The University of Hull

Department of Computer Science

MSc Technical Progress Report

Medical Image Segmentation using Immersive Visualization

Olivier Tilloy

March 2006

Research Supervisor: Professor Roger Phillips
Chairperson: James Ward

Contents

List of Figures

Chapter 1

Introduction

1.1 Scope and context

Image segmentation is an image analysis process (Wikipedia: the free encyclopedia 2006*c*) that aims at partitioning an image into several regions according to a homegeneity criterion (a complete mathematical definition can be found in section 2.1).

Image segmentation is a very complex task, which benefits from computer assistance, and yet no general algorithm exists. It has been a research field in computer science for more than 40 years now, and the early hope to find general algorithms that would achieve perfect segmentations independently from the type of input data has been replaced by the active development of a wide range of very specialised techniques. Most of the existing segmentation algorithms are highly specific to a certain type of data (Haralick and Shapiro 1985, Spirkovska 1993, Lucchese and Mitra 2001), and some research is pursued to develop generic frameworks integrating these techniques (Zouagui et al. 2004).

Segmentation can be a fully automatic process, but it achieves its best results with semi-automatic algorithms, i.e. algorithms that are guided by a human operator (Bhanu and Fonder 2000, Blake et al. 2004, Hinshaw and Brinkley 1997). This concept of semi-automatic process naturally involves an environment in which the human operator will interact with the algorithms and the data in order to produce optimal segmentations. The simplest example of the need of a human intervention during the task of segmentation results from the specificity of the existing algorithms: depending on the type of input data, the operator will have to carefully pick the best adapted algorithm, which most of the time cannot be done in an automatic way. The subjective point of view of the human is required.

Image segmentation has become an essential tool in the medical field (Hinshaw and Brinkley 1997, Elliott et al. 1992, Warfield et al. 1998). With the generalisation of diagnosis using Magnetic Resonance Imaging (referred to as MRI), image segmentation is often required to allow doctors and surgeons to analyse the patients' data, e.g. prior to surgery to determine the exact location of an organ or a tumour.

MRI, also known as Nuclear Magnetic Resonance, is a method used mainly (but not only) in medical applications to visualise the insides of a patient in a harmless fashion. It relies on the relaxation properties of excited hydrogen nuclei in water (Wikipedia: the free encyclopedia 2006*b*) after the body or part of the body to image has been placed in a powerful and uniform magnetic field. The obvious benefit of this technique is its harmless character, compared to other techniques such as CT scans and X-rays in which the patient is exposed to ionising radiations. 3D and 4D (3D + time) MRI is increasingly used in diagnosis and therapy (Elliott et al. 1992).

The data acquired from MRI is generally presented as a volumetric image that can be viewed as a series of slices following one of the 3 axes : sagittal (following the X axis), coronal (following the Y axis) or axial (following the Z axis). The aim of the segmentation process is to locate with the highest accuracy possible the boundaries of a special part of the image (an organ, a tumour...), thus allowing the use of 3D information for planning of treatment, navigation or visualisation. Once this task is achieved, simulation, pre-computations or training can be performed to prepare the real operation.

Haptic is a word derived from the Greek *Haptai* and refers in its general meaning to the sense of touch. From a technological point of view, haptics aim at providing interfaces between humans and machines via the sense of touch (Wikipedia: the free encyclopedia 2006*a*). More specifically, haptics in virtual reality applications add to the traditionnal visual and audio worlds the possibility to feel objects and to interact with them by touching, and possibly dragging or deforming them. Today's applications mainly use haptic styli (a complete description of the haptic device used in this research work is given in section 5.1).

This research project is undertaken within the Simulation and Visualization Research Group of the University of Hull's Computer Science department. The research being pursued within the group includes a wide range of innovative technologies with special expertise in computer graphics, visualization, mathematical modelling, computer-based simulation, digital cartography, terrain/seabed visualization, virtual environments, position sensing, haptic devices, imaging and their skills and industrial experience in the engineering of large complex computer-based systems (University of Hull's Computer Science Department 2005). This research project naturally integrates into the wide range of the medical group's interests, among which are image guided surgery, surgical training, imaging and diagnosis, and anatomical simulation.

1.2 Justification of research

The hypothesis of this MSc research work is whether the suitable use of an immersive environment can improve (both in terms of accuracy, speed and user experience) classic 2D slice-based image segmentation techniques. By *immersive environment*, it is meant an environment that makes use of virtual reality devices and techniques to provide the user an improved experience (Vidal et al. 2004). Such an environment, in the case of this research work, is composed of a 3D visual space in which the user can interact with entities with a haptic device (here a stylus with six degrees of freedom).

The emphasis will be put on trying to demonstrate the benefits of the use of a haptic device rather than on building a revolutionary 3D segmentation environment. The idea behind is that 2D slice by slice segmentation works quite well and does not need a lot of training to achieve very good results, but is time consuming. On the other hand, in general full 3D environments are able to display much more information but are thus rarely intuitive and easy to familiarise with. By adding the sense of touch to a 2D slice by slice based segmentation environment, it is hoped to dramatically improve the user experience and reduce the time needed to perform a segmentation, while providing the user an instinctive and easy to use tool.

1.3 Aims and objectives

As stated in section 1.2, the aim of this research work is to investigate whether building a haptically enabled medical image segmentation environment can notably improve the overall process of image segmentation in general, enhance the user experience and reduce the processing time in particular. The project is split up into a set of objectives, the results of which, together, will help verify, or invalidate, the hypothesis.

These objectives are:

- Propose a number of possible uses of haptics that could improve the user feedback and experience;
- Map these defined uses onto a selected segmentation technique;
- Develop a segmentation environment implementing these haptic techniques coupled with a segmentation algorithm;
- Assess the benefits by defining an evaluation framework. This framework will have to allow comparisons at different points of view and a real valuation of the improvements and/or drawbacks of the tool and compare with existing segmentation techniques.

All of these objectives are tightly linked. Therefore, they must be seen as a whole, and the fulfilment of each of them will contribute to the overall aim of the project.

1.4 Thesis plan

The organisation of this thesis is as follows:

Chapter 2 presents in detail the background knowledge required to undertake this piece of research: the mathematical definition of image segmentation, a review of existing segmentation techniques, an overview of existing segmentation environments and an introduction to the theory of user interaction in the segmentation of medical images.

Chapter 3 details the integration of haptics into the process of image segmentation by analysing the stakes of a successful integration and studying this integration with a region-growing algorithm.

Chapter 4 deals with specific considerations about the design of a haptically enabled segmentation environment. The interface between the human operator and the machine is defined together with the mapping of the data with haptic forces, and a section is dedicated to user interface considerations.

Chapter 5 reviews the hardware and software bases of the project: the haptic device used, the API handling haptics and graphics and the toolkit in charge of the segmentation of images.

Chapter 6 presents the architecture of the project from an engineering point of view and the developed tools.

Chapter 7 applies the developed tools in a purposefully designed framework to assess the benefits of the project by comparing it to classic 2D segmentation environments.

Finally, chapter 8 draws conclusions about the work done and proposes directions for future work.

Chapter 2

A review of background knowledge

2.1 A mathematical definition of segmentation

The following is a very general definition of image segmentation (Spirkovska 1993). It uses a homogeneity predicate $P()$ that helps formalising the notion of homogeneity in an image: a region R is homogeneous if and only if $P(R) = True$. Therefore, the homogeneity can be defined in an infinity of different ways: on the grey levels, on the textures or even on non-obvious properties of the image.

Definition 1 (segmentation) *Let I be the set of pixels (the input image) and P() the homogeneity predicate defined on groups of connected pixels.*
A segmentation S of I is a partitioning set of image regions $\{R_1, R_2, \ldots, R_n\}$ *such that*

$$\bigcup_{i=1}^{n} R_i = I \ and \ R_i \cap R_j = \emptyset \ \forall \ i \neq j \tag{2.1}$$

and such that the homogeneity predicate P() satisfies

$$P(R_i) = True\ \forall\ i \tag{2.2}$$

$$P(R_i \cup R_j) = False\ \forall\ R_i\ adjacent\ to\ R_j \tag{2.3}$$

$$(R_i \subset R_j) \wedge (R_i \neq \emptyset) \wedge (P(R_j) = True) \Rightarrow (P(R_i) = True) \tag{2.4}$$

Equation 2.1 is a mathematical definition of a partition: the union of all the regions form the whole image and all the regions are distinct.

Equation 2.2 signifies that the homogeneity predicate is valid on every region.

Equation 2.3 signifies that the union of two adjacent regions cannot satisfy the homogeneity predicate, i.e. two adjacent regions *must* be distinct regarding the homogeneity predicate.

Finally, Equation 2.4 signifies that the homogeneity predicate is valid on any sub-region of a region where it is verified.

2.2 A review of existing segmentation techniques

As stated in section 1.1, a wide range of very specialised segmentation techniques currently exist and since the research is very active in this field, the panel of available techniques and algorithms constantly evolves. Therefore, a complete study that would review all the state-of-the-art techniques is not relevant in the context of this document. Instead, this section tries to present a simple yet homogeneous and relevant classification of the existing techniques into a number of families. For each family the general functional philosophy is analysed and a non-extensive list of algorithms is presented, with a short explanation of the specificities for each of them.

There are numerous types of classifications proposed in the specialised literature, each of which is relevant respectively to the point of view required by the study. Since this research project deals with medical image segmentation, where a large majority of the acquired data is grey-scaled, all the techniques concerning colour images will be left aside.

Spirkovska (1993) separates the techniques into three main families:

- Pixel based techniques (also known as histogram thresholding);
- Edge based techniques;
- Region based techniques.

This classification is very commonly encountered in numerous papers.

Dr Singh (2006) defines the following more detailed classification:

- Histogram thresholding;
- Edge based segmentation;
- Tree/graph based approaches;
- Region growing;
- Clustering;
- Probablistic and Bayesian approaches;

- Neural networks segmentation;
- Other approaches.

In a well-known article, Haralick and Shapiro (1985) review a large number of existing segmentation techniques, including spatial clustering, thresholding, region growing and split and merge.

Ibáñez et al. (2005) choose to divide the segmentation techniques in four families:

- Region growing;
- Segmentation based on watersheds;
- Level set segmentation;
- Hybrid methods.

Most of the classifications listed above have in common certain families of techniques. The classification proposed by Spirkovska (1993) has the advantage of being simple; yet it allows the review of most of the techniques of interest in the scope of this research work. Therefore the following sections will focus on the presentation of the techniques according to this classification.

2.2.1 Histogram thresholding

The pixel-based family of techniques is probably the most simple one, it essentially consists in finding an acceptable threshold in the grey levels of the input image in order to separate the object(s) from the background. It is often referred to as histogram thresholding since the grey-levels histogram of an ideal image will clearly show two distinct peaks assimilable to Gaussians (which can be obtained by applying a filter to the image) representing the distribution of grey levels for one object and its background. This kind of histogram is sometimes referred to as *bimodal*.

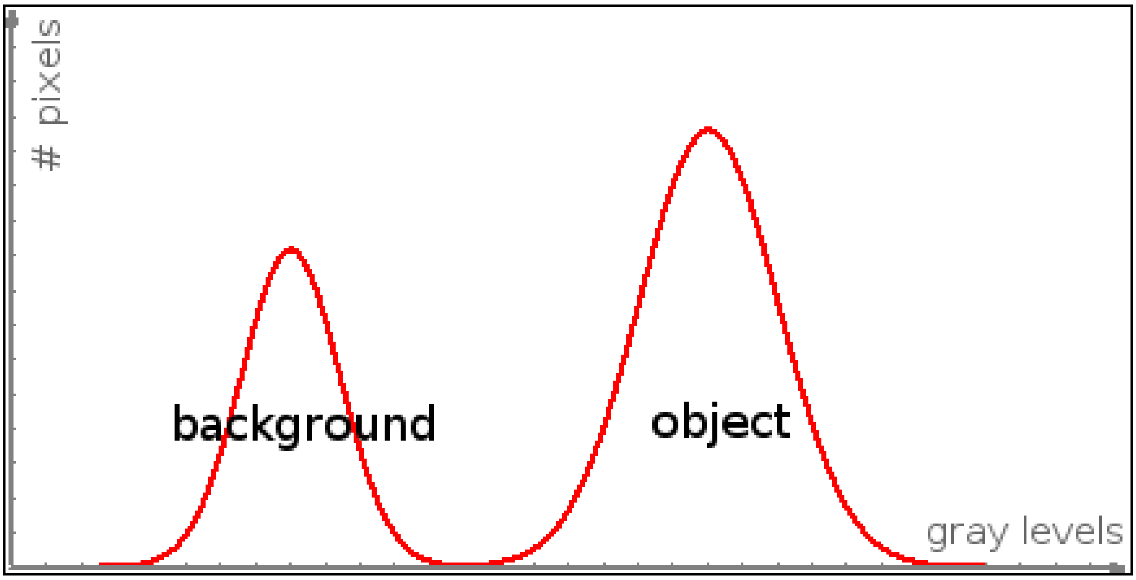

Figure 2.1: A bimodal histogram

The threshold value for an image can be computed by using several different methods, like the Gaussian filtering (Jain and M-P. 1992) or Otsu's (1979) method.

2.2.2 Edge-based segmentation

The edge-based family of techniques tries to detect edges in an image so that the boundaries of the objects can be inferred. The most simple method of this type is known as *detect and link*: the algorithm first tries to detect local discontinuities and then tries to build longer ones by connecting them, hopefully leading to closed boundaries which circumscribe the objects in the image. The main disadvantage of this technique lies in the fact that, depending on the quality of the input image, the algorithm is not guaranteed to produce closed edges. As a consequence, the image will not sharply split into regions. Some improvements for this method have been proposed in order to overcome this type of issue (Yanowitz and Bruckstein 1989, Parker 1991).

2.2.3 Region-based segmentation

The region-based family of techniques fundamentally aims at iteratively building regions in the image until a certain level of stability is reached.

The region growing algorithms start from well chosen seeds (usually defined by the user). They then expand the seed regions by annexing their homogeneous neighbours. The process is iterated until all the pixels in the image have been classified.

The region splitting algorithms use the entire image as a seed and splits it into regions until no more heterogeneity can be found.

An algorithm that associates the advantages of both methods, called the *Split, Merge and Group* (SMG) algorithm, has been developed by Horowitz and Pavlidis (1976).

2.3 Segmentation environments

An image segmentation environment is a set of combined tools that provide the human operator (a technician, a phycisist, a doctor, a computer scientist, etc.) everything needed to perform an actual segmentation on an input data set and produce a usable result. The type of tools included can vary, but one will usually find:

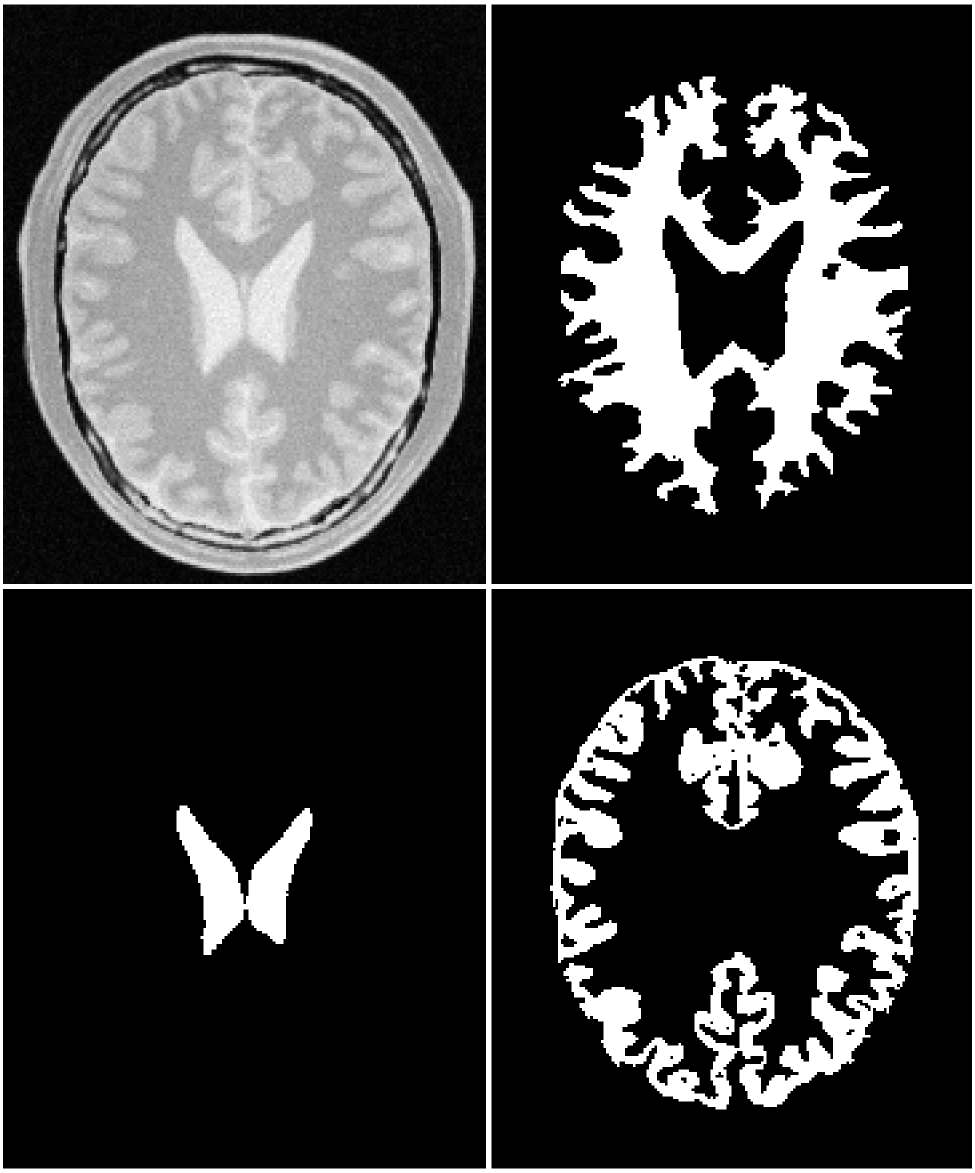

Figure 2.2: Execution of a region growing algorithm with different initial seeds. Top-left: the original image. Top-right: the white matter. Bottom-left: a ventricle. Bottom-right: the grey matter.

- A data reader, able to read and load in memory the input data;
- A data visualiser, in order to help the operator view the data, possibly from different viewpoints and/or different modalities (2D or 3D);
- Some filters, useful in case the input data is noisy (i.e. it has artifacts due to the quality of the acquisition);
- One or more segmentation algorithms to select from;

- An interaction procedure by which the operator *guides* the algorithm in case of an interactive segmentation process;
- A segmentation visualiser;
- Some refining tools, to allow the user to manually edit the results in case of local segmentation errors;
- An output writer, to export the result in a format that can be later processed by other tools.

Until fairly recently, this type of environment was exclusively in two dimensions, i.e. making use of classic devices such as a screen, a keyboard and a mouse. With the recent breakthrough of haptics (allowed by the increasing capacities of hardware), researchers have developed 3D segmentation environments (Vidholm and Agmund 2004, Vidholm et al. 2004, Harders and Székely 2002, Senger 2005).

2.3.1 2D segmentation environments

Two-dimensional segmentation environments were until recently the only existing solution. The limitations of a 2D space normally restrict working to a slice-by-slice basis. A 3D volumetric image is then divided in slices along an axis (see section 1.1) and each slice, being a 2D pixel map, has to be segmented individually. Then the overall result can be assembled to compute 3D surfaces marking the boundaries of the objects of interest.

An example of this type of environment is **amira** (*amira 3.1 User's Guide and Reference Manual* n.d.). **amira** is in fact rather a visualisation environment for 3D data sets that includes basic segmentation tools. An interesting feature provided by **amira** is the possibility to interpolate the segmentation on a range of slices. It uses the fact that very often two consecutive slices in a data set slightly differ from each other. By carefully selecting correctly segmented key slices, the segmentation can be propagated to the whole set, saving a lot of time.

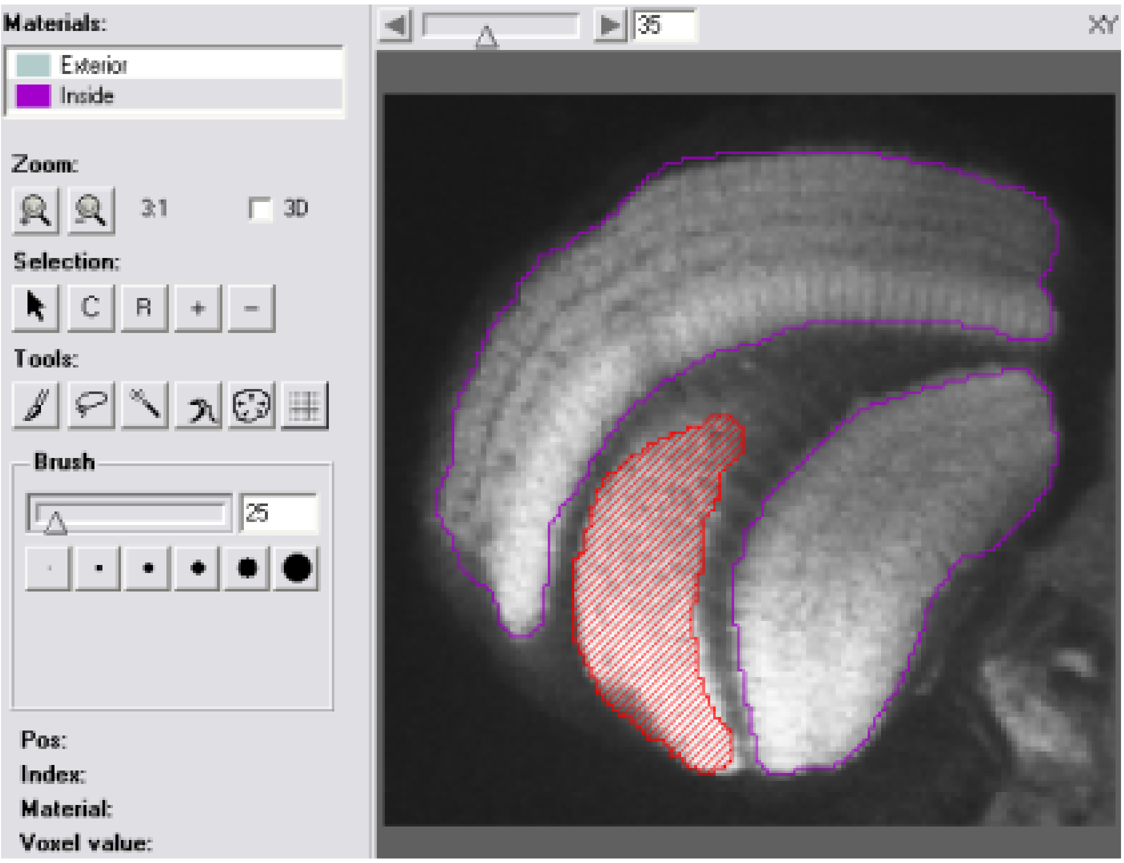

Figure 2.3: **amira**'s segmentation tool

The main advantage of such a type of segmentation environment is its relative intuitiveness. The human mind is used to seeing images in 2D. Viewing and understanding an image as a volume requires some experience which can only be acquired by training. Therefore most of the time this type of tool is relatively easy to familiarise with.

The major obvious drawback is the time-consuming character of the process. It is a negligible factor when segmenting a single image, but when dealing with large volumetric data sets with hundreds of slices, it becomes tedious and even sometimes unacceptable in terms of time.

2.3.2 3D segmentation environments

In the past five years, with the fast gains in computational power allowed by the increasing capacities of hardware, haptic devices have found new applications which were unimaginable a few years ago. Among those applications is the segmentation of volumetric data sets using the haptic feedback.

A distinction has to be made between real immersive environments and 3D environments used on classical devices. Indeed, a 3D scene projected and displayed on a flat screen that can be browsed through with classical input devices such as a mouse can already be considered as a 3D environment. In the context of this research however, a 3D environment will always refer to an *immersive environment* (Vidal et al. 2004) that provides an enhanced experience through the use of stereoscopic vision and a haptic device for interaction.

The haptic device being very often coupled to a 3D capable display, the user, equipped with special glasses that provide stereoscopic vision, can really view through the data and interact with it using the haptic device (often a stylus). The haptic capabilities can be used in numerous different ways; the following shortly describes some research works dealing with the subject.

Vidholm and Agmund (2004) developed a haptically enabled application that allows the visualisation of the data in 3D, its segmentation and possibly the manual editing of the result. In this context, the haptic device is used to initialise a fast marching (region growing) segmentation algorithm by placing a seed, and then to manually edit the result by using the stylus as a tool that can draw, erase, erode or dilate regions directly in the resulting volume.

Harders and Székely (2002) define a framework for the segmentation of tubular structures (such as bowels). The haptic device is used to extract the centerline of a tubular structure, and from this centerline the tool generates a cylinder with a varying diameter that can be used as the initial guess for a deformable surface.

Senger (2005) proposes a complete haptically enabled segmentation and visualisation environment. This environment allows the use of two distinct segmentation algorithms, where the haptic feedback is used to provide a sense of touch over the segmented objects and to dynamically guide the segmentation process, by selecting the direction in which a region should expand in the case of the seeded region growing algorithm.

The development of 3D segmentation environments using haptic feedback is a major aspect of the research on medical image segmentation. Some promising improvements have been made in the past few years, but these techniques are still highly experimental and under development. The benefits brought by the use of such environments are numerous, they allow a true immersion into the data, thus giving a better perception of it, and they introduce the sense of touch which, if correctly used, can greatly enhance the perception as well as the segmentation process itself.

On the other hand, the majority of specialists are used to segmenting data slices in 2D, therefore the adaptation necessary to learn to use a totally new segmentation environment is not negligible. As stated by Harders and Székely (2002): "*Editing, controlling and interacting in three dimensions often overwhelms the perceptual powers of a human operator*".

2.4 Theory of user interaction

Image segmentation techniques can be classified from an interaction point of view into three families:

- Manual segmentation where all the work is manually done by the operator;
- Fully automatic segmentation where a data set and some parameters are given as input and the segmentation is computed as output, requiring no intervention from the operator;
- Semi-automatic segmentation where the operator can interact with the algorithm in charge of the segmentation during its execution, allowing to benefit from the operator's knowledge of the data.

The aim of this study is to combine haptic interaction with semi-automatic segmentation. When speaking of *user interaction*, one often has a general idea of the concepts involved. Too rarely are the exact mechanisms precisely defined, whereas such a definition is fundamental to achieve a good understanding of the interaction.

Olabarriaga (1999) provides an extensive study on the interaction in image segmentation, particularly when dealing with medical applications. This study includes a review of the types of interaction inputs and a discussion on the purpose of interaction.

Interactive image segmentation can be viewed as a pipeline. At its front end, the input data and some input parameters, and at its other end, the output result. Along it, the segmentation algorithm and the user interacting.

Basically, the operator can interact in five differents ways. He can:

- Set the parameters at the initialisation (requires a knowledge of how the parameters are used by the algorithm);
- Judge the result by accepting or rejecting it (binary choice);
- Correct the result by manually editing it;
- Tune some parameters to locally improve the result in a selected area;
- Compose the result, i.e. the initial algorithm provides a set of primitives and lets the user combine these primitives.

To the above five possibilities it is possible to add the one described by Senger (2005) in which the operator dynamically guides the growing of a region. To some extent though, it can be seen as a multiple-choice question answered by the operator in a more instinctive way.

All these types of interactions will be studied and possibly integrated into the technical developments of this project in chapters 3 and 4. As Olabarriaga (1999) suggests, it is already possible to define some guidelines regarding the emerging trends that should ensure the quality of the developed tool:

- Use pictorial input for instinctive interaction;
- Use high-level interaction tools for intuitive operation;
- Keep the user in control of the segmentation process to guarantee faithful results;
- Involve the user in the initialisation of the segmentation process to provide information that can bootstrap or lead the method to the desired segmentation result more quickly;
- Learn from user interventions for long-term efficiency.

2.5 Summary

Image segmentation is often considered as a low-level operation (Lucchese and Mitra 2001), sometimes as a bridge between low-level vision and high-level vision (Spirkovska 1993). It is a preliminary process to higher level operations: image segmentation is a tool that has applications in many varied fields.

Different classifications have been proposed in the technical literature. A simple one has been presented in this chapter, which allows the review of most of the key concepts and techniques that will be used in the following chapters of this thesis.

To perform image segmentation, a technique or an algorithm is generally not enough: one needs a segmentation environment that provides a set of useful tools to efficiently process the data. A review of some of the existing types of segmentation environments has been presented in this chapter in order to list the advantages and drawbacks of each of them and help in the conception of an improved immersive image segmentation environment.

User interaction is at the heart of this project, as the quality of the proposed interface will directly influence the quality of the whole environment and thus the success of the project. A general theory of user interaction has been presented and the next chapters will apply this theory to the concrete conception and implementation of an immersive image segmentation environment.

Chapter 3

Integrating haptics into image segmentation

This chapter discusses the objectives of integrating the haptic feedback into an image segmentation process and the means that will be employed to successfully reach these objectives, first from a theoretical point of view, then with more details, the segmentation technique being a seeded region growing algorithm (this choice being justified in the concerned section).

3.1 The stakes of a successful integration

The first question that naturally arises concerns the definition of *integration*. The task of integrating haptics should not be seen as a completely separate work that comes after the development of the whole environment. Due to haptic interaction being a key feature of this project, it will be built around the core haptic functionalities. To this extent, the word *integration* is inappropriate. However, the project is also based on a segmentation process, and to this extent the haptics will indeed be integrated into this existing basis.

The haptic feedback is sensed by the operator through a physical probe, in this case a stylus with six degrees of freedom (see section 5.1). The stylus forms the interface between the machine and the human operator (see section 4.1). There are numerous ways of using it, most of which are not so intuitive due to the fact that human operators are used to interacting with simple interfaces such as a mouse (2D positioning + binary states of the buttons) or a keyboard (binary states of the keys, possibly several keys handled at the same time).

The big difference lies in the usability of the haptic stylus both for input *and* output. This is due to the active character of the device (as opposed to the passive characters of a mouse or a keyboard). The input and output are therefore tightly linked and their processing should be thought as a whole where an input can always target some appropriate output and vice versa.

The input part of the user interaction is, despite its 3D character, quite easy to conceive: the end of the stylus can be seen as a cursor (such as the one of a mouse) moving in a 3D space, typically used to select entities (the haptic stylus is equipped with a lateral button) and interact with them or define positions in space.

The output part needs more reflection. The possibilities are almost infinite, only limited by the imagination and the capacity of the operator to interpret the haptic output. The haptic feedback can be used in a *passive* way to help the operator feel some properties of the input data such as the grey level intensity of the pixels in an image by applying an appropriate force, or the gradient of this image at a given position. It can also be used to tune some parameters specific to an algorithm by defining a parameters space (in 3D) where the position of the probe would define a unique combination of initial parameters. Here again, by defining and applying appropriate forces, the tuning of these parameters would hopefully become more instinctive than just selecting a value on a slider bar for a meaningless parameter. As described by Senger (2005), the haptic feedback can be used interactively as input and output when guiding the direction in which the algorithm should next *grow* (in the case of a region growing algorithm, but this concept could be extended to more general families of interactive algorithms). Here the algorithm has its own decision patterns that can be countered by the operator by applying a force in the desired direction. Finally, the output can be used to browse through the result as it was first used to browse through the input data.

3.2 Application to the region growing model

3.2.1 The seeded region growing algorithm

The seeded region growing algorithm is an algorithm first proposed by Adams and Bischof (1994). It belongs to the family of region based algorithms. A certain number of initial regions is manually defined, and these regions are grown until all the pixels in the image have been processed. The definition of the initial regions is achieved by a mechanism of seeds: a seed is a pixel or a small group of pixels contained in a region and whose properties are representative of the characteristics of the region. The seeds are selected by a human operator who should place as many seeds as the desired final number of regions in the segmented image.

The algorithm starts with a set of n seed regions $\{S_1, S_2, \dots, S_n\}$. Each iteration of the main loop of the algorithm will add an unallocated pixel to one of the regions.

Let U be the set of unallocated pixels which border at least one of the regions:

$$U = \left\{ x \in \bigcup_{i=1}^{n} S_i \mid \bigcup_{i=1}^{n} S_i \bigcap N(x) \neq \emptyset \right\}$$

where $N(x)$ is the set of neighbours of the pixel x. The criterion for neighbourhood can vary (4-connectivity, 8-connectivity...). If, for $x \in U$, $N(x)$ meets only one of the S_i then $i(x) \in \{1, 2, \ldots, n\}$ is defined as the index such that $N(x) \cap S_{i(x)} \neq \emptyset$ and $\delta(x)$ is defined as the measure of the difference (in terms of grey levels) between x and $S_{i(x)}$. A simple definition for $\delta(x)$ is:

$$\delta(x) = |g(x) - mean_{y \in S_{i(x)}}[g(y)]|$$

where $g(x)$ is the grey level value of the pixel x. If $N(x)$ meets two or more, $i(x)$ is defined as the value of i that minimises $\delta(x)$. Another possibility introduced by the authors is to classify x as a boundary pixel, which can be useful for display purposes. Then $z \in U$ is chosen such that:

$$\delta(z) = \min_{x \in U} \ \delta(x)$$

and appended to S_i. This process is repeated until all pixels in the image have been classified.

This algorithm is quite stable, simple and efficient, and the only input parameter is the placement of the seeds at the initialisation. This choice of seeds should be carefully carried out. Adams and Bischof (1994) state: "*it is recommended that small seed areas be used (instead of single pixels) when segmenting noisy images. Each seed area should be sufficiently large to ensure that a stable estimate of its region's mean is obtained.*".

The major drawback of the algorithm is its dependency on the order of pixel processing: raster order and anti-raster order processing of the pixels in the image will lead to different resulting regions. This problem has been solved in an improved version of the algorithm proposed by Mehnert and Jackway (1997), in which the dependencies are identified and appropriately eliminated.

3.2.2 Browsing through the data

Among all the previously identified possible uses of the haptic medium, the first one that can be applied to the seeded region growing algorithm (and in fact to any type of algorithm) is a *passive* effect to *render* the data. This type of haptic interaction is called passive in the way that it is only an output, no input from the user being expected. It more precisely consists in giving a haptic rendering of the image data by transcripting the image characteristics (in the simplest way the grey levels of its pixels) into forces that will make the operator literally *feel* the image.

The most obvious and instinctive way to visualise data has long been (and still is) the graphical way. The human vision is, in terms of computer interaction, the most accessible sense. Moreover, the human vision is sensitive and sharp enough to allow the visualisation of very detailled structures. However, the sense of touch is also well developed and can be used as a good supplement (and even as a complete alternative for the visually impaired when it comes to reading and understanding the surrounding environment). The haptic rendering, combined with the graphic rendering, should enhance the perception of the data by adding a sensory dimension.

The computation of the appropriate forces is discussed in details in section 4.2. It makes use of the concept of *haptic texture*.

3.2.3 Placing the seeds

The second type of haptic interaction is *active*: it involves input from the operator. It is, as opposed to the first one, specific to the seeded region growing algorithm. The haptic stylus is seen as a pencil: the operator literally draws on the image. In the case of the seeded region growing algorithm, the expected input is a set of seed regions, which are defined as a pixel or a small group of pixels contained in a region and whose properties are representative of the characteristics of the region (see section 3.2.1).

To concretely implement this interaction, the lateral button featured on the haptic stylus will be used: during the initialisation phase in which the operator has to place the seeds, a continuous pressure on the button will mark the pointed area in the image. Once the button released, the stylus goes back to its passive role described in section 3.2.2. The marked areas should be graphically (and possibly haptically) distincted from the rest of the image. This distinction can be achieved by colouring the area and by applying a specific uniform haptic force in it.

3.2.4 Growing the regions

This section deals with what could be called the real interactive part of the algorithm. The aim is to give the operator a decision role *during* the segmentation process. While the original seeded region growing algorithm can run all by itself after the seeds have been placed, and despite the improvements described by Mehnert and Jackway (1997) that remove the order dependencies from the algorithm, the idea is to introduce a user interaction, to allow this user to control the flow of the algorithm.

At first this idea can seem unjustified: indeed, it slows the process down. But this drawback is largely compensated by the benefit gained from the operator's high level of reasonning and understanding of the image: if this artificially introduced interaction can help us take advantage of the *a priori* knowledge of the operator on the image, the result can be greatly enhanced. Indeed, in the original algorithm, the only human knowledge that is injected into the process is the position of the seeds, and even today's state-of-the-art techniques cannot replace a good human experience in the process of image segmentation.

The implementation of this interaction involves the introduction in the process of some kind of decisional pattern governed by the operator. The idea is to let the user control the growth of the regions. At a given point of the process, the selection of the next region to grow, instead of being left to the algorithm, should be given to the user. For the selected region, the operator should also be able to select the area of growth, by pointing out a specific boundary of the region and giving a direction.

This integration has to be done in a continuous fashion, otherwise the risk of ending up with a counter-intuitive and unusable environment is high. When no input is given by the user, the algorithm will carry on its process by itself: it is meant to be a *semi-automatic* algorithm.

3.2.5 Browsing through the result

This phase of the segmentation process is meant to allow the operator to visualise the resulting computed regions. *Visualise* implies graphic and haptic visualisation, as previously described in section 3.2.2.

The graphic rendering for the visualisation of the segmentation results has to display two layers:

- The first layer displays the original grey levels image;
- The second layer, on top of the first one, has to be semi-transparent in order to allow the visualisation of the original image data. It displays the resulting regions (and possibly the boundary pixels) by assigning each region a unique colour.

The haptic rendering should also highlight the partition of the image into regions. Inside a given region, the haptic force applied has to be uniform, and the boundaries between two regions should be tough to cross with the stylus, this toughness obtained by applying appropriate tangential forces opposite to the direction of the stylus.

3.2.6 Editing the regions

After the normal process of the algorithm has terminated correctly, the result is displayed in a convenient way for the operator. A possibility is to allow the operator to manually edit the result in order to correct the contingent errors generated by the noise in the image or the bad quality of the data. This possibility is another step towards a better integration of the human operator's knowledge: the expert eye of the operator will be able to detect the segmentation aberrations and correct them if the appropriate tools are provided.

This type of manual editing is studied by Vidholm and Agmund (2004) who propose a 3D segmentation environment as well as a set of tools to manually edit the resulting segmented objects. Although a 2D segmentation environment is studied here (and therefore a 2D editing tool), some similarities in the conception can be found.

Assuming that the number of generated regions is correct (and this assumption should always be true since the operator himself defines the seed regions), the editing of the result is only a matter of deforming regions, making some of them expand and others retract. The dual of the problem of deforming regions can be naturally expressed as the displacement of the boundaries: the dual way of describing a partition in an image by its regions is the description by the boundaries. Therefore, two possible editing tools can be imagined:

- A tool based on the regions: the operator can deform the regions by pushing the boundaries from inside a region to expand it;
- The dual tool based on boundaries: the operator can pick an edge and move it to locally adjust the form of the two involved regions.

In both cases, the edition is done at an interactive rate, so that the operator can immediatly see the result of his modifications, and the editing is of course haptically enabled, meaning that damper forces are applied to make the editing process as accurate as possible. Similarly to what is explained in section 3.2.5, a boundary should not be too easy to deform. This effect can be obtained by applying appropriate tangential forces opposite to the direction of the deformation. The values of the applied forces can depend on several parameters and will have to undergo some fine tuning in order to find their optimal values.

3.3 Summary

A successful integration of haptics into the image segmentation process raises a lot of issues, most of which have been covered in this chapter.

These issues and the proposed solutions are specific to a seeded region growing algorithm. This algorithm was selected for this project for its simplicity, robustness, efficiency and the multiple interesting ways in which the haptic interaction can be integrated into it.

The next chapter deals with the design considerations related to this integration.

Chapter 4

Design considerations

This chapter discusses some design considerations for the development of an effective haptically enabled image segmentation environment. The segmentation environment to be developed will make use of different techniques described in section 2.3. Section 4.1 discusses the interface between the human operator and the system from a theoretical point of view. Section 4.2 describes in a detailled way how the haptic forces are computed and mapped with the data to give the operator a realistic immersing experience. Section 4.3 deals with some graphic and haptic issues to determine the most effective layout of the different entities and tools in the environment.

4.1 The human/system interface

From a physical point of view, the interface between the human operator and the system is the haptic device and its associated graphic system (namely the *Reachin Device*, described in section 5.1).

From a more theoretical point of view, the interface is a set of predefined possible interactions between the user and the system; each of these interactions triggers one or more actions (and possibly other interactions). For example, when browsing through the input data (see section 3.2.2), the displacement of the pointer on the surface displaying the data is an *input*. This input updates the position and computes the new appropriate force to apply to the stylus. This force is sent to the haptic device and the user receives it as an *output*. The interface can therefore be seen as a continuous flow of inputs and outputs, the inputs using mainly a single medium (the haptic stylus), and possibly a keyboard as a supplement, and the outputs being of two different natures: haptic and graphic. The next section details the haptic outputs.

4.2 Mapping the data with forces

The computation of the forces applied to the haptic device as an output feedback for the operator is crucial: they condition the realism of the sensory experience and thus the accuracy of the performed operations. The general method for computing and applying a force is first explained, and then the different forces identified in section 3.2 are studied.

4.2.1 Computation of a force

The whole context of this project deals with the interaction of a stylus pointing objects and data on a 2D plane surface (this surface displaying one slice of the image data).

The first force to take into account is the resistance of the plane, supposed to be solid and thus impossible to cross. This force, which is called $\overrightarrow{R}$, is normal to the plane and proportional to the force applied by the operator. The stylus theoretically cannot go through the plane. However, due to mechanical limitations of the haptic device, if the force applied by the operator is too strong, the stylus will cross the plane because the device cannot oppose a force strong enough to counter the operator. In normal conditions of use, this *must* be avoided.

The other forces applied can be disjointed into two components. Let $\overrightarrow{F_i}$ be one of these forces. It can be written as $\overrightarrow{F_i} = \overrightarrow{N_i} + \overrightarrow{T_i}$ where $\overrightarrow{N_i}$ is the normal component and $\overrightarrow{T_i}$ is the tangential component of $\overrightarrow{F_i}$. Several forces resulting from different causes can be applied at the same time. These forces will be called $\overrightarrow{F_i}$, $i \in [1; n]$ where $n \geq 1$ is an integer. The resulting total force can be expressed as:

$$\overrightarrow{F} = \sum_{i=1}^{n} \overrightarrow{F_i} = \sum_{i=1}^{n} (\overrightarrow{N_i} + \overrightarrow{T_i}) \tag{4.1}$$

As $\overrightarrow{R}$ is always applied, the resulting total force sent to the haptic device as an output for the operator is expressed as:

$$\overrightarrow{H} = \underbrace{\overrightarrow{R} + \sum_{i=1}^{n} \overrightarrow{N_i}}_{normal} + \underbrace{\sum_{i=1}^{n} \overrightarrow{T_i}}_{tangential} \tag{4.2}$$

4.2.2 Rendering the data: haptic texture

The haptic rendering of the input data aims at providing the user a realistic and meaningful feedback on the data. In order to achieve this, forces based on the image data have to be applied to the haptic device depending on the position of the pointer

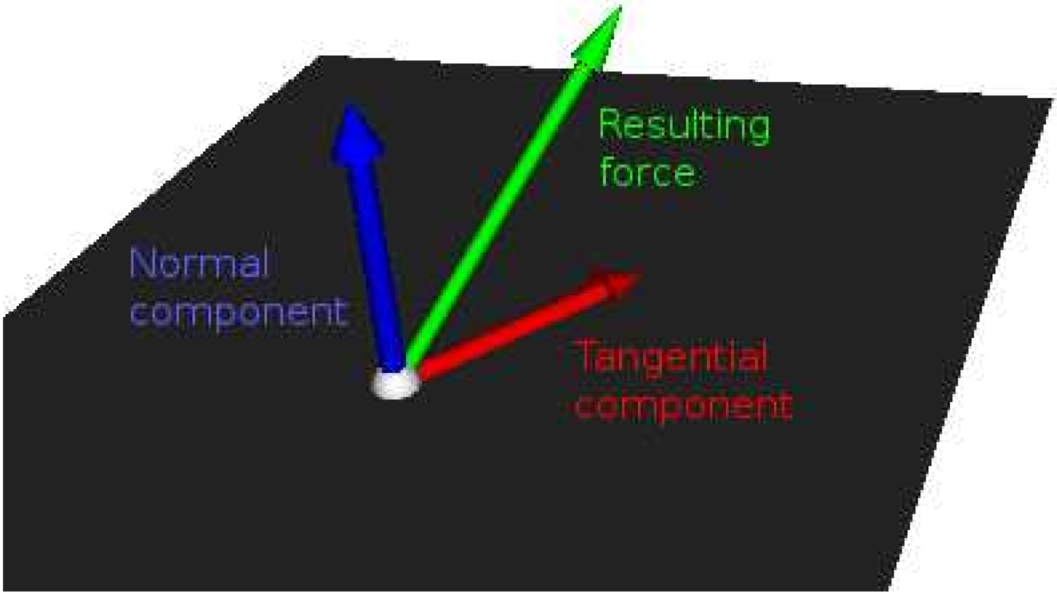

Figure 4.1: The different components of the resulting force

in the image. Several solutions have been proposed over the past few years by very productive research teams with the rapid development of haptic simulators in order to solve the non trivial problem of rendering an image with a haptic device.

Shi and Pai (1997) present a solution based on volume rendering: they use stereo reconstruction to build volume out of stereo images pairs. This volume is then haptically rendered using classical techniques in which the forces are computed with the surfaces normals. This solution is unusable in the case of this research since a plane surface is effectively needed, not a volume.

The other solution many other research works have come up with introduces the concept of *haptic texture*. In (Wikipedia: the free encyclopedia 2006*d*), a texture is defined as “*the properties held and sensations caused by the external surface of objects received through the sense of touch*”. A texture is in other fields often defined as a set of properties that stimulate one of our senses. In computer science, a texture often refers to a graphic texture. In a strict way, the expression *haptic texture* is circumlocutory since haptics involves the sense of touch, but it will be used to avoid any confusion with graphic textures. Moreover in this work graphic textures are also used: each slice of the input data set is an image texture. The aim of this section is to describe how this image texture can be mapped with a haptic texture.

Pai and Siira (1996) state that "*the haptic texture is defined to be all the effects which are not explicitely accounted for by traditional rigid body contact normal (constraint) and lateral (friction) forces*".

A method that seems perfectly adapted to the needs of this research project is reviewed by Theoktisto et al. (2005): it renders the haptic force using a *bump-mapping* algorithm (also called *force mapping*). The algorithm calculates the force to apply to the haptic device when a collision between the pointer and the surface is detected: the direction and the magnitude of the force are determined by a vector which components are the Red, Green and Blue values of the pointed pixel's intensity. This haptic rendering technique is often used in concert with a bump-mapping graphic rendering that visually simulates the volume, and it is stated by Theoktisto et al. (2005) to achieve a correct perception of surface roughness when the distortion is not very high. In this case however, there is no need to use such a technique for the graphic rendering because, again, *flat* slices extracted from a 3D volume are used.

As a conclusion about haptic texture rendering, Choi and Tan (2004) assert that the development of a realistic haptic texture-rendering system requires the appropriate design of a texture renderer, the stable control of a haptic interface and a better understanding of the somatosensory system. All these requirements make this development a highly technical problem in itself, and in this case supplemental haptic feedback will be considered as sufficient: it does not necessarily have to be highly realistic, it just needs to provide a haptic experience that enhances the overall perception of the data.

4.2.3 The growing process

The haptic interface will also be integrated into the segmentation process in itself as described in section 3.2.4. The idea is to guide the hand of the human operator by applying an appropriate force to the haptic device in the direction the algorithm decides is best. Of course the operator keeps full control over the growing process: the applied forces only have a guidance role. By moving the haptic device in another direction, the operator will modify the local decision pattern of the algorithm at a given step of its execution, interfering on the final result.

The suggested growing direction can be easily modelled as a vector tangent to the surface: the forces involved here have a nil normal component. The hint force is taken into account if and only if the haptic cursor is in contact with the surface. That way, the operator is given a useful means to let the algorithm know he does not want to intervene in the decision process.

4.2.4 Rendering the result

The rendering of the result, again, is obtained in two supplementary ways: graphic and haptic.

The graphic rendering does not really innovate in comparison with the existing segmentation environments: the resulting regions are coloured in different colours to allow a rapid and easy distinction. The boundaries of each region should be clearly visible (by using a darker variation of the region's colour for example). An important issue is to give the operator a double-depth visibility: he needs to be able to see the regions as well as the original image in the same rendering context. To achieve that, the coloured regions have to be semi-transparent.

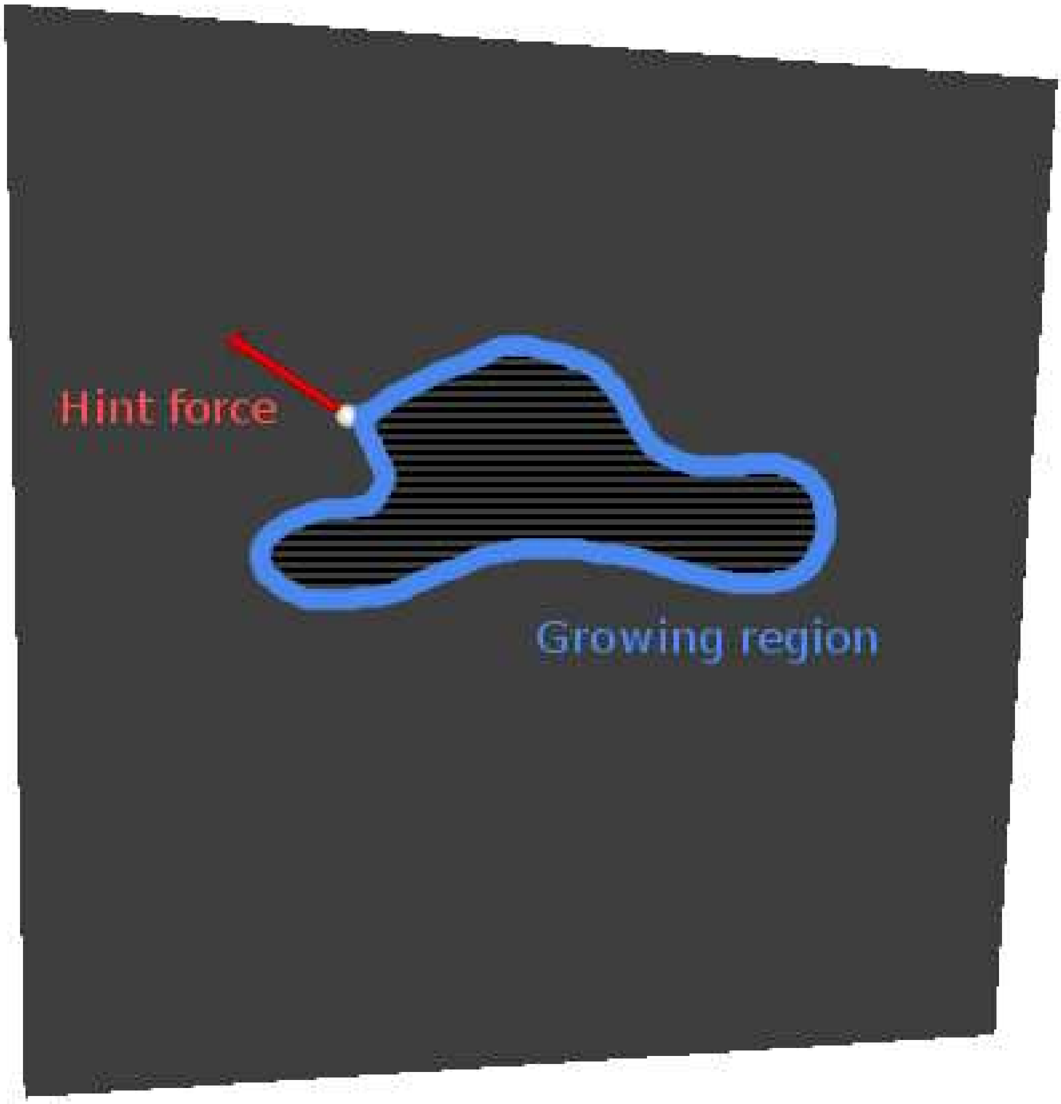

Figure 4.2: The hint force applied to suggest the growing direction

The haptic rendering presented here is based on the boundaries of the resulting regions. The idea is that feeling a region can be faithfully achieved by simply feeling its borders; more generally at a topological theoretical level, a closed region can be defined by its sole boundaries. Here again as in the previous section, the forces to apply are only tangential. If the pointer is in contact with the surface and if it tries to cross a region boundary, an opposing force is applied. The force direction is computed with the vector normal to the curve formed by the boundary, and its magnitude is proportional to the strength applied by the operator to the stylus. Of course if no precautions are taken this could lead to blocking situations, possibly damaging the fragile mechanics of the haptic device. This is why a limit magnitude should be defined where the pointer finally crosses the boundary to avoid any material problem.

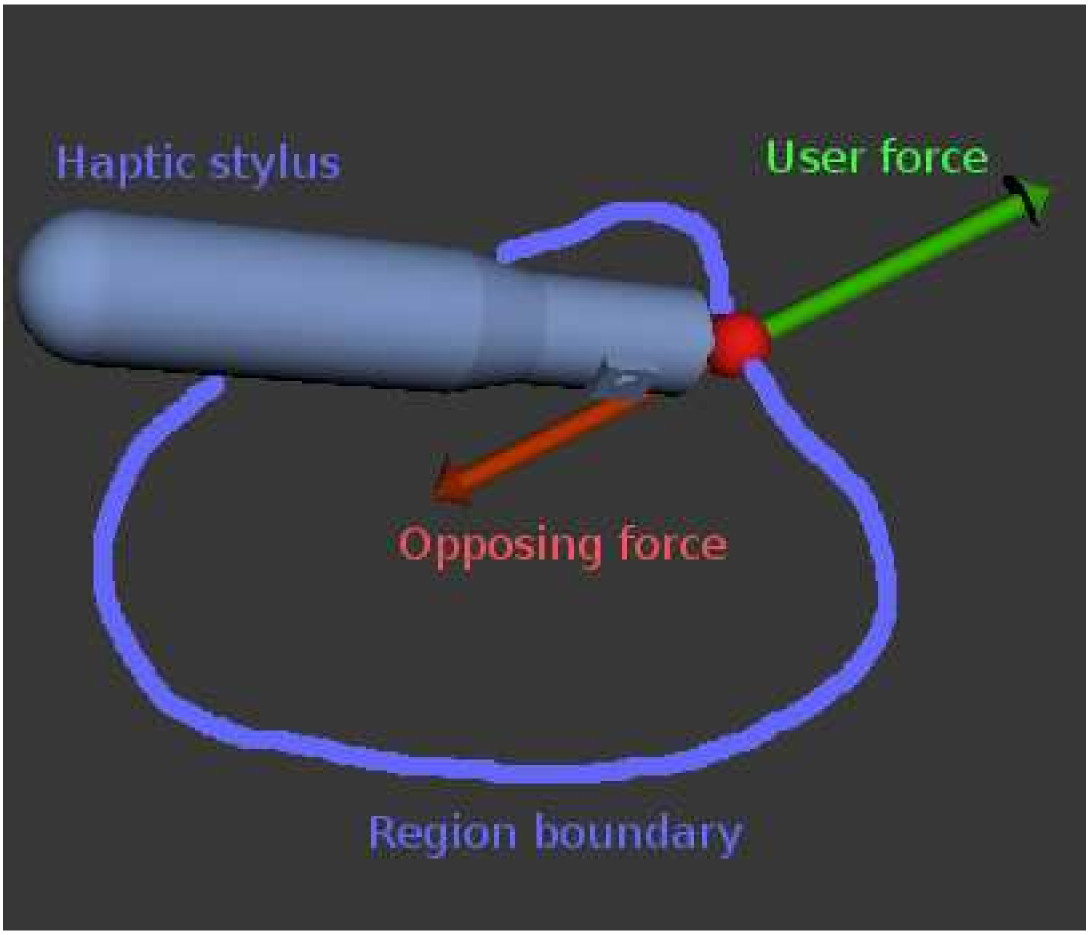

Figure 4.3: The force applied to haptically delineate the boundary of a region

4.2.5 The edition process

The last identified useful integration of haptics is in the edition process, when the user freely brings modifications to the regions. This phase is traditionally tricky when performed with only a visual feedback: an untrained hand can easily slip and make mistakes due to the imprecisions in the pointing device (a meaningful example of this is the task of drawing with a traditional mouse: it requires a lot of training before giving acceptable results).

It is therefore postulated that a haptic feedback combined with the traditional visual one would greatly enhance the accuracy of the manipulations. The edition process consists mainly, as stated in section 3.2.6, in deforming the regions' common boundaries. Therefore, the notion of opposing force presented in the previous section can be extended to complete the goal of this research project: the deformation of a boundary is subject to an opposing force that emulates a viscosity effect. By resisting enough (but not too much) to the operator's pressions, it is hoped to give him a better control over the adequation of the deformations to his expectations.

In this case the applied force is a non-blocking one: the operator *can* expand a region as much as he thinks is necessary. Therefore the issue of crossing a border does not exist any more.

4.3 User Interface considerations

Two types of user interfaces among those listed in (Wikipedia: the free encyclopedia 2006*e*) are used in this project:

- A Graphical User Interface (GUI) that can take input from a mouse and/or a keyboard and provides a graphical output on a screen. In this project however, the graphical user interface will be mainly used for its output facilities, the input being managed by the second type of interface;
- A Tactile Interface (sometimes also referred to as Haptic User Interface, HUI) that both takes inputs and provides outputs.

This section, to be completed, will deal with practical considerations to determine the most effective layout of the different entities and tools in the environment.

Chapter 5

Hardware and software bases

The development of an effective segmentation environment integrating haptics requires a hardware device for the graphical and haptic interaction, and APIs and toolkits to respectively deal with the haptic capabilities and the image segmentation in itself.

5.1 The Reachin device

This section will present the Reachin Display device (Reachin Technologies 2005) composed of a Phantom Premium 1.5 force feedback device, a stereo monitor and a mirror to co-locate haptics and graphics.

5.2 Haptics with H3D API

This section will present the H3D Application Programming Interface that combines high-level haptic and graphic rendering of a 3D scene-graph (SenseGraphics AB 2004*b*, SenseGraphics AB 2004*a*).

5.3 Segmentation with ITK

This section will present the Insight ToolKit (ITK) that provides a high-level and abstracted framework for image manipulation and segmentation (Ibáñez et al. 2005).

Chapter 6

Software architecture

This chapter will detail the architecture of the developed segmentation environment, the technical choices and the algorithms used.

6.1 Preliminary: a calibration tool

This section will present the calibration tool developed for the H3D API, the reasons why such a tool is needed, the theory behind its conception and the results produced.

Chapter 7

Assessing the benefits of the tool

This chapter will present a framework designed to assess the benefits of the developed tool by comparing it to classic 2D segmentation environments. This framework will include a certain number of identified key-points, each of which will help quantify the gain obtained with the new tool. User tests will be performed with a sample of testers from a wide range of professional backgrounds.

Chapter 8

Conclusions and future work

As a conclusion regarding the current research on this project, all the key objectives have been clearly identified together with the technical background and techniques needed to achieve these objectives.

The development of the application has started and will now occupy most of the research planning. The biggest challenge so far has been the clean integration of H3D and ITK due to the complexity of the two APIs, though promising results were eventually obtained.

Appendix A gives a time plan for the remaining time allocated to the project.

References

Adams, R. and Bischof, L. (1994), ‘Seeded Region Growing’, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(6), 641–647.

amira 3.1 User's Guide and Reference Manual (n.d.).

Bhanu, B. and Fonder, S. (2000), Learning Based Interactive Image Segmentation, *in* ‘ICPR’, Vol. 1, IEEE Computer Society, Washington, DC, USA, pp. 1299–1302.

Blake, A., Rother, C., Brown, M., Pérez, P. and Torr, P. H. S. (2004), Interactive Image Segmentation Using an Adaptive GMMRF Model, *in* ‘ECCV’, pp. 428–441.

Choi, S. and Tan, H. Z. (2004), ‘Toward Realistic Haptic Rendering of Surface Textures’, *IEEE Computer Graphics and Applications* **24**(2), 40–47.

Dr Singh, M. (2006), ‘Image segmentation studies’, `http://www-staff.lboro.ac.uk/~coms/Segmentation-review.htm`. Accessed 1 March 2006.

Elliott, P., Knapman, J. and Schlegel, W. (1992), ‘Interactive image segmentation for radiation treatment planning’, *IBM Systems Journal* **31**(4), 620–634.

Haralick, R. and Shapiro, L. (1985), ‘Image Segmentation Techniques’, *CVGIP* **29**(1), 100–132.

Harders, M. and Székely, G. (2002), Improving Medical Segmentation with Haptic Interaction, *in* ‘VR’, pp. 243–250.

Hinshaw, K. P. and Brinkley, J. F. (1997), Shape-based interactive three-dimensional medical image segmentation, *in* K. M. Hanson, ed., 'SPIE', Vol. 3034, pp. 236–242.

Horowitz, S. and Pavlidis, T. (1976), 'Picture Segmentation by a Tree Traversal Algorithm', *JACM* **23**(2), 368–388.

Ibáñez, L., Schroeder, W., Ng, L. and Cates, J. (2005), *The ITK Software Guide*, second edn, Kitware, Inc.

Jain, A. and M-P., D. (1992), 'Segmentation of X-Ray and C-Scan Images of Fiber Reinforced Composite Materials', *Pattern Recognition* **25**(3), 257–270.

Lucchese, L. and Mitra, S. (2001), 'Color Image Segmentation: A State-of-the-Art Survey', *Indian National Science Academy (INSA-A): Image Processing, Vision and Pattern Recognition* **67**(2), 207–221.

Mehnert, A. and Jackway, P. (1997), 'An improved seeded region growing algorithm', *Pattern Recogn. Lett.* **18**(10), 1065–1071.

Olabarriaga, S. D. (1999), Human-Computer Interaction for the Segmentation of Medical Images, PhD thesis, Advanced School for Computing and Imaging.

Otsu, N. (1979), 'A Threshold Selection Method from Gray-Level Histograms', *IEEE Trans. on Systems, Man and Cybernetics SMC* **9**(1), 62–66.

Pai, D. K. and Siira, J. (1996), Haptic Texturing - A Stochastic Approach, *in* 'ICRA', Vol. 1, pp. 557–562.

Parker, J. (1991), 'Gray Level Thresholding in Badly Illuminated Images', *PAMI* **13**(8), 813–819.

Reachin Technologies (2005), 'Reachin Products: the Reachin Display', `http://www.reachin.se/products/Reachindisplay/`. Accessed 9 March 2006.

Senger, S. (2005), Integrating Haptics into an Immersive Environment for the Segmentation and Visualization of Volumetric Data, *in* 'WHC', pp. 487–490.

SenseGraphics AB (2004*a*), 'H3D.org - Open Source Haptics', `http://www.h3d.org/`. Accessed 9 March 2006.

SenseGraphics AB (2004*b*), ‘Open Source Haptics: Products and services’, `http://www.sensegraphics.se/products.html`. Accessed 9 March 2006.

Shi, Y. and Pai, D. K. (1997), Haptic Display of Visual Images, *in* ‘VRAIS ’97: Proceedings of the 1997 Virtual Reality Annual International Symposium’, IEEE Computer Society, Washington, DC, USA, p. 188.

Spirkovska, L. (1993), A Summary of Image Segmentation Techniques, Technical report, NASA, Ames Research Center.

Theoktisto, V., Fairén, M., Navazo, I. and Monclús, E. (2005), ‘Rendering detailed haptic textures’, *Workshop On Virtual Reality Interaction and Physical Simulation* .

University of Hull’s Computer Science Department (2005), ‘Simulation and Visualization Research Group’, `http://www.dcs.hull.ac.uk/simvis`. Accessed 11 March 2006.

Vidal, F., Bello, F., Brodlie, K., John, N., Gould, D., Phillips, R. and Avis, N. (2004), Principles and Applications of Medical Virtual Environments, *in* C. Schlick and W. Purgathofer, eds, ‘STAR âĂŞ State of The Art Report’, pp. 1–35.

Vidholm, E. and Agmund, J. (2004), Fast surface rendering for interactive medical image segmentation with haptic feedback, *in* S. Seipel, ed., ‘SIGRAD’, Vol. 13, Linköping University Electronic Press, pp. 33–39.

Vidholm, E., Tizon, X., Nyström, I. and Bengtsson, E. (2004), Haptic guided seeding of MRA images for semi-automatic segmentation, *in* ‘ISBI’, Mira digital Publishing, pp. 288–291.

Warfield, S. K., Jolesz, F. A. and Kikinis, R. (1998), Real-time image segmentation for image-guided surgery, *in* ‘Supercomputing ’98’, IEEE Computer Society, Washington, DC, USA, pp. 1–7.

Wikipedia: the free encyclopedia (2006*a*), ‘Haptic’, `http://en.wikipedia.org/wiki/Haptics`. Accessed 28 February 2006.

Wikipedia: the free encyclopedia (2006*b*), ‘Magnetic resonance imaging’, `http://en.wikipedia.org/wiki/Magnetic_Resonance_Imaging`. Accessed 28 February 2006.

Wikipedia: the free encyclopedia (2006*c*), 'Segmentation (image processing)', `http://en.wikipedia.org/wiki/Segmentation_%28image_processing%29`. Accessed 9 December 2005.

Wikipedia: the free encyclopedia (2006*d*), 'Texture', `http://en.wikipedia.org/wiki/Texture`. Accessed 8 March 2006.

Wikipedia: the free encyclopedia (2006*e*), 'User interface', `http://en.wikipedia.org/wiki/User_interface`. Accessed 9 March 2006.

Yanowitz, S. and Bruckstein, A. (1989), 'A New Method for Image Segmentation', *CVGIP* **46**(1), 82–95.

Zouagui, T., Benoit-Cattin, H. and Odet, C. (2004), 'Image segmentation functional model', *Pattern Recognition* **37**(9), 1785–1795.

Appendix A

Future plans

Here is a list of the remaining tasks and a Gantt chart showing the progress of the work through the course of the rest of the year:

1. First version of the environment: data browser
2. Second version: region growing algorithm
3. Third version: result browser
4. Fourth version: editing tool
5. Validation framework
6. Poster writing
7. Poster presentation
8. Draft thesis writing
9. Rehearsal of presentation
10. Presentation and demonstration
11. Thesis writing

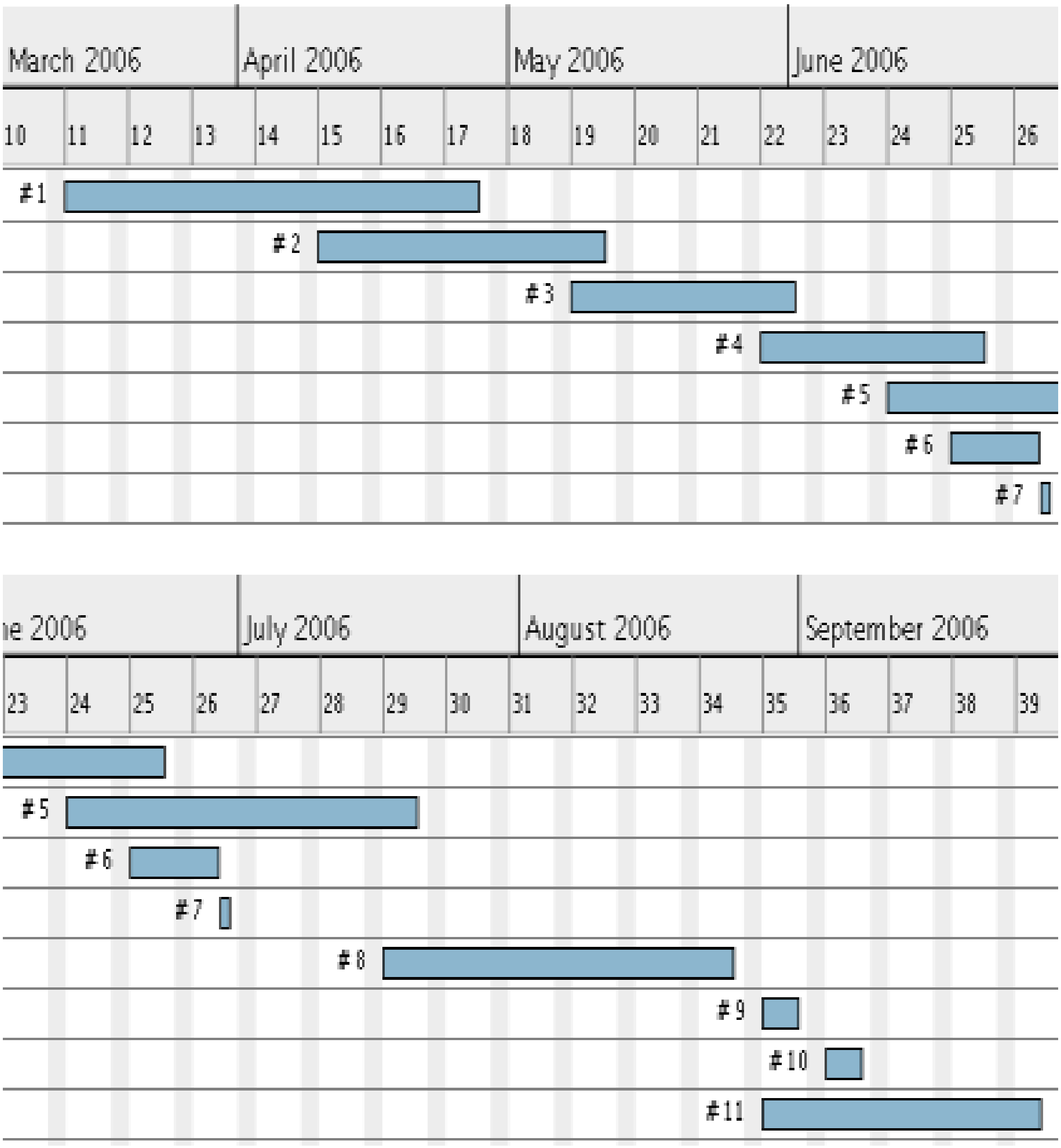

Figure A.1: Gantt chart of the time plan