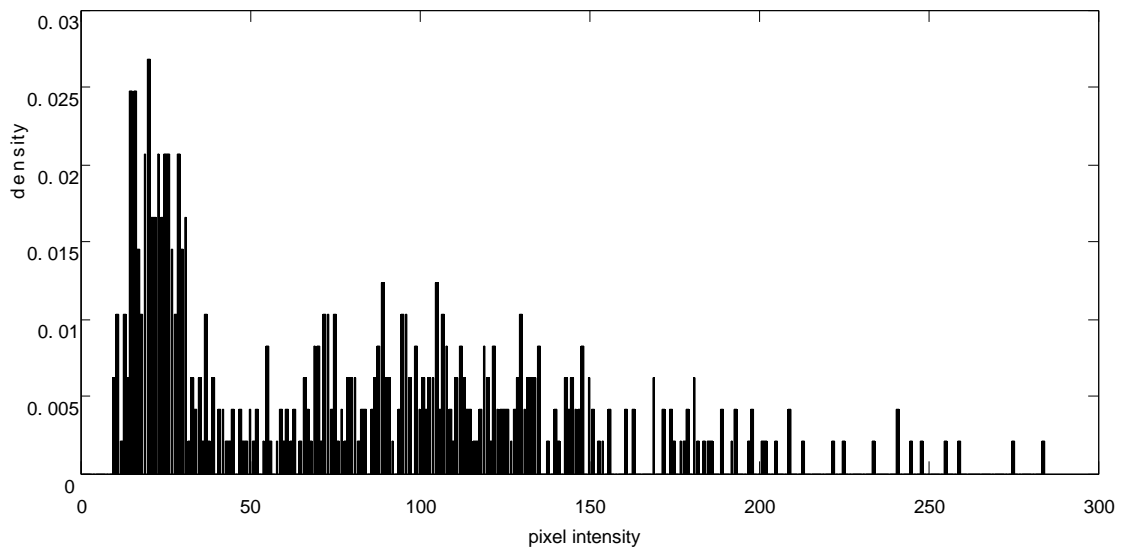
(c) Histogram of the window

Figure 3.5: Histogram of a small window located near the colon wall in image 2

3.4 Results

We present the results of our method in Figure 3.6. We notice an improvement compared to the results obtained in Figure 3.3. The contour of the segmented regions manages to reach the real edge of the colon. However, the contour looks jagged. If we zoom in to the contour as in Figure 3.7(b), we observe many variations whereas a smooth curve would be expected. Despite changing the size of the window $W(q)$ (Figure 3.7), there is no significant improvement.

These results can be explained by the influence of noise in our algorithm. Indeed, even though we reduce the noise to find the approximate contours in the first part, the refinement with adaptive local thresholding uses the original image directly. Threfore, we are still affected by the presence of noise near the edges.

In order to solve this problem, we would like to have an image where the noise is reduced and which would be accurate enough to be used at every step of our segmentation procedure. Therefore, we need to find a better preprocessing algorithm than Gaussian blurring.

3.5 Conclusion

The segmentation method presented in this first chapter can be summarized by the chart in Figure 3.8.

We showed that our segmentation procedure gives jagged contours due to the presence of noise. In the next chapter, we will propose a different preprocessing algorithm to replace Gaussian blurring. This algorithm is called anisotropic diffusion.
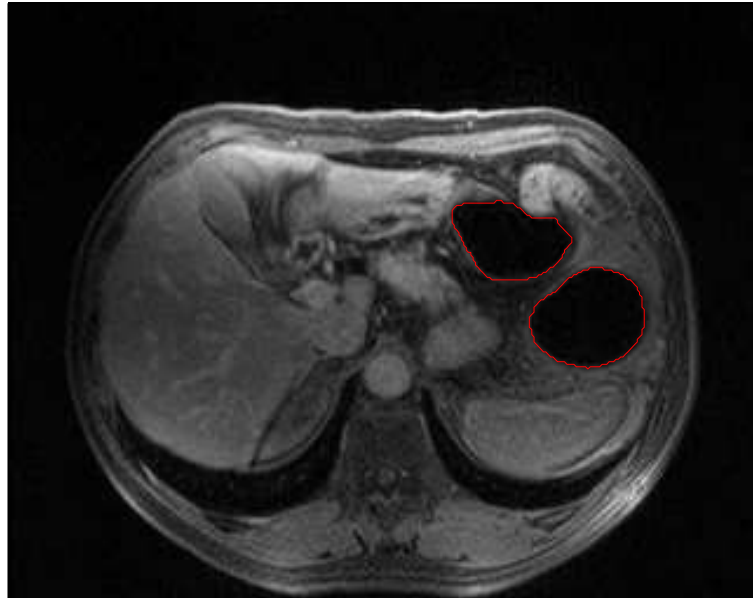
(a) Contour of segmented region on image 1
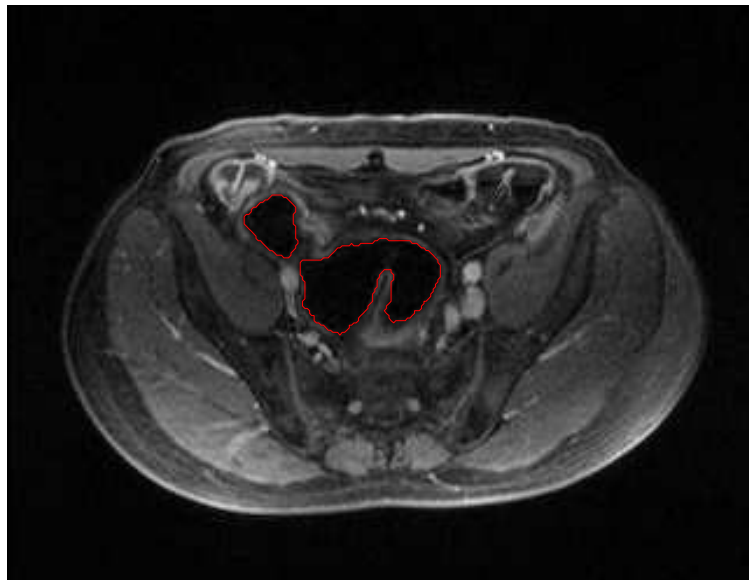
(b) Contour of segmented region on image 2

Figure 3.6: Final contours after adaptive local thresholding
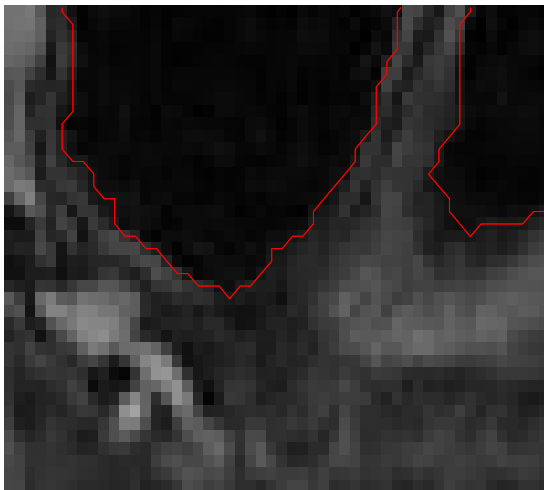
(a) Window sicze = 10x10 pixels
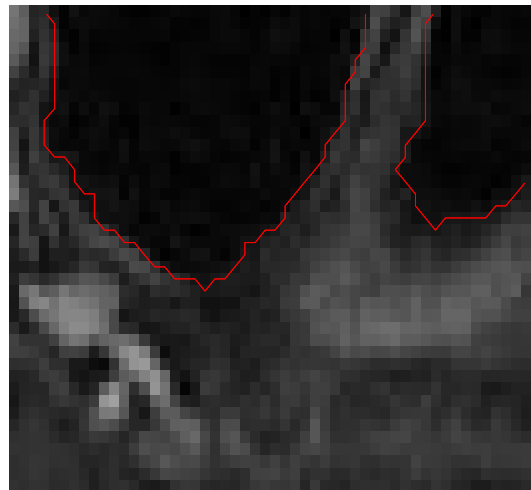
(b) Window size = 20x20 pixels
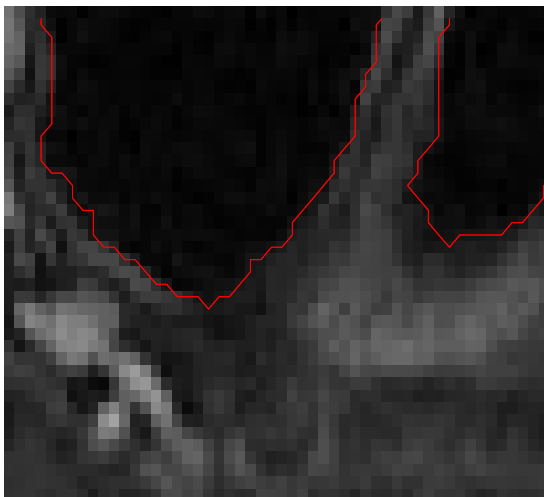
(c) Window size =30x30 pixels
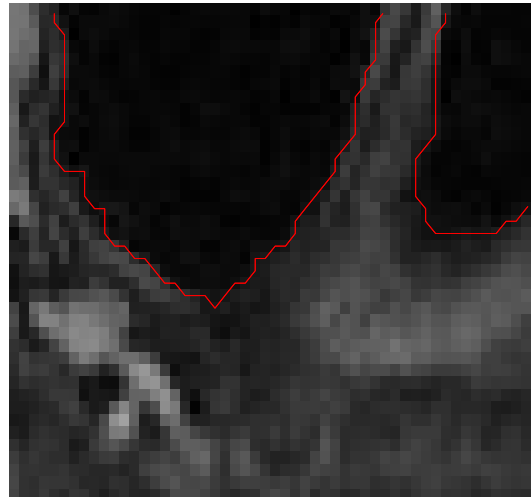
(d) Window size = 40x40 pixels

Figure 3.7: Zoom on the contour for different sizes of the window $W(q)$
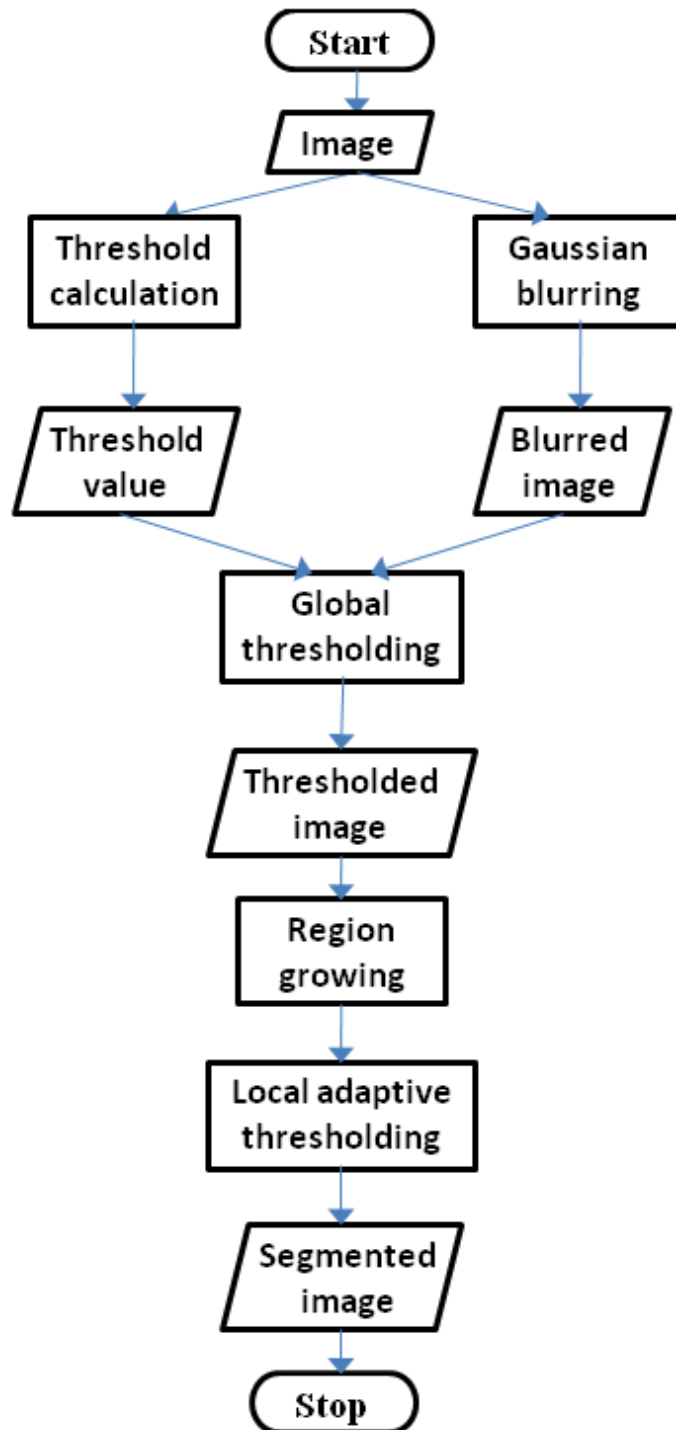
Figure 3.8: Segmentation procedure with thresholding methods

4. Anisotropic diffusion

4.1 Introduction

This chapter presents the anisotropic diffusion algorithm as a preprocessing step to reduce noise while enhancing edges in MR images of the abdomen. After a theoretical presentation of the algorithm, we will focus on the implementation and the optimization of the parameters. Finally, we will see how the segmentation procedure studied in the previous chapter can be adjusted to perform well on images preprocessed by anisotropic diffusion.

4.2 Presentation of anisotropic diffusion

4.2.1 History

Anisotropic diffusion was first introduced by Perona and Malik [54] in 1990 to detect edges in noisy images. If the efficiency of anisotropic diffusion is recognized by the scientific community from the beginning, the first theoretical study was made only in 1992 by Catté *et al.* [55] and then later completed by Weickert [56] in 1998. The first application of anisotropic diffusion for MR images was described by Gerig *et al.* [57] in 1992. In this paper, the algorithm of Perona and Malik was used to enhance edges in MR images of the brain.

Anisotropic diffusion is still present in the recent literature. Montagnat *et al.* [58] use 4D anisotropic diffusion (3D+time) in processing ultrasound images of the heart. They used the model proposed by Weickert as a preprocessing step in their segmentation pro-

cedure to observe deformations of the heart.

More recently, Le Manour [25] proposed the use of anisotropic diffusion as a prepro-cessing step for the segmentation of the colon in MR images. In this chapter, we use the same method as Le Manour and improve it by proposing a quantitative procedure in the choice of the parameters.

4.2.2 Theoretical background

Perona and Malik model

In the first chapter, we used Gaussian blurring as a preprocessing method to reduce the noise in MR images. The main drawback of this method is to blur the edges. Anisotropic diffusion, unlike Gaussian blurring, has the ability to enhance edges while reducing the noise.

The first concept in anisotropic diffusion is to consider the image not only as a func-tion of the position but as a function of time too. We use the notation $I(x, y, t)$. Therefore, the original image is the image at $t = 0$ and all its transformations are considered as be-ing the image at a different time $t > 0$. The image is transformed, or diffused, according to the diffusion equation :

$$\frac{\partial I}{\partial t} = \operatorname{div}(D.\nabla I) \tag{4.1}$$

where ∇ is the gradient operator and D is called the diffusion tensor. We notice that if D is a scalar, Equation 4.1 becomes:

$$\frac{\partial I}{\partial t} = D.\nabla^2 I \tag{4.2}$$

where ∇^2 is the laplacian operator. We recognize the heat equation, of which solutions are well known:

$$I(t) = G_{\sqrt{2kt}} * I(0) \tag{4.3}$$

where $G_{\sqrt{2kt}}$ is a Gaussian kernel of variance $\sqrt{2kt}$. In other words, the diffusion equa-tion with D constant is equivalent to a Gaussian blurring.

The diffusion equation becomes interesting if we allow D to vary according to the

position of the image. Perona and Malik [54] choose $D = g(|\nabla I|)$ where g is an application defined on $\mathbb{R}^+$ and taking its values in $[0, 1]$. The choice of g greatly influences the result of the algorithm. We can impose some constraints on g to achieve our goal. We can consider that a high value for $|\nabla I|$ corresponds to a real edge in the image, whereas a low value corresponds to noise. Therefore, we expect g to have the following behaviour:

$$\lim_{x \to +\infty} g(x) = 0 \tag{4.4}$$

$$g(0) = 1 \tag{4.5}$$

The two functions proposed by Perona and Malik are:

$$g(|\nabla I|) = \exp\left(\frac{|\nabla I|}{K}\right)^2 \tag{4.6}$$

$$g(|\nabla I|) = \frac{1}{1 + \left(\frac{|\nabla I|}{K}\right)^2} \tag{4.7}$$

where K is a constant which influences the sensitivity of the algorithm to edges. The higher K is, the more edges are diffused.

We compare the results of the diffusion equation with D constant and $D = g(|\nabla I|)$ (using Equation 4.6) on a 1D signal. The results are presented on Figure 4.1. For each graph, we display the evolution of the signal according to the diffusion time. It clearly appears that the Perona and Malik diffusion algorithm preserves the edges whereas Gaussian blurring tends to flatten the entire signal.

Weickert's model

Although the results obtained by Perona and Malik are impressive, several researchers raised the point that no analysis of the condition of stability has been made. Catté *et al.* [55] show that if the noise in the image is significant, the algorithm proposed by Perona and Malik can lead to instabilities and tends to enhance the noise. They solve this problem in a simple and elegant way, by adding a regularization function. Equation 4.1 becomes:

$$\frac{\partial I}{\partial t} = \mathrm{div}\left(g(|\nabla G_\sigma * I|).\nabla I\right) \tag{4.8}$$
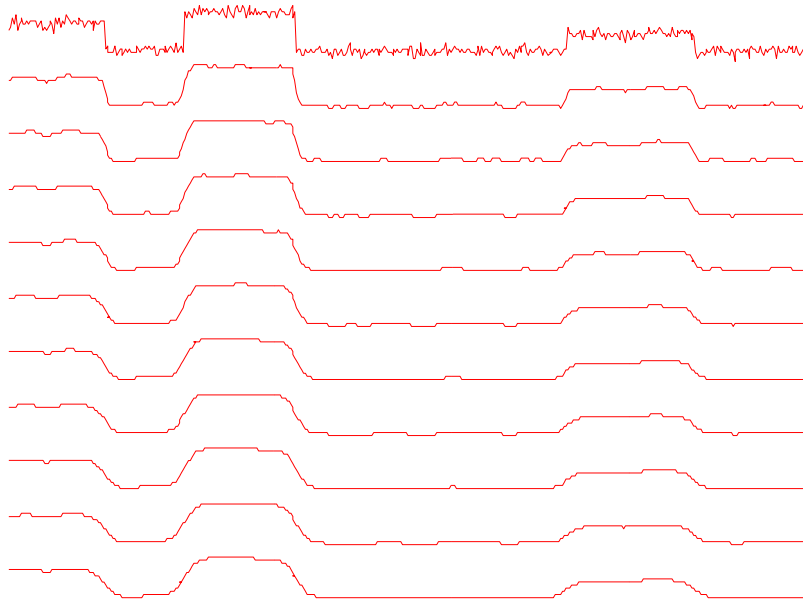
(a) 1D diffusion with D constant
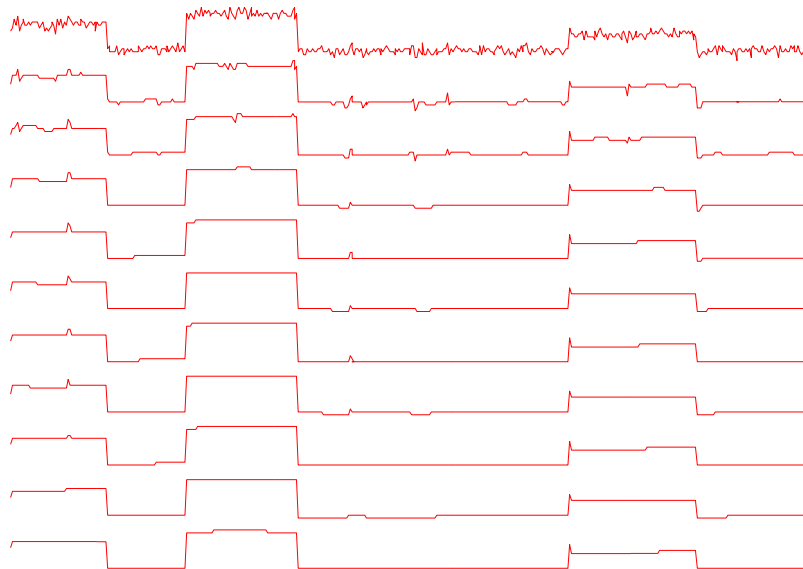
(b) 1D diffusion with D=g(|▽I|)

Figure 4.1: Comparison between anisotropic diffusion and Gaussian blurring on a 1D signal

where G_σ is a Gaussian kernel. They prove that with this simple modification, enhancement of noise is avoided in general. They also prove uniqueness of the solution in this case.

In [59], Weickert studied in detail the mathematical background of anisotropic diffusion. He established a series of conditions that g must fulfill for stability. He showed that the Perona and Malik functions do not fulfill those conditions and proposed a new diffusion function:

$$g(|\nabla I|) \quad = \quad 1 - \exp\left[\frac{-C}{\left(\frac{|\nabla I|}{\lambda}\right)^4}\right] \tag{4.9}$$

where C is a constant chosen so that $xg(x)$ is increasing for $x < \lambda$ and decreasing for $x \geq \lambda$. The parameter λ plays a similar role as the parameter K in the Perona and Malik's equations.

Improvement of Weickert's model

Li *et al.* [60], in their analysis of the Perona and Malik's model improved by Catté, show that anisotropic diffusion tends to enhance noise if the amplitude of noise is similar to the amplitude of edges in the image. We tried to see if the Weickert's model led to the same issue. In Figure 4.2(a), we apply the Weickert's model to a 1D noisy signal which contains one edge and one noisy peak of similar amplitude. We note that the noisy peak is enhanced. Therefore, the Weickert's model does not solve the issue raised by Li *et al.*

The solution proposed by Li *et al.* consists of making the parameters K and σ in Perona and Malik's equation (Equation 4.6 depend on the diffusion time t. A similar action with the Weickert density function (Equation 4.9) would be to make λ and σ dependent on t. The idea is to start with higher values for λ and σ so that high edges are enhanced and the noise is reduced efficiently (in our 1D signal example, a high value of σ can help flatten the peak). As the process evolves, the noise almost disappears and σ becomes unnecessary. But small edges must also be preserved and enhanced. Therefore, we make λ decrease with the diffusion time. We take the example on Li *et al.* to choose
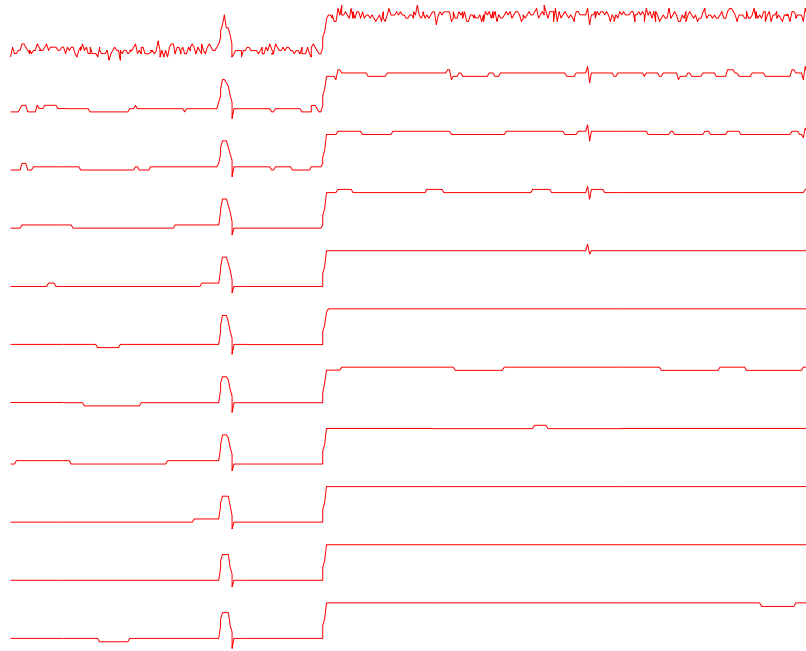
(a) 1D diffusion with constant parameters
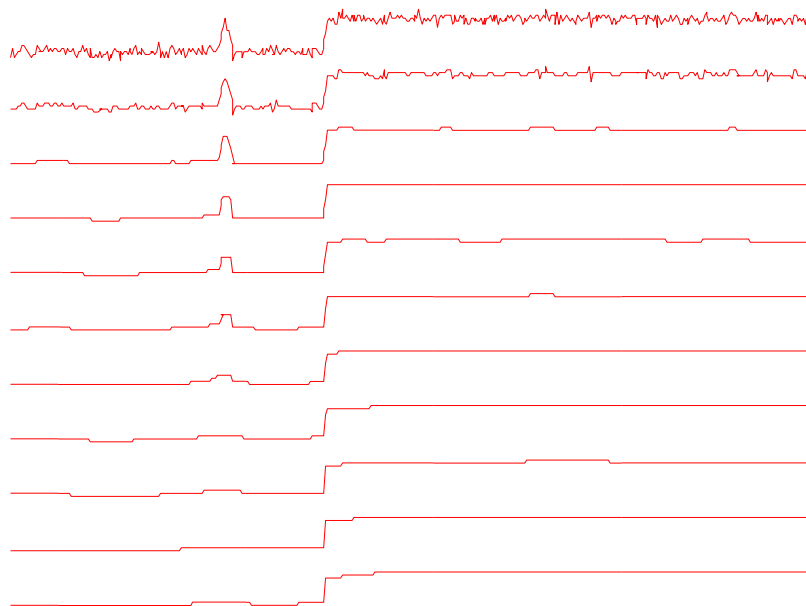
(b) 1D diffusion with variable parameters

Figure 4.2: Diffusion of a 1D signal with a noisy peak
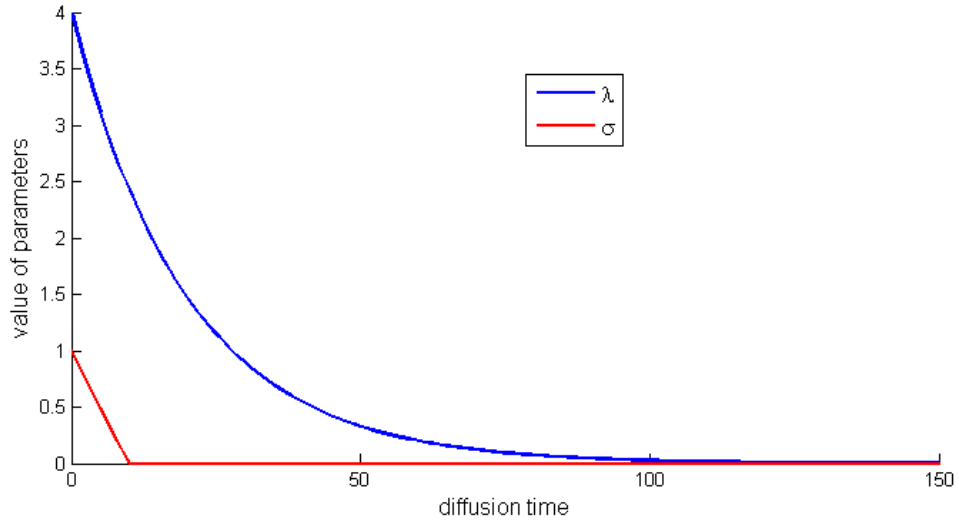
Figure 4.3: Evolution of parameters λ and σ

the following expressions for λ and σ:

$$\lambda = \lambda_0 \exp \frac{-t}{20} \tag{4.10}$$

$$\sigma = \mathcal{L}^+ \left(\sigma_0 \left(1 - \frac{t}{10} \right) \right) \tag{4.11}$$

where $\mathcal{L}^+$ is an operator defined as:

$$\mathcal{L}^+(x) = x \quad \text{if } x \geq 0 \tag{4.12}$$

$$\mathcal{L}^+(x) = 0 \quad \text{if } x < 0 \tag{4.13}$$

The evolution of parameters λ and σ according to the diffusion time are shown in Figure 4.3.

We apply the algorithm with those parameters on the 1D signal (Figure 4.2(b)). The improvement is obvious; the noisy peak is no longer enhanced while the edge is still preserved.

In the rest of this work, we will use the diffusion Equation 4.8 with the Weickert's model defined in Equation 4.9 and variable λ and σ as defined in Equations 4.10 and 4.11.

4.3   Implementation

Now that the theory about anisotropic diffusion has been established, we need to discretize Equation 4.8 to implement it. We will just give here the most important ideas of this implementation. The reader can refer to the works of Le Manour [25] and Weickert [56] [59] for more details.

Since I is defined as a function of space and time, the discretization must be done in both space and time. The spatial discretization is straightforward. The pixel structure of the original image gives the rectangular grid on which the image is projected. Central differences are used in order to calculate the divergence operator. Equation 4.8 can be written as:

$$\frac{dI}{dt} = \mathbf{A}(I).I \tag{4.14}$$

where $\mathbf{A}$ is a $(n \times n)$ matrix.

The time discretization requires more attention. We discretize the time space with a time step τ. We use the notation $I_k = I(k\tau)$. The usual explicit scheme leads to:

$$\frac{I_{k+1} - I_k}{\tau} = \mathbf{A}(I_k).I_k \tag{4.15}$$

However, this scheme is stable only if τ satisfies some conditions. Weickert shows that the stability can be guaranteed in any case for $\tau < \frac{1}{2}$. This condition is too restrictive for our application. Weickert proposes to use a semi-implicit scheme:

$$\frac{I_{k+1} - I_k}{\tau} = \mathbf{A}(I_k).I_{k+1} \tag{4.16}$$

which leads to

$$[\mathbf{I} - \tau\mathbf{A}(I_k)]I_{k+1} = I_k \tag{4.17}$$

where $\mathbf{I}$ is the identity matrix. This scheme is unconditionally stable. However, it requires more computation since the matrix $\mathbf{B} = \mathbf{I} - \tau\mathbf{A}(I_k)$ needs to be inverted at each iteration. Since this matrix is tridiagonal, it can be easily inverted with efficient algorithms.

4.4 Choice of parameters

4.4.1 Review of previous work

Now that the implementation scheme has been established, we still need to set the parameters of anisotropic diffusion. Four parameters have an influence on the result:

- λ: this parameter is inherent to the diffusion equation. It influences the sensitivity of the algorithm to edges. The higher λ is, the more edges are diffused.

- σ: the variance of the Gaussian kernel in the regularization process.

- τ: the timestep. As we mentioned before, the stability of the algorithm is not affected by its value, but the accuracy of the result can be affected.

- N: the total diffusion time or in other words, the number of times that Equation 4.17 is applied.

An analysis of the best value for the timestep τ is made by Weickert in [56]. He observes that for $\tau \leq 5$, the results are not affected. For $\tau \geq 10$, the image starts having distortions that are not acceptable. Therefore, we agree with the value $\tau = 5$.

N must be high enough for the algorithm to modify the image. However, it must not be too high, otherwise the image will be too diffused. Whitaker *et al.* [61] show that it is impossible for the image to stabilize unless it is wider than 150,000 pixels. We decide to choose $N = 50$. We will discuss this choice later.

We still have two parameters to set, λ and σ. More precisely, since we have defined λ and σ as variables of the diffusion time t (Equations 4.10 and 4.11), we need to set λ_0 and σ_0. In the literature about anisotropic diffusion, those parameters are usually chosen based on visual interpretation. Le Manour [25] makes an attempt at choosing λ and σ based on the signal to noise ratio (SNR) on a computer generated image. The image is corrupted with Gaussian and speckle noise. Diffusion anisotropic is applied to the corrupted image and the parameters are chosen in order to obtain the highest SNR. The method chosen by Le Manour is debatable. Indeed, measuring the SNR is good way to know how anisotropic diffusion reduces the noise in the image. But it does not give any information about how edges are preserved. Therefore, we develop a new method to evaluate the best parameters according to the two features we are interested in : noise

(a) image3   (b) image4
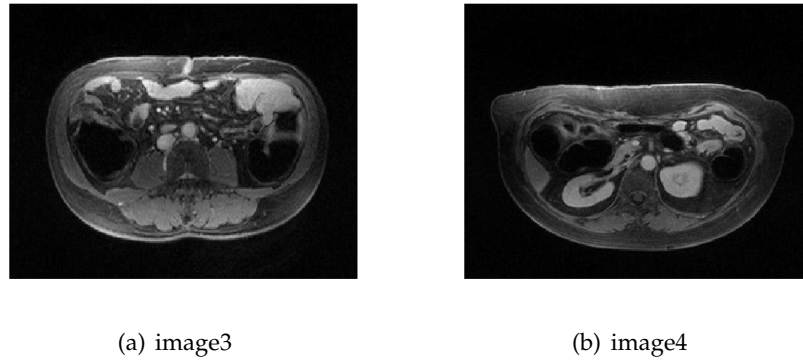
Figure 4.4: MR images from two other datasets

Figure 4.5: Computer generated image

reduction and edge enhancement.

4.4.2 Creation of a computer generated image

Our first task is to create an image that would be representative of real images while being relatively simple. We decide to use a 200×200 image with a dark shape on a brighter background. The dark shape represents the colon while the background represents the gray tissues. We choose the value of the pixel intensities for each of those regions according to our observation of real images.

For this purpose, we use the two images that were previously mentioned in this work (Figures 3.1(a) and 3.1(c)) as well as two other images coming from two different datasets (Figure 4.4). We select manually an 8×8 area inside each colonic air pocket. Then, we calculate the mean value of all pixels. We find a mean value of 17.98 (with a

standard deviation of 6.37). Therefore, we decide to set the intensity of pixels inside the dark shape equal to 20.

Setting a representative value for the background is not as straightforward as the range of intensities for gray tissues is wide. The contrast between the colon and surrounding gray tissues can be very high as in Figure 3.1(a) or very low as in Figure 4.4(b). We decide to set the background value at 50, which represents a low contrast. Our assumption is that if the algorithm can perform well in the worst cases, it will perform well in more favorable cases. Our computer generated image is shown in Figure 4.5.

The next step is to find a model for the errors that naturally affect MR images. Those errors can be classified into two categories:

- small scale errors or noise, and

- large scale errors or bias. The main consequence of this error is that the mean value for one particular kind of tissue can vary spatially

We use a model described by Guillemaud *et al.* [62]. Its mathematical formulation is

$$I_{\text{measured}} = I_{\text{original}}B + N \tag{4.18}$$

I_{measured} is the acquired MR image. I_{original} is the image which is supposed to represent exactly the reality. B and N represent the bias and noise, respectively. Our goal is to find expressions for B and N which are representative of the bias and noise that usually affect MR images. Once we have those expressions, we will be able to corrupt the computer generated image which will serve as the input for anisotropic diffusion.

According to Guillemaud *et al.*, noise in MR images usually follows a Rayleigh or a Rice distribution, but a classical Gaussian distribution with zero mean can be assumed to evaluate the performance of algorithms. We make this assumption. A value for the standard deviation of the noise σ_N needs to be determined. Kaufman *et al.* [63] propose to estimate the standard variation of the noise in MR images by measuring the mean M and the standard deviation SD of the intensity in regions of real MR images where there is no information, for example the air surrounding the abdomen. Due to the fact that the intensity of MR images is a complex value and that we only have access to the magnitude in the displayed images, they found that M and SD are, respectively, an

overestimation and an underestimation of σ_N:

$$M \quad = \quad 1.253\sigma_N \tag{4.19}$$

$$SD \quad = \quad 0.655\sigma_N \tag{4.20}$$

According to Gerig *et al.* [57], the measurement of M and SD could be affected by the bias, unless it is done in small regions of the image. Therefore, they consider 8×8 regions in homogeneous parts of real MR images and calculate M and SD for each of those regions. We use a similar strategy by taking 8×8 regions in the four images that we used before. Those regions are always taken in the air surrounding the abdomen. We use a total of 320 regions, 80 in each image, and we calculate M and SD for each of those regions. For M, we find an average value of 11.51, the minimum value being 8.41 and the maximum value being 19.83. For SD, we find an average value of 3.37, the minimum value being 1.98 and the maximum value being 9.73. Following what we did previously in setting the contrast in our artificial image, we will consider the worst case and take the maximum values. Using the equations of Kaufman (Equations 4.19 and 4.20), we obtain:

$$\sigma_{N,M} \quad = \quad 19.83/1.253 \tag{4.21}$$

$$= \quad 15.82 \tag{4.22}$$

$$\sigma_{N,SD} \quad = \quad 9.73/0.655 \tag{4.23}$$

$$= \quad 14.86 \tag{4.24}$$

where $\sigma_{N,M}$ and $\sigma_{N,SD}$ are estimations of σ_N from the calculation of M and SD respectively. After considering those values, we choose $\sigma_N = 15$.

Estimating the bias B is a challenging task and has been the topic of many papers ([62] and [64] for example). Moreover, the bias can have different forms according to the image and it is not possible to define a unique model which would be representative of every different kind of bias. We decide to take our inspiration from Guillemaud *et al.* [62], who use a sinusoidal bias in the y direction on their artificial image. The period of the sinusoid is set so that the height of the image represents slightly more than one period. Furthermore, Meyer *et al.*[64] indicate that the variation in intensity values due

Figure 4.6: Computer generated image corrupted by noise and bias

to the bias can reach 30%. Therefore, we decide to use the expression:

$$B(x,y) \quad = \quad 1 + 0.15\sin\left(\frac{3\pi y}{200}\right) \tag{4.25}$$

Finally, the computer generated image corrupted with N and B is displayed in Figure 4.6.

4.4.3 Description of the method

Now that we have our corrupted image, we need to find a method to choose the optimal parameters. We define an error function as follows:

$$Error = \sqrt{\frac{1}{S}\sum_{q}(I_{\text{original}}(q) - I_{\text{diffused}}(q))^2} \tag{4.26}$$

S represents the total number of pixels in the image and summation is made on all the pixels q. We choose the set of parameters that minimize the error.

Our error function simply compares the diffused image with the original image. Therefore we will select the parameters that restore the image the closest to the original image. Parameters that do not reduce the noise efficiently will lead to a high value for the error. Similarly, parameters that diffuse the edges will increase the value of the error. The optimal parameters will be the best compromise between noise reduction and edge enhancement

(a) Diffused image with constant parameters, N=50

(b) Diffused image with variable parameters, N=50

Figure 4.7: Results of anisotropic diffusion on computer geenrated image

Our strategy is to choose the optimal values among a finite set of values:

$$\lambda_0 \in \{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\} \tag{4.27}$$

$$\sigma_0 \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\} \tag{4.28}$$

Therefore, we have $7 \times 8 = 56$ possible combinations of parameters. We calculate the error for each of those combinations and we choose the set of parameters giving the lowest error.

4.4.4 Results

We obtain:

$$\lambda_0 = 4 \tag{4.29}$$
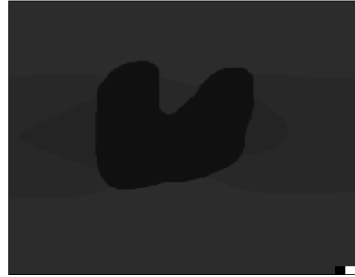
$$\sigma_0 = 1 \tag{4.30}$$

The value of the error E obtained with those parameters is:

$$E = 3.97 \tag{4.31}$$

We have already discussed one advantage of variable λ and σ in section 4.2.2. Another advantage can be shown here.

In Figure 4.7, we compare the computer generated image diffused with constant λ

(a) Diffused image with constant parameters, N=100

(b) Diffused image with variable parameters, N=100

(c) Diffused image with constant parameters, N=200

(d) Diffused image with variable parameters, N=200

(e) Diffused image with constant parameters, N=500

(f) Diffused image with variable parameters, N=500

Figure 4.8: Evolution of the results according to the diffusion time N

and σ^1 and the image diffused with variable λ and σ.

The differences are not visually significant for $N \geq 50$ However, if we apply the algorithm for other values of $N \geq 50$ (Figure 4.8), we observe that with constant parameters, the edges start disappearing for $N \geq 200$ and the image is completely diffused for $N \geq 500$. On the other hand, with variable parameters, the image becomes stable. Therefore, using variable values for λ and σ rather than constant values. removes the constraint of choosing an optimal value for N.

Finally, we apply anisotropic diffusion with variable λ and σ on the four MR images that we mentioned before (Figure 4.9). The images appear less noisy than the original ones. The edges also look sharper. In those conditions, it seems that the segmentation of those images will be easier and more accurate, as we will see in the next section.

4.5 Modification of the segmentation procedure

The preprocessing of MR images of the abdomen with anisotropic diffusion has a radical effect on the histograms of images. In Figure 4.10, we represent the histograms of the same images as we used in the previous chapter (Figure 3.1) after diffusion. We can see that our assumption that the histogram is the sum of two Gaussian distributions cannot be applied to the histogram of the diffused image.

The same observation can be made about the local histograms in adaptive thresholding. Figures 4.11 and 4.12 display the histogram of the same windows as in Figures 3.4 and 3.5 after diffusion.

We decide not to change the global thresholding algorithm. We still use the original image to determine the threshold. However, we use this threshold to classify the pixels of the diffused image. A region growing algorithm is then applied to exclude extra-colonic areas.

The adaptive local thresholding is however completely changed. This time, we work directly on the non-normalized histogram. Our goal is to find a way to choose the local thresholds $T(q)$ for each pixel $q \in B$ as defined in section 3.3. We note that there are usually two or more peaks on the local histograms, the first one being the highest and corresponding to colonic air. Our strategy is to select all the intensity values which

---

[1]We choose $\lambda = 2$ and $\sigma = 0.75$, which are the optimal constant values, using the same method as the one we described in the previous section
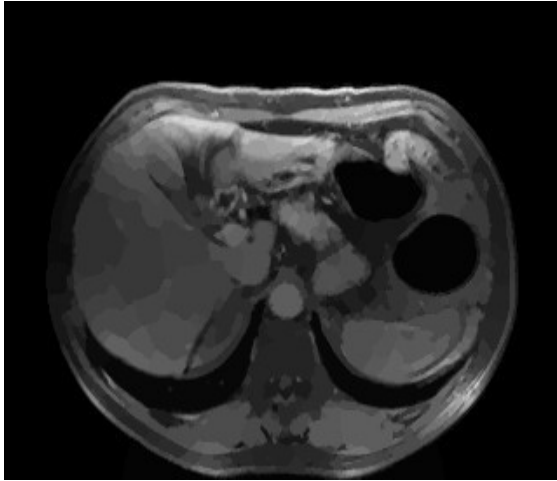
(a) Image 1 diffused
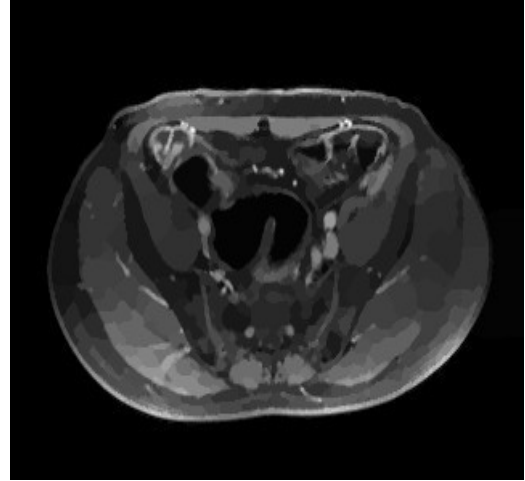
(b) Image 2 diffused
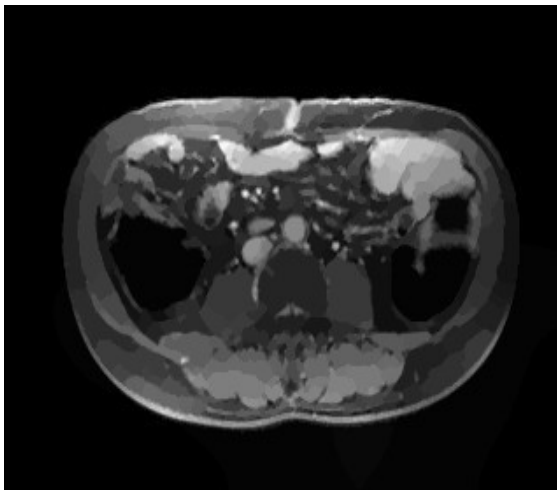
(c) Image 3 diffused
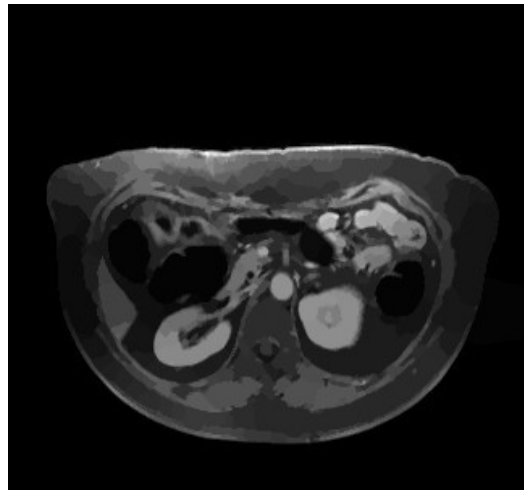
(d) Image 4 diffused

Figure 4.9: MR images after anisotropic diffusion
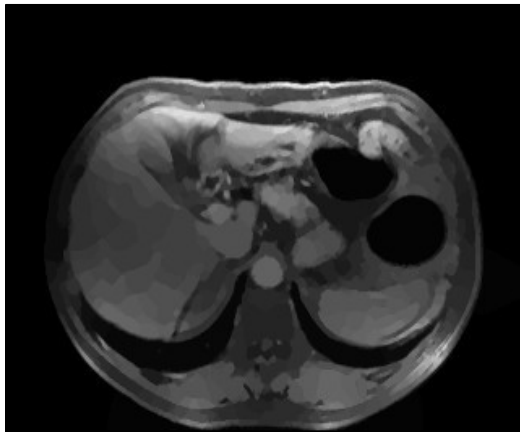
(a) Image 1 diffused
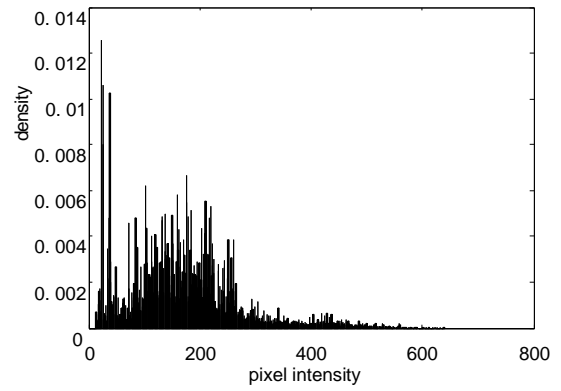
(b) Histogram of image 1 diffused
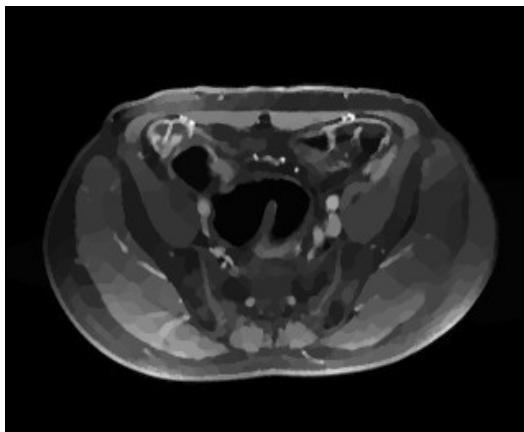
(c) Image 2 diffused
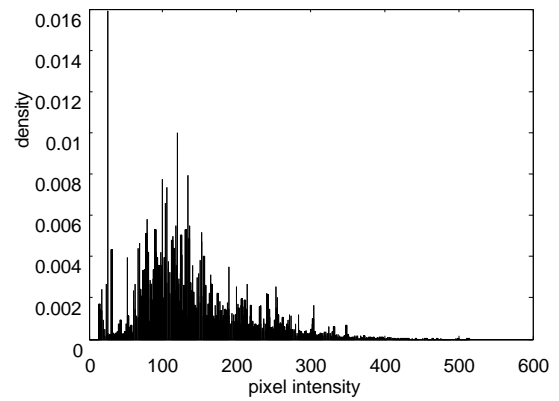
(d) Histogram of image 2 diffused

Figure 4.10: Histogram after anisotropic diffusion
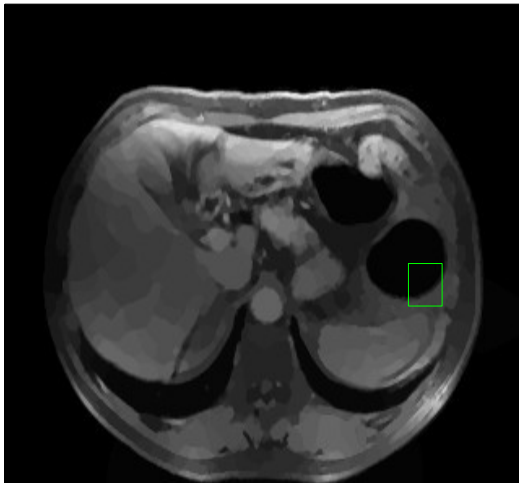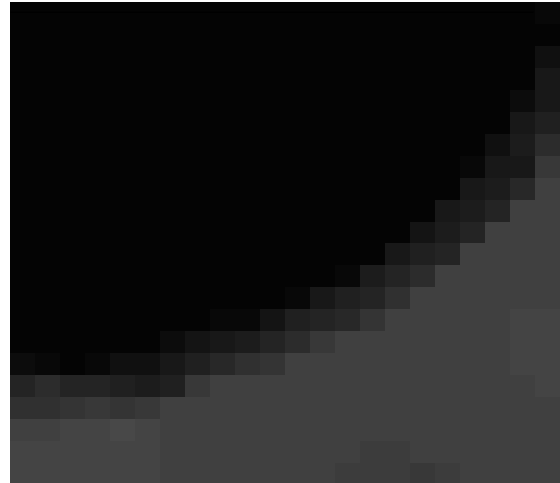
(a) Window on image 1
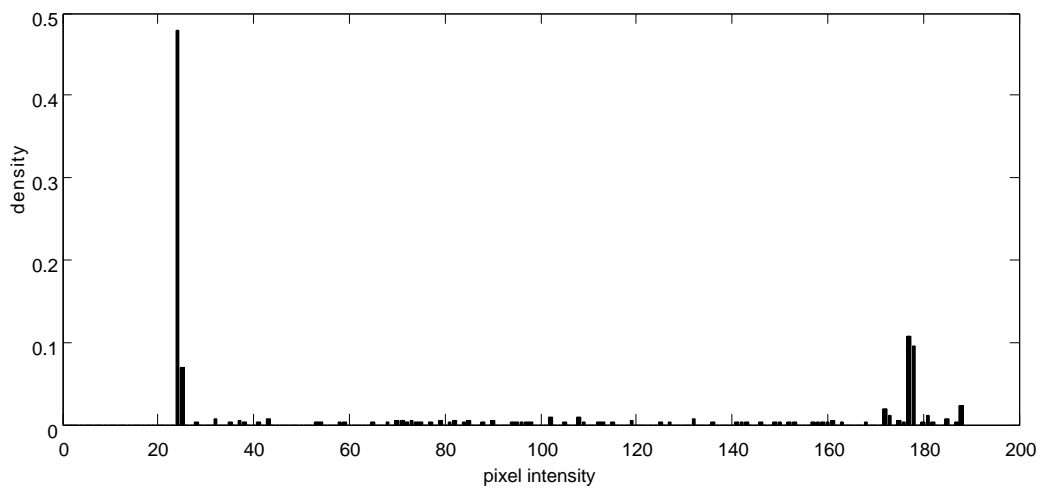
(b) Zoom

(c) Histogram of the window

Figure 4.11: Histogram of a small window located near the colon wall in image 1 after anisotropic diffusion
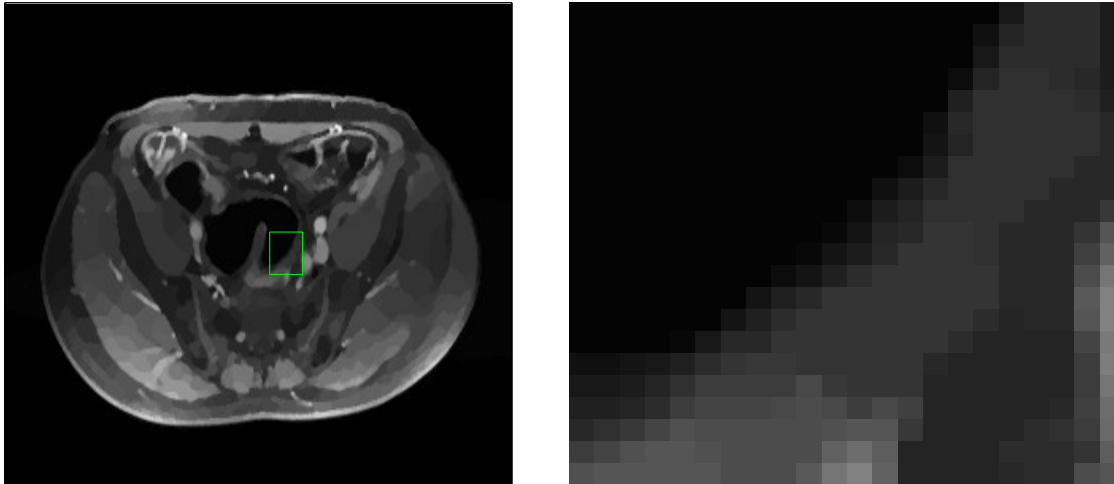
(a) Window on image 2

(b) Zoom
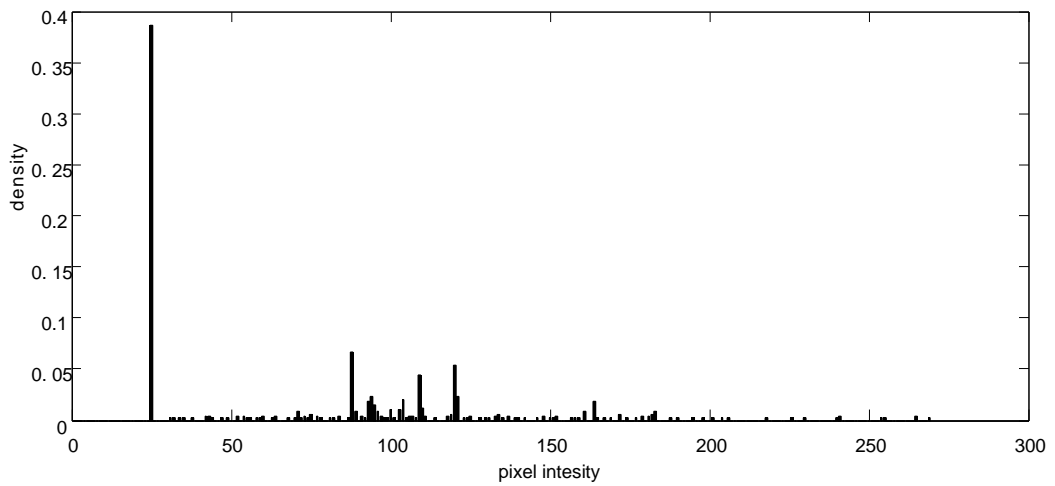
(c) Histogram of the window

Figure 4.12: Histogram of a small window located near the colon wall in image 2 after anisotropic diffusion
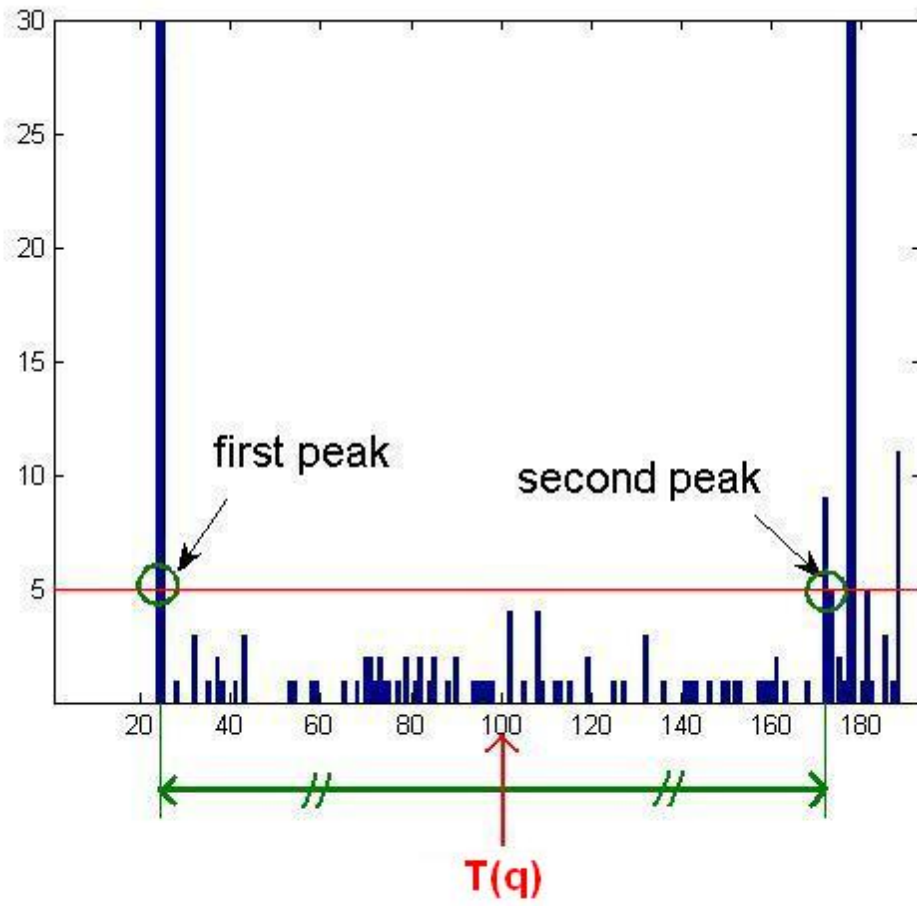
Figure 4.13: Local threshold determination

are represented by 5 pixels or more in the window. We retain the first two values and set $T(q)$ as the middle value between those two values. An example is given in Figure 4.13. On this histogram, the first intensity value represented by more than 5 pixels is 27. The second intensity value represented by more than 5 pixels is 173. Therefore, we choose $T(q) = \frac{27+173}{2} = 100$.

4.6 Results

In Figure 4.14, we present the results of our segmentation procedure with anisotropic diffusion as a preprocessing step. In terms of accuracy, the contour represents accurately the real contour of the colon. Moreover, if we zoom in to the contour, it appears more regular than the contour we obtained in the previous chapter, as it is shown in Figure 4.15.

4.7 Conclusion

Figure 4.16 illustrates the entire segmentation procedure of this chapter. Anisotropic diffusion appears as an attractive method to reduce noise in MR images. The results are better than those we obtained with Gaussian blurring as a preprocessing step. In the next chapter, we will take a completely different approach to segment the colon with the use of deformable models or snakes. The results of the two approaches, thresholding methods and snakes, will then be compared.
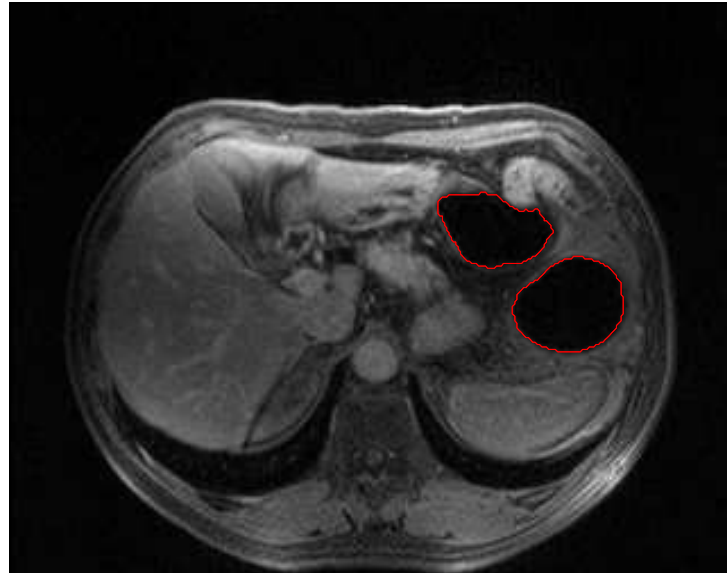
(a) Contour of segmented region in image 1

(b) Contour of segmented region in image 2

Figure 4.14: Contour of segmented region using thresholding methods and anisotropic diffusion
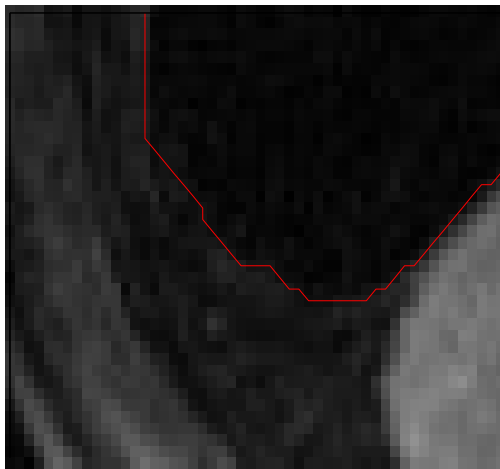
(a) Contour obtained with anisotropic
diffusion
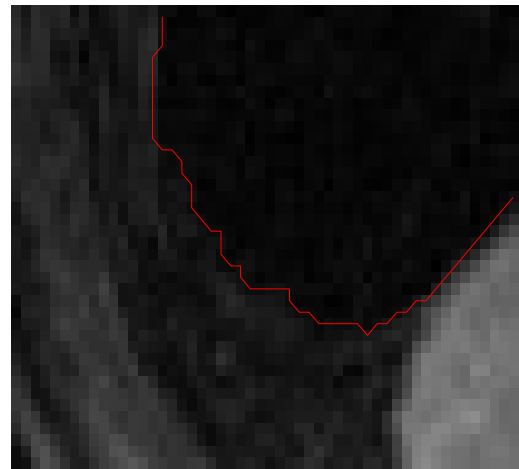
(b) Contour obtained without anisotropic
diffusion
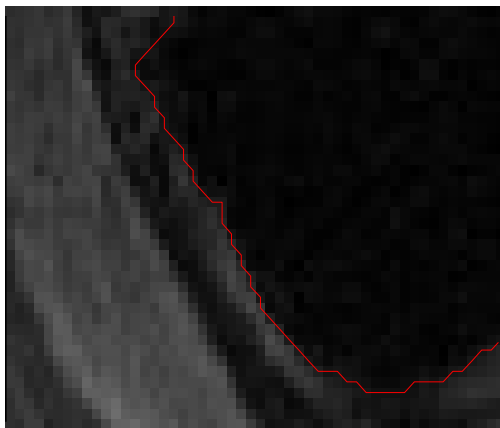
(c) Contour obtained with anisotropic
diffusion
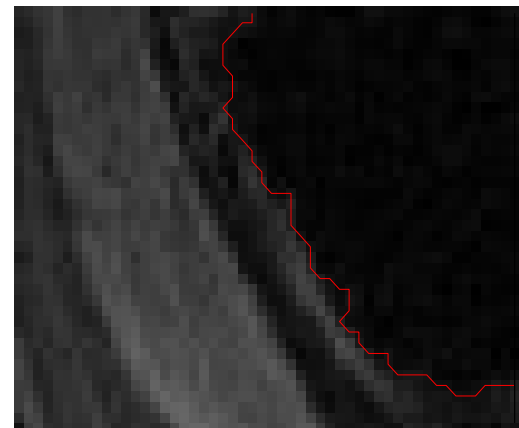
(d) Contour obtained without anisotropic
diffusion
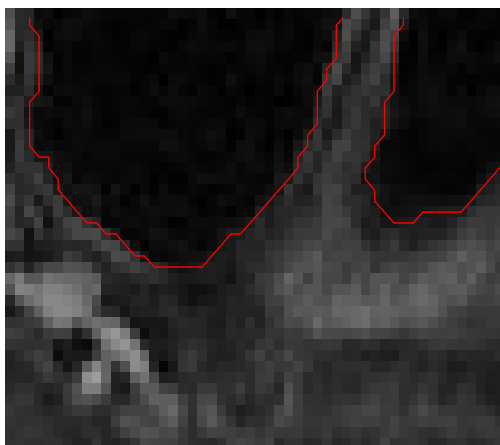
(e) Contour obtained with anisotropic
diffusion
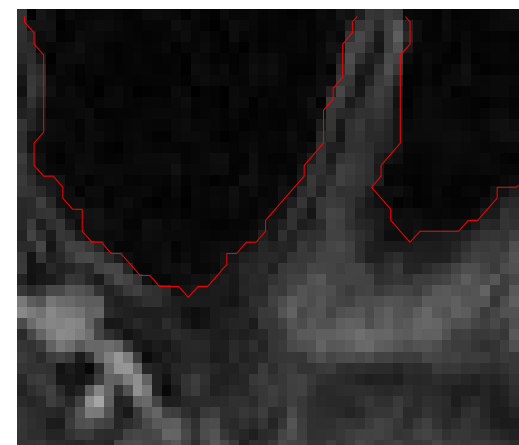
(f) Contour obtained without anisotropic
diffusion

Figure 4.15: Comparison of the contours obtained with different preprocessing methods
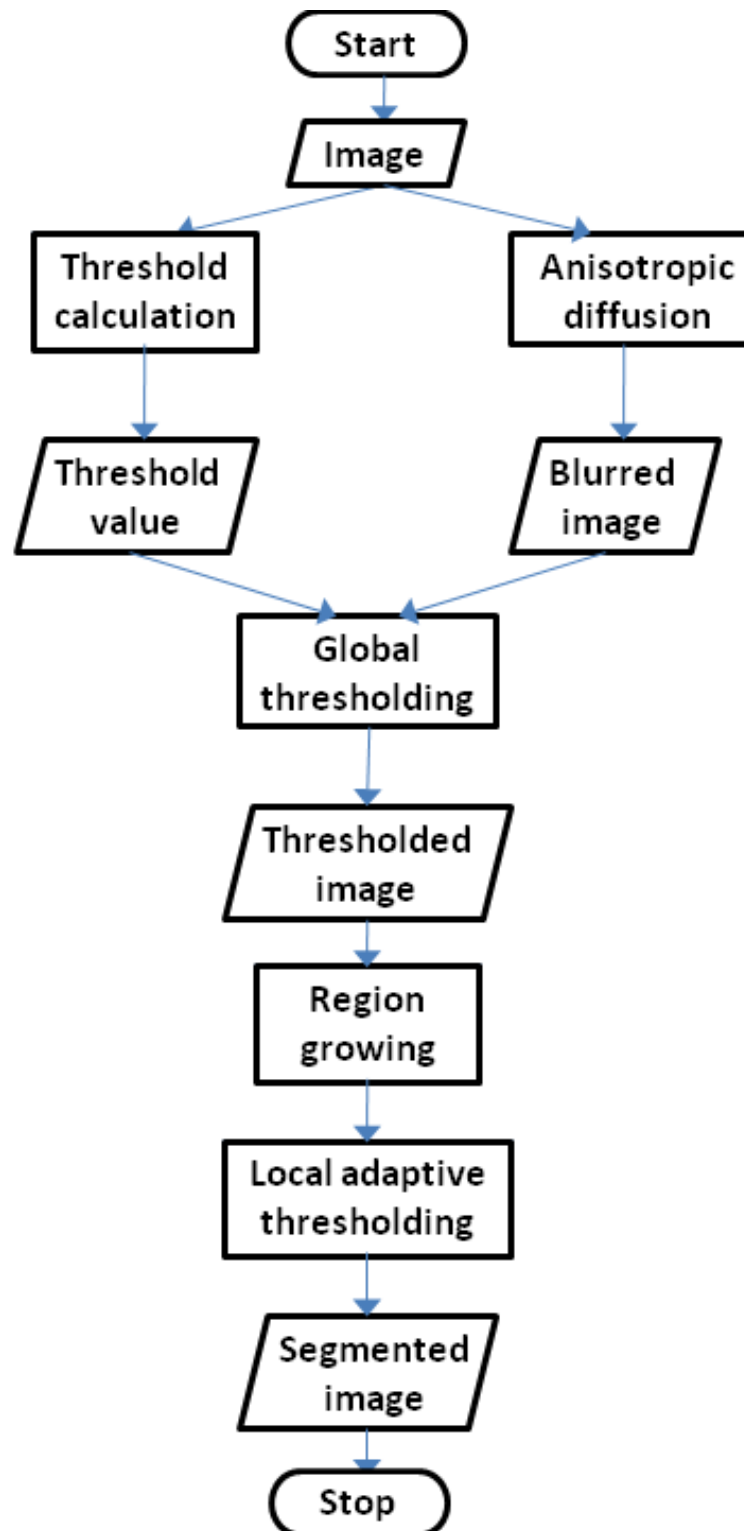
Figure 4.16: Segmentation procedure with thresholding methods and anisotropic diffusion

5. Snakes

5.1 Introduction

In this chapter, we investigate a different approach for segmentation problems. We start from a closed contour, which can be deformed according to the information given by the image, as well as some other constraints defined by the user. The goal is to match this contour with the real contour of the colon. Such a contour is called deformable model or snake.

After presenting the history and the theory of deformable models, we will propose a first implementation based on finite difference approximation. We will show the limitations of this model and present an implementation based on B-snakes. Finally, we will see how those two models are linked to each other and how they may be integrated in a single implementation scheme.

5.2 History

Snakes were first introduced by Kass *et al.* [65] in 1988. they presented the concept of a dynamic curve which could be deformed under the constraints of internal and external forces. Internal forces impose a smoothness constraint on the curve while external forces attract the curve to some features of the image, typically lines or edges. Due to its dynamic behavior, they name the curve a snake.

The first application to medical images appeared in 1990, when Cohen *et al.* [66] used snakes for the segmentation of the left ventricle in ultrasound and MR images. But

the potential of snakes in medical imaging applications was established by McInerney *et al.* in [67]. Since then, snakes have proven to be an efficient method of segmentation, particularly for noisy medical images such as MR images and ultrasound images.

Improvements of the initial model have been a popular topic in the literature since Kass *et al.* first introduced the idea. In their first paper, Cohen *et al.* propose a new implementation based on finite elements. Simultaneously, Menet *et al.* [68] developed a B-spline implementation of snakes, commonly referred to as B-snakes. Both finite elements and B-splines require fewer discretization points and are more accurate than the initial finite difference model proposed by Kass *et al.* (more details about the superiority of B-snakes over finite difference snakes will be given later).

Other models of snakes can be found in the literature. We list here some of the most important developments:

- Staib *et al.* [69] developed a probabilistic snake based on a Fourier decomposition of the boundary;

- Gavrila [70] used a Hermitian deformable model whose implementation is comparable in theory to B-snakes;

- Caselles *et al.* [71] developed a geometric deformable model which can easily handle topological transformations (splitting and merging).

As stated in the introduction, medical image processing should be as automatic as possible. Therefore, an important issue must be addressed, namely, their initialization. Indeed, the first models require the snake to be initialized close to edges to converge. This condition is not practical in medical imaging applications because it would require too much user interaction.

Various efforts have been made to overcome this problem. The first attempts tried to modify the external forces to have a wider range of attraction. An example is the work of Xu *et al.* [72] who created a new external force field called the Gradient Vector Flow (GVF). GVF will be studied in detail later. Another approach to this problem is to develop a complete initialization algorithm.

The history and main challenges of snakes have been presented. We will now study in detail the mathematical foundations.

5.3 Theoretical background

A snake is a 2D curve:

$$[0,1] \longrightarrow \mathbb{R}^2 \tag{5.1}$$

$$s \longmapsto v(s) = (x(s), y(s)) \tag{5.2}$$

This curve moves through the spatial domain of the image in order to minimize the energy functional:

$$\mathcal{A} \longrightarrow \mathbb{R} \tag{5.3}$$

$$v \longmapsto E(v) = \int_0^1 [E_{int}(v(s)) + E_{ext}(v(s))]ds \tag{5.4}$$

$\mathcal{A}$ represents the space of admissible snakes. $E_{int}(v(s))$ is the internal energy associated with the snake. It guarantees the smoothness of the snake. A commonly accepted form for $E_{int}(v(s))$ is:

$$E_{int}(v(s)) = \underbrace{\frac{1}{2}\alpha v'(s)^2}_{\text{membrane energy}} + \underbrace{\frac{1}{2}\beta v''(s)^2}_{\text{thin-plate energy}} \tag{5.5}$$

where $(\alpha, \beta) \in \mathbb{R}^{+2}$ and the primes denote differentiation with respect to s. The membrane energy discourages stretching and discontinuity while the thin-plate energy avoids the bending of the curve. To give more flexibility to the snake, it is common to take into consideration only the membrane energy (by setting $\beta = 0$). Therefore, the snake can become second-order discontinuous and develop corners [65].

$E_{ext}(v(s))$ is the external energy. This energy derives form the image and is meant to attract the snake toward the features we are interested in. Unlike the internal energy, many expressions exist for this term.

The functional (5.5) can be written as

$$E(v) = \int_0^1 [\alpha v'(s)^2 + E_{ext}(v(s))] \, ds \tag{5.6}$$

We denote

$$f(s, v(s), v'(s)) \quad = \quad \frac{1}{2}\alpha v'(s)^2 + E_{ext}(v(s)) \tag{5.7}$$

We know that a necessary condition to minimize $E(v)$ is to solve the Euler-Lagrange equations:

$$\frac{\partial f}{\partial x} \quad = \quad \frac{d}{ds}\left(\frac{\partial f}{\partial x'}\right) \tag{5.8}$$

$$\frac{\partial f}{\partial y} \quad = \quad \frac{d}{ds}\left(\frac{\partial f}{\partial y'}\right) \tag{5.9}$$

The first equation in the x direction leads to

$$\frac{\partial f}{\partial x} \quad = \quad \frac{d}{ds}\left(\frac{\partial f}{\partial x'}\right) \tag{5.10}$$

$$\frac{\partial E_{ext}(v(s))}{\partial x} \quad = \quad \frac{1}{2}\frac{d}{ds}(2\alpha x'(s)) \tag{5.11}$$

$$\frac{\partial E_{ext}(v(s))}{\partial x} \quad = \quad \alpha x''(s) \tag{5.12}$$

$$\alpha x''(s) - \frac{\partial E_{ext}(v(s))}{\partial x} \quad = \quad 0 \tag{5.13}$$

Similarly, in the y direction,

$$\alpha y''(s) - \frac{\partial E_{ext}(v(s))}{\partial y} \quad = \quad 0 \tag{5.14}$$

We make Equations 5.13 and 5.14 dynamic by adding a time variable. We obtain[1]:

$$\frac{\partial x(s,t)}{\partial t} \quad = \quad \alpha\frac{\partial^2 x(s,t)}{\partial s^2} - \frac{\partial E_{ext}}{\partial x} \tag{5.15}$$

$$\frac{\partial y(s,t)}{\partial t} \quad = \quad \alpha\frac{\partial^2 y(s,t)}{\partial s^2} - \frac{\partial E_{ext}}{\partial y} \tag{5.16}$$

The terms $-\frac{\partial E_{ext}}{\partial x}$ and $-\frac{\partial E_{ext}}{\partial y}$ can be seen as the components of an external force $F_{ext} = (F_x, F_y)$ pushing the snake toward the boundary of the colon. Therefore, we denote

$$F_{ext} = -\nabla E_{ext} = \begin{pmatrix} F_x \\ F_y \end{pmatrix} \tag{5.17}$$

---

[1]from now we will write E_{ext} in place of $E_{ext}(v(s,t))$ to simplify the notation

Similarly, $\alpha\frac{\partial^2 x(s,t)}{\partial s^2}$ and $\alpha\frac{\partial^2 y(s,t)}{\partial s^2}$ are the components of an internal force F_{int}. We can write

$$F_{int} = \alpha\frac{\partial^2 v(s,t)}{\partial s^2} \tag{5.18}$$

The theoretical background having been set, we will now see in detail how to implement snakes. This implementation can be divided into three steps:

- Initialization

- Choice of external forces

- Discretization

In the following chapter, we will propose our model for each of those steps.

5.4 Initialization

In the early works on snakes, initialization was not a major focus. This task was usually done manually. The level of interaction between the user and the algorithm was quite high because the initial snake had to be accurate enough for the algorithm to converge.

The common approach to solve this problem is to use a pre-segmentation algorithm. This algorithm is usually very basic since we only require the initial snake to be at a reasonable distance from the edges. Several authors have used this strategy. Medina *et al.* [73] use a Canny edge detector associated with a region growing algorithm. Rahnamayan *et al.* [74] combine global thresholding and morphological operators.

If we refer to Chapter 3 of our work, we have already developed a basic segmentation algorithm. This algorithm is based on global thresholding, region growing and adaptive local thresholding. Since we just need a rough approximation of the contour, the adaptive local thresholding part is not necessary. Figure 5.1 is a flowchart of our initialization procedure.

As stated in Section 3.2, the actions required from the user are:

- Selecting a point inside the colon,

- Selecting a region inside the colon and another region in the gray matter surrounding the colon.

Those actions are simple and do not require much time for execution.
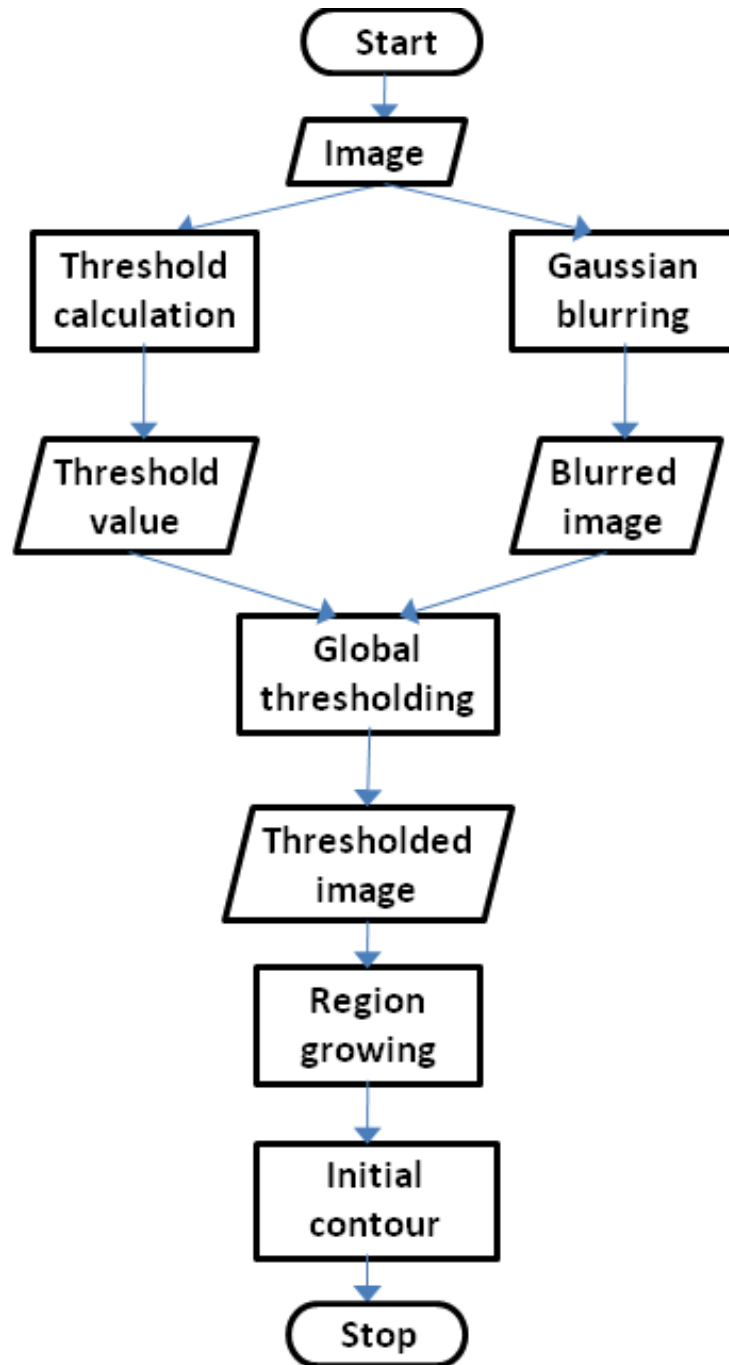
Figure 5.1: Initialization of the snake
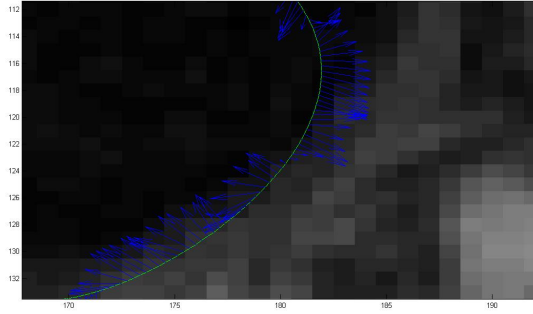
Figure 5.2: External forces applied to a snake

5.5 External force field

5.5.1 First model

The external energy term is meant to attract the snake toward the colon wall. This energy must therefore be minimal at the edges. It seems natural to choose

$$E_{ext} = -||\nabla I||^2 \tag{5.19}$$

The corresponding force F_{ext} can be written as

$$F_{ext} = \nabla||\nabla I||^2 \tag{5.20}$$

In Figure 5.2, we represent this force field applied on a snake[2].

The main problem with the energy defined in Equation 5.19 is that its capture range is very small. In other words, there is a high chance that the snake does not converge toward the edges, even with a very good initialization. Figure 5.3(a) gives an example of such a situation.

Cohen [66] proposes a solution to this problem which consists in normalizing the forces and adding an inflating term. F_{ext} becomes

$$F_{ext} = k_1 \frac{\nabla P}{||\nabla P||} + k_2 \vec{n}(s,t) \tag{5.21}$$

where $P = -||\nabla I||^2$ and $\vec{n}(s,t)$ is the normal vector to the snake at point $v(s,t)$. The role of the term $k_2 \vec{n}(s)$ is to inflate the snake when the gradient values are small and do not

---

[2]what is represented is the interpolation of the force field to discrete points of the snake

(a) Traditional external forces   (b) Cohen's model with inflating force
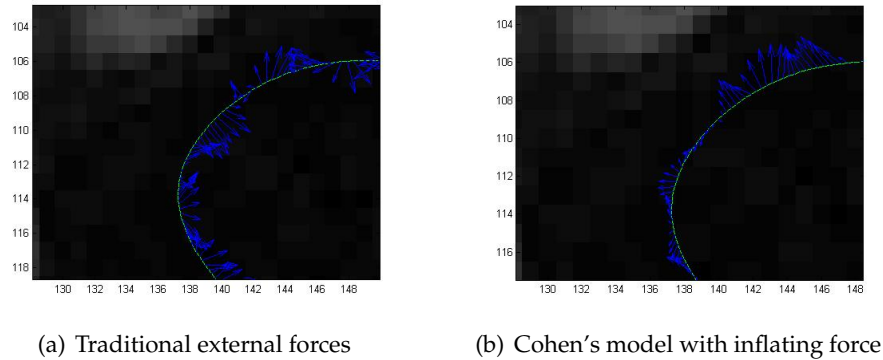
Figure 5.3: Comparison between the traditional model and Cohen's model of external forces
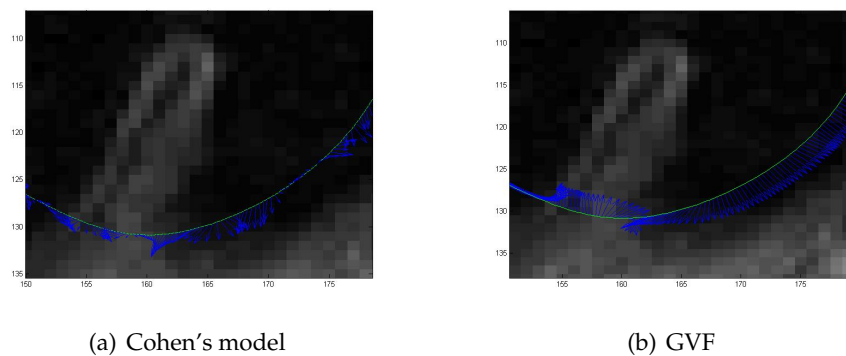
(a) Cohen's model   (b) GVF

Figure 5.4: Comparison between Cohen's model and the GVF near a haustral fold of the colon

give information about the location of the edges (small gradient values are generally due to noise). If we choose $k_1 > k_2$, the gradient force will become predominant as the snake gets close to the edges, allowing the snake to converge. We can see the improvement brought by this force model in Figure 5.3(b).

Cohen's model is powerful in most applications in general. However, it does not give good results if the region to be segmented has concave parts. This is a real problem in our application because the colon has many concave parts called haustral folds. In Figure 5.4(a), we can see that Cohen's force tends to push the snake away from the concave area. This problem can be solved by using another model of external forces called the gradient vector flow (GVF).

5.5.2 GVF

The first step to compute a GVF is to define an edgemap:

$$e = ||\nabla I||^2 \tag{5.22}$$

We notice that this edgemap is the opposite of our first definition of the external energy (Equation 5.19). The GVF is defined as the vector field $F_{ext} = (F_x, F_y)$ which minimizes the functional

$$\mathcal{E} = \iint_I \underbrace{\mu \left(\left(\frac{\partial F_x}{\partial x} \right)^2 + \left(\frac{\partial F_x}{\partial y} \right)^2 + \left(\frac{\partial F_y}{\partial x} \right)^2 + \left(\frac{\partial F_y}{\partial y} \right)^2 \right)}_{1} + \underbrace{|\nabla e|^2 |F_{ext} - \nabla e|^2}_{2} \, dx \, dy$$

When there is no information from the data (∇e small), the functional is dominated by the partial derivatives of the vector field (first term), thus making the field smooth. μ appears as a regularization parameter. The noisier the image is, the higher μ should be. On the other hand, when ∇e is high, the second term dominates the functional and is minimized by taking $F = \nabla e$. Then, the vector field points toward the high gradients of the image, including the edges of the colon. Similarly to the snake functional, the minimization of the the function $\mathcal{E}$ can be done by solving the Euler Lagrange equations. Details of the implementation of the GVF can be found in Appendix A. In Figure 5.4(b), we show the GVF on a portion of snake near a haustral fold. We observe that unlike Cohen's force, the GVF attracts the snake inside the concavity.

5.6 Discretization with finite differences

The most natural way to discretize Equations 5.15 and 5.16 is the finite difference method. As we discussed before in the chapter on anisotropic diffusion, the discretization must be done both in space and time. From now we will focus on Equation 5.15 as the arguments are exactly the same for Equation 5.16. We start with the discretization in the space domain. We use the notation $x_i(t) = x(hi, t)$ where h is a space step. Therefore, we have

$$\frac{\partial x_i(t)}{\partial t} = \alpha \frac{x_{i+1}(t) - 2x_i(t) + x_{i-1}(t)}{2h^2} + F_{x_i}(t) \tag{5.23}$$

Then, we discretize Equation 5.23 in the time domain. We define $\gamma = 1/\tau$ as the inverse of the time step. We note $x_i(t_k) = x_i(k\tau)$. We use an implicit scheme for the internal force term and an explicit scheme for the external force term as follows:

$$\gamma\left(x_i(k+1) - x_i(k)\right) \quad = \quad \alpha \frac{x_{i+1}(k+1) - 2x_i(k+1) + x_{i-1}(k+1)}{2h^2} + F_{x_i}(k)$$

This equation can be written in matrix form. Assuming that we have $n+1$ discretization points $(x_0 \dots x_n)$, we define a vector $\mathbf{X}$ as:

$$\mathbf{X} = \begin{bmatrix} x_0 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} \tag{5.24}$$

Similarly, we define $\mathbf{F}_X$ as:

$$\mathbf{F}_X = \begin{bmatrix} F_{x_0} \\ \vdots \\ F_{x_i} \\ \vdots \\ F_{x_n} \end{bmatrix} \tag{5.25}$$

We introduce a matrix $\mathbf{A}$ of size $((n+1) \times (n+1))$:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 0 & \dots & \dots & 0 & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 0 & \dots & \dots & \dots & 1 & -2 \end{bmatrix} \tag{5.26}$$

Therefore, we have:

$$\gamma\left(\mathbf{X}(k+1) - \mathbf{X}(k)\right) \quad = \quad \alpha\mathbf{A}\mathbf{X}(k+1) + \mathbf{F}_X(k) \tag{5.27}$$

$$(\gamma\mathbf{I} - \alpha\mathbf{A})\mathbf{X}(k+1) \quad = \quad \gamma\mathbf{X}(k) + \mathbf{F}_X(k) \tag{5.28}$$

$$\mathbf{X}(k+1) \quad = \quad (\gamma\mathbf{I} - \alpha\mathbf{A})^{-1}(\gamma\mathbf{X}(k) + \mathbf{F}_X(k)) \tag{5.29}$$

Despite the simplicity of the implementation, we found that this method was not optimal for several reasons:

- First, the spatial discretization requires a small space step to conserve the smoothness of the snake. This condition leads to an important number of discretization points, increasing the computation time.

- Then, in our construction, the distance between the discretization points is the same for every point [3]. Two problems may be encountered:

  - If the colon wall has a complicated shape at some place, there is a possibility that the number of points is locally insufficient to represent the shape accurately.

  - Similarly if the colon wall has a regular shape at some place, there might be an over-representation of the shape.

  In other words, the number of discretization points does not adjust to the local shape properties of the colon wall.

Considering those limitations of the finite difference model, we investigate another implementation method called B-snakes.

5.7 B-snakes

5.7.1 Theoretical background

A B-spline is a linear combination of polynomial functions with finite support called splines. Given $m+1$ values $\underline{s}_i \in [0, 1]$ called knots, a two-dimensional B-spline of degree n is a parametric curve:

$$[\underline{s}_0, \underline{s}_m] \longrightarrow \mathbb{R}^2 \tag{5.30}$$

$$\underline{s} \longmapsto v(\underline{s}) = \sum_{i=0}^{m-n-1} P_i b_{i,n}(\underline{s}) \tag{5.31}$$

---

[3] the notion of distance must be here considered for the parameter s. This distance is not the euclidean distance in the 2D space of the image

where P_i are called control points ($P_i \in \mathbb{R}^2 \quad \forall i \in [0, m - n - 1]$) and $b_{i,n}$ are defined recursively as

$$b_{i,0}(\underline{s}) \;=\; \begin{cases} 1 & \text{if} & \underline{s}_i \le \underline{s} < \underline{s}_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{5.32}$$

$$b_{i,n}(\underline{s}) \;=\; \frac{\underline{s} - \underline{s}_i}{\underline{s}_{i+n} - \underline{s}_i} b_{i,n-1}(\underline{s}) + \frac{\underline{s}_{i+n+1} - \underline{s}}{\underline{s}_{i+n+1} - \underline{s}_{i+1}} b_{i+1,n-1}(\underline{s}) \tag{5.33}$$

We consider the particular case of uniform B-splines where $\underline{s}_0 = 0$, $\underline{s}_{m+1} = 1$ and $\underline{s}_{i+1} - \underline{s}_i = \Delta\underline{s} = $ constant.

To better understand the concept of B-splines, let us first consider a B-spline of degree 1. In this case, there are $m - 1$ control points ($P_0 \ldots P_{m-2}$). $v(\underline{s})$ can be written as

$$v(\underline{s}) \;=\; \sum_{i=0}^{m-2} P_i b_{i,1}(\underline{s}) \tag{5.34}$$

From the definition of the basis functions, we have $\forall i \in \{0 \ldots m - 2\}$,

$$b_{i,1}(\underline{s}) \;=\; \begin{cases} \frac{\underline{s} - \underline{s}_i}{\Delta\underline{s}} & \text{if} & \underline{s}_i \le \underline{s} < \underline{s}_{i+1} \\ \frac{\underline{s}_{i+2} - \underline{s}}{\Delta\underline{s}} & \text{if} & \underline{s}_{i+1} \le \underline{s} < \underline{s}_{i+2} \\ 0 & \text{otherwise} \end{cases} \tag{5.35}$$

If we consider a particular $\underline{s} \in [\underline{s}_i, \underline{s}_{i+1}]$, $i \in \{1 \ldots m - 2\}$, we notice that all the basis functions are equal to 0 except $b_{i,1}$ and $b_{i-1,1}$. Consequently, $v_i(\underline{s})$, which is the restriction of $v(\underline{s})$ to $[\underline{s}_i, \underline{s}_{i+1}]$ can be written as

$$v_i(\underline{s}) \;=\; P_{i-1} b_{i-1,1}(\underline{s}) + P_i b_{i,1}(\underline{s}) \tag{5.36}$$

$$v_i(\underline{s}) \;=\; P_{i-1} \frac{\underline{s}_{i+1} - \underline{s}}{\Delta\underline{s}} + P_i \frac{\underline{s} - \underline{s}_i}{\Delta\underline{s}} \tag{5.37}$$

At this point we introduce a new variable $s = \frac{\underline{s}_{i+1} - \underline{s}}{\Delta\underline{s}}$. A rapid calculation shows that $\frac{\underline{s} - \underline{s}_i}{\Delta\underline{s}} = 1 - s$. If we write v_i according to variable s, we obtain

$$v_i(s) \;=\; P_{i-1}s + P_i(1 - s) \quad \forall s \in [0, 1] \tag{5.38}$$

We recognize this as the equation of a straight line joining P_{i-1} and P_i. It can be easily shown that in the cases $i = 0$ and $i = m - 1$, v_i is a line joining $(0, 0)$ and P_0, and P_{m-1} and $(0, 0)$, respectively. Finally, v appears as a broken line joining all the control points and the origin. Therefore, v is continuous (and even C^∞) everywhere except at the knots. Moreover, by construction, if we move one control point, the shape of the curve only changes locally (only the two lines linking the control point to its two neighbours change).

If we want to work with a closed curve (which is what we will need to do), we simply need to use the sequence of control points $(P_1 \ldots P_{m-1}P_1)$ and we restrict the parameter $\underline{s}$ to $[\underline{s}_1, \underline{s}_{m-1}]$.

Having shown the main properties of a B-spline of degree 1, it is easy to show that in the general case, a B-spline of degree n:

- is a polynomial function of degree n and is therefore C^∞ between knots,

- is C^{n-1} at the knots,

- only changes locally when a control point is moved,

- can be extended to a closed curve by using the sequence of control points $(P_1 \ldots P_{m-n-1}P_1 \ldots P_n)$ and parameter $\underline{s}$ restricted to $[\underline{s}_n, \underline{s}_{m-1}]$.

Those properties make B-splines a perfect tool for segmentation applications where both regularity and flexibility of the curve are required. The possibility of a local control of the curve is also a great advantage as it allows the curve to adjust to local properties of the image. B-splines used as snakes in segmentation applications are often referred to as B-snakes.

In our application, we choose a B-spline of dimension 3 or cubic B-spline. Using the same change of variables as before, it can easily be shown that on each interval $[\underline{s}_i, \underline{s}_{i+1}]$, v can be written in matrix form:

$$
v_i(s) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{6} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ \frac{-1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}
\tag{5.39}
$$

5.7.2 Discretization

With the matrix formulation in Equation 5.39, the discretization of the B-snake can be done by sampling the interval $[0, 1]$. We use a sampling step h and we note $s_j = jh$. Let $\mathbf{V}_i$ be a matrix such as $\mathbf{V}_i(j) = v_i(s_j)$. If we assume that there are p sampling points, $\mathbf{V}_i$ is of size $(p \times 2)$. Then, we have $\forall i \in \{1 \ldots m - 3\}$[4]:

$$
\mathbf{V}_i =
\begin{bmatrix}
0 & 0 & 0 & 1 \\
\vdots & \vdots & \vdots & \vdots \\
s_j^3 & s_j^2 & s_j & 1 \\
\vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
\frac{-1}{6} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{6} \\
\frac{1}{2} & -1 & \frac{1}{2} & 0 \\
\frac{-1}{2} & 0 & \frac{1}{2} & 0 \\
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0
\end{bmatrix}
\begin{bmatrix}
P_{i-1} \\
P_i \\
P_{i+1} \\
P_{i+2}
\end{bmatrix}
\tag{5.40}
$$

We simplify the notation by defining:

$$
\mathbf{H} =
\begin{bmatrix}
0 & 0 & 0 & 1 \\
\vdots & \vdots & \vdots & \vdots \\
s_j^3 & s_j^2 & s_j & 1 \\
\vdots & \vdots & \vdots & \vdots \\
1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
\frac{-1}{6} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{6} \\
\frac{1}{2} & -1 & \frac{1}{2} & 0 \\
\frac{-1}{2} & 0 & \frac{1}{2} & 0 \\
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0
\end{bmatrix}
\tag{5.41}
$$

$\mathbf{H}$ is of size $(p \times 4)$. $\mathbf{V}_i$ can then be written:

$$
\mathbf{V}_i = \mathbf{H}
\begin{bmatrix}
P_{i-1} \\
P_i \\
P_{i+1} \\
P_{i+2}
\end{bmatrix}
\tag{5.42}
$$

---

[4]we assume that $P_{m-3} = P_0$, $P_{m-2} = P_1$ and $P_{m-1} = P_2$

we can go further and write the whole B-snake in a single matrix formulation. We first define the following vectors and matrices:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_i \\ \vdots \\ \mathbf{V}_{m-3} \end{bmatrix} \tag{5.43}$$

$\mathbf{V}$ is of size $(p(m-3) \times 2)$. We write the matrix $\mathbf{H}$ defined in (5.41) as $\mathbf{H} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \mathbf{H}_3 \ \mathbf{H}_4]$ where each $\mathbf{H}_k \ k \in \{1 \dots 4\}$ is of size $(p \times 1)$. We define a new matrix $\mathbf{M}$:

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 & \dots & \dots & \dots & \dots \\ 0 & \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \dots \\ \mathbf{H}_4 & \dots & \dots & \dots & \dots & \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 \\ \mathbf{H}_3 & \mathbf{H}_4 & \dots & \dots & \dots & \dots & \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 & \dots & \dots & \dots & \dots & \mathbf{H}_1 \end{bmatrix} \tag{5.44}$$

We set $\mathbf{M}$ so that its size is $(p(m-3) \times (m-3))$. Finally we define the vector $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} P_0 \\ \vdots \\ P_i \\ \vdots \\ P_{m-4} \end{bmatrix} \tag{5.45}$$

which is simply the vector of control points of size $((m-3) \times 2)$. The set of discrete points $\mathbf{V}$ representing the B-snake can be easily written as

$$\mathbf{V} \quad = \quad \mathbf{M}\mathbf{P} \tag{5.46}$$

From a set of $m-3$ points control points, we are able to reconstruct a curve with $p(m-3)$ discretization points. We recall that for the finite difference methods, we needed to record all the discretization points in order to reconstruct the curve (no difference was

made between control points and discretization points). Therefore, the assumption that the contour of the colon can be approximated by polynomial functions of degree 3 linked together in a C^2 way allows us to reduce the quantity of information that needs to be recorded to define a particular snake. If we manage to find a way to interact directly with those control points to deform the snake, we could reduce the computational load. This is exactly what we will do in the next section.

5.7.3 Deformation of the B-snake

As we mentioned earlier, the snake is deformed under the influence of internal forces and external forces. We will see how those forces can be applied directly to the control points.

First, we recall the expression of internal forces (derived for the membrane energy):

$$F_{int} \quad = \quad \alpha \frac{\partial^2 v(s,t)}{\partial s^2} \tag{5.47}$$

This force uses the second derivative of v with respect to the variable s. Since the B-snake that we created in Section 5.7.1 is at worst C^2, the second derivative exists at every point on the B-snake. More precisely, we realize that if we use the following matrix:

$$\mathbf{H}'' = \begin{bmatrix} 0 & 1 \\ \vdots & \vdots \\ s_j & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 1 & -2 & 1 & 0 \end{bmatrix} \tag{5.48}$$

in place of $\mathbf{H}$ in Subsection 5.7.2, we can construct a matrix $\mathbf{M}''$ following the same principle as in Equation 5.44 to finally obtain a good approximation of the second derivative of the B-snake with the following expression:

$$\mathbf{V}'' = \mathbf{M}''\mathbf{P} \tag{5.49}$$

The discretized internal force can be written as

$$\mathbf{F}_{int} = \alpha\mathbf{M}''\mathbf{P} \tag{5.50}$$

We focus now on the external force. We assume that we have already computed an external force field such as GVF. We simply interpolate this force field on each discretized point of the B-snake to create a $(p(m-3) \times 2)$ vector $\mathbf{F}_{ext}$.

Finally, the equation of deformation of the snake discretized in the spatial domain can be written as

$$\mathbf{M}\frac{\partial \mathbf{P}}{\partial t} = \alpha \mathbf{M}''\mathbf{P} + \mathbf{F}_{ext} \tag{5.51}$$

For the discretization in the time domain, we use a semi-implicit method similar to the one we used in the finite difference model. We obtain

$$\gamma \mathbf{M}\left(\mathbf{P}(k+1) - \mathbf{P}(k)\right) = \alpha \mathbf{M}''\mathbf{P}(k+1) + \mathbf{F}_{ext} \tag{5.52}$$

$$(\gamma \mathbf{M} - \alpha \mathbf{M}'')P(k+1) = \gamma \mathbf{M}\mathbf{P}(k) + F_{ext} \tag{5.53}$$

For reasons that will become clearer in the next chapter, we would like to find an expression for $\Delta \mathbf{P}(k) = \mathbf{P}(k+1) - \mathbf{P}(k)$. From Equation 5.53, we easily obtain:

$$\gamma \mathbf{M}\left(\mathbf{P}(k+1) - \mathbf{P}(k)\right) = \alpha \mathbf{M}''\mathbf{P}(k+1) + \mathbf{F}_{ext} \tag{5.54}$$

$$(\gamma \mathbf{M} - \alpha \mathbf{M}'')(\mathbf{P}(k+1) - \mathbf{P}(k)) = \gamma \mathbf{M}\mathbf{P}(k) + \mathbf{F}_{ext} - (\gamma \mathbf{M} + \alpha \mathbf{M}'')\mathbf{P}(k)$$

$$(\gamma \mathbf{M} - \alpha \mathbf{M}'')\Delta \mathbf{P}(k) = \alpha \mathbf{M}''\mathbf{P}(k) + \mathbf{F}_{ext} \tag{5.55}$$

The matrix $\mathbb{M} = \gamma \mathbf{M} - \alpha \mathbf{M}''$ is of dimension $(p(m-3) \times (m-3))$. Therefore, it is not a square matrix and Equation 5.55 is not directly invertible. To solve this problem, we use the Moore-Penrose pseudo inverse defined as:

$$\mathbb{M}^+ = (\mathbb{M}^T \mathbb{M})^{-1}\mathbb{M}^T \tag{5.56}$$

Now we can solve Equation 5.55:

$$\mathbb{M}\Delta \mathbf{P}(k) = \alpha \mathbf{M}''\mathbf{P}(k) + \mathbf{F}_{ext} \tag{5.57}$$

$$\mathbb{M}^T \mathbb{M}\Delta \mathbf{P}(k) = \mathbb{M}^T(\alpha \mathbf{M}''\mathbf{P}(k) + \mathbf{F}_{ext}) \tag{5.58}$$

$$\Delta \mathbf{P}(k) = (\mathbb{M}^T \mathbb{M})^{-1}\mathbb{M}^T(\alpha \mathbf{M}''\mathbf{P}(k) + \mathbf{F}_{ext}) \tag{5.59}$$

$$\Delta \mathbf{P}(k) = \mathbb{M}^+(\alpha \mathbf{M}''\mathbf{P}(k) + \mathbf{F}_{ext}) \tag{5.60}$$

5.7.4 Stopping criterion and control point insertion

Until now, we have omitted an important aspect when implementing a dynamic model: the stopping criterion. The most natural idea would be to stop the algorithm when the variation $\Delta \mathbf{P}$ becomes small. As we saw before, $\Delta \mathbf{P}(k)$ is a $((m-3) \times 2)$ matrix which represents the displacement of $\mathbf{P}$ at the time k. We can compute a norm for this matrix as

$$\|\Delta \mathbf{P}(k)\| \quad = \quad \max_{i \in \{0...m-4\}} \sqrt{(\Delta \mathbf{P}(k))^2_{i,1} + (\Delta \mathbf{P}(k))^2_{i,2}} \tag{5.61}$$

With this expression, we simply calculate the largest displacement among all the control points. If we set a value $\delta_1 > 0$, we can decide to stop the algorithm at time k_f which verifies:

$$\|\Delta \mathbf{P}(k_f)\| \quad < \quad \delta_1 \tag{5.62}$$

$$\|\Delta \mathbf{P}\| \quad < \quad \delta_1 \tag{5.63}$$

$$\|\mathbf{F}\| \quad < \quad \delta_2 \tag{5.64}$$

$$\mathbf{P} \quad = \quad \mathbf{P} + \Delta \mathbf{P} \tag{5.65}$$

However, there might be situations where a small displacement does not mean that the forces applied to the snake are weak. For example, it is possible that the forces along a portion of the snake between two control points compensate each other, resulting in a small displacement. In such a situation, one might want to have more control points to better estimate the forces at this location.

We develop a strategy to insert a control point at the location where the forces are the strongest. We define a norm for the term $\mathbf{F}(k) = \alpha \mathbf{M}''\mathbf{P}(k) + \mathbf{F}_{ext}$ which is similar to the norm for the displacement matrix:

$$\|\mathbf{F}(k)\| \quad = \quad \max_{j \in \{0...pm\}} \sqrt{(\Delta \mathbf{F}(k))^2_{j,1} + (\Delta \mathbf{F}(k))^2_{j,2}} \tag{5.66}$$

Then, we define a value $\delta_2 > 0$. If $\|\mathbf{F}(k)\| \geq \delta_2$, a control point is inserted where the force is the strongest.

We still need to indicate how to insert a control point exactly. The goal is to insert the
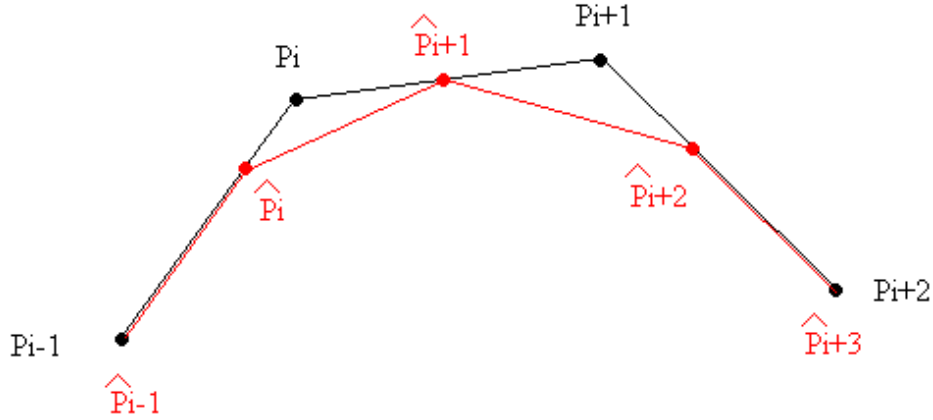
Figure 5.5: Control point insertion

point without changing the shape of the snake. Let us imagine that we have a sequence of $m - 3$ control points $\mathbf{P}$ and that the maximal force has been found on the curve controlled by points $(P_{i-1}, P_i, P_{i+1}, P_{i+2})$ at the parameter s. We create a new sequence of $m - 2$ control points $\widehat{\mathbf{P}}$ so that

$$\forall j \in 0 \ldots m - 2, \quad \widehat{P}_j = (1 - \alpha_j)P_{j-1} + \alpha_j P_j \tag{5.67}$$

$$\tag{5.68}$$

where α_j is given by

$$\alpha_j = \begin{cases} 0 & \text{for} & j \leq i - 1 \\ \frac{s+2-(i-j)}{3} & \text{for} & i \leq j \leq i + 2 \\ 1 & \text{for} & j \geq i + 3 \end{cases} \tag{5.69}$$

Figure 5.5 illustrates the control points insertion strategy.

5.8 Comparison of the two models

We have presented the two models (finite difference and B-snake) in a classical way, which makes them difficult to compare at first sight. Following a similar approach to Terzopoulos *et al.* [51], we will try to unify the two models in the same implementation.

More precisely, we will show how the finite difference model can be interpreted as a particular B-snake.

The finite difference snake model can be seen as a B-snake of degree 1. Indeed, the finite difference model uses a sequence of discretization points which are linked together with straight lines. As we mentioned before, the discretization points corresponds to the control points. By analogy with the B-snake, the finite difference snake can be written as

$$\mathbf{V} \quad = \quad \mathbf{MP} \tag{5.70}$$

M being the identity matrix $\mathbf{I}_{m-1}$ Obviously, unlike the cubic B-spline, the finite difference snake is not differentiable. Therefore, the discretization of the internal force $F_{int}(s) = \alpha \frac{\partial^2 v(s,t)}{\partial t^2}$ is not straightforward. However, the only thing that we need to do is to find a matrix $\mathbf{M}''$ which would act as a second derivative operator by analogy with the B-snake implementation. A suitable matrix is the one we defined in Equation 5.26. We simply change the dimensions for the analogy. So:

$$\mathbf{M}'' = \begin{bmatrix} -2 & 1 & 0 & \dots & \dots & 0 & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 0 & \dots & \dots & \dots & 1 & -2 \end{bmatrix} \tag{5.71}$$

$\mathbf{M}$ and $\mathbf{M}''$ having been defined, the snake deformation equation is exactly the same as the equation for the B-snake:

$$\Delta \mathbf{P} \quad = \quad \mathbb{M}^+(\alpha \mathbf{M}'' \mathbf{P}(k) + \mathbf{F}_{ext}) \tag{5.72}$$

We now compare the two models in terms of computational load. Let $m_{FD} - 1$ be the number of control points for the finite difference snake and $m_{BS} - 3$ be the number of control points for the B-snake. Let p be the number of sampling points in each interval $[0, 1]$ as defined before for the B-snake. In the two cases, the pseudo inverse A^+ can be

calculated beforehand. In the finite element case, the complexity is equal to:

$$C_{FD} = 2(m_{FD} - 1)^2 + (m_{FD} - 1) + 2(m_{FD} - 2)^2 \tag{5.73}$$

$$C_{FD} \sim 4m_{FD}^2 \tag{5.74}$$

For the B-snake model,

$$C_{BS} = 2p(m_{BS} - 3)^2 + p(m_{BS} - 3) + 2p(m_{BS} - 3)^2 \tag{5.75}$$

$$C_{BS} \sim 4pm_{BS}^2 \tag{5.76}$$

As stated before, a B-snake requires fewer control points than a finite difference snake. More precisely, if we want a equivalent representation of the two models, we should take $m_{FD} \sim pm_{BS}$. C_{BS} can be expressed according to m_{FD}:

$$C_{BS} \sim 4p\frac{m_{FD}^2}{p^2} \tag{5.77}$$

$$C_{BS} \sim \frac{4m_{FD}^2}{p} \tag{5.78}$$

A common value for p is $p \approx 10$. We can see that the computational complexity can be reduced by a factor 10 with the B-snake model.

At this point, since we have transformed the finite difference model in a B-snake model, we could also generalize the stopping criterion developed in the previous chapter. There would be no problem to generalize the first part, namely the evaluation of $\|\Delta \mathbf{P}(k)\|$. However, the control point insertion would have no sense for the finite difference model. Indeed, in this model, since control points and discretization points are the same, we do not measure the force between the control points as for the B-snake model. Therefore, a situation as the one we mentioned in Subsection 5.7.4 (a compensation of forces yielding to a small displacement) can not happen. This is one of the reasons why B-snakes are more flexible and accurate than finite difference snakes.

5.9 Conclusion

In this chapter, a complete procedure of segmentation based on snakes has been presented. The theory about snakes has been explained in details and a correspondence
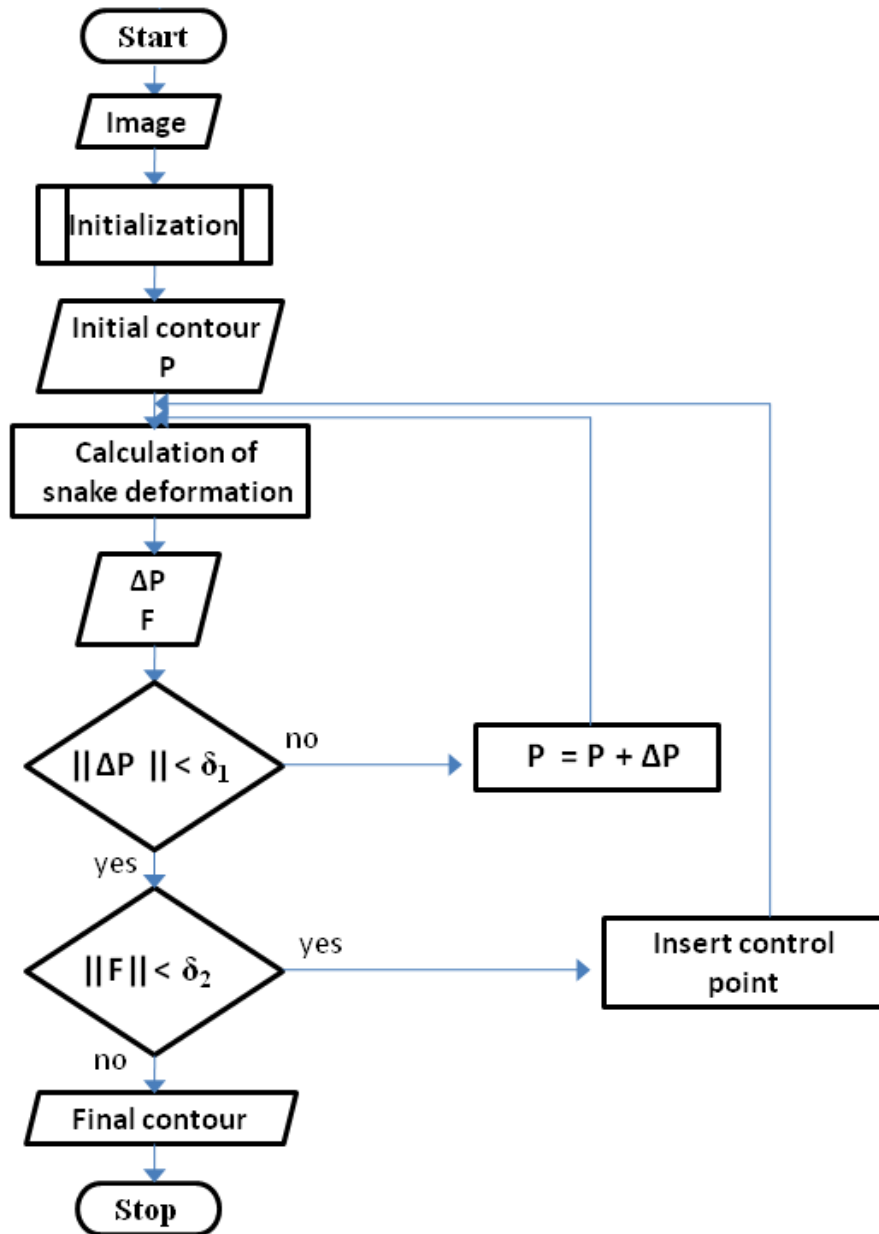
Figure 5.6: Segmentation procedure with snakes

between the traditional finite difference implementation model and B-snakes has been established. Figure 5.6 summarizes the entire snake algorithm. The reader can refer to Section 5.4 and Subsection 5.7.4 for more details about the initialization and the stopping algorithm test respectively.

Based on the initial results that we have presented, snakes seem to give a smoother contour than the thresholding methods presented before. In the next chapter, we will compare in detail the two methods on two sets of 2D MR images of the colon.

6. Results and discussion

6.1 Presentation of the data

The full procedure for a complete MR colonography examination comprises several acquisitions:

- T2-weighted

- T2-weighted with colon inflation (the colon remain inflated for all the following acquisitions)

- T1-weighted in coronal view

- T1-weighted in coronal view with contrast agent (the contrast agent remains for all the following acquisitions)

- T1-weighted in axial view, patient in prone position

- T1-weighted in axial view, patient in supine position

The T1-weighted images with colon inflated and contrast agent are usually used the ones used by the radiologist to detect polyps because they offer the best resolution. In our work, we tested our algorithms on the T1-weighted images in axial view, the patient being in prone position. We used images coming from two datasets, both from the National University Hospital (NUH) in Singapore.

The first dataset comprises 176 images against 168 for the second dataset. In the two datasets, the images are of size 288×320 with an intra-slice resolution of 1.25 mm and
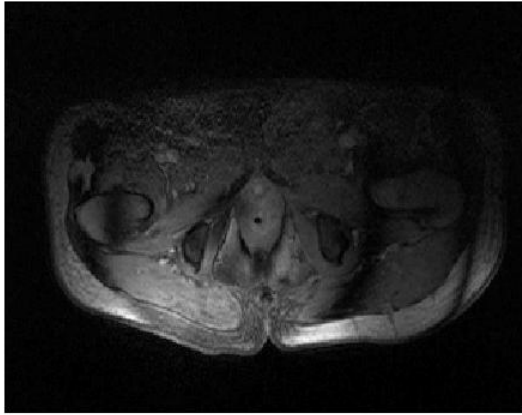
(a) Image with no colon
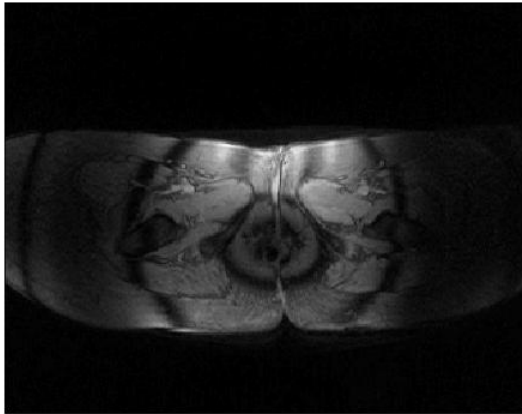
(b) Another image with no colon

Figure 6.1: Examples of images not retained for testing our algorithms

an inter-slice resolution of 2.5 mm.

Some images were not usable for our experiment. The reason is that the colon simply did not appear in some of the images at the extremity of the dataset (Figure 6.1(a)). Therefore, we kept 143 images in the first dataset and 152 images in the second dataset.

6.2 Presentation of the method

6.2.1 Quantitative evaluation

The most common method to evaluate an segmentation algorithm is to compare the result with a manually segmented image. We denote by Ω_M, Ω_{TM} and Ω_{BS} the set

of colonic pixels segmented manually, using thresholding methods (TM) and using B-snakes (BS), respectively. We use the Jaccard's measure to compare the sets:

$$J_{TM} = \frac{\text{card}(\Omega_{TM} \cap \Omega_M)}{\text{card}(\Omega_{TM} \cup \Omega_M)} \tag{6.1}$$

$$J_{BS} = \frac{\text{card}(\Omega_{BS} \cap \Omega_M)}{\text{card}(\Omega_{BS} \cup \Omega_M)} \tag{6.2}$$

The set of manually segmented pixels is considered as ground truth. Therefore, the validity of this method depends closely on the quality of manual segmentation. However, due to the relatively low quality of MR images, our confidence in manual segmentation was not strong in many images. Then we evaluated our algorithms on a set of 30 images chosen among the two datasets for which we were confident about the manual segmentation.

6.2.2 Qualitative evaluation

In the previous section we saw that a quantitative evaluation was not possible for all images in the dataset. However, we would like to be able to give an appreciation of the performance of our algorithms for all the images of the dataset where the colon appears.

Therefore, we applied the algorithms on all the images of the dataset and evaluated the results visually. A segmented contour was considered as good when the contour gave an acceptable representation of the colon according to the information in the image. It was considered as bad when the contour was obviously not representative of the reality.

This method is not as accurate as the quantitative method since it depends strongly on the appreciation of the author. However, it allows an evaluation of the whole dataset. By combining the results of the quantitative and the qualitative methods, we can have a good idea of the performances of our algorithms.

6.3 Results

All our algorithms were implemented in Matlab on a PC with a 3.6 GHz processor and 2GB of RAM. The processing time with thresholding methods for each image is about 10 seconds, including anisotropic diffusion. The processing time with B-snakes is usually

	mean	max	min
TM	$93,8\%$	96.5%	$88,4\%$
BS	$93,7\%$	96.7%	$89,7\%$

Table 6.1: Quantitative results

	Total	TM Success	BS Success
Number of images	143	94	108
Rate	100%	66%	76%

Table 6.2: Qualitative results obtained with dataset 1

around 3 seconds.

6.3.1 Quantitative results

The results of the quantitative evaluation on 30 images are summarized in Table 6.1. We observe that thresholding methods and B-snakes both perform well in cases where we were confident about manual segmentation. Figure 6.2 shows the contours obtained with the three segmentation methods on one image.

6.3.2 Qualitative results

In our qualitative evaluation, we found that thresholding methods gave accurate results in 94 images in the first dataset and 126 images in the second dataset. B-snakes gave accurate results in 108 images in the first dataset and 127 images in the second dataset. The better results obtained with the second dataset can be explained by the better quality of images. Tables 6.2 and 6.3 summarize the results obtain for the two datasets, respectively. Table 6.4 gives the results for all images, regardless of the dataset. In those tables, a success means that the segmented contour obtained was considered as good. We observe that B-snakes achieve a slightly better performance than thresholding methods (75% against 80%).

In cases where both thresholding methods and B-snakes give accurate results, we observe an important difference in the regularity of the contour between the two

	Total	TM Success	BS Success
Number of images	152	126	127
Rate	100%	83%	84%

Table 6.3: Qualitative results obtained with dataset 2

(a) Manual segmentation

(b) Thresholding methods

(c) B-snakes

Figure 6.2: Comparison between manual thresholding and automatic thresholding

	Total	TM Success	BS Success
Number of images	295	220	235
Rate	100%	75%	80%

Table 6.4: Overall qualitative results
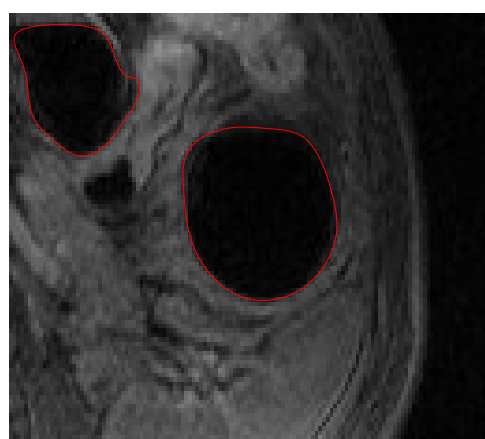
(a) contour obtained with B-snake
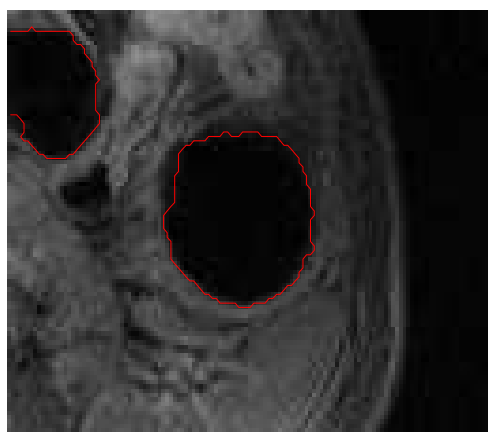
(b) contour obtained with B-snake

(c) contour obtained with thresholding methods

(d) contour obtained with thresholding methods

Figure 6.3: Comparison of regularity of contours (1)
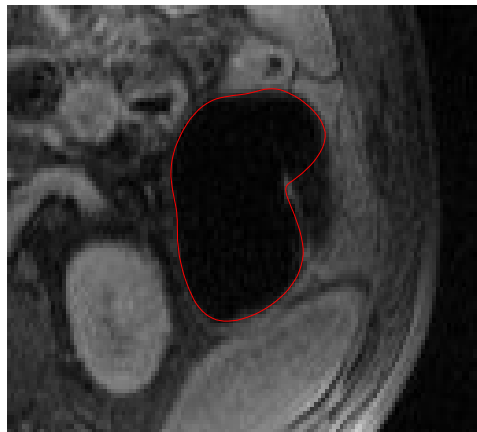
(a) contour obtained with B-snake
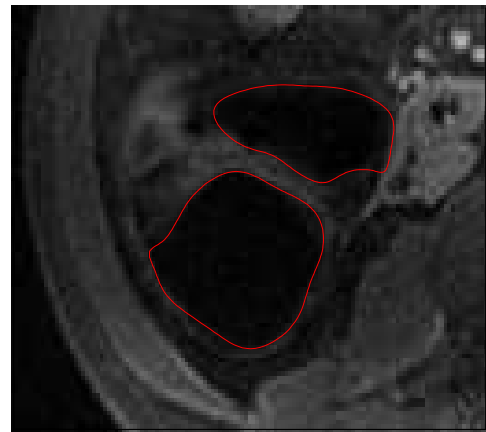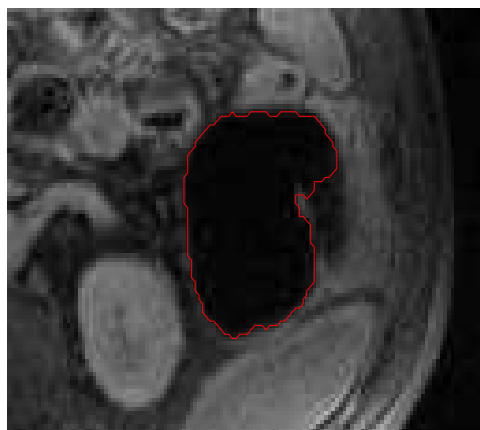
(b) contour obtained with B-snake

(c) contour obtained with thresholding
methods
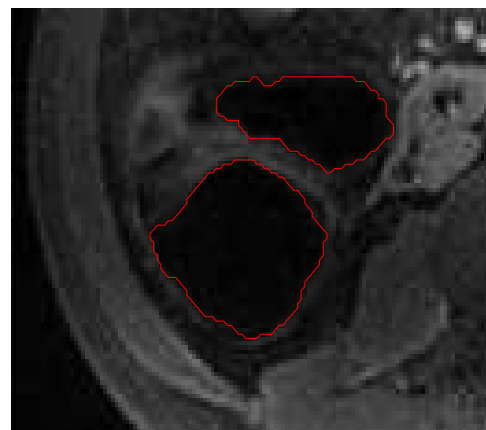
(d) contour obtained with thresholding
methods

Figure 6.4: Comparison of regularity of contours (2)

methods. B-snakes give a regular contour in most cases, whereas thresholding methods, despite the use of anisotropic diffusion, give jagged contours if the image is very noisy. We illustrate this result in Figures 6.3 and 6.4.

We identified two categories of images where B-snakes and thresholding methods both fail to segment the colon properly. The first category consists of images with sharp details in the colon wall. Those sharp details usually represent haustral folds. Figure 6.5 shows different kinds of contours that we obtained with B-snakes. We observe that the details are completely ignored in some cases (Figures 6.5(c) and 6.5(d)) and the snake intersects itself in other cases (Figures 6.5(a) and 6.5(b)). Figure 6.6 shows the results obtained with thresholding methods. Details are also ignored in some cases (Figures 6.6(c) and 6.6(d)). In some other cases, a small part of the detail is isolated from the rest of the contour (Figures 6.6(a) and 6.6(b)).

The second category of images that our algorithms failed to process well are images where the contrast colon-gray tissues was very low. The results obtained with both algorithms were usually contours that leaked out of the colon. Figures 6.7(a) and 6.7(b) show some results obtained with B-snakes while Figures 6.7(c) and 6.7(d) show the contours given by thresholding methods.

6.4 Interpretation of the results

6.4.1 Improvement on the regularity of contours with B-snakes

As mentioned before, the results obtained with B-snakes have a smoother aspect than the contours obtained with thresholding methods. There are two main reasons to explain this observation. The first reason comes from the implementation of the snake itself: the presence of internal forces, as well as the constraint of C^2 regularity ensures that the contour remains smooth. The second reason is more subtle; the accuracy of the thresholding methods is limited to the size of the pixels, whereas the snake can achieve sub-pixel accuracy. To better understand this issue, let us consider an edge as represented in Figure 6.8[1]. In red and magenta, we show the contours given by the thresholding methods. For the magenta contour, pixels at the inner border of the thresholded region have been considered whereas the red contour represent pixels located at

---

[1]this image has been artificially generated
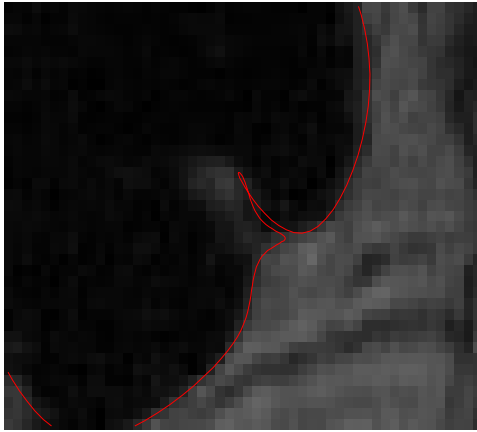
(a) B-snake intersecting with itself
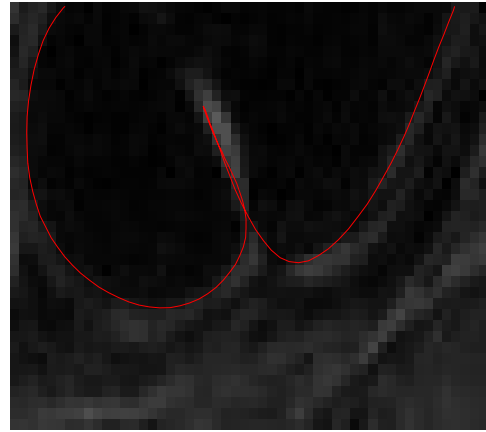
(b) B-snake intersecting with itself
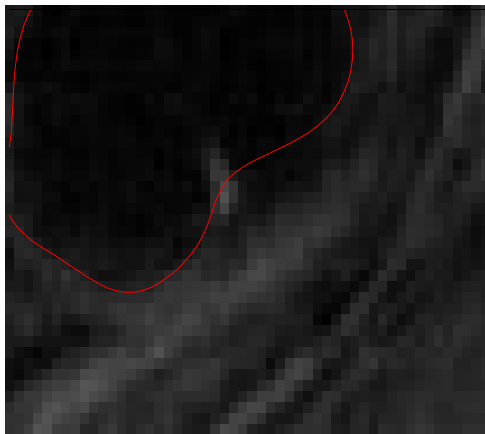
(c) B-snake missing a sharp detail
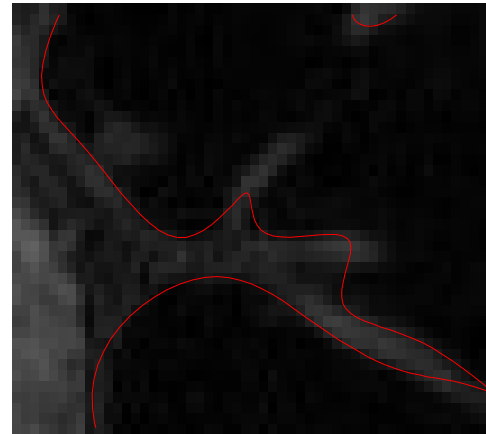
(d) B-snake missing a sharp detail

Figure 6.5: Problems encountered with B-snakes on sharp details
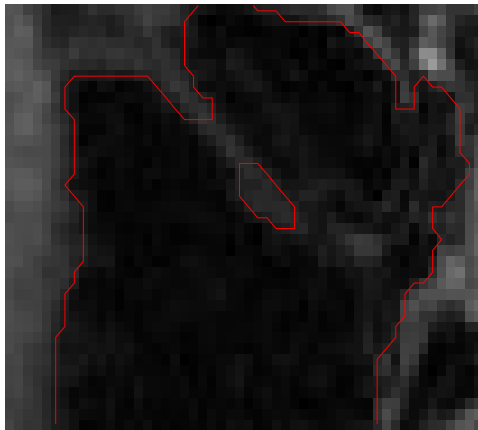
(a) sharp detail isolated with
thresholding methods
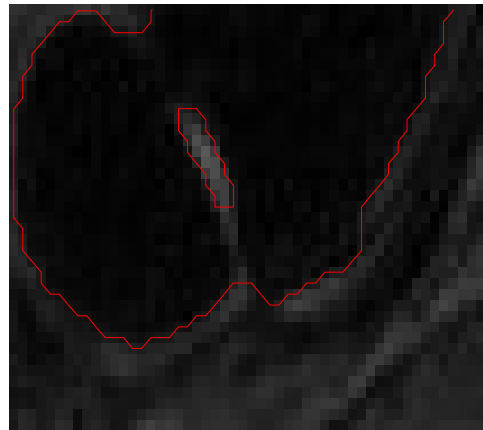
(b) sharp detail isolated with
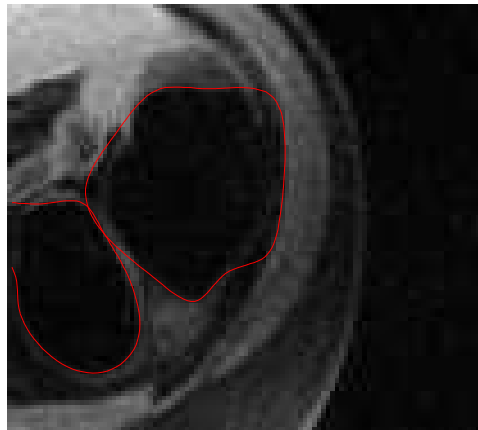thresholding methods
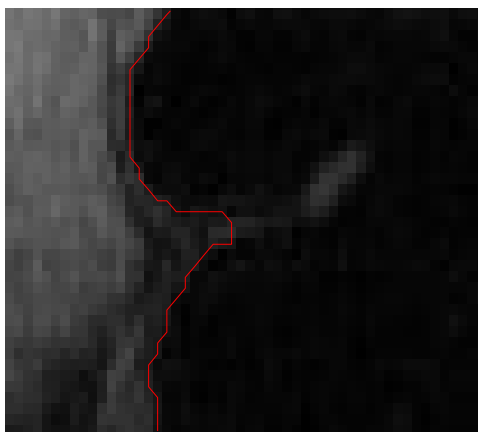
(c) thresholding methods missing a
sharp detail
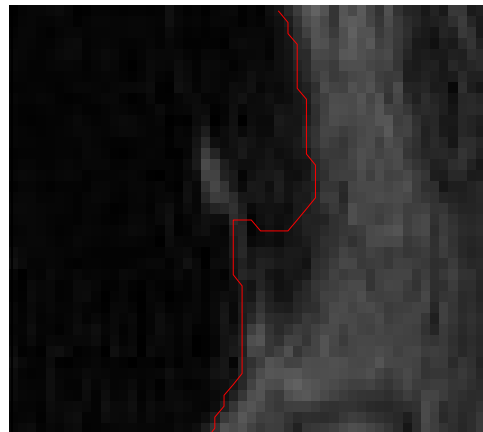
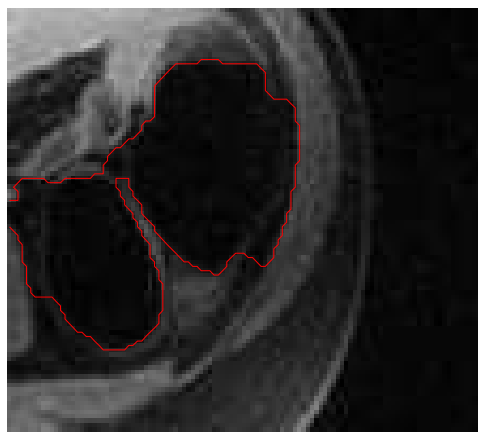(d) thresholding methods missing a
sharp detail

Figure 6.6: Problems encountered with thresholding methods on sharp details

(a) leakage with B-snake

(b) leakage with B-snake

(c) leakage with thresholding methods
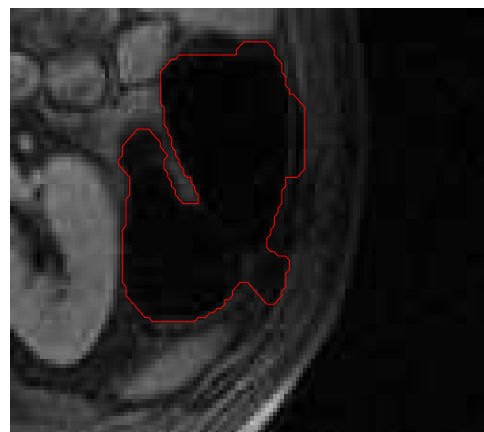
(d) leakage with thresholding methods

Figure 6.7: Problems with low contrast images
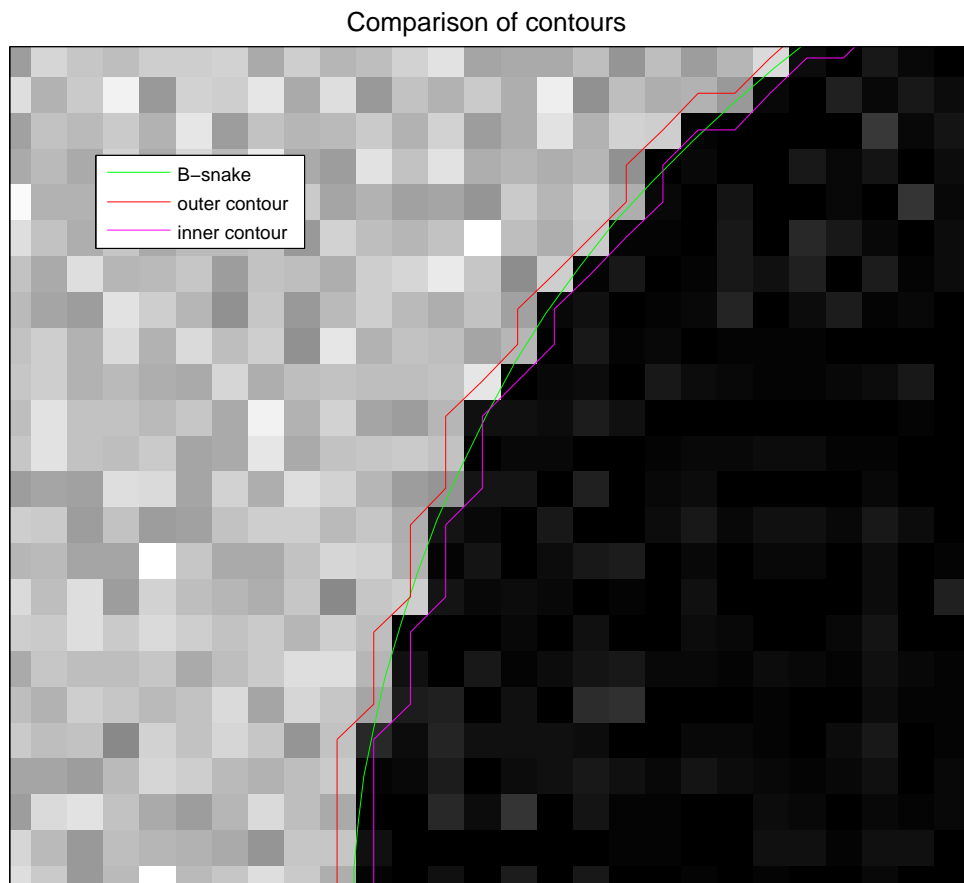
Figure 6.8: Comparison of contours

the outer border of the thresholded region. The blue contour is the result given by the B-snake algorithm. As we can see, the contours obtained with the thresholding methods are limited by the pixel size. We can say that the space of admissible contours is discretized. The magenta and red contours are the best approximation for the ideal contour which is located somewhere in between. On the contrary, the space of admissible B-snakes is continuous, allowing the snake to get closer to the ideal contour and to adopt a smoother aspect.

Globally, B-snakes manage to cope with most of the difficulties brought by MR images as thresholding methods do. The power of B-snakes is to impose a constraint of regularity which is not too restrictive and can still adjust to many different shapes. Our strategy of control points insertion offers even more flexibility to the model. Furthermore, the processing time is smaller for B-snakes since it does not require any sophisticated preprocessing algorithm. This quality is particularly significant for big datasets like the one we use for virtual colonoscopy in general.

6.4.2 Difficulties encountered with some images

We showed that both B-snakes and thresholding methods did not give accurate results for some images, namely images with sharp details and images with low contrast between the colon and gray tissues. Those problems can be partly explained due to the fact that we work on 2D images. Due to the complexity of the colon shape, the boundary of the colon is sometimes difficult to find in 2D images. If we look at the series of consecutive images in Figure 6.9 we see on the right side of the image that one part of the colon splits into two different parts. This can represent a haustral fold or a change of direction of the colon at that location. In some images, the boundary is so thin that it is difficult to determine if there is only one or two parts. Those images are typically images containing sharp details which our algorithms have difficulty in segmenting. One could surmise that these problems could be solved if we use the whole 3D dataset. Indeed, we would then have access to the information carried by adjacent slices, which could be helpful in such cases.

However, working in 2D is not the only issue and the inherent quality of MR images is also a critical factor. Indeed, we recall the fact many images are difficult to segment manually. Some improvement in MR imaging technology is expected in order to take
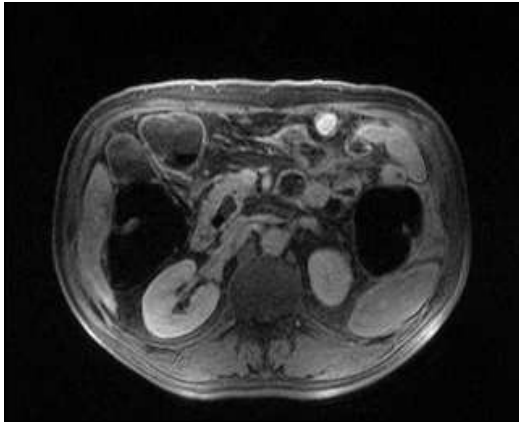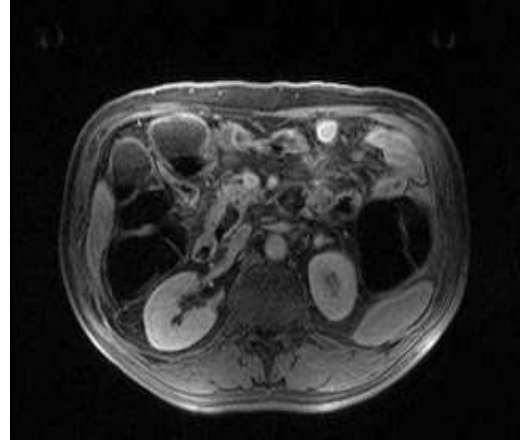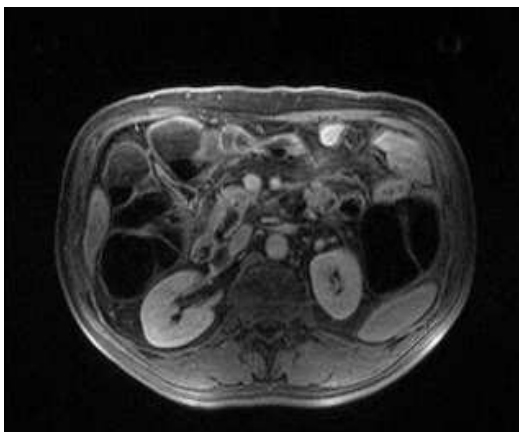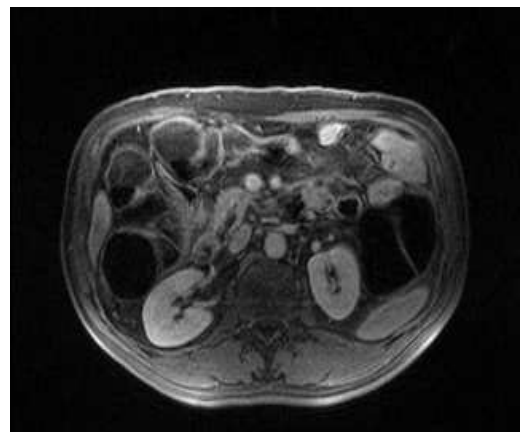
(a) Image 1
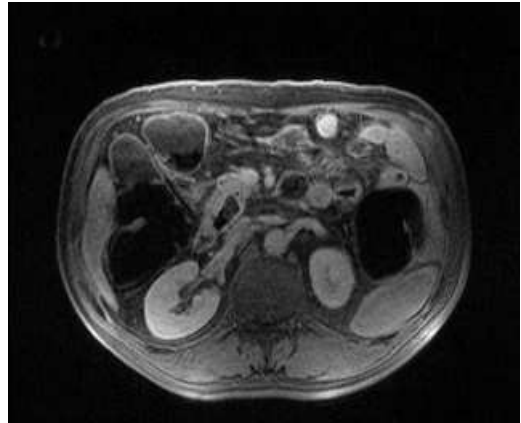
(b) Image 2
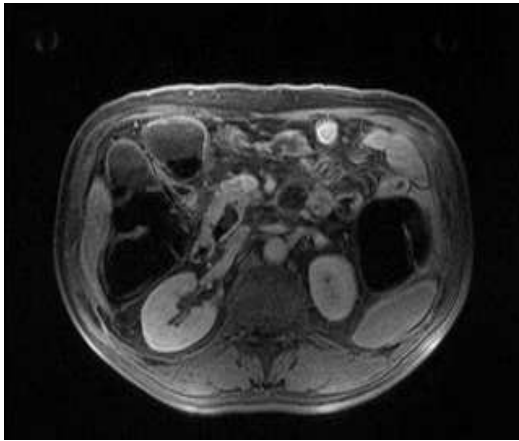
(c) Image 3

(d) Image 4

(e) Image 5

(f) Image 6

Figure 6.9: Series of images where the colon splits into two different parts

full advantage of MR images.

6.5 3D reconstruction

Although we only work on 2D images, a 3D reconstruction is possible by agglomerating the 2D contours. We chose a series of 40 images and segmented each image with a B-snake. The results for each image are shown in Appendix B.

The agglomeration of B-snakes can be easily done using the theory that we developed in Chapter 5. Recalling Equation 5.39, we now use two parameters (s, t) and on each interval $[s_i, s_{i+1}]$ and $[t_i, t_{j+1}]$, the snake can be written as

$$
v_{i,j}(s, t) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{6} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 \\ \frac{-1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \end{bmatrix} \begin{bmatrix} P_{i-1,j-1} & P_{i-1,j} & P_{i-1,j+1} & P_{i-1,j+2} \\ P_{i,j-1} & P_{i,j} & P_{i,j+1} & P_{i,j+2} \\ P_{i+1,j-1} & P_{i+1,j} & P_{i+1,j+1} & P_{i+1,j+2} \\ P_{i+2,j-1} & P_{i+2,j} & P_{i+2,j+1} & P_{i+2,j+2} \end{bmatrix}
$$

$$
\dots \begin{bmatrix} \frac{-1}{6} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & 0 & \frac{2}{3} \\ \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{6} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}
$$

$$(6.3)$$

The results of this 3D reconstruction are shown in Figures 6.10.

The main limitation of this method is the difficulty to handle the changes of topology as the one we mentioned in section 6.4.2. In that particular case, we would need the ability to split one contour into two distinct contours. Our algorithm does not currently handle this kind of situation.
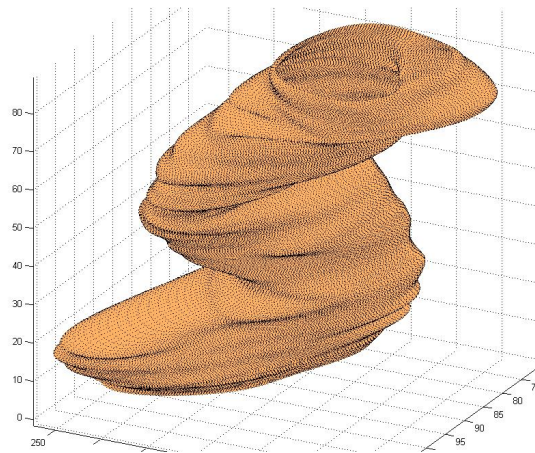
(a) 3D reconstruction (view 1)
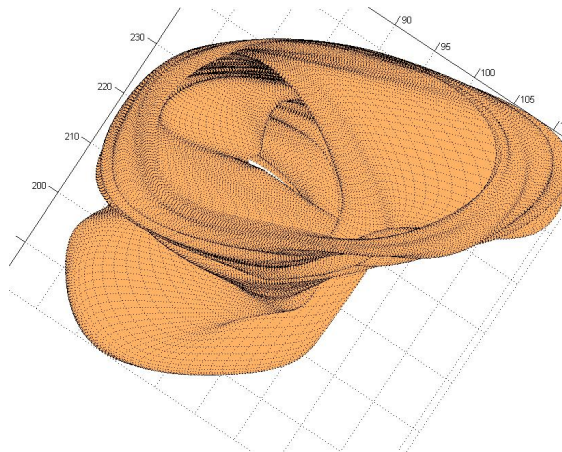
(b) 3D reconstruction (view 2)
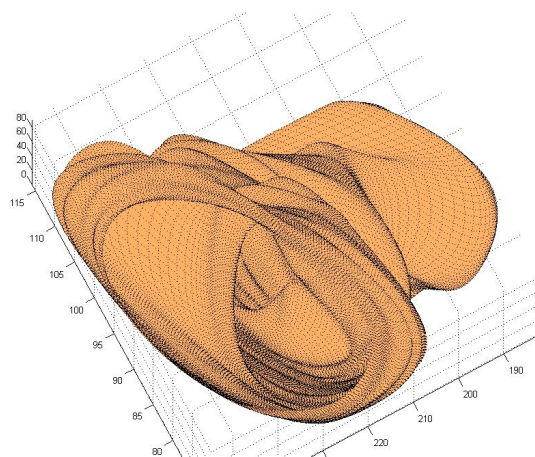
(c) 3D reconstruction (view 3)

Figure 6.10: 3D recontruction of the colon from 40 planar images

7. Conclusion

7.1 Summary of contributions

In this work, we have first established a coherent procedure of segmentation based on thresholding methods and anisotropic diffusion. We have combined different well known algorithms and improved some of them to fit our application. A particular effort has been made for the optimization of the parameters in the anisotropic diffusion algorithm.

Then, we have presented another segmentation procedure based on snakes. We have first mentioned the problems of initialization and we have solved them by using some basic thresholding methods presented in the first part. Two different models of snakes have been presented, namely the finite difference snakes and the B-snakes. A simple and efficient implementation has been proposed for the two models. A stopping criterion and a control point insertion algorithm have been included to further improve the B-snake model.

Finally we compared the results of thresholding methods and snakes. We showed that B-snakes and thresholding methods perform equally well in general. B-snakes proved to be superior to thresholding methods in terms of regularity of the contour and time processing.

7.2 Future work

In our work, we developed an empirical method to obtain the best parameters for MR images of the colon. One drawback of our method is that the parameters that we obtain are optimal for our application only. The same parameters would probably not be optimal for ultrasound images for example. A current trend in the research about anisotropic diffusion is to find a method to obtain adjustable optimal parameters. An interesting approach can be found in [75]. In this paper, Castellanos *et al.* describe an iterative procedure to determine optimal parameters for the diffusion equations of Perona and Malik. They apply their algorithm to MR images of the brain. It would be interesting to analyze the performance of their algorithm on different types of images, and to implement a similar procedure on other diffusion equations like the improved Weickert's equation that we used in our work.

We have also proven the feasibility of colon segmentation with B-snakes in 2D MR images. A 3D reconstruction obtained from the agglomeration of 2D B-snakes has been shown. But as mentioned before, this method has some limitations. The next step would be to implement the B-snakes directly in 3D. The main challenges in this approach would be to redefine the initialization and the control points insertion algorithms. The problem comes from the fact that the order of control points matters for snakes (the same control points taken in a different order can give a totally different result). Although this problem can be easily handled in 2D, it is much more challenging in 3D.

Another direction of research would be to evaluate the ability of snakes to represent polyps accurately. Many polyp detection algorithms are based on an analysis of contour curvature. Since snakes have an explicit mathematical formulation, access to the curvature can be done easily and accurately. A preliminary approach can be found in [76] where B-snakes are used to determine contact angles of water drops.

Finally, different models of snake could be investigated, for example, geometric deformable models, which have the advantage of not using a parametric representation. We introduced them in section 2.3. Geometric models are based on level-set methods. In other words, the contours are represented as the zero level of a higher-dimensional function. Therefore, unlike B-snakes, geometric models can easily handle changes of topology. Figure 7.1 illustrates this property. The function Φ is the higher-dimensional

$$contour\, C : \Phi = 0$$
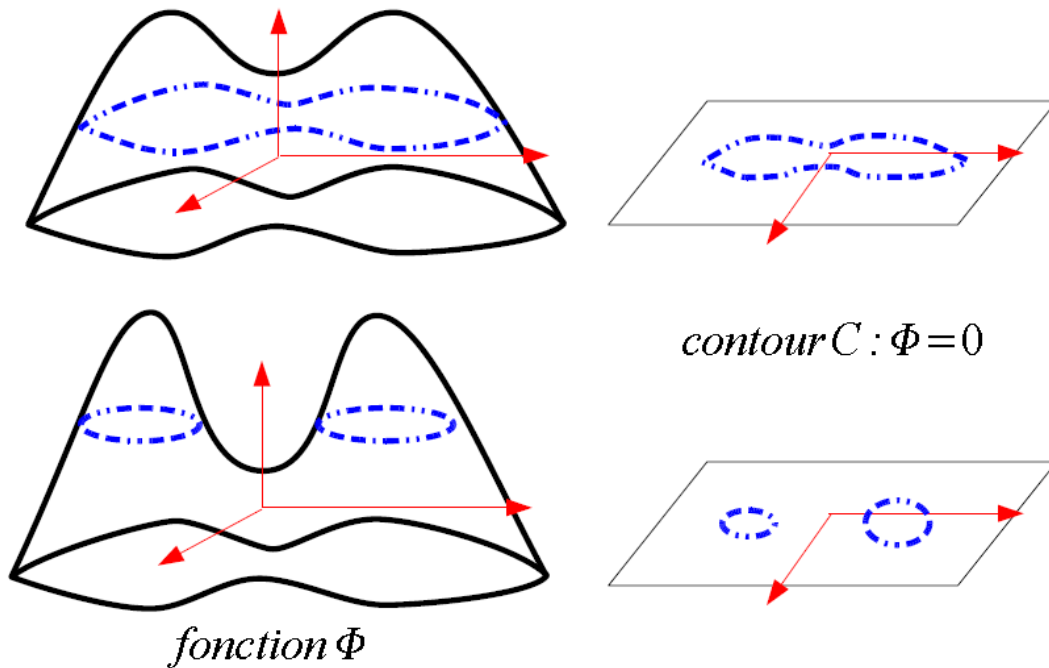
$$fonction\, \Phi$$

Figure 7.1: Principle of level-set method [3]

function and C represents the zero-level. In this figure, we can see that by changing the function Φ, a single contour can easily be split into two contours.

More details about geometric models can be found in [71]. Recent examples of their application for medical images can be found in [77] for the segmentation of lungs in CT images, or in [78] for the segmentation of the colon in CT images too.

Until now, we have focused on finding a method to segment the colon. However, the ultimate goal of virtual colonoscopy is to automatically detect polyps accurately. Most methods available in the literature try to detect polyps according to their shape. Yeo [26] gives an example of such a method. However, the complexity of the colon shape is an issue for those methods, and haustral folds often give false-positive results [79]. An interesting direction for future work would be to detect the polyps according to the contrast of luminosity in MR images. Some promising observations have been made by Hartmann *et al.* [37]. In their work, they visually detect polyps from the contrast between polyps and the surrounding tissues. It would be interesting to integrate this feature in an automatic detection algorithm.

Bibliography

[1] U.S. National Library of Medicine. MedinePlus Encyclopedia. `http://www.nlm.nih.gov/medlineplus/ency/presentations/100089_1.htm`.

[2] `www.answers.com/topic/colonoscopy`.

[3] R. Teina. Méthodes de traitement de l'image: Modèles déformables. `http://www.teina.org/Enseignement/2006-2007/TIF/`.

[4] World Health Organization. Cancer, July 2008. `http://www.who.int/mediacentre/factsheets/fs297/en/`.

[5] American Cancer Society. Cancer Facts and Figures 2008. Technical report, American Cancer Society, 2008.

[6] A. Seow, W.P. Koh, K.S. Chia, L. Shi, H.P. Lee, and K. Shanmugaratnam. Trends in Cancer Incidence in Singapore 1968–2002. Technical Report 6, Singapore Cancer Registry, 2004.

[7] http://en.wikipedia.org/wiki/Colon.

[8] H. Gray. *Anatomy of the Human Body*, chapter XI. Lea & Febiger, 20th edition, 1918. `http://www.bartleby.com/107/249.html`.

[9] S. Holland. Colorectal polyp. `http://commons.wikimedia.org/wiki/File:Polyp.jpeg`.

[10] M.-Y. Su, Y.-P. Ho, C.-M. Hsu, C.-T. Chiu, P.-C. Chen, J.-M. Lien, S.-Y. Tung, and C.-S. Wu. How can colorectal neoplasms be treated during colonoscopy? *World Journal of Gastroenterology*, 11(18):2806–2810, 2005.

[11] S.J. Winawer, A.G. Zauber, M.N. Ho, M.J. O'Brien, L.S. Gottlieb, S.S. Sternberg, J.D. Waye, M. Schapiro, J.H. Bond, and J.F Panish. Prevention of Colorectal Cancer by Colonoscopic Polypectomy. *The New English Journal of Medicine*, 329(27):1977–1981, Dec 1993.

[12] A. Figuerido, R.B. Rumble, J.Maroun, C.C. Earle, B. Cummings, R. McLeod, L. Zuraw, and C. Zwaal. Follow-up of Patients With Curatively Resected Colorectal Cancer : A Practice Guideline. *BMC Cancer*, 3(26), Oct 2003.

[13] American Cancer Society. Colorectal Cancer Facts and Figures 2005 Special Edition. Technical report, American Cancer Society, 2005.

[14] M. Häfner. Conventional Colonoscopy : Technique, Indicators, Limits. *European Journal of Radiology*, 61:409–414, 2007.

[15] G.A. Rollandi, E. Biscaldi, and E. DeCicco. Double Contrast Barium Enema: Technique, Indications, Results and Limitations of a Conventional Imaging Methodology in the MDCT Virtual Endoscopy Era. *European Journal of Radiology*, 61:382–387, 2007.

[16] A. O'Hare and H. Fenlon. Virtual Colonoscopy in the Detection of Colonic Polyps and Neoplasms. *Best Practice and Research Clinical Gastroenterology*, 20(1):79–92, 2006.

[17] Siemens Healthcare. SOMATOM Sensation. `http://www.medical.siemens.com`.

[18] Siemens Healthcare. MAGNETOM Avento 1.5T. `http://www.medical.siemens.com`.

[19] S. Romano and T. Mang. Imaging of the Colon : State of the Art Conventional Studies, New Techniques, Issues and Controversies. *European Journal of Radiology*, 61:375–377, 2007.

[20] `http://www.viatronix.com/`.

[21] M.S. Juchems, J. Ehmann, and H.-J. Brambs. A Retrospective Evaluation of Patient Acceptance of Computed Tomography Colonography in Comparison With Con-

ventional Colonoscopy in an Average Risk Screening Population. *Acta Radiologica*, 46:664–670, 2005.

[22] Canadian Agency for Drugs and Technologies in Health. Magnetic Resonance Colonography for Colorectal Polyp and Cancer Detection. *Health Technology Update*, 6, May 2007.

[23] D.J. Brenner and M.A. Georgsson. Mass Screening With CT Colonography : Should the Radiation Exposure Be of Concern. *Gastroenterology*, 129(1):328–337, Jul 2005.

[24] B. Saar, A. Beer, T. Rösch, and E.J. Rummeny. Magnetic Resonance Colonography : A Promising New Technique. *Current Gastroenterology Reports*, 6(5):389–394, Oct 2004.

[25] F. Le Manour. Application of Diffusion Techniques to the Segmentation of MR 3d Images For Virtual Colonoscopy. Master's thesis, National University of Singapore, 2006.

[26] E.T. Yeo. Virtual Colonoscopy Software. Master's thesis, National University of Singapore, May 2004.

[27] D.J.Vining, D.W. Gelfand, R.E. Bechtold, E.S. Scharding, E.K. Grishaw, and R.Y. Shifrin. Technical Feasibility of Colon Imaging With Helical CT and Virtual Reality. *American Journal of Roentgenology*, 162(Suppl):104, 1994.

[28] A.K. Hara, C.D. Johnson, J.E. Reed, D.A. Ahlquist, H. Nelson, R.L. Ehman, C. McCollough, and D.M. Ilstrup. Detection of Colorectal Polyps by Computed Tomographic Colography : Feasibility of a Novel Technique. *Gastroenterology*, 110(1):284–290, Jan 1996.

[29] A.K. Hara, C.D. Johnson, J.E. Reed, D.A. Ahlquist, H. Nelson, R.L. MacCarty, and D.M. Ilstrup. Detection of Colorectal Polyps With CT Colography : Initial Assessment of Sensitivity and Specificity. *Radiology*, 205(1):59–65, Jan 1997.

[30] J.G. Fletcher, C.D. Johnson, T.J. Welch, R.L. MacCarty, D.A. Ahlquist, J.E. Reed, W.S. Harmsen, and L.A. Wilson. Optimization of CT Colonography Technique Prospective Trial in 180 Patients. *Radiology*, 216(3):704–711, Sep 2000.

[31] G. Spinzi, G. Belloni, A. Martegani, A. Sangiovanni, C. Del Favero, and G. Minoli. Computed Tomographic Colonography and Conventional Colonoscopy for Colon Diseases a Prospective Blinded Study. *The American Journal Of Gastroenterology*, 96(2):394–400, Feb 2001.

[32] P.J. Pickhardt, J.R. Choi, I. Hwang, J.A. Butler, M.L. Puckett, H.A. Hildebrandt, R.K. Wong, P.A. Nugent, P.A. Mysliwiec, and W.R. Schindler. Computed Tomographic Virtual Colonoscopy to Screen for Colorectal Neoplasia in Asymptomatic Adults. *The New English Journal of Medicine*, 349(23):2191–2220, Dec 2003.

[33] D.C. Rockey, T.M. Zarchy, J.R. Uribe, S. Pais, C. Bongiorno, P.O. Katz, G.S. Thomas, and P.J. Pickhardt. Virtual Colonoscopy to Screen for Colorectal Cancer. *The New English Journal of Medicine*, 350(11):1148–1150, Mar 2004.

[34] A.K. Hara, C.D. Johnson, R.L MacCarty, and T.J. Welch. Incidental Extracolonic Findings at CT Colonography. *Radiology*, 215(2):353–357, May 2000.

[35] W. Luboldt, P. Bauerfeind, P. Steiner, M. Fried, G.P. Krestin, and J.F. Debatin. Preliminary Assessment of 3d Magnetic Resonance Imaging for Various Colonic Disorders. *The Lancet*, 349:1288–1291, May 1997.

[36] W. Luboldt, P. Steiner, P. Bauerfeind, P. Pelkonen, and J.F. Debatin. Detection of Mass Lesions with MR Colonography : Preliminary Report. *Radiology*, 207(1):59–65, Apr 1998.

[37] D. Hartmann, B. Bassler, D.Schilling, H.E. Adamek, R. Jakobs, B. Pfeifer, A. Eickhoff, C. Zindel, J.F. Riemann, and G. Layer. Colorectal Polyps Detection With Dark-Lumen MR Colonography versus Conventional Colonoscopy. *Radiology*, 238(1):143–149, Jan 2006.

[38] J. Florie, S. Jensch, R.A.J. Nievelstein, R.A.J. Nievelstein, J.F. Bartelsman, L.C. Baak R.E. van Gelder, B. Haberkorn, A. van Randen, M.M. van der Ham, P. Snel, V.P.M. van der Hulst, P.M.M. Bossuyt, and J. Stoker. Colonography With Limited Bowel Preparation Compared With Optical Colonoscopy in Patients at Increased Risk for Colorectal Cancer. *Radiology*, 243(1):122–131, Apr 2007.

[39] F. Vos, I. Serlie, R. van Gelder, J. Stoker, H. Vrooman, and F. Post. A Review of Technical Advances in Virtual Colonoscopy. *Studies in health technology and informatics*, 84(2):394–400, 2001.

[40] W. Luboldt, J.G. Fletcher, and T.J. Vogl. Colonography Current Status, Research Directions and Challenges. *European Radiology*, 12(3):502–524, Mar 2002.

[41] J. Wessling, R. Fischbach, N. Meier, T. Allkemper, J. Klusmeier, K. Ludwig, and W. Heindel. CT Colonography : Protocol Optimization with Multi-Detector Row CT : Study in an Anthropomorphic Colon Phantom. *Radiology*, 228(3):753–759, 2003.

[42] C.L. Wyatt, Y. Ge, and D.J. Vining. Automatic Segmentation of the Colon for Virtual Colonoscopy. *Computerized Medical Imaging and Graphics*, 24:1–9, Oct 1999.

[43] R.M. Summers, C.F. Beaulieu, L.M. Pusanik, J.D. Malley, R.B. Jeffrey, D.I. Glazer, and S. Napel. Automated Polyp Detector for CT Colonography : Feasibility Study. *Radiology*, 216(1):284–290, 2000.

[44] R. Van Uitert, I. Bitter, and R.M. Summers. Detection of Colon Wall Outer Boundary and Segmentation of the Colon Wall Based on Level Set Methods. *Proceedings of the 28th IEEE EMBS Annual International Conference*, pages 3017–3020, Sept 2006.

[45] M. Franaszek, R.M. Summers, P.J. Pickhardt, and J.R. Choi. Hybrid Segmentation of Colon Filled With Air and Opacifed Fluid for CT Colonography. *IEEE Transactions On Medical Imaging*, 25(3):358–368, Mar 2006.

[46] C.L. Wyatt, Y. Ge, and D.J. Vining. Segmentation in Virtual Colonoscopy Using a Geometric Deformable Model. *Computerized Medical Imaging and Graphics*, 30:17–30, 2006.

[47] F. Admasu, S. Al-Zubi, K. Toennies, N. Bodammer, and H. Hinrichs. Segmentation of Multiple Sclerosis Lesions From MR Brain Images Using the Principles of Fuzzy-Connectedness and Artificial Neuron Networks. *Proceedings of the International Conference on Image Processing*, 2:1081–1084, Sep 2003.

[48] Y. Chenoune, E. Deléchelle, E. Petit, T. Goissen, J. Garot, and A. Rahmouni. Segmentation of Cardiac Cine MR Images and Myocardial Deformation Assessment

Using Level Set Methods. *Computerized Medical Imaging and Graphics*, 29:607–616, Sep 2005.

[49] A. Gupta, L. von Kurowski, A. Singh, D. Geiger, C.-C. Liang, M.-Y. Chiu, L.P. Adler, M. Haacke, and D.L. Wilson. Cardiac MR Image Segmentation Using Deformable Models. *Proceedings on Computers in Cardiology*, pages 747–750, Sep 1993.

[50] S. Ranganath. Contour Extraction from Cardiac MRI Studies Using Snakes. *IEEE Transactions On Medical Imaging*, 14(2):328–338, Jun 1995.

[51] J. Liang, T. McInerney, and D. Terzopoulos. United Snakes. *Medical Image Analysis*, 10:215–233, Nov 2006.

[52] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, January 2004.

[53] Y. Nakagawa and A. Rosenfeld. Some experiments on variable thresholding. *Pattern Recognition*, 11:191–204, 1978.

[54] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 12(7):629–639, Jul 1990.

[55] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, Feb 1992.

[56] J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever. Efficient and Reliable Schemes for Nonlinear Diffusion Filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, Mar 1998.

[57] G. Gerig, O. Kubler, R. Kikinis, and F.A. Jolesz. Nonlinear Anisotropic Filtering of MRI Data. *IEEE Transactions On Medical Imaging*, 11(2):221–232, Jun 1992.

[58] J. Montagnat, M. Sermesant, H. Delingette, G. Malandin, and N. Ayache. Anisotropic Filtering for Model-Based Segmentation of 4D Cylindrical Echocardiographic Images. *Pattern Recognition Letters*, 24:815–828, 2003.

[59] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, Universität Kaiserslautern, Jan 1996.

[60] X. Li and T. Chen. Nonlinear Diffusion with Multiple Edginess Thresholds. *Pattern Recognition*, 27(8):1029–1037, 1994.

[61] R.T. Whitacker and S.M. Pizer. A Multi-scale Approach to Nonunifor Diffusion. *CVGIP: Image Understanding*, 57(1):99–110, Jan 1993.

[62] R. Guillemaud and M. Brady. Estimating the Bias Field of MR Images. *IEEE Transactions On Medical Imaging*, 16(3):238–251, Jun 1997.

[63] L. Kaufman, M. Kramer, L.E. Crooks, and D.A. Ortendahl. Measuring Signal-to-Noise Ratios in MR Imaging. *Radiology*, 173(1):265–267, Oct 1989.

[64] C.R. Meyer, P.H. Bland, and J. Pipe. Retrospective Correction of Intensity Inhomogeneities in MRI. *IEEE Transactions on Medical Imaging*, 14(1):36–41, Mar 1995.

[65] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision,*, pages 321–331, 1988.

[66] L.D. Cohen and I. Cohen. A finite element method applied to new active contour models and 3D reconstruction from cross sections. Technical Report 1245, INRIA, 1990.

[67] T. McInerney and D. Terzopoulos. Deformable Model in Medical Image Analysis. *Medical Image Analysis*, 1(2), 1996.

[68] S. Menet, P. Saint-Marc, and G. Medioni. Active contour models: overview, implementation and applications. *IEEE International Conference on Systems, Man and Cybernetics*, 1990.

[69] L.H. Staib and J.S. Duncan. Boundary Finding with Parametrically Deformable Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, Nov 1992.

[70] D.M. Gavrila. Hermite Deformable Contours. *Proceedings of the International Conference on Pattern Recognition*, pages 130–135, 1996.

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

[72] C. Xu and J.L. Prince. Gradient Vector Flow: A New External Force for Snakes. *IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 66–71, 1997.

[73] V. Medina, R. Valdés, O. Yañez-Suárez, M. Garza-Jinich, and J.-F. Lerallut. Automatic Initialization for a Snakes-Based Cardiac Contour Extraction. *Proceedings of the 22nd Annual EMBS International Conference*, pages 1625–1628, 2000.

[74] S. Rahnamayan, H.R. Tizhoosh, and M.M.A. Salama. Automated Snake Initialization for the Segmentation of the Prostate in Ultrasound Images. *Proceedings of the International Conference on Image Analysis and Recognition*, pages 930–937, 2005.

[75] J. Castellanos, K. Rohr, T. Tolxdorff, and G. Wagenknecht. Automatic Parameter Optimization for De-noising MR Data. *Medical Image Computing and Computer-Assisted Intervention*, 3750:320–327, 2005.

[76] A.F. Stalder, G. Kulik, D. Sage, L. Barbieri, and P. Hoffmann. A snake-based approach to accurate determination of both contact points and contact angles. *Colloids and surfaces. A, Physicochemical and engineering aspects*, 286:92–103, 2006.

[77] M. Lee, S. Park, W. Cho, S. Kim, and C. Jeong. Segmentation of medical images using a geometric deformable model and its visualization. *Canadian Journal of Electrical and Computer Engineering*, 33:15–19, 2008.

[78] S.F. Hamidpour, A. Ahmadian, R.A. Zoroofi, and J.H. Bidgoli. Hybrid segmentation of colon boundaries CT images based on geometric deformable model. *IEEE International Conference on Signal Processing and Communications*, pages 967–970, 2007.

[79] R.M. Summers J. Yao. Detection and segmentation of colonic polyps on haustral folds. *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 900–903, 2007.

Appendices

A. Gradient Vector Flow

The GVF is defined as the vector field $F_{ext} = (F_x, F_y)$ which minimizes the function

$$\mathcal{E} = \iint_I \mu \left(\left(\frac{\partial F_x}{\partial x} \right)^2 + \left(\frac{\partial F_x}{\partial y} \right)^2 + \left(\frac{\partial F_y}{\partial x} \right)^2 + \left(\frac{\partial F_y}{\partial y} \right)^2 \right) + |\nabla e|^2 |F_{ext} - \nabla e|^2 \, dx \, dy$$

We wish to minimize the functional according to the function F_{ext} defined as

$$
\begin{aligned}
F_{ext}: \quad \mathbb{R}^2 \quad &\longrightarrow \quad \mathbb{R}^2 \\
(x,y) \quad &\longmapsto \quad F_{ext}(x,y) = (F_x(x,y), F_y(x,y))
\end{aligned}
\tag{A.1}
$$

To simplify the notation, the partial derivative of any function ϕ according to any variable u will be denoted by

$$\partial_u \phi \quad = \quad \frac{\partial \phi}{\partial u} \tag{A.2}$$

We now define a function g such as

$$
\begin{aligned}
& g\left(F_{ext}, x, y, \partial_x F_x, \partial_y F_x, \partial_x F_y, \partial_y F_y \right) = \\
& \mu \left((\partial_x F_x)^2 + (\partial_y F_x)^2 + (\partial_x F_y)^2 + (\partial_y F_y)^2 \right) + |\nabla e|^2 |F_{ext} - \nabla e|^2
\end{aligned}
\tag{A.3}
$$

with:

$$|\nabla e|^2 \quad = \quad (\partial_x e)^2 + (\partial_y e)^2 \tag{A.4}$$

$$|F_{ext} - \nabla e|^2 \quad = \quad (F_x - \partial_x e)^2 + (F_y - \partial_y e)^2 \tag{A.5}$$

The function $\mathcal{E}$ is minimal if the equations of Euler-Lagrange are verified:

$$\frac{\partial g}{\partial F_x} - \frac{\partial}{\partial x}\left(\frac{\partial g}{\partial(\partial_x F_x)}\right) - \frac{\partial}{\partial y}\left(\frac{\partial g}{\partial(\partial_y F_x)}\right) = 0 \tag{A.6}$$

$$\frac{\partial g}{\partial F_y} - \frac{\partial}{\partial x}\left(\frac{\partial g}{\partial(\partial_x F_y)}\right) - \frac{\partial}{\partial y}\left(\frac{\partial g}{\partial(\partial_y F_y)}\right) = 0 \tag{A.7}$$

Equation A.6 yields:

$$2\left((\partial_x e)^2 + (\partial_y e)^2\right)(F_x - \partial_x e) - 2\mu(\partial_{x^2} F_x + \partial_{y^2} F_x) = 0 \tag{A.8}$$

$$\left((\partial_x e)^2 + (\partial_y e)^2\right)(F_x - \partial_x e) - \mu\nabla^2 F_x = 0 \tag{A.9}$$

Similarly, equation A.7 gives

$$\left((\partial_x e)^2 + (\partial_y e)^2\right)(F_y - \partial_y e) - \mu\nabla^2 F_y = 0 \tag{A.10}$$

In order to solve those two equations, we make them dynamic by adding a time variable. For example, on the x direction, we obtain

$$\partial_t F_x = \left((\partial_x e)^2 + (\partial_y e)^2\right)(F_x - \partial_x e) - \mu\nabla^2 F_x \tag{A.11}$$

We discretize in the time domain using an explicit scheme:

$$\gamma(F_x(k+1) - F_x(k)) = \left((\partial_x e)^2 + (\partial_y e)^2\right)(F_x(k) - \partial_x e) - \mu\nabla^2 F_x(k) \tag{A.12}$$

where γ is the invert of the time step. The discretization in the space domain is done using the pixel grid of the image. We use classical discrete operators for the calculation of the derivatives.

$B.$ 2D projections of the 3D B-snake

The series of 40 images used in Section 6.5 to construct the 3D model of the colon are represented below. On each image, the 2D contour that was used in the reconstruction of the model is represented.
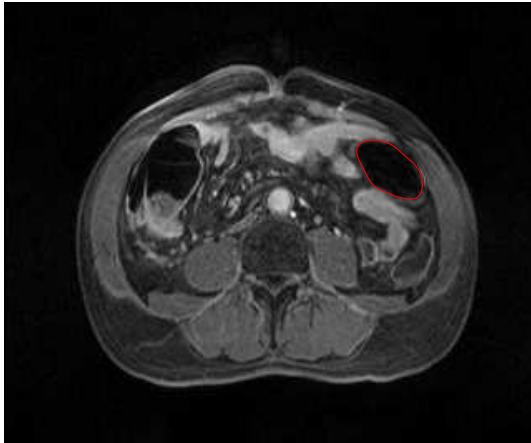
1
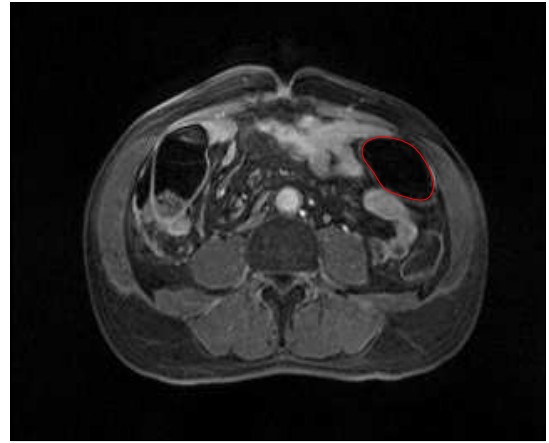
2
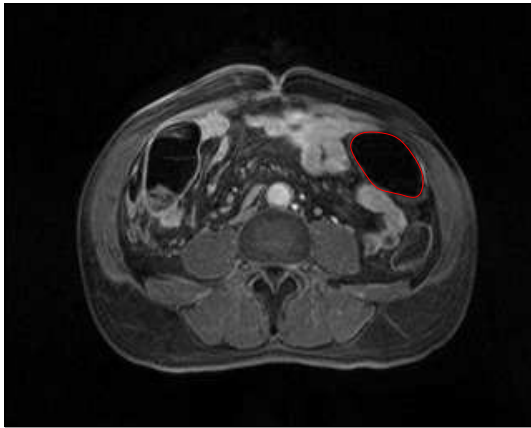
3
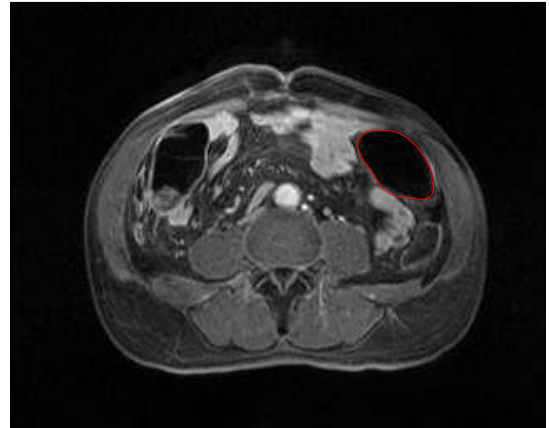
4
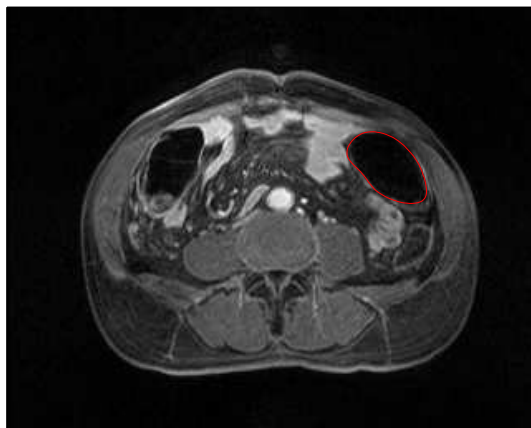
5
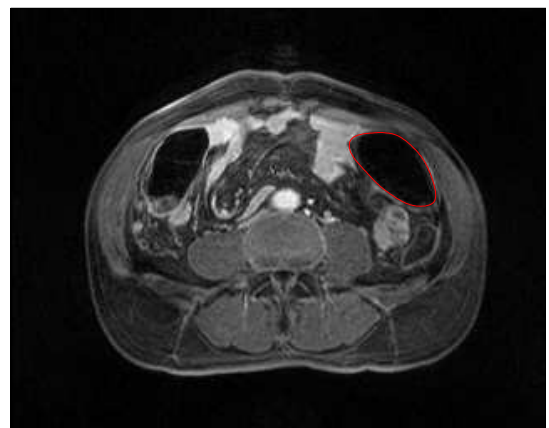
6

Figure B.1: 2D projections (1)
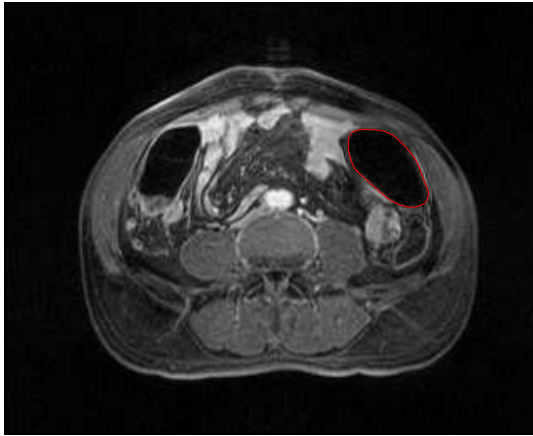
7
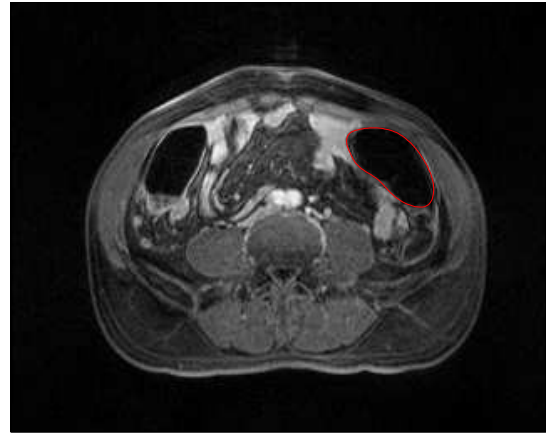
8
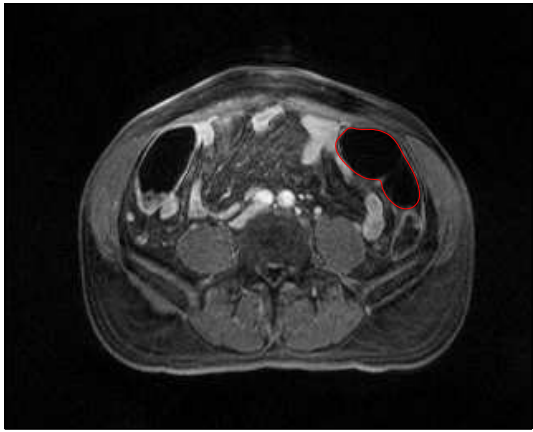
9
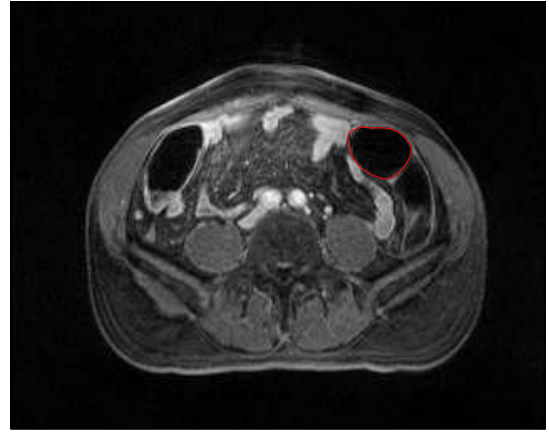
10
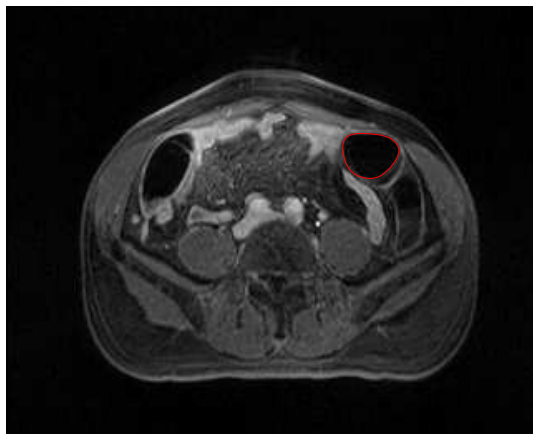
11
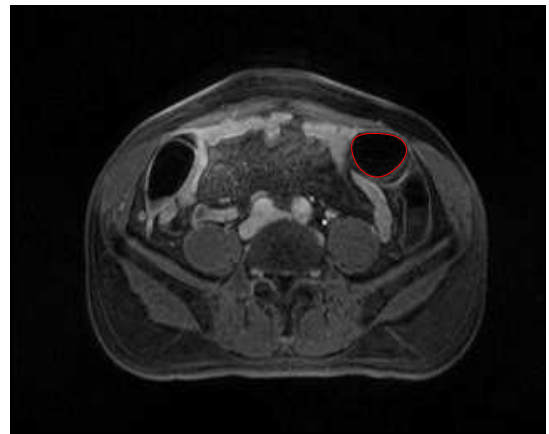
12

Figure B.2: 2D projections (2)
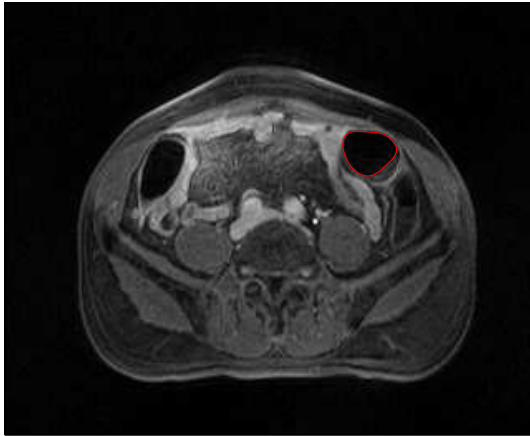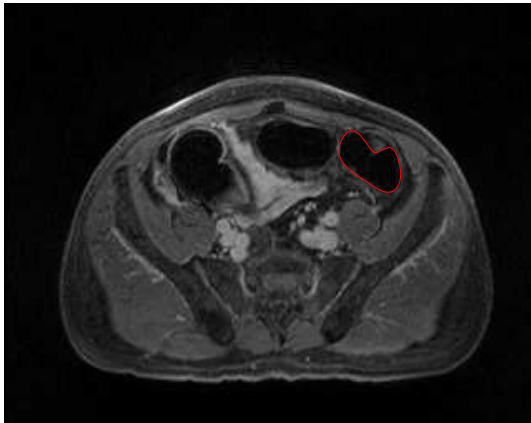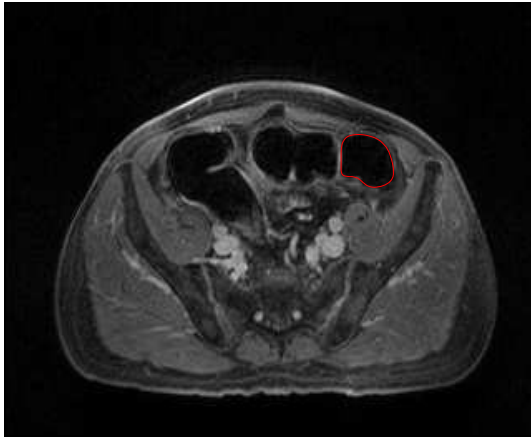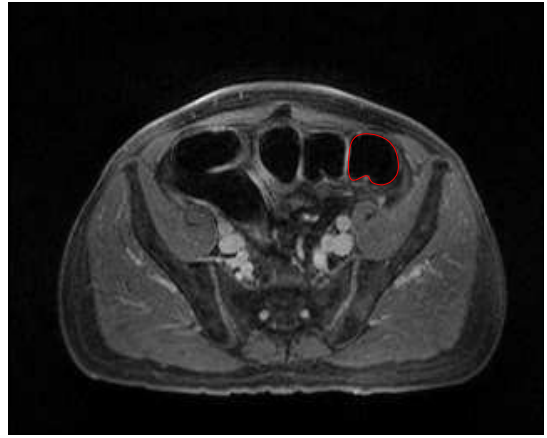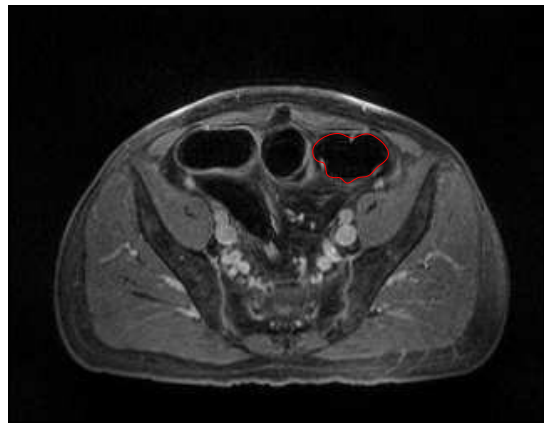
13
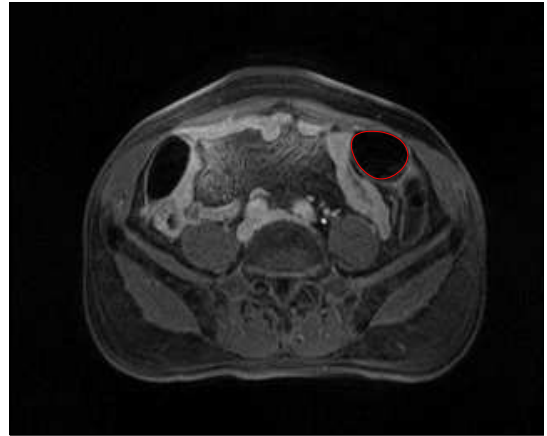
14
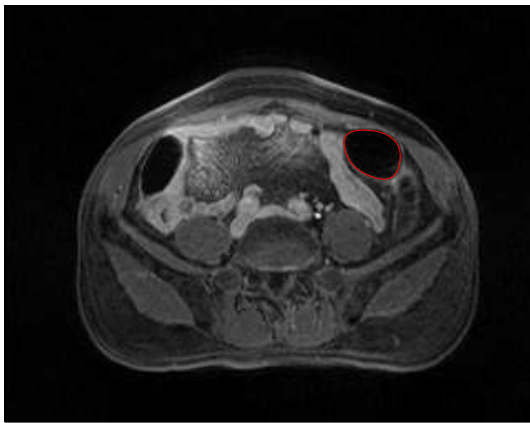
15
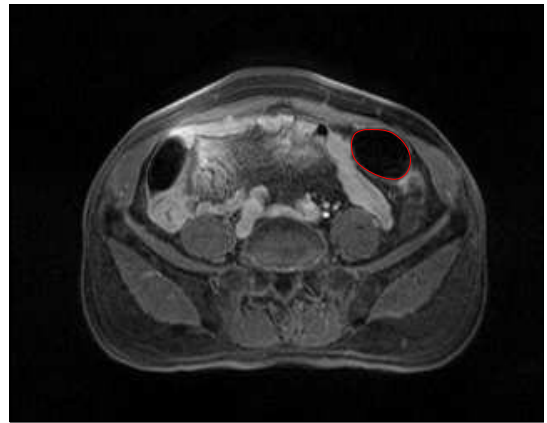
16
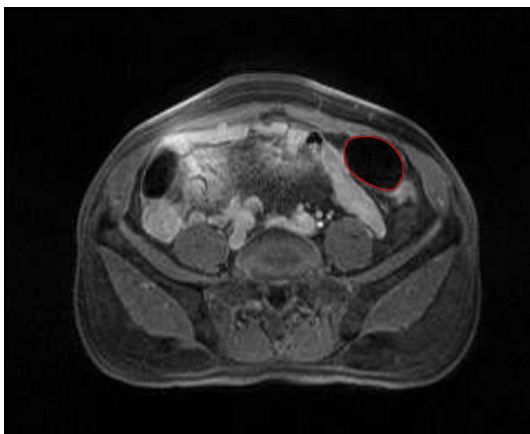
17
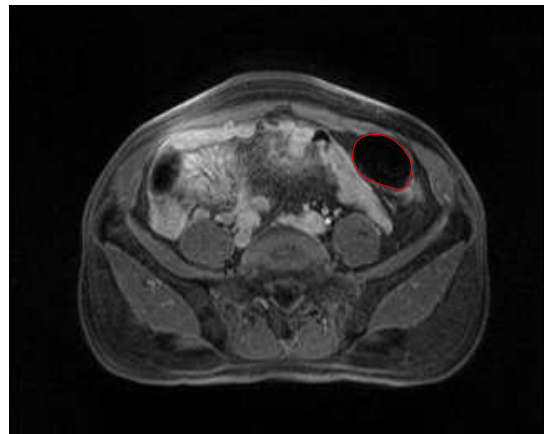
18

Figure B.3: 2D projections (3)
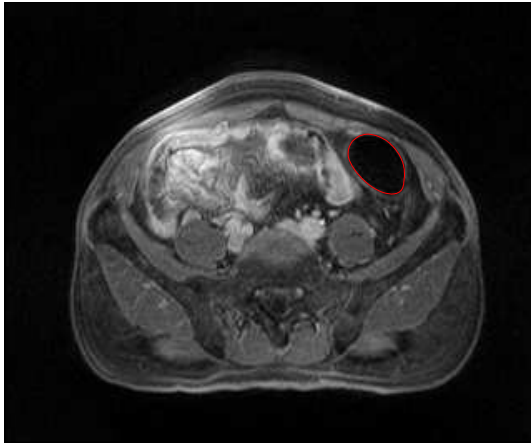
19
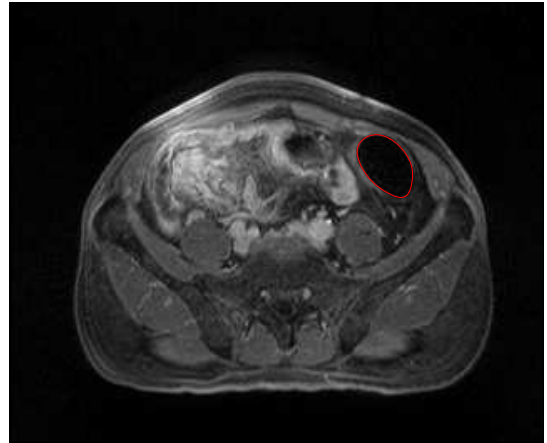
20
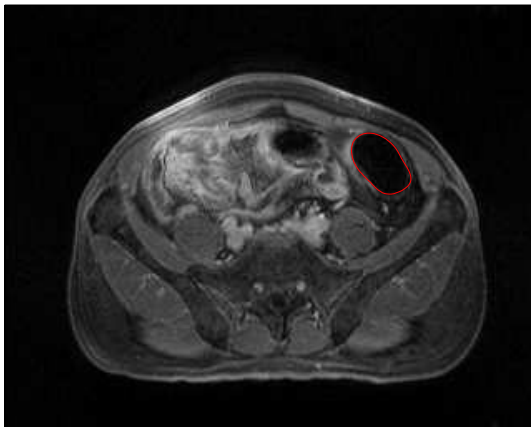
21
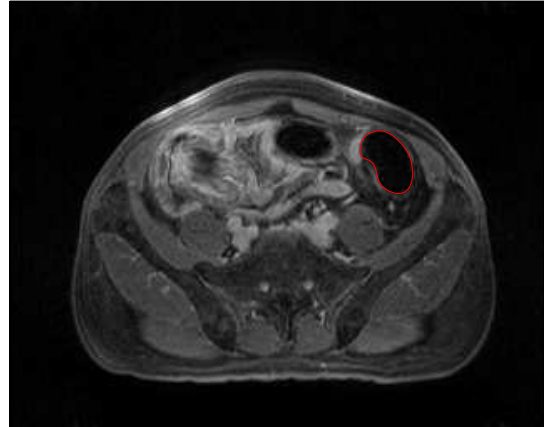
22
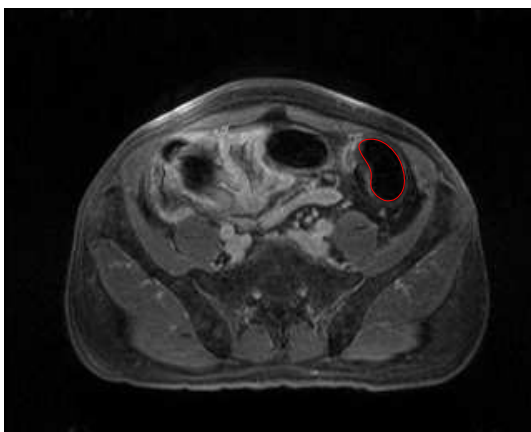
23
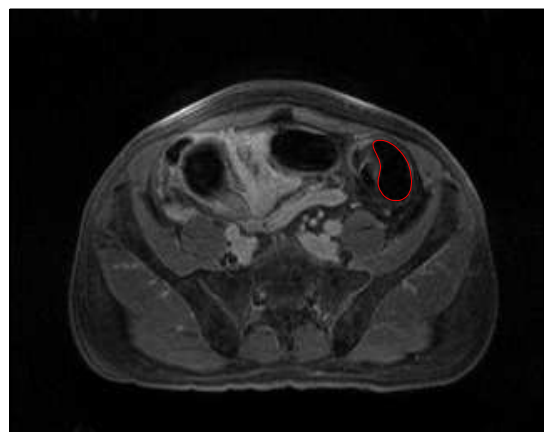
24

Figure B.4: 2D projections (4)
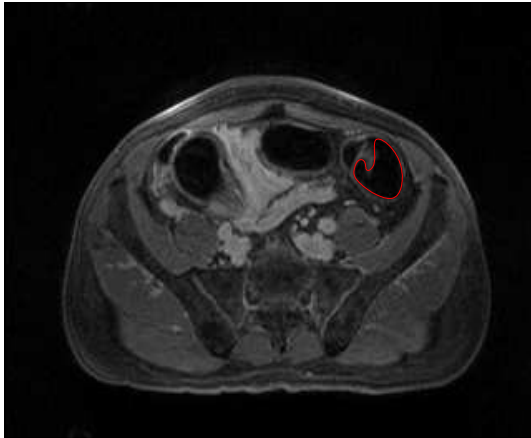
25
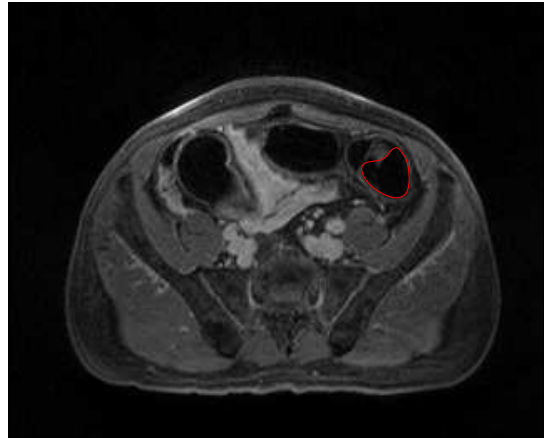
26

27
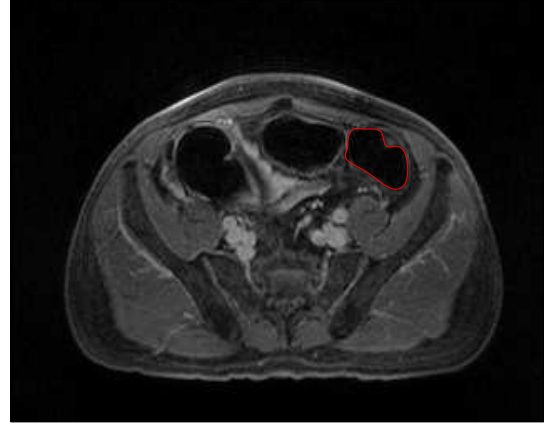
28
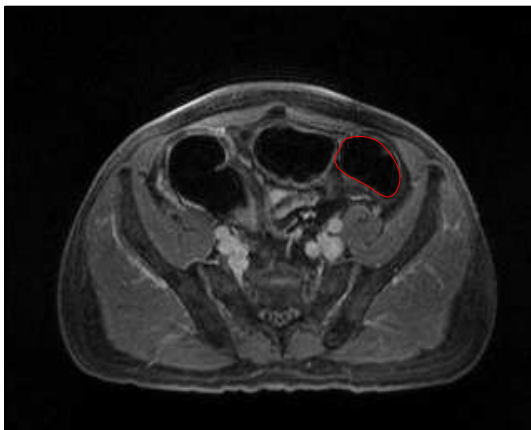
29
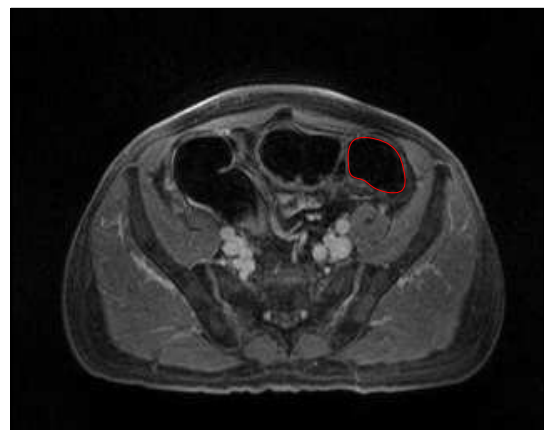
30

Figure B.5: 2D projections (5)
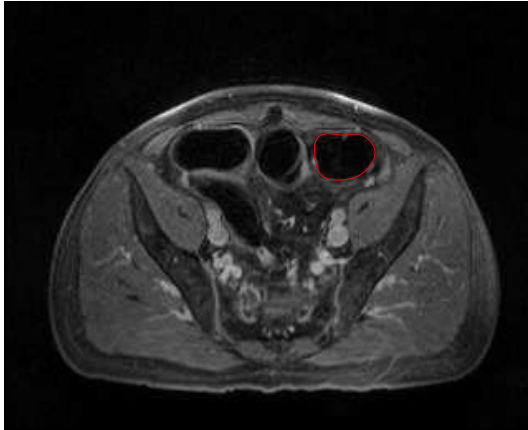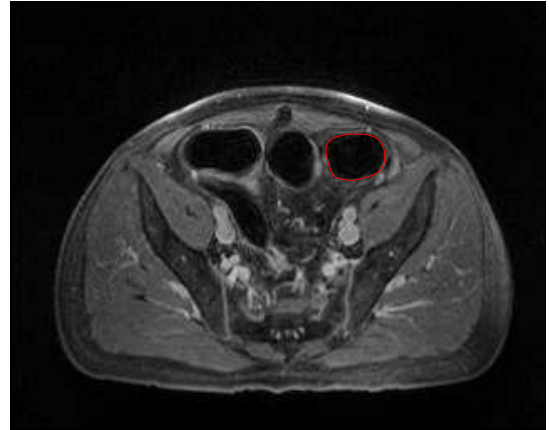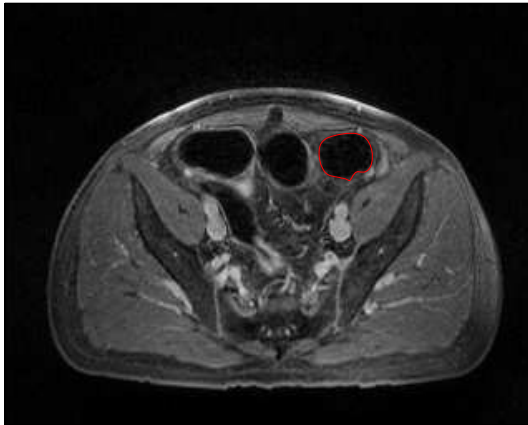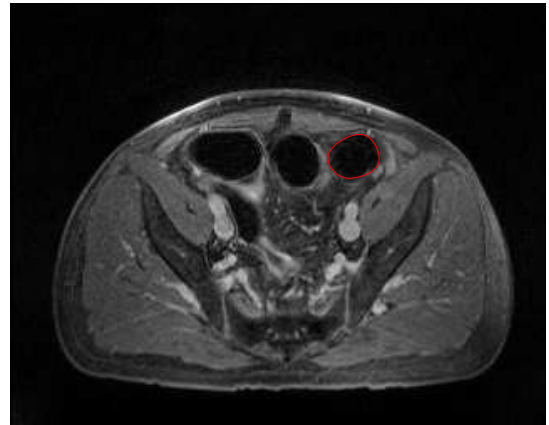
31

32
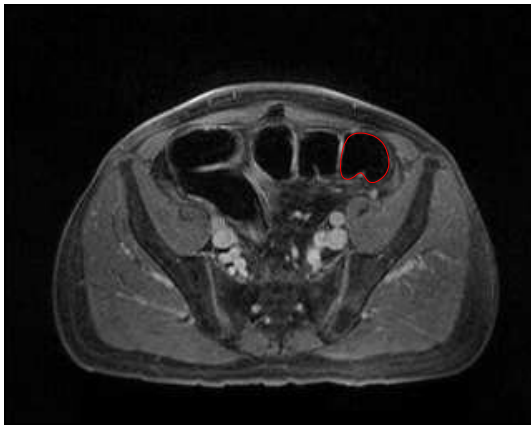
33
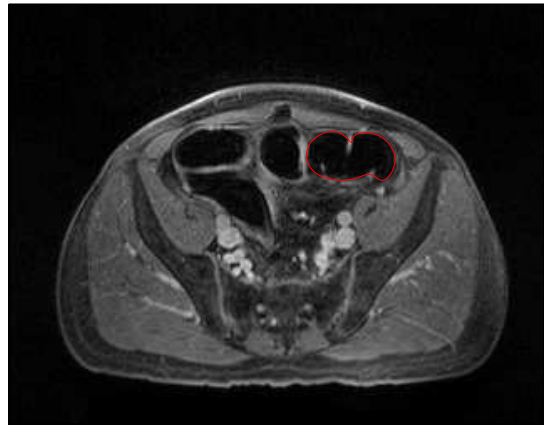
34
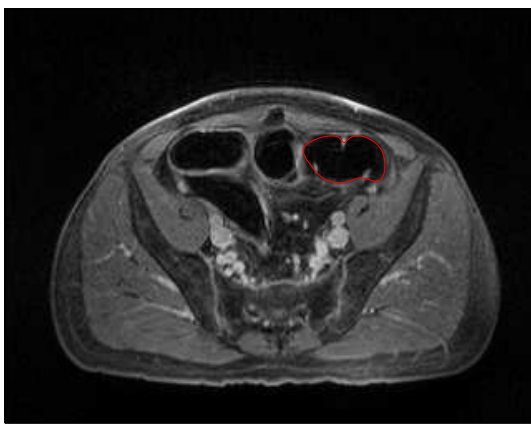
35

36

Figure B.6: 2D projections (6)

37

38

39

40

Figure B.7: 2D projections (7)