The University of Hull
Department of Computer Science

A Multilayered Agent Society for Flexible Image Processing

being a Thesis submited for the Degree of Doctor of Philosophy

in the University of Hull

by

Qais Mahmoud Hassan

September 2008

"The only way of discovering the limits of the possible is to venture a little way past them into the impossible"

Sir Arthur Clarke

Acknowledgements

During my PhD research I received much needed technical help and emotional support from a number of exceptional people, without them this thesis would not have been possible. Firstly and foremost, I would like to thank my dear father Mahmoud, mother Basima and siblings, for their continued love, understanding and support and for encouraging me during my studies at University of Hull.

I am grateful to my supervisor Prof. Roger Phillips for accepting me in his research group during the difficult time of changing supervisors and for giving me the freedom to explore my own research ideas. During the various stages of the research, Roger provided constant support and guidance. I want to also thank him for the important advice and comments during our meetings which have enriched both this research and my personal experience.

I would like to thank the Department of Computer Science for providing some financial support. I am grateful for Castle Hill Hospital for providing the cardiac datasets used in the medical case study. I also want to thank Dr. Qingde Li and Dr. Len Bottaci for providing important feedback during my Supervisory Panel meetings.

Finally, I want to thank all the good friends I have made at the University whom provided me with endless support and made this journey more enjoyable.

Abstract

Medical imaging is revolutionising the practise of medicine, and it is becoming an indispensable tool for several important tasks, such as, the inspection of internal structures, radiotherapy planning and surgical simulation. However, accurate and efficient segmentation and labelling of anatomical structures is still a major obstacle to computerised medical image analysis. Hundreds of image segmentation algorithms have been proposed in the literature, yet most of these algorithms are either derivatives of low-level algorithms or created in an ad-hoc manner in order to solve a particular segmentation problem.

This research proposes the Agent Society for Image Processing (ASIP), which is an intelligent customisable framework for image segmentation motivated by active contours and MultiAgent systems. ASIP is presented in a hierarchical manner as a multilayer system consisting of several high-level agents (layers). The bottom layers contain a society of rational reactive MicroAgents that adapt their behaviour according to changes in the world combined with their knowledge about the environment. On top of these layers are the knowledge and shape agents responsible for creating the artificial environment and setting up the logical rules and restrictions for the MicroAgents. At the top layer is the cognitive agent, in charge of plan handling and user interaction. The framework as a whole is comparable to an enhanced active contour model (body) with a higher intelligent force (mind) initialising and controlling the active contour.

The ASIP framework was customised for the automatic segmentation of the Left Ventricle (LV) from a 4D MRI dataset. Although no pre-computed knowledge were utilised in the LV segmentation, good results were obtained from segmenting several patients' datasets. The output of the segmentation were compared with several snake based algorithms and evaluated against manually segmented "reference images" using various empirical discrepancy measurements.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Image Segmentation for Medical Structures

Recent advancements in image and video capture devices have provided an abundance of high quality digital images. These images range from internal organs scans obtained by MRI devices to images of civil structures acquired via remote sensing techniques. The human visual system is an excellent image analysis device, especially for extracting high-level information and understanding the content of digital images. However, most imaging applications cannot entirely depend on human observers for the analysis, in particular when processing large amounts of data. Since, it will be in general time consuming, tedious and prone to user error. Therefore, the true value of many digital images can not be fully realised unless automatically analysed by computers.

Digital image analysis, is the process of finding objects present in images [94]. One of the important steps in most image analysis systems is the assignment of a group of pixels in the image to distinct objects or the background [66], i.e. image segmentation. Many digital images contain irrelevant data to a particular image analysis application. For example, consider a face detection system, before attempting to recognise the face, the image should be segmented to several regions such as the background, face, eyes, etc. After the unrelated objects are discarded, the appropriate objects are then matched to those in a database. There is a wide range of image segmentation applications, such as the detection of cancerous tumours from MRI scans [73], locating airports from remote sensing data [1], optical character recognition [129] and surveillance systems [63].

In medicine, image analysis allows physicians to detect potentially life-saving information without the need of invasive surgery. Furthermore, medical imaging is expanding beyond its traditional role as an instrument for basic visualisation and inspection of anatomic structures. It is becoming an indispensable tool for several important tasks in medicine, such as, radiotherapy planning and treatment [93], surgical planning and simulation [101] and monitoring treatment efficiency [116]. However, accurate and efficient extraction of internal structures from medical images is still limited, due to several difficulties associated with image segmentation. The following are some issues that make image segmentation for medical structures a challenging problem.

- The complexity and variability of anatomic shapes. Real life objects have complex shapes in general, with no fixed geometric shape. Also, anatomical structures shapes differ considerably from one patient to another and from a healthy tissue to an abnormal one. This results in, limiting the use of a priori shape knowledge and rigid shape constraints.

- Defects in the images, such as noise, spatial aliasing and sampling artifacts originating from either poor equipments, transmission or processing errors. These shortcomings could cause the boundaries of structures to be indistinct and disconnected. Also, some image modalities such as ultrasound machines produce low quality images, which make it difficult to differentiate a medical structure from its neighbours.

- The large size of the medical datasets. For example, an average patient cardiac MRI for a single heartbeat contains around 220 slices. The use of manual and most semi-automatic methods is not efficient when applied to large datasets. Therefore, the only convenient solution is the use of automatic segmentation. However, automatic segmentation algorithms present a more challenging problem when compared with semi-automatic methods.

- The existence of different image types from several sources. Medical images originate from several image modalities such as CT, MRI and ultrasound. Also, they have different image types such as binary, greyscale and colour.

Since the nature of information stored in each image type differs from the other, it is difficult to create a solution that works well for all modalities. For example, a segmentation solution tailored for greyscale image might need adjustments when applied to colour images

Some of these issues, such as low image quality and noise, will have less impact in the future as medical imaging hardware continues to evolve. Currently, image segmentation is an active key research area in medical imaging since accurate segmentation is essential for many applications, and it determines the quality of the final results of image analysis. Also, the number of realistic image segmentation applications is most likely to increase, since the quality of medical images is improving and image segmentation algorithms are becoming more capable.

1.2 Thesis Motivation, Aim and Objectives

Hundreds of image segmentation algorithms have been proposed in the literature. Zaidi [132] provides some interesting data about image segmentation algorithms in the medical domain. Observing the number of publications of medical image segmentation from 1993 to 2005, Zaidi determined that the number of segmentation algorithms almost doubles every two years. For example, the number of peer-reviewed publications of medical image segmentation in a PUBMED search for 1993 was around 50 and for 1995 was around 100, in 2003 it was around 250 publications and for 2005 it was around 460 publications. Zaidi indicates that there are as many segmentation algorithms as there are researchers in the field, perhaps more since many researchers have proposed multiple methods. Nonetheless, to the author's knowledge, no single algorithm can be considered suitable for all sorts of images and applications. Moreover, most of these algorithms are either derivatives of low-level algorithms or created as ad-hoc in order to solve a particular segmentation problem.

Traditional low-level algorithms, such as thresholding [108], edge detectors [80] and region growing [47], do not benefit from the available knowledge about structures and only considers local image information. This can result in making incorrect assumptions during the segmentation process and in generating wrong object boundaries. Also, these low-level techniques are not suitable for automatic segmentation as they usually require considerable amounts of human intervention. Several medical segmentation algorithms are ad-hoc in nature, that have been proposed to solve a particular application. Application specific algorithms usually perform well by taking into account prior knowledge. Nevertheless, it can be argued that these algorithms provide limited benefit to the general segmentation problem. Since, these algorithms are usually not customisable to other applications and in some cases are not applicable to different image types. Creating application specific algorithms then generalising them in order to solve several segmentation problems, might not be a suitable method to create customisable image segmentation frameworks. Customisability should not be regarded as an isolated feature that can be added at a later step, it should be an integral part and considered while designing the framework.

Active Contour Models (ACM) [68, 83], also referred to as snakes, provide a promising model-based approach to medical image segmentation. The recognized importance of ACM originate from their ability to segment anatomic structures by utilising bottom-up constraints derived from the image data (external forces) together with top-down a priori knowledge (internal forces) about the size, shape and location of these structures. However, as pointed out by Kass [68], "snakes do not try to solve the entire problem of finding salient image contours. They rely on other mechanisms to place them somewhere near the desired contour". Traditionally, the "other mechanisms" are the guidance of knowledgeable users, for example, in [44] the user draws an initial contour around the target medical structure and interacts with the ACM while it attaches itself to high gradient points. Several attempts were made to improve and automate the original snake algorithm and enhance its ability to locate noisy and missing boundaries, for example see [22, 23, 128]. Nevertheless, these attempts have not been successful in dealing with the vast variety of structures and image data. Since, most of these improvements are application specific or they introduced new external and internal forces that require fine-tuning by users for optimal performance.

The aim of this research is to create a customisable automatic image segmentation framework. The research proposes to achieve this aim through the integration of MultiAgent systems and active contours. MultiAgent systems, an extension of distributed artificial intelligence, aim to create intelligent systems by utilising several autonomous, collaborative and rational entities which share a common goal. A generic active contour will be implemented as a society of reactive MicroAgents inhabiting a virtual world. In addition, several high-level agents will act on behalf the “other mechanisms” or expert users by initialising and guiding the MicroAgent society through the segmentation process. The use of AI techniques in the segmentation framework allows the MicroAgents to adapt to different environments (medical applications) without the need for extensive modification from the user. Also, the MultiAgent design allows each agent to specialise in a specific aspect of the image segmentation problem such as planning, shape constraints or knowledge management. Therefore, each agent can be designed, implemented and maintained independently from the other agents in the system. Using AI techniques in image segmentation is not a novel approach, several AI methods, most notably genetic algorithms [105] and neural networks [33], have been previously proposed in the literature. However, the use of MutliAgent Systems to solve low-level machine vision problems, such as image segmentation, is still considerably new. Therefore, the research will also assess the usability of utilising rational agents in the creation of image segmentation frameworks.

1.3 Thesis Outline

The thesis contains seven chapters and two appendices. The layout of the thesis is the following.

The next two chapters are mostly introductory, providing a general background on the related domains. The literature review is not restricted to the introductory chapters as other chapters include more specific literature when appropriate. Chapter 2 reviews commonly used image segmentation algorithms with a focus on active contour models and related technologies. Since the image segmentation domain is both vast and diverse it is not possible to survey all existing segmentation algorithms in a single chapter. For this reason, the chapter focuses on the advantages active contours possess over traditional segmentation solutions. In addition, it will aim to rationalise on the benefits of utilising active contour concepts in the MultiAgent framework. Understanding the advantages and limitations of existing segmentation algorithms are essential for the creation of the generic segmentation framework as it inspires the main motivation for this research.

Chapter 3 provides a general overview on Artificial Intelligence (AI), emphasising on MultiAgent systems. AI is a large field containing several ”schools of thought” proposed to create artificial minds. Although, these schools all share the same goal of creating intelligent entities, they are fundamentally different, since each school has their unique perspective/definition of intelligence. Ranging from schools that believe that the road for machine intelligence is building systems that think like humans (cognitive science), act like humans (Turing test), think rationally (symbolic AI) to systems that act rationally (agent approach). Since the AI research is both big and disparate, the chapter cannot cover all the different AI approaches. Therefore, the chapter concentrates on the MultiAgent systems as a tool for creating intelligent entities. Also, it provides some of the perceived advantages of using AI techniques in Image segmentation.

Chapter 4 proposes a generic framework for image segmentation. The framework is not intended to provide a single solution to all image segmentation problems, such a goal is not achievable. Instead the framework aims to provide a generic shell that can be customised for many image segmentation problems. The generic framework contains several collaborative agents (layers), each agent specialising in a single area of the segmentation process. The design of the framework as a multilayered (MultiAgent) architecture is inspired by Brooks work on the subsumption architecture [8] [9] [10]. In addition to presenting the MultiAgent framework, the chapter addresses the desired requirements that should exist in a segmentation framework in order to be customisable.

Chapter 5 populates the proposed framework with a medical case study. Many newly proposed segmentation algorithms are tested using synthetic images. Synthetic images even when corrupted with artificial noise provide little challenge when compared to real life applications. Therefore the decision was taken to populate the framework with a challenging medical problem, which is the segmentation of the left ventricle from a 4D dataset. In essence, Chapter 5 demonstrates how the segmentation framework can be customised for medical problems whilst benefiting from the available local knowledge.

Chapter 6 evaluates the proposed framework using several empirical discrepancy methods. Several newly developed segmentation algorithms are often subjectively compared with some particular algorithms and a few hand-selected images [133]. However, the human evaluation of segmentation results is subjective, qualitative, time consuming and might lack validity. Therefore, there is need for objective machine based evaluation of the segmentation results.

Appendix A contains description about the developed software used to implement the framework and tackle the medical problem. The Agents in the framework are represented using objects and are implemented using a quasi-parallel approach. The software was programmed using the C# language and the .NET framework. Appendix B contains visual results from the segmentation of the left ventricle along with the reference image utilised in the evaluation of the framework. Finally, the future work and conclusion are presented in Chapter 7.

Chapter 2

From Pixels to Objects Boundaries

2.1 Introduction

Digital image segmentation is one of the primary steps in image analysis, since accurate segmentation is essential for many image analysis applications and highly influences the quality of the final output. Jones and Metaxas [66] define image segmentation as the process of assigning a group of pixels in an image to distinct objects or the background. As mentioned earlier in Chapter 1, there is a wide range of image segmentation applications, such as the detection of cancerous tumours from MRI scans [73], locating airports from remote sensing data [1], optical character recognition [129] and surveillance systems [63]. In the medical domain, image analysis allows physicians to detect potentially life-saving information without the need of invasive surgery. Furthermore, medical imaging is expanding beyond its traditional role as an instrument for basic visualisation and inspection of anatomical structures. It is becoming an indispensable tool for several important tasks in medicine, such as, radiotherapy planning and treatment [93], surgical planning and simulation [101] and monitoring treatment efficiency [116].

This chapter provides an overview of commonly used image segmentation algorithms with a focus on Active Contour Models (ACM) and related technologies. Since the image segmentation domain is both vast and diverse it is not possible to survey all existing segmentation algorithms in a single chapter. In addition, a complete description of the presented algorithms is beyond the scope of this chapter and the readers are referred to the references for additional details. This chapter contains two main parts, the first part aims to provide the reader with a general overview on image segmentation methods. While the second part focuses on ACM and the advantages they possess over traditional segmentation solutions.

2.2 Medical Image Segmentation Difficulties

As previously mentioned in Chapter 1, accurate and efficient extraction of internal structures from medical images is still limited, due to several difficulties associated with image segmentation. The following are some issues that make image segmentation for medical structures a challenging problem.

- The complexity and variability of anatomic shapes. Real life objects have complex shapes in general, with no fixed geometric shape. In addition, anatomical structures shapes differ considerably from one patient to another and from a healthy tissue to an abnormal one. This results in, limiting the use of a priori shape knowledge and rigid shape constraints.

- Defects in the images, such as noise, partial volume effect, spatial aliasing and sampling artifacts either originating from poor equipments, transmission or processing errors. These shortcomings could cause the boundaries of structures to be indistinct and disconnected. Figure 2.1 provides an example of sampling errors producing partial volume effects, where several tissues contribute to a single pixel resulting in blurring intensity levels across boundaries. In addition, some image modalities such as ultrasound machines produce low quality images, which make it difficult to differentiate a medical structure from neighbouring structures and surroundings.

- The large size of the medical datasets. With the increasing size and number of medical images, the use of automated methods to analysis them has become a necessity. For example, an average patient cardiac MRI for a single heartbeat contains around 220 slices. The use of manual and most

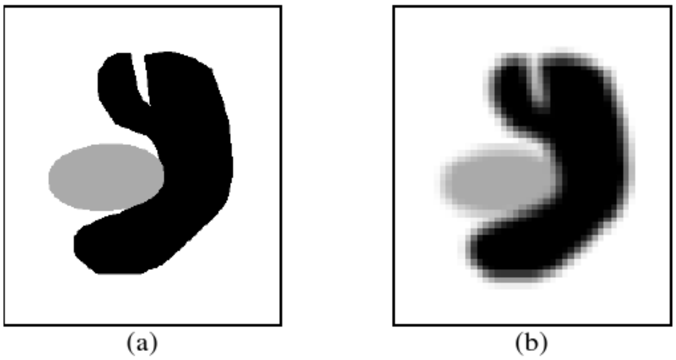


Figure 2.1: Illustration of partial volume effect: (a) Ideal image, (b) acquired image, adapted from [96].

semi-automatic methods is generally not efficient when applied to large datasets. Therefore, the only convenient solution is the use of automatic segmentation. However, automatic segmentation algorithms present a more challenging problem when compared with semi-automatic methods.

- The existence of different image types from several sources. Medical images originate from several image modalities such as CT, MRI and ultrasound. In addition, they have different image types such as binary, greyscale and colour. Since the nature of information stored in each image type differs from the other, it is difficult to create a solution that works well for all modalities. For example, a segmentation solution tailored for greyscale image might need adjustments when applied to colour images

2.3 Classification of Image Segmentation Techniques

Hundreds of image segmentation algorithms have been proposed in the literature, ranging from low-level algorithms such as region growing to more complex model based solutions such as deformable models. No single segmentation algorithm can be considered suitable for all sort of images and applications. In general, choosing the "best" segmentation algorithm to solve a particular problem depends heavily on the image analysis application. Usually, while choosing a suitable method for image segmentation there might be a trade-off between performance and interaction. Manual segmentation can improve the accuracy by incorporating the operator prior knowledge about the target object. However, for large datasets, this can be time consuming, tedious and prone to user error. On the other hand, semi-automatic segmentation algorithms, where the user initialises the algorithm, considerably reduce the time needed for segmentation. However, they are still time consuming when dealing with large studies and subject to inter and intra observer variability. Fully automated systems for image segmentation are the most desired method for segmentation as they are the fastest and require no (or minimal) user interaction. Nevertheless, automatic segmentation algorithms are more sensitive to noise and initial parameters which can significantly affect their performance.

Most segmentation algorithms can be classified into one of the following three groups.

- Region extraction techniques
- Boundary detection techniques
- Feature thresholding or clustering

Details on these techniques are provided further in the following sections.

2.3.1 Region Extraction Techniques

Region techniques are based on the assumption that there exists some similarity property (i.e. homogeneity criteria) between the pixels. For example, pixels sharing a homogeneity criterion such as intensity, texture or colour are considered part of the same region. There is two main approaches for region extraction techniques, “split and merge” [20] and “region growing” [136]. Although both approaches share the same concept of homogeneity, their approach for segmentation is different. A classical split and merge algorithm consists of two main steps. In the first step, the whole image is considered as one region. If this region does not satisfy a homogeneity criterion then the region is split into several subregions and each subregion is tested in the same way. The process is repeated until every subregion contains only homogeneous pixels. In the second step, all neighbouring regions with similar attributes are merged following other (or the same) homogeneity criteria. On the other hand, region growing algorithms start with one or more pixels, usually referred to as seeds. The local neighbouring pixels of the seeds are evaluated based on a homogeneity criterion. If they are similar to the seed, they are merged into the region. The process is repeated, with the new merged pixels behaving as seeds, until all the pixels in the image are assigned to one or more regions.

2.3.2 Boundary Detection Techniques

Boundary based methods are based on some discontinuity property of the pixels, such as intensity. The most common method of boundary detection is edge based techniques such as Canny [13] and Sobel [111]. Edges in digital images correspond to a noticeable change of intensity levels. The first derivative of the image function has a maximum at the position corresponding to the edge in the image, and the second derivative have a zero at that position. Generally speaking, active contours can be also considered as a boundary based technique. Since the aim of most active contours is to locate boundaries and the external energy function can depend on the image gradient. Active contours are discussed in depth in the following sections.

2.3.3 Feature Thresholding and other Techniques

Thresholding approaches [108] segment images by creating a binary partitioning of the image intensities or other quantifiable features. For example, a threshold value can be used to separates the desired classes in an image. The segmentation is then performed by grouping all pixels with intensity levels greater than the threshold value into one class, and all other pixels into another class. Thresholding is a simple concept, yet it can be an effective tool for segmenting an image where different objects have noticeable contrasting intensities.

Clustering algorithms [59] segment images by searching for distinct groups in the feature space. The algorithm expects that these groups belong to different structures that can be clearly differentiated. For example, the K-means clustering algorithm [24] groups data by iteratively computing a cluster centre for each class and segments the image by classifying each pixel to the cluster that minimizes the variance between the pixel and the cluster centre. Several challenges are associated with clustering algorithms, such as, choosing the best number of clusters and determining the validity of clusters.

Several other segmentation techniques also exist in the literature. Such as the use of Artificial Neural Networks (ANN) in clustering methods [100], or the integration of region and boundary based techniques [17, 43].

2.4 Active Contour Models

Active Contour Models (ACM), also referred to as snakes, deformable models or deformable contours, provide a model-based approach to medical image segmentation. ACM were first introduced by Kass [68] as an alternative to the classical view in machine vision research. The classical view suggests that low-level tasks such as segmentation are autonomous bottom-up processes, that should only make use of image data without relaying on any higher-level information [81]. This rigid view implies that any errors made by the low-level processes will be propagated without the opportunity for correction. The recognized importance of ACM originate from their ability to segment anatomic structures by utilising bottom-up constraints derived from the image data (external forces) together with top-down a priori knowledge (internal forces) about the size, shape and location of these structures.

ACM have several advantages over existing pure statistical and region based techniques such as thresholding and region growing. In addition to combining both knowledge and image data in a single model, they respond to applied simulated forces in the same way objects respond to forces in the physical world. Thus, making the manipulation of them with the forces natural and intuitive. In addition, ACM are fundamentally dynamic, therefore they unify the analysis of shape and motion. This makes them favourable to studying natural objects in motion, for example beating hearts, moving lips, arms movement, and so on.

2.4.1 Active Contour Definition

Active contours are physics-based curves or surfaces defined within the image domain, they deform (move) under the influence of internal and external forces (or energies). Internal forces are defined within the model itself and they restrict its behaviour, they are used to keep the model smooth during deformation. On the other hand, external forces are computed from the image data itself. Both internal and external forces are defined with the intention that the ACM will conform to an object boundary or other desired features within an image. The deformation is achieved by minimising an energy function designed so its local minimum is obtained at the desisted feature of the object. ACM are used in many applications, including image segmentation [5], shape modelling [46] and motion tracking [4]. Figure 2.2 demonstrate an example of using ACM in segmentation of medical images, in the figure the initial contour is plotted in grey, while the result is plotted in white.

There are two main types of ACMs, parametric active contours [68, 125, 127, 128] and geometric active contours [14, 79]. The first type, which is also the oldest approach, represents active contours in an explicit manner corresponding to the Lagrangian formulation. The second type represents active contours in an

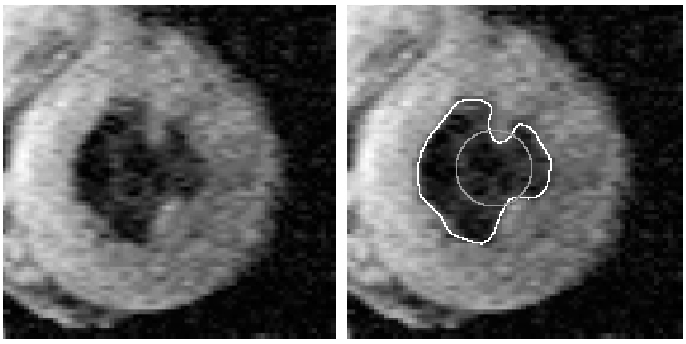


Figure 2.2: Example of using ACM in segmentation of medical images. Left: A 2D MRI image of the heart left ventricle. Right: Using ACM to segment the left ventricle, the initial contour is plotted in grey, while the final result is plotted in white, adapted from [127].

implicit manner corresponding to the Eulerian framework. Despite the fundamental difference between the two approaches, the underlying principles of both parametric and geometric ACM are similar. The main advantage of the geometric ACM is its ability to automatically handle topological changes, such as splitting or merging parts, during the deformation. However, they tend to be computationally more complex since they evolve a surface instead of a curve. Also, it is difficult to introduce shape prior-knowledge into them since the curve representation is implicit. Table 2.1 compares the general features of both parametric and geometric ACM.

In general, the choice between parametric or geometric active contours depends on the application. One of the most important factors to consider is the topological changes requirements. If the number of target objects is unknown and the contour needs to split or merge from several starting contours, then geometric

	Parametric Contours	Geometric Contours
Efficiency	***	*
Ease of Implementation	***	**
Topology Change	No	Yes
Open Contours	Yes	No
Interactivity	Good	Poor

Table 2.1: Parametric and geometric ACM comparison, adapted from [29].

ACM should be a better option. In this research, significant topological changes are neither required nor desired, since it presumed that the framework is aware about the segmentation process and target objects. Further comparison between parametric and geometric active contours is provided in [29]

2.4.2 Mathematical Model of ACMs

McInerney and Terzopoulos [83] describe the mathematical foundation of the ACM as one that represent the confluence of geometry, physics, and approximation theory. Geometry is utilised to represent the shape and it usually support broad shape coverage by employing geometric representations that involve several degrees of freedom, such as splines. Physics constrains the shape variability over space and time. Approximation theory supplies the underpinnings of mechanisms for fitting the ACM to the data.

A parametric snake is a curve $\mathbf{x}(s) = [x(s), y(s)], s \in [0, 1]$, that deforms over a series of iterations (time steps) through the spatial domain of an image to minimise the following energy functional [68].

$$E_{snake} = \int_0^1 E_{int}(\mathbf{x}(s)) + E_{ext}(\mathbf{x}(s)) + E_{con}(\mathbf{x}(s))ds \tag{2.1}$$

Where E_{int} represent the internal energy of the contour. This energy enforces smoothness on the contour, any change in the shape will increase its internal energy causing a restitution force that tries to restore it to its original shape. E_{ext} is the external energy also referred to as image forces, this energy attracts the contour towards salient features computed directly from the image data, such as lines and edges. E_{con} is the external constraint (interactive) forces, they originate from high-level sources such as human users and they are used to push the contour close to the desired solution or out of a local minimum. Interactive forces are useful in semi-automatic segmentation, where an operator needs to guide the model during deformations. Examples of these forces includes Kass [68] spring and volcano forces. Usually constraint forces are considered part of the snake external forces, and are omitted from the energy function.

The internal energy can be written as.

$$E_{int} = \frac{1}{2}[\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2] \tag{2.2}$$

Where α and β are weighting parameters that control the snake tension and rigidity. They determine the degree to which the contour is allowed to stretch or bend at a specific point. $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ denote the first and second derivatives of $\mathbf{x}$(s) with respect to s. The first term makes the snake act like a membrane while the second term makes it act like a thin plate. Setting α as 0 at a point will make a discontinuity occur at that point, while setting β to 0 allows a corner to develop.

The external (potential) energy function E_{ext} is computed from the image so that it has low values at the features of interest such as edges. Representing a grey-level image as I(x,y) as the intensity function of the image, a typical image energy function designed to move the snake towards step edges is [127].

$$E_{ext}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2 \tag{2.3}$$

Where $G_\sigma(x, y)$ is a 2D Gaussian function with standard deviation σ, ∇ is the gradient operator, and * is the 2D image convolution operator. Several edge detectors have been used to compute the potential energy field such as the Canny and Sobel edge detectors. The Canny detector is very good in eliminating noise but not good at identifying weak edges, on the other hand, Sobel filter is good at identifying weak edges but is bad in reducing noise. To improve the performance of the Sobel detector, a 2D Gaussian smoother can be used to remove the noise and small-scale texture from the image.

2.4.3 Numerical Implementation

Several numerical implementations of ACM have been proposed in the literature. The first ACM model, along with other snake models, utilise the variational calculus approach, in which the finite difference method [68] and the finite element method [23] are used to solve the energy minimising for curves and surfaces, respectively. The variational methods lack numerical stability and require estimates of higher order derivatives of the discrete data. Also, the image forces need to be differentiable in order to guarantee convergence, making it not possible to include hard constraints, such as the minimum distance between points introduced in [125].

To address these problems a dynamic programming method was introduced by Amini et al. [2], the method thoroughly searches all admissible solutions, and each iteration results in a locally optimal contour. However, the dynamic programming method is considered slow and costly to compute having a complexity of $O(nm)^3$, where n: number of points in the contour, m: size of neighbourhood in which a point can move. In order to enhance the performance of the dynamic programming approach, Willams and Shah [125] introduced the greedy snakes. The greedy snake is faster and simpler than the dynamic programming approach, it is referred to as "greedy" since the energy minimisation is performed at each step. During the greedy snake deformation, the algorithm searches for a lower energy location at each control point by comparing the control point energy and its eight local neighbours. If a lower energy location was found the control point is moved to the new local minimum energy location.

2.4.4 ACM Limitations

There is number of known problems associated with using ACM in image segmentation. The following is a list of the most important limitations that affect the classical snake and other parametric active contour models.

Initialisation: The final extracted contour largely depends on the shape and position of the initial contour due to the presence of many local minima in the energy functional. The initial contour must be placed near the desired object, otherwise the external energy will not be strong enough to pull the contour towards the target object and it will either collapse on itself or it will get stuck on a local minimum. This limitation was pointed out by Kass [68], "snakes do not try to solve the entire problem of finding salient image contours. They rely on other mechanisms to place them somewhere near the desired contour". Traditionally, the "other mechanisms" are the guidance of knowledgeable users. Some researchers tried solving this problem from within the ACM itself without relying on the "other mechanisms". For example, Xu et al [127, 128] introduced a new external force called the

Gradient Vector Flow (GVF). This new force is mainly intended to solve two problems found in the classical model, initialisation and poor convergence to boundary concavities. The GVF increases the attraction range by diffusing the gradient of an edge map in regions distant from the boundary. The amount of diffusion adapts according to the strength of edges to avoid distortions of the object boundaries. GVF is similar in theory to Cohen and Cohen [23] distance potential force, which is based on the concept that each model point should be attracted to the nearest edge point. The external energy function is defined using a distance map, the value of the distance map at each pixel is obtained by calculating the distance between the pixel and the closest boundary point. Although, both the distance potential force and GVF provide a long attraction force, they are more sensitive to noise than other traditional image forces.

Poor convergence to boundary concavities: The ACM internal forces tend to push the contour out of boundary concavities ('U' shaped objects), since the aim of internal forces is keeping the shape as smooth as possible. Cohen [22] introduced a new pressure force for ACM that can push contours into boundary concavities. This pressure force inflates/deflates the ACM (like a balloon) in order to make it pass over weak edges. The balloon model is efficient when the initial ACM lies inside the object and the object is smooth. However, it is not well suited for complex rough objects. The balloon force along with the image forces function as external forces affecting the contour and pushing it into strong edges, the modified external force equation is.

$$F = balloonforce - imageforce \tag{2.4}$$

The balloon force is expressed as.

$$balloonforce = k_1 \overrightarrow{n}(s) \tag{2.5}$$

Where $\overrightarrow{n}(s)$ is the normal unitary vector to the curve at point v(s) and k_1 is the amplitude of this force. Changing the sign will result in a deflating effect for the balloon instead of inflating.

Sensitivity to noise: ACMs are sensitive to noise, since noise can appear as gradient maxima and attract the ACMs towards it instead of the real edges. Noise reduction techniques such as Gaussian smoothing might weaken and blur the edges. In Cohen's balloon snake, the pressure force will allow the ACM to pass over isolated noise locations until it reaches strong edges and stops. However, the balloon ACM is hard to control, it should be instructed when to inflate or deflate, based on whether it is inside or outside the desired object. In addition, it is not suitable for structures with weak edges and might leak out of them.

Parameter fitting: Most active contours contain several parameters that require fine-tuning by the user to obtain good results. For example, in the ACM of Kass there is no clear guidelines for choosing the values of α and β coefficients in the internal energy equation. Several ACM algorithms use a constant value for them, these values are important and should be chosen carefully as they change the behaviour of the ACM.

2.5 Discussion

The image segmentation domain is an active research field with hundreds of proposed algorithms. Nonetheless, to the author's knowledge, no single algorithm can be considered suitable for all sorts of images and applications. Traditional low-level techniques, such as thresholding, edge detectors and region growing, do not benefit from the available knowledge about structures and only considers local image information. This can result in making incorrect assumptions during the segmentation process and in generating wrong object boundaries. In addition, these low-level techniques are not suitable for automatic segmentation, as they usually require considerable amounts of human intervention. Active contours provide a promising model-based approach to medical image segmentation through the integration of bottom-up constraints (external forces) along with top-down a priori knowledge (internal forces). However, active contours do not solve the entire problem of image segmentation as they rely on other mechanisms, such as the user, to place them near the desired object.

Several attempts have been made to improve and automate the original snake algorithm and enhance its ability to locate noisy and missing boundaries, for example see [5, 22, 23, 50, 62, 119, 128]. Nevertheless, these attempts have not been successful in dealing with the vast variety of structures and image data. Since, most of these improvements are either application specific or they introduced new restrictive external and internal forces that require fine-tuning by users for optimal performance. Although, some of the new external and internal energies are useful to some specific applications, they are not universal in the sense that they can be applied to many segmentation problems. The author believes, that the current research trend of creating more "enhanced" energy forces is not necessarily the best approach to improve active contours. Instead, active contours require better management and integration with higher intelligent systems that can control and guide them during the deformation process.

2.6 Summary

This chapter has provided an overview of image segmentation algorithms focusing on active contour models. Most segmentation algorithms can be classified into region based, boundary based or thresholding techniques. Low-level segmentation algorithms are simple to implement, still they do not benefit from the available knowledge about structures and only considers local image information. On other hand, active contours provide a model-based approach to image segmentation through the integration of image data along with a priori knowledge about target objects. There is a number of key challenges associated with using active contours in image segmentation, such as, initialisation, handling concave shapes and parameter fitting. Nonetheless, active contours are more suitable than most low-level algorithms for integration into a generic image segmentation framework.

A number of surveys on image segmentation are available including [43, 91, 96], also, review studies focusing on active contour models are provided in [58, 83, 84, 127].

Chapter 3

AI and MutliAgent Systems

3.1 Introduction

A MultiAgent System (MAS) is a society of autonomous and intelligent entities, collaborating in order to achieve the same goal and solve a common problem. MAS concepts evolved from Distributed Artificial Intelligence (DAI) systems, and are influenced by Brooks' work on the "subsumption architecture" [8, 9, 10]. Brooks opposes the traditional symbolic AI view, which states that symbolic representation is the key for building intelligent systems. Instead, Brooks proposes a system with incremental layers of intelligence, which operate in parallel. Each layer has its own purpose or goal, and combines both the peripheral systems and control systems, there is no central control for the system. In addition, each layer must decide when to act for themselves (autonomous), and not to be told how by other layers. The advantage of using the subsumption architecture is that it provides an incremental path from very simple systems to complex autonomous intelligent systems. At each stage, it is only essential to build one piece, and interface it to an existing and working intelligence.

The aim of this chapter is to provide a general overview on MultiAgent systems. Artificial Intelligence (AI) is a large and diverse domain containing several "schools of thought" proposed to create artificial minds. Although, these schools all share the same goal of understanding/creating intelligent entities, they are fundamentally different, as each school has their unique perspective/definition of intelligence. Since the AI research is both big and disparate, a single chapter cannot cover all the different AI approaches. Therefore, this chapter will concentrate on MultiAgent systems as a tool for creating intelligent entities.

3.2 Artificial Intelligence

The notion of Artificial Intelligence (AI) might seem controversial for some readers and lead to one of the following questions.

- Is AI possible? Could there possibly be artificial minds?
- Does a mind require a body? Do we need to build a robot in order to achieve machine intelligence?

- Can we create autonomous entities? Do humans even possess a free will?

Although, these philosophical debates are important to the study of AI, they are well outside the scope of this thesis. Therefore, it will be assumed that there are three levels of intelligence, human, animal and machine (AI). In addition, the level of intelligence possessed by animals and machines will not be directly compared to the human intelligence standards. Instead, it will be measured against an ideal concept of intelligence, which is rationality. A system is rational if it does the "right thing", given the available information and computational resources. Some behaviours demonstrated by machines and animals should be labelled as a form of intelligent behaviour. Observe for example, an animal obtaining food by a complex series of actions never preformed before, or a machine winning a chess match against a world champion [12]. These forms of behaviour can only be described as intelligent.

For more insights into the nature of intelligence and philosophical AI debates, the reader is refereed to Franklin's book "Artificial Minds" [42], which also serves as a good introduction to AI.

3.2.1 Roadmap for Intelligence

There are several proposed approaches for understanding and creating intelligent entities, these approaches are usually referred to in the literature as, "schools of thought" or "recipes for intelligence". These schools are fundamentally different, as each school has their unique perspective/definition of intelligence, which eventually resulted into several methods for building intelligent machines. Russell et al. [107] categorised these approaches, based on their definition of AI, into the following four groups.

1. Acting humanly: Alan Turing [115] proposed a test to provide an operational definition of intelligence. During the test, a human interrogator asks a system located in another room a set of questions. The machine is considered intelligent if the human observer could not distinguish it from another human. Nevertheless, it is debatable if a system that passes this test is really intelligent. Duplicating the human model requires a lot of work and might not result in intelligent systems. Russell et al. [107] provides a good argument against pursuing this approach, "The quest for artificial flight succeeded when the Wright brothers and others stopped imitating birds and learned about aerodynamics".

2. Thinking humanly: The cognitive modelling approach combines computer models from AI with experimental techniques from psychology in order to construct testable theories of the human mind. However, a system might follow the human reasoning and still perform poorly, or vice versa. In addition, cognitive science is based on experimental investigation of actual

humans or animals, which might have some ethical implications.

3. Thinking rationally: The "law of thought" or symbolic AI approach attempts to solve any solvable problem described in logical notations. There are two main obstacles to this approach. First, the transduction problem, that of transforming the real world information into a precise, sufficient symbolic description in time for that description to be useable. Second, the representation/reasoning problem, that of how to symbolically represent information about complex real world entities and processes, and how to reason with this information in time for the results to be useful.

4. Acting rationally: This approach attempts to produce intelligent entities by creating rational agents. A rational agent is one that acts in order to achieve the best outcome, given the available information and computational resources. There are several other attributes that differentiate agents from "standard" programs, such as, operating under autonomous control, perceiving the environment and adapting to change. The rational agent approach and MultiAgent systems are discussed in detail in the following sections.

3.3 Rational Agents and MultiAgent Systems

In the late 80's and early 90's, the field of Distributed Artificial Intelligence (DAI) has produced a new paradigm, rational agents and MultiAgent Systems (MAS). Software agents came into sight to meet the raising increase in applications complexity and the need for a higher level of abstraction for the analysis, design and implementation of intelligent software systems. MAS could be described as collaborative problem solvers, where each problem solver has the required experience and knowledge to solve a sub-problem within the system. In addition, these problem solvers are required to work together as a team in order to achieve the common goal of the system. There are several advantages for the agent-based approach over other AI approaches. For example, the approach is more flexible than the symbolic AI approach, since correct deduction is just one of the several methods for achieving rationality. In addition, it is more suitable to scientific development than approaches based on either human behaviour or human thought.

3.3.1 Definitions of Agents and MultiAgents

Currently, there is no agreed definition for the term “agent”, as it is researched from different branches of computer science, such as, AI, software engineering and cognitive science. Therefore, several definitions for the term agent have emerged. Fox et al. [41] introduced a general definition of a *Unified Autonomous Agent*

(UAA) based on both the behavioural and cognitive capabilities of an agent. A UAA is an entity that exists in some sort of physical or informational environment. This environment must behave to an extent that is independent of the actions of that agent. Jennings and Wooldridge [61] suggested a set of behavioural features of agents instead of trying to define what an agent is or what it is not. They characterised agents as being:

Autonomous: Agents should be able to make the majority of their decisions without direct intervention from other agents or humans, also they should have a degree of control over their actions.

Social: Agents should have the ability to interact with other self-interested agents and humans in order to solve their problems or help others in solving there problems.

Responsive: Agents should be able to perceive their environment and collect information from it, they should also respond to changes in the environment or their goals in a timely fashion.

Proactive: Agents should show the ability to exhibit goal-directed behaviour and take the initiative when possible.

Jennings et al. [60] define a MultiAgent System (MAS) as, a society of agents that cooperate and communicate in order to achieve the same goal and solve a common problem. MAS possess the following characteristics.

- Each agent has incomplete information or capabilities for solving the problem.
- There is no global system control.
- Data is decentralised.
- Computation is asynchronous.

One of the difficulties in designing MAS is balancing between the agents goals and desires and between the overall aim of the system. Since, it is possible that some agents have a goal different from the MAS. Therefore, a good design should be able to solve conflicts between agents and the MAS. The term "MacroAgent" is usually used to refer to the whole agent society, while the term "MicroAgent" is used to describe a single agent in the society.

3.3.2 Differences between Agents and Objects

The term agent is currently employed in several domains in order to describe different entities. When dealing with agents from a software engineering prospective, an important debate rises on whether or not an agent is simply an object with extra capabilities or is it in an entirely different paradigm. This debate exists because many obvious similarities are found between agents and objects, for instance both of them have internal parameters; agents and objects interact with surrounding elements and none of them has complete knowledge about their environment. Nevertheless, there are fundamental differences that separate the two notions apart.

According to Jennings et al. [60] there are three major differences between agents and objects. The first is the degree of autonomy agents and objects possess. In object-oriented systems you invoke a method upon an object, in other words you instruct the object what to do. This results in objects loosing control upon their own actions. On the other hand, when dealing with agents you request actions from them, and they are free to accept or refuse. Jennings et al. [60] summarise the autonomy difference using a descriptive slogan: "objects do it for free; agents do it for money".

The second distinction between object and agent systems is with regard to the concept of flexible autonomous behaviours that includes the agents ability to be reactive, pro-active and social. Although an object-oriented system can be built that incorporates these behaviours, the classical object-oriented programming model has nothing to do with these types of behaviours.

The third important difference between agent systems and the standard object model is that each agent is considered to have its own thread of control. While in the standard object model there is a single thread of control in the system. Having their own thread of control will result in agents being independent from other agents in the system, if an agent encountered a problem a MultiAgent System will most probably continue functioning while if an object encounters an error the system will most likely fail. Recently a lot of work has been dedicated to concurrency in object-oriented systems. Although, multi-threading capabilities could be added to objects in modern languages such as Java and C#. These objects will not necessarily have the ability to exhibit flexible autonomous behaviour.

Other researchers view agents as a special type of objects. Jeff Bradshaw [7] views agents as "objects with an attitude" in the sense of agents being objects with extra added capabilities such as being reactive, pro-active, mobile and social. The same viewpoint can lead us to view objects as agents without these extra agent characteristics [90]. It could be debated that the additional agent capabilities can be added to objects. Nonetheless, it will not necessary make them agents. In addition, some basic concepts in the object-oriented paradigm such as inheritance are meaningless in agent programming.

Although, agents share some similarities to objects, there are enough differences that entitle them to a separate classification. A more detailed discussion on the similarities and differences between agents and objects can be found in [90].

3.3.3 Agent Architectures

The area of agent architectures addresses the issues of constructing computer systems that satisfy the properties specified by the different agent theories. According to a popular classification by Wooldridge and Jennings [126], there are three general types of agent architectures, deliberative, reactive or hybrid.

3.3.3.1 Deliberative Architectures

This is the classical approach for building agents; it is based on viewing agents as a particular type of knowledge-based systems. The majority of the research in this area is being influenced by the symbolic AI paradigm. Symbolic AI researchers believe that building intelligent systems can only be done through decomposition of the system into independent information processing units, which must interface with each other by the use of symbolic representation. Symbolic systems often employ the concept of a central system with perceptual modules as inputs and action modules as outputs. The perceptual modules produce a symbolic description of the world while the action modules use a symbolic representation of the desired actions and perform them in the world. The purpose of the central system then is to process the symbolic information based on logical reasoning.

The deliberative agent can be viewed as a specific type of symbolic architectures. Wooldridge and Jennings [126] define the deliberative agent architecture as "one that contains an explicitly represented symbolic model of the world, and in which decisions (for example about what actions to perform) are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulation". The same authors warn that when using this type of architecture for building agents there are at least two important problems to be solved.

1. The transduction problem: that of transforming the real world information into a precise, sufficient symbolic description in time for that description to be useable. This problem has been mainly encountered in machine vision, learning, speech understanding and so forth.

2. The representation/reasoning problem: that of how to symbolically represent information about complex real world entities and processes, also how to get agents to reason with this information in time for the results to be useful. This problem has led to work on knowledge representation, automatic planning, automated reasoning and so forth.

Despite the massive amount of work that these problems have generated, most researchers agree that neither is anywhere close to solving. As a result, some researchers have explored the use of other techniques for building agents. These techniques will be discussed in the following sections.

3.3.3.2 Reactive Architectures

The problems associated with symbolic AI have led some researches to question the practicability of the whole paradigm and whether or not these problems can be ever solved, a criticism of symbolic AI can be found in [10]. In addition, it led to the development of an alternative architecture known as the reactive agent architectures. In [126] the reactive architecture is defined as "one that does not include any kind of central symbolic world model, and does not use complex symbolic reasoning".

Examples of reactive architectures for building agents include the previously mentioned "subsumption architecture" proposed by Brooks [11]. This architecture consists of several autonomous layers, each layer at its own is capable of interacting with the environment and producing purposeful behaviour. Improvements to the systems is achieved by adding new independent layers. In his research on the alternative approach for intelligence [8] [9] [10], Brooks argues that symbolic AI is fundamentally flaw, and "Intelligent behaviour can be generated without explicit representations of the kind that symbolic AI proposes". In addition, Brooks views intelligence as an " emergent property of certain complex systems", and reason that "intelligence is in the eye of the beholder; it is not an innate, isolated property".

At the same time, other researchers have reached similar conclusions about the shortfall of the symbolic representation. They also suggested other approaches which used no symbolic representation, such as Maes' agent network architecture [78], and Rosenschein and Kaelbling's situated automata [104].

3.3.3.3 Hybrid Architecture

Some researchers argue that neither a pure deliberative nor a pure reactive approach is the best method for constructing agents. They have suggested a hybrid agent architecture, which tries to combine the strong feature of both approaches. The hybrid architecture is composed of a deliberative subsystem and a reactive subsystem. The reactive subsystem has a simpler architecture and can respond faster to external events in the world. On the other hand, the deliberative part can be used for maintaining a symbolic model for the world and for planning for future actions. The hybrid architecture usually consists of multiple layers, with the fast simple reactive components at the bottom level, and the complex deliberative components at the top levels.

An example of the hybrid architecture includes the Touring machines developed by Ferguson [36]. This architecture consists of three independent control layers, embedded in a control framework, which mediate between the layers to solve possible action conflicts. The reactive layer generates quick responses to unpredictable events in the world that happens too fast for other layers to deal with. The planning layer is responsible for generating plans and executing actions that will accomplish the agent's main goal. While, the modelling layer contains a symbolic representation of the state of other entities in the agent world. The symbolic models are used to identify and resolve goal conflict situations. The three layers communicate between each other through message passing.

3.3.4 Agent Programming and Communication Languages

In recent years, a number of agent programming languages have been proposed in order to help researchers implement agents easier. These languages aim to provide a higher level of abstraction for the concepts produced by agent theorists. Many aspects in the agent languages are influenced by concurrent object languages. Concurrent languages provide self-contained parallel executing objects, which have their own private internal states and can communicate with each other.

Agent oriented programming, was proposed by Shoham [110] as "a new programming paradigm, based on a societal view of computation". Shoham argues that any fully developed agent oriented programming language should have at least three components.

1. A logical system to define the mental state of agents.

2. An interpreted programming language for developing agents.

3. An agentification process to compile agents programs into system executable binaries.

Shoham proposed a language called AGENT0 [109], which satisfies the first two requirements. The logical component is a quantified multi-modal logic and contains three parts: belief, commitment and ability. Corresponding to this logic the agent has a set of initial beliefs and commitments, a set of capabilities (actions the agent can do) and a set of rules that determines how the agent acts.

KIF (Knowledge Interchange Format) and KQML (Knowledge Query and Manipulation Language) are two languages proposed for exchanging information and knowledge between agents [38]. Both languages form a standard for agent communication, this is needed when dealing with open MultiAgent systems, where agents may originate from various environments and developers. Other agent communication languages includes The Foundation for Intelligent Physical Agents (FIPA) ACL [39] which is based on speech act theory and similar to KQML.

3.4 Agents in Image Analysis

The use of agents in image analysis solutions is still noticeably modest, particularly for what is considered low-level image processing operations such as image segmentation. This is mainly because MultiAgent concepts did not mature until the mid 1990s, and since there is still confusion in the literature as to what makes a system truly agent-based. Some of the proposed agent-based approaches in image analysis merely re-implemented traditional image processing algorithms using agent terminology. For example, in both [118] and [72] a variant of the split and merge algorithm have been proposed. In addition, in [6] an agent-based region growing algorithm was introduced for the segmentation and the analysis of living cells. There is no clear benefit from using the agent approach in these algorithms, also, it is questionable if they actually provide better results than the traditional split and merge or region growing algorithms. Other proposed approaches, utilised agents as part of the solution. For instance, in [45] a Multi-Agent system was integrated into a traditional framework. Thus, missing out on the benefits a MultiAgent system might provide.

There are several possible advantages from the use of agents in image analysis. Perhaps the most significant advantage is the ability to design systems based on decoupled units. Monolithic systems are harder to design, implement and most importantly maintain. Since, the highest cost in any software solution mainly consists of maintenance expenses. Crevier [27] indicated the following list of advantages an agent-based approach could provide to image analysis systems.

Inherently parallel: The ability to benefit from parallel computing.

Ease of construction and maintenance: It is easier to build and maintain semi-independent entities than a single large program.

Heterogeneous problem solving: The methods suitable to a part of a problem may not be suitable for another part. Control strategies can choose the best processing methods for different data situations.

Reliability: An agent can be corrected by other agents. Agents can discuss there answers and agree upon the best answer.

Focussing ability: Not all of the knowledge is needed for all tasks. Different agents can focus on different objects in the image.

3.5 Summary

In this chapter, an overview on the rational agent approach and MultiAgent Systems has been provided. There are several proposed approaches for creating intelligent systems. These Range from approaches that believe that the road for machine intelligence is building systems that think like humans (cognitive science), act like humans (Turing test), think rationally (symbolic AI) to systems that act rationally (agent approach). A MultiAgent System (MAS) is a society of autonomous and rational entities that cooperate and communicate in order to achieve the same goal and solve a common problem. The MAS approach is more flexible than the symbolic AI approach, since correct deduction is just one of the several methods for achieving rationality. In Addition, it is more suitable to scientific development than approaches based on either human behaviour or human thought. The agent philosophy of constructing incremental layers of intelligence is a favourable method for the creation of a generic image segmentation framework. Since, it is easier to build and maintain semi-independent entities than a single large program.

Chapter 4

Agent Society for Image Processing: Generic Framework

4.1 Introduction

Image segmentation is the process of assigning a group of pixels in an image to distinct objects or the background [66]. Segmentation is usually a perquisite for many higher-level tasks in image analysis such as reconstruction, matching and recognition. Therefore, the quality of the segmented output significantly affects the subsequent steps in image analysis. Satisfactory segmentation output provides a better rate of success for later stages, on the other hand, poor quality almost always guarantees failure or significant inaccuracy for these stages.

As mentioned earlier in Chapter 2, hundreds of image segmentation algorithms have been proposed in the literature, ranging from low-level algorithms such as thresholding to more complex model based solutions such as deformable models. Nevertheless, to the author's knowledge, no single algorithm can be considered suitable for all sorts of images and applications. Moreover, most of these algorithms are either derivatives of low-level algorithms or created in an ad-hoc manner in order to solve a particular segmentation problem. The classical view in machine vision research considers image segmentation as a low-level operation, which should only depend on the image data. In a statement of this traditional view, Marr and Nishihara [81] state that the problem starts out with an intensity array and concludes with a description that only depends on that array. This view combined with the difficulty of providing a general-purpose solution to the segmentation problem has created several limitations with the traditional image segmentation algorithms, such as the following.

- Most algorithms do not benefit from the widely available domain knowledge about the objects to be segmented. Algorithms that do benefit tend to have the knowledge closely integrated within.

- Most algorithms are limited to a specific set of applications or image modalities, algorithms designed for ultrasound images are usually not suitable for MRI images. This has resulted in a large literature base on image segmentation containing hundreds of different algorithms.

- The motivation for most segmentation solutions was to tackle a specific application. This resulted in the design and implementation being application driven as opposite to be driven be the general problem of segmentation. Adapting the solution for another application is considered as a later step, and not taking into consideration during the initial design. Nonetheless, customisability should not be regarded as an isolated feature that can be added at a later step, it should be an integral part and considered while designing the framework.

The aim of this chapter is to propose and discuss a generic agent framework for image segmentation. The framework is not intended to provide a single solution to all image segmentation problems, such a goal is not achievable. Instead, the framework aims to provide a generic shell that can be customised for many image segmentation problems. The framework is based on some of the original motivations of active contour models combined with the intelligent collaboration and problem solving abilities of MultiAgent Systems.

4.2 Generic Segmentation Frameworks

Designing and implementing a customisable image segmentation framework is obviously not a trivial task. Even the selection of a suitable algorithm for a specific type of image modality is considered a difficult problem. Several medical segmentation algorithms are ad-hoc in nature, which have been proposed to solve a particular application. In addition, as pointed out by [96], application specific algorithms usually perform better than most general algorithms by taking into account prior knowledge. Nevertheless, it can be argued that these algorithms provide limited benefit to the general segmentation problem. Since these algorithms are usually not customisable to other applications and in some cases are not applicable to different image types. Therefore, there is more need for generic segmentation frameworks that are customisable for a wide range of application. Since the number of realistic segmentation applications is on the increase through major advancement in imaging devices especially in the medical domain. Accordingly, the focus should be on trying to solve the general problem of finding objects in a digital image and not special case applications.

To classify a segmentation algorithm as generic and suitable for many applications, it should include most of the following characteristics.

- The algorithm should incorporates prior knowledge about the objects into the segmentation process. Simple knowledge such as object location, grey-level intensity and relation with surrounding structures can enrich the segmentation process through the compensation of noisy and missing image data situations.

- It should incorporate shape restrictions about the object such as size, circularity and concavity. Alternatively, have some sort of shape matching facilities that can work as a feedback mechanism to ensure that the segmented object resembles the target shape. Shape restrictions and matching are useful in noisy environments and incomplete and missing image features situations.

- Customising the framework to different applications should not be a tiresome process. The algorithm should minimise the number of user-tuned parameters, in opposite to the fine-tuning problem endured while using most active contours.

- It should not depend on a large data set for learning or for creating the object profile. As access to a large study data might be restricted or collecting a large sample might be impractical for some real life objects. This requirement rules out Artificial Neural Networks (ANN) that requires a large number of images for training and testing. ANNs have been successful in some areas of image processing such as pre-processing and filtering. Nevertheless, a solution solely based on neural nets is not suitable for general segmentation frameworks. In addition to the large number of images problem, Egmont-Petersen et al. [33] has identified several ANN limitations that make them unsuitable for image understanding such as choosing the best ANN architecture, overtraining and the black-box problem, details are left for the respective paper.

- The general-purpose algorithm should possess both a reliable bottom-up data-driven process along with a strong top-down knowledge-driven process within a single robust framework [30].
- The framework should be assembled from semi-independent units organised in a flexible structure. Each semi-independent unit should be specialised in solving one problem. In addition, they should communicate and collaborate to achieve the common goal of segmentation. Semi-independent units add flexibility to the framework as it is easier to design, implement and maintain semi-independent units than a single large unit.
- Future addition of the semi-independent units should be taken into consideration while designing the system. Adding or removing a unit should not break the system.
- The framework should possess a level of intelligence and learning capabilities to make it adapt to different domains easily. Thus, the system should benefit from the experience gained while segmenting an object in later segmentations.
- The generic framework should possess self-awareness about the current segmentation task. This awareness will help the framework decide when to

trust an image feature and ignore the constraint information or vice versa.

4.3 Intelligent Multilayered Approach

This thesis proposes the Agent Society for Image Processing (ASIP), which is a MultiAgent framework for intelligent image segmentation. The goal of this model is to automatically locate and identify objects boundaries in a digital image. This goal is accomplished by dividing the problem into smaller problems and having an expert agent working on each individual sub-problem. This “divide and conquer” policy is essential when designing segmentation systems that rely on an approach that combines knowledge and data. Since it provides greater control and flexibility over the whole system by focusing on a single sub-problem at a time. The framework is divided into five layers based on the different type of functionality each layer provides with relation to the image segmentation process. The top layers can be viewed as the mind of the framework and they possess the planning and control capabilities. On the other hand, the lower layers are thought of as the body of the framework with simple reactive behaviours, which will perform the actual segmentation. Figure 4.3 demonstrates the general architecture of the model. The framework is designed based on the MultiAgent architecture, where each layer represents a high-level agent, specialised in a specific aspect of the image segmentation problem such as planning, shape constraints or knowledge management. The high-level agents communicate and collaborate using message passing techniques, while the low-level MicroAgents communicate using a shared blackboard system.

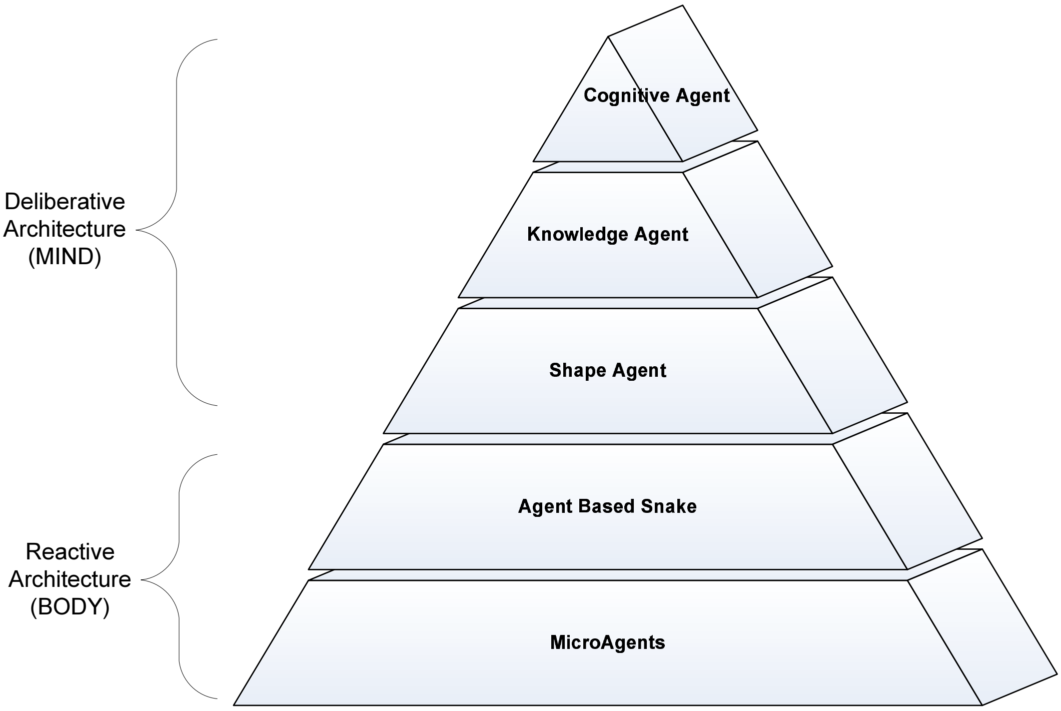


Figure 4.3: Agent Society for Image Processing Framework.

The motivation of designing the framework as a multilayered architecture originate from Brooks work on the subsumption architecture [8, 9, 10], which was previously presented in Chapter 3, and Terzopoulos and Hamarneh work on Artificial Life systems [51, 52, 53, 114]. Brooks opposes the traditional symbolic AI school that state that symbolic representation is required for building intelligent systems. Instead, Brooks proposes a system with incremental layers of intelligence that operates in parallel.

Terzopoulos and Hamarneh propose a virtual organism for medical image analysis based on an ALife modelling approach. The artificial animal possesses the key components of its real counterpart. It contains a body that contain muscle actuators, sensory organs (eye, nose, etc.) and a brain that is responsible for cognition operations such as planning and learning. To manage the complexity of Artificial Life models they are best organised in hierarchical layered approach, with each successive layer adding to the functionality of lower layers.

Although the proposed framework can be described in ALife terms as having a body and mind, it is not based on an ALife modelling approach. The ALife method lacks flexibility for a general-purpose segmentation framework. Since the different layers are closely coupled making it difficult to modify or append layers without affecting the other layers. In addition, the ALife single agent framework does not allow multiple organisms to work collectively on a different object in the same image. This is of high importance since many segmentation applications require to locate more than one object, usually these objects share a boundary making it useful to share information between two parallel processes. In addition, the proposed framework differs from Brooks's architecture in the sense that it has a hybrid architecture in terms of artificial intelligence schools. The top three layers benefit from the deliberative architecture. On the other hand, the lower two layers are based on a simpler reactive architecture. There is a need for the higher layers to have control over the segmentation process, which a pure reactive model does not provide.

4.4 Framework Overview

The description of the framework will commence in a bottom-up fashion. This will serve to illustrate how the framework was designed in an incremental way. In addition, it will permit better realisation of the framework's intelligent nature. First, the Agent Based Snake (ABS) and the MicroAgents will be discussed together because of their close nature. Next, the shape and knowledge agents responsible for the shape restrictions and knowledge integration will be described. Finally, a discussion of the cognitive agent responsible for controlling the framework is given.

4.4.1 Agent Based Snake Model

Current solutions for solving the snake energy minimising problem such as variational calculus and dynamic programming are not suitable for integration in an intelligent flexible system. Intelligence should not be regarded as an isolated feature of the upper layers, but it should also come from within the lower layers. The research proposes re-implementing the active contour as a society of rational collaborating agents. Rational agents are agents that do the right thing under the available information and computational resources. The concept of rational agency, as discussed in Chapter 3, is considered as a leading candidate for the development of systems that demonstrates intelligence [106][107].

An Agent Based Snake (ABS) is a society of collaborative MicroAgents inhabiting a virtual world. The term micro serves to illustrate that these agent have a simple architecture and they are part of a higher macro agent. The society itself is comparable to an enhanced ACM, where the MicroAgent resembles semi-independent snaxels or control points. The MicroAgent possesses reactive behaviours that are stimulated by the semi-dynamic environment that can change by responding to the MicroAgents actions. The higher layers are responsible for populating and initialising the environment. The MicroAgents goal is to locate image features by finding minimum energy locations and maximising its performance measure. The ABS contains four entities as illustrated in Figure 4.4, namely the virtual environment, MicroAgents, snake segments and a blackboard system for agent communication and collaboration.

4.4.1.1 Environment

The virtual agent world V, of size NxM is a 2D discrete semi-dynamic environment. The environment contains the pre-processed image data view of the framework. Each location L or energy pixel contains the image features needed to attract MicroAgents and can be inhabited by at most one MicroAgent at the same time. The energy pixels can be populated to attract the agents to edges

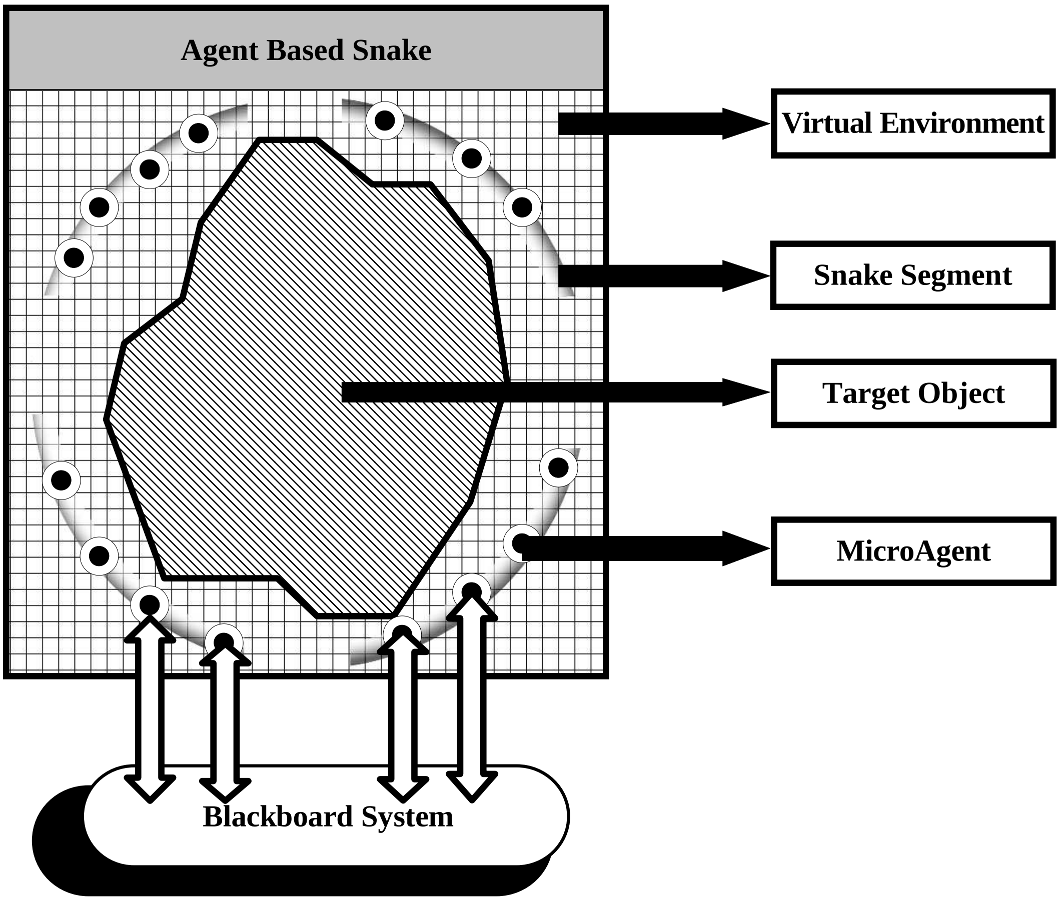


Figure 4.4: Agent Based Snake (ABS) Model.

by processing the image by an optimal edge detector such as Canny or Sobel. The environment is comparable to the ACM external energy term. However, it is more flexible by allowing the MicroAgents to combine or choose from multiple image features or energies at the same time. For example, the energy pixels can contain the gradient of the input image along with the image intensity value. At the same time, the MicroAgents can be guided to search for a strong image gradient that belongs to an image region with a specified range of image intensity values. Hence, multiple image features can be combined in a similar fashion to make it more robust.

According to Russell and Norvig [107] proposed classifications of task environments, this virtual world has the following characteristics.

Partially observable: Each MicroAgent has a partial view of the environment at each point in time. The sensors return energy pixels in the current agent neighbourhood. For example, typical sizes of the neighbourhood would be 3x3 and 5x5. The size is relative to the input image resolution and features sought after.

Strategic: The next state of the environment can be determined by the current state and actions executed by a single MicroAgent. Nevertheless, the environment is strategic since it is partially observable and other MicroAgents can independently change the state of environment.

Episodic: The MicroAgent experience is divided into atomic episodes or iterations. In each episode, the agent perceives the environment then performs the action. The next action does not rely on actions taken in previous episodes, thus, there is no need for the MicroAgents to plan ahead.

Semi-dynamic: The environment energy pixels content does not change over time unless acted on by MicroAgents. MicroAgents can move and inhabit energy pixels, which results in marking them as unusable locations for other MicroAgents.

4.4.1.2 MicroAgents

Each MicroAgent consist of sensors to perceive the environment, beliefs about its internal status, performance measure to determine how well it is doing and actuators to act on the environment. After populating the society and initialising the first MicroAgents they search the world for a better location to inhabit that contains the desired image feature. If the search resulted in the discovery of a better location the MicroAgent moves to it. The movement is influenced and restricted by the following factors (energies):

- Image features such as lines, edges, etc. This information is stored in the environment energy pixels.
- Knowledge about the desired target object such as average object intensity, statistical region information, etc. The knowledge is stored in the blackboard system.
- Other MicroAgents experience and discovery about the world, this contains information such as the other agent's distance from the centre and average energy values.

- Shape restriction stored in the form of snake segments. The segments are computed from the object profile and each segment can have a different flexibility classification that affects the level of freedom the MicroAgent in terms of dealing with the shape restrictions. Snake segments are covered in the shape agent section.

These factors are not computed as a traditional energy minimising equation. In other words, the best location is not a summation of the four energies with each multiplied by a parameter. Instead, the MicroAgents first look for more stable image features and then uses the other factors to decide if it should trust this feature or not. This will allow it to deal with missing boundaries situation and overcome noisy environments.

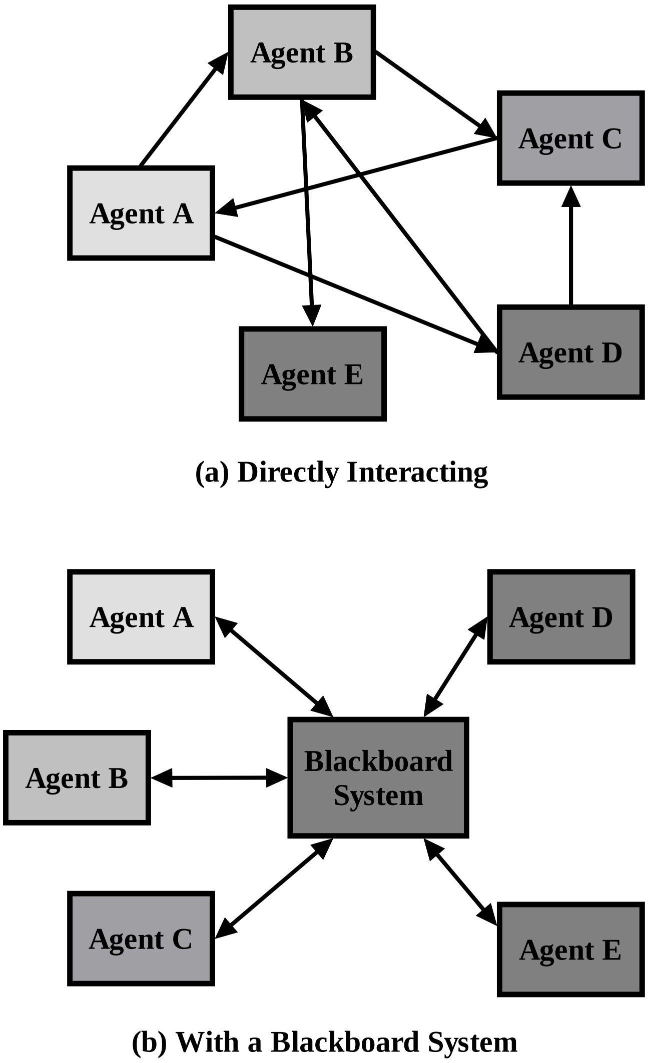
The performance measure is used to evaluate the MicroAgent and to determine whether it is acting in a rational way. A good MicroAgent finds a minimum energy location in the lowest amount of iterations. Otherwise, the MicroAgent is considered superfluous for the society and is then terminated. This will ensure that society size stays relatively small to avoid unnecessary computation. Table 4.2 summaries the basic set of MicroAgent behaviours that will enable the MicroAgent to combine the available data/knowledge efficiently and find the desired image feature.

Behaviour	**Description**	**Trigger**
Move	Moves to a new location in the neighbourhood	The MicroAgent sensed an empty minimum energy location better than the current location
Breed	The parent MicroAgent creates new independent MicroAgents	Several suitable locations were found these locations all meet the requirements
Destruct	MicroAgent is no longer needed and vanishes from the world	Failed to find and move to a good location after some time
Communicate	Read/write the world experience to blackboard	Reading is done at the beginning of the step and writing at the end
Sleep	Waits until other MicroAgents finish. When all of them are sleeping the segmentation ends	The current location is a minimum energy location and no other locations are better

Table 4.2: MicroAgent Behaviours.

4.4.1.3 Blackboard System

Mainly there are two methods for agent communication and collaboration. The first is direct interaction between agents (Figure 4.5(a)), it is referred to in the literature as direct message passing [25]. Agent ‘A’ sends a message to agent ‘B’ which agent ‘B’ will respond to in a direct way. This method requires that all agents involved in the communication are aware of each other and have decided on what knowledge/results to share. The second method is using an indirect method of communication such as the blackboard system (Figure 4.5(b)). A blackboard [35] is a shared memory structure that is available to all agents and it serves as a communication and repository medium. Agents using the blackboard system are not directly aware of other agents. They broadcast their message for everyone to see. For more details on different collaboration methods between agents refer to [25].



(a) Directly Interacting

(b) With a Blackboard System

Figure 4.5: Communication methods between agents.

In the ABS model, a blackboard system is used for indirect communication between MicroAgents. The communication will enable the MicroAgents to exchange information and findings (i.e. experience) about the world. The blackboard system also contains knowledge about the target object (inserted by higher agents) to use in the segmentation process. The choice of using a blackboard system for MicroAgents communication is for two reasons. The first is that it is faster to communicate with one entity (blackboard) than several entities (other MA). The second is that a single MA does not need to know about the identity and numbers of other MAs in the world.

Another level of communication occurs in the society between snake segments. After the MicroAgents in each segment finish searching for an energy minimum. It becomes necessary to merge (link) the segments. This is done by negotiating the merging between two segments at the same time. Each segment merges with the closest segments that preferably share a common area in the world. In this case, some MicroAgents will be destroyed if they exist in both segments. On the other hand, if there were no common area new micro agents will be introduced to fill the gap between the two segments. This will be repeated until there is only one segment remaining. The communication is achieved using direct message passing as for the number of the segments is predefined for each object type.

4.4.2 Shape Agent

The shape agent is a collection of predefined snake segments, its main function is to provide shape constraints for MicroAgents. The snake segments act like attraction forces for the MicroAgents enforcing the final segment to be similar to a previously defined segment in terms of size, location and curvature. In addition to the shape constraints, the shape agent also deals with missing or incomplete boundaries which results from noise or poor image quality.

4.4.2.1 Importance of Shape Constraints

Image gradient alone is not a sufficient measure to use while segmenting real life objects. Boundaries are not always well defined through edge detectors; they might have discontinuity, noise, and multiple objects can share the same boundary. Most ACM internal energy constraints penalise high curvature and force the contour to become smooth. Nevertheless, smoothness is not always a desirable feature, shapes can be convex or concave and there is no single constraint that can be useful when applied to all the shapes.

A more logical approach is to use the available shape knowledge about the target object and integrate it into the segmentation process. The integration can be done in two ways, the first is to use shape matching techniques [120] to check the intermediate segmentation result and act as a feedback mechanism. The second method is to introduce shape restrictions directly into the MicroAgents environment. The first method involves expensive computational operations such as translation, rotation and scaling. In addition, when matching the similarity between two patterns a threshold decision problem arises, namely, how to choose the right threshold that will help decide if the dissimilarity is smaller than the threshold? On the other hand, the second method involves introducing relaxed shape restrictions based on an approximation of the target boundary. These shape restrictions work as an attraction force for the MicroAgents pushing them into keeping a similar shape as that of the profile shape. Combined with the image gradient and statistical data the shape constraints helps the MicroAgents locate the target boundary. A major benefit of this method is that shape knowledge is integrated in each MicroAgent who decides based on the environment and logical rules on what to do next.

4.4.2.2 Shape Agent Structure

Instead of having shape constraints that affect the whole shape a structural decomposition approach is taken. The shape agent is divided into segments or regions, each segment can have different properties and constraints than the other segments. This results in making each segment semi-independent and more flexible in terms of handling complex shapes. Dividing the shape into segments is done based on their quasi-convex/concave properties a method proposed by

Nishida [88][89]. Although Nishida used this method for structural indexing and classifications, it is a beneficial method to be used when adapted to the shape agent. Representing a shape based on convex/concave structures that incorporates quantised-directional features provides a simple compact shape representation yet still containing rich features. The method details are left for the respective references [88][89].

After dividing the shape into several segments, each segment is then classified as follows.

Rigid: when there is high certainty of the target shape, size and location. The shape agents are expected to give the shape constraints a high weight.

Medium: constraints apply but there is a certain level of variations allowed.

Flexible: the shape segment is highly deformable. Freedom is given for the MicroAgents to ignore the shape restrictions and focus on other restrictions.

This classification is then used by the MicroAgents to solve disputes and make logical decisions about the best location to move to when the several forces point in different directions.

4.4.3 Knowledge Agent

Many researchers classify the knowledge used in image segmentation and understanding frameworks into two main groups [27]. The first is general-purpose knowledge that is independent of any application domain. For example, knowledge about the imaging tools in use, planning and control knowledge. The second is specialised knowledge about the application domain that includes object position, shape and size. In the proposed segmentation framework, three layers deal with knowledge and they are the cognitive, knowledge and shape agents. The cognitive agent possesses general-purpose knowledge, i.e. planning and control knowledge. It is aware of other agents' existence and capabilities enabling the cognitive agent to create a plan to solve a specific segmentation task. Despite the fact that both the knowledge and shape agent deal with application domain knowledge they were separated into two agents for the following reasons.

Specialisation: the shape agent deals with geometric knowledge about the target. On the other hand, the knowledge agent deals with region based statistical knowledge.

Representation: each of them influences the Agent Based Snake in a different approach. The shape agent creates snake segments, which imposes shape restrictions on the society. While the knowledge agent stores knowledge directly into the society blackboard.

Flexibility: different types of application knowledge are not always available or needed in all segmentation tasks. This division allows either agent to be dropped from the segmentation plan. For example, if the target object geometric structure is highly unpredictable but statistical region knowledge is available, then the shape agent is dropped from the segmentation task.

Applications that only depend on local edge-based image features such as the classical snake are usually sensitive to noise and the initial estimation. Utilising application specific knowledge in the segmentation process makes the framework more robust to noise and missing image features situations. The kind of domain knowledge to use is based on the application and image modality in use. Domain knowledge suitable for remote sensing applications might not be appropriate for a medical problem. The use of statistical region knowledge to aid in image segmentation has been reported several times in the literature. By way of example, Boscolo et al. [5] used location, size and image intensity values to segment the chest wall from CT scans. Chesnaud et al. [21] used a Maximum Likelihood Estimation (MLE) approach based on the statistics of the inner and the outer regions of an object.

Two main issues exist in the process of integrating domain knowledge into a generic segmentation framework. The first is how to represent the knowledge into the system? The second is how to collect the application knowledge? In the proposed framework, the region knowledge will be represented as energy parameters stored by the knowledge agent into the ABS blackboard. These energy parameters will then act as attractions forces pulling the MicroAgents into a region that matches a statistical parameter. There can be negative and positive knowledge energy parameters, positive parameters push the MicroAgents into a region while negative parameters pull them from a region. For example, using the image intensity as a knowledge indicator for the target object and the background, the object's positive energy parameters attracts the MicroAgents into the object region, while the negative parameters push the MicroAgents out from the background. As for collecting the application knowledge, this can be achieved through two methods. Either semi-automate the process through the user interface or having an expert user manually collect it. While automating the knowledge collecting process is a highly desirable feature it is outside of the scope for this research because of time constraints.

In addition to initialising the ABS and populating the blackboard with region-based knowledge, the proposed knowledge agent is responsible in handling experience gained from a segmentation task. After a successful segmentation, the knowledge agent updates the region statistics with the current region information and this newly learned knowledge is then used to segment the next slice in a time series dataset or stored back into the object profile for later segmentation tasks.

4.4.4 Cognitive Agent

To deal with different applications, planning knowledge about solving a specific task domain is needed. In addition, in order to create the plan, knowledge about the available tools (agents) is also needed. The cognitive agent can be compared to an expert human user, it is aware of other agents' existence and capabilities enabling the cognitive agent to create a plan to solve a specific segmentation task. As indicated by Crevier and Lepage [27] planning in image understanding can be achieved mainly by two methods. The first is to use a search-based planning procedure. For example, after the user has specified a goal such as finding a rectangle in a given region in the image. A graph where its nodes represent image features (edge image, line, rectangle) is searched. The directed arcs of the graph correspond to image operators, such as gradient calculation, edge linking or noise removal. The plan then consists of the path through this graph, starting from the raw image and ending at the image feature sought after. The second method is to use a case-based strategy, where a number of plans or subplans used to solve earlier problems are stored in a case database. The plan or subplans most suitable to the current segmentation task is selected and modified as required.

In the proposed agent framework, a case-based strategy is more suitable for two main reasons. First, the number of agents (operators) in the framework is limited, there are four agents apart from the cognitive agent. Second, a case-based strategy is more extensible than search-based methods, adding more applications are done by simply adding more cases. Figure 4.6 displays a typical process flow for the system. After the user loads the images and the object profile, the cognitive agent creates the plan and distributes the tasks between the knowledge and shape agents, whom in turn prepare the virtual world and the blackboard for the MicroAgents.

In addition to the cognitive agent planning and managing role, it also handles the following tasks in the framework.

User Interaction: all the user interactions are handled by the cognitive agent. The user inputs the raw image data and object profile through a GUI. After the plan has finished the cognitive agent collects the output from the required agents and presents it to the user. In a semi-automatic segmentation, the user can have a greater role by judging and verifying intermediate results.

Low-level image processing: basic image processing operations such as noise reduction and gradient computation are performed by this agent. These operations are basic and generate a single output therefore; they are integrated into the cognitive agent and not into a separate agent.

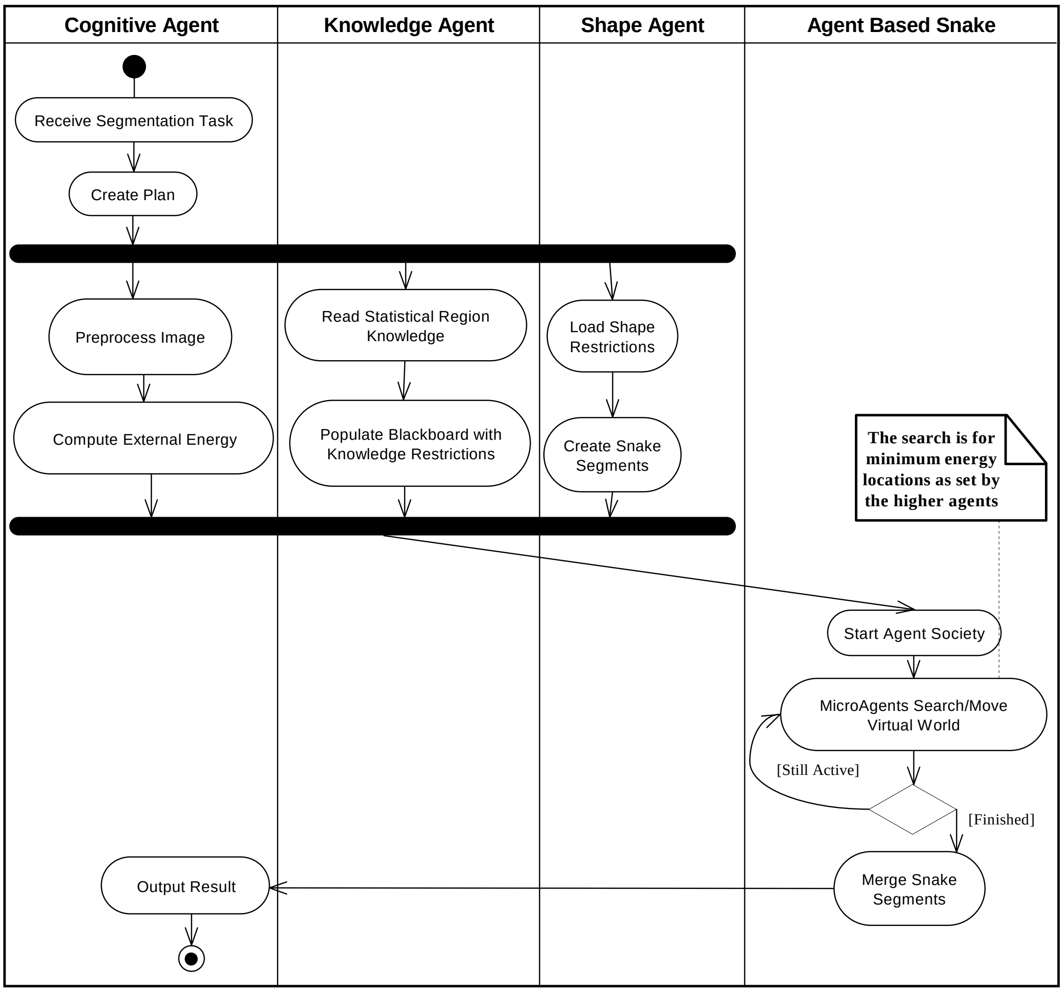


Figure 4.6: Simplified UML Activity Diagram for the Framework.

4.5 Summary

This chapter proposed the Agent Society for Image Processing (ASIP), which is a generic framework for intelligent image segmentation motivated by the active contour model and MultiAgent Systems. ASIP is presented in a hierarchical manner as a multilayer system consisting of several high-level agents (layers). The bottom layers contain a society of rational reactive MicroAgents that adapt their behaviour according to changes in the world combined with their knowledge about the environment. On top of these layers are the knowledge and shape agents responsible for creating the artificial environment and setting up the logical rules and restrictions for the MicroAgents. At the top layer is the cognitive agent, in charge of plan handling and user interaction. The framework as a whole is comparable to an enhanced active contour model (body) with a higher intelligent force (mind) initialising and controlling the active contour.

In the next chapter, the proposed ASIP framework is customised for the segmentation of the left ventricle from a 4D MRI dataset. Appendix A contains description about the software realisation of the framework and the application developed for the medical case study.

Chapter 5

Segmentation of the Ventricle using the Proposed Framework

5.1 Introduction

Cardiovascular Diseases (CVD) are the primary cause of death in the UK [95], and account for 233,000 deaths in 2003 (around 1/3 of all deaths). Fortunately, most of the CVDs are curable when diagnosed early. The detection of CVD is performed by using either invasive or noninvasive methods. Invasive methods such as diagnostic cardiac catheterization [70] and Transesophageal Echocardiography (TOE) [37] consist of inserting objects into a patient's body and can present a risk to the patient's life. On the other hand, most noninvasive diagnostic imaging methods can be safely used for the diagnosis and treatment monitoring of many

CVDs. Images acquired through cardiac imaging allow accurate measurements of cardiac factors such as left ventricular size, myocardial mass and myocardial perfusion [124].

Current methods used in cardiac imaging include ultrasound, Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). While ultrasound is a relatively cheap and noninvasive system, it gives low contrast when compared to MRI and CT and has a low signal-to-noise ratio (SNR). CT utilizes an x-ray based technique to acquire several 2D images (slices) of internal structures. These slices can be used to reconstruct a 3D image representation of the structures [67]. Although CT is capable of both high resolution and high speed, cardiac CT is considered as a health risk as it is equivalent to 500 chest X-rays in terms of radiation [123].

Cardiac Magnetic Resonance Imaging (CMRI), which is used in this chapter, is a established and commonly used imaging modality in diagnosing heart diseases [77]. CMRI is a safe noninvasive modality that can be used to acquire high resolution 4D images (3D + time) of the heart. The produced images have a high contrast difference between the blood and the myocardium muscle without the need of a contrast agent. However, the images can suffer from noise and variation of grey value between adjacent slices [69] [77].

Obtaining the cardiac measurements from the various image modalities involves segmentation of a large dataset (around 200 slices in CMRI for a single cardiac cycle). When segmentation is performed manually, the process suffers from the disadvantages of being time consuming, tedious and prone to user error. In this chapter, the agent-based framework proposed in Chapter 4 is employed to automatically segment the Epicardium (outer layer) and Endocardium (inner layer) from a short-axis MRI dataset. The aim of this is chapter is to demonstrate an example of populating the proposed framework through tackling a challenging medical case study.

5.2 Medical Problem Analysis

Cardiovascular imaging is an important tool for the diagnostic of heart diseases as it allows the detection of diseases without the need for risky invasive surgery. However, analysing the medical images in order to retrieve the cardiac measurement is not a trivial task. One of the essential steps in the image analysis process involves segmenting the digital images. The datasets for this medical case study were obtained using a 1.5 Tesla GE MRI scanner in the short axis view and is stored in Digital Imaging and Communications in Medicine (DICOM) file format. DICOM is a standard for storing, handling and transmitting media in medical imaging. DICOM allows information obtained from different hardware vendors and different image modalities to be dealt with in a uniform way. A typical DICOM file contains two parts, the raw image data and a header containing descriptive tags such as patient, modality and slice information.

The short axis view is obtained across the heart cutting through the two ventricles. The images were obtained at different locations from the base to the apex of the heart for the cardiac cycle. The resulting 4D data is stored as a set of 2D slices in separate DICOM files with each slice having an image resolution of 256*256 pixels. Figure 5.7 demonstrates a slice from the dataset near the centre of the heart. Each patient's dataset contains around 10 locations and 20 phases (i.e. time steps) in the cardiac cycle resulting in a total of some 200 slices.

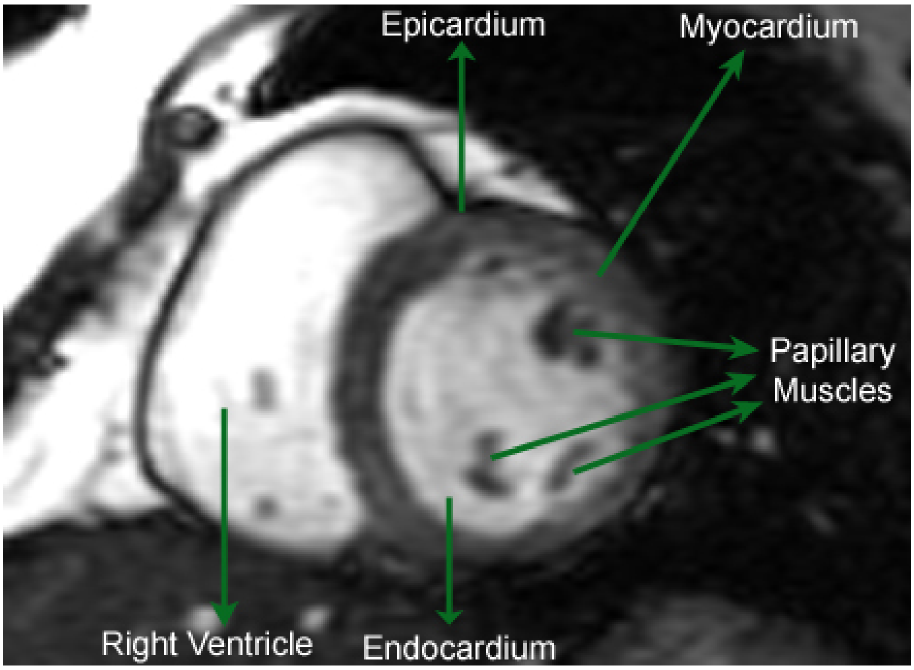


Figure 5.7: Annotated 2D slice from the dataset obtained in the short axis view.

Automatic segmentation of the left ventricle from a 4D dataset is a challenging problem. In addition to the general noise, partial volume effect and the cardiac motion, several other factors contribute in making this medical case study difficult:

1. The variety in shape and size of the endo/epicardium structures between adjacent slices and between patients. Figure 5.8 contains four different slices, the upper two are from the same dataset while the lower two are from a different patient dataset. Notice the obvious variety in shape and size between slices. Figure 5.8(a) demonstrates the standard shape in short axis view for the endo/epicardium where they appear to have a circular form. Nevertheless, due to anatomical variations between patients, heart defects and noise they can possess more complex shapes as displayed in Figure 5.8(b-d). This variety makes it difficult to impose strict shape and size constraints on a segmentation model. Moreover, it limits the use of shape based segmentation models such as ASM [102] and AAM [32].

2. The variation of grey-level values for the same object between adjacent slices. This problem as indicated by [77] is caused by the MRI scan itself. Figure 5.9 shows endocardium histograms for two adjacent slices. Although, the two adjacent slices bare great visual resemblance their histograms clearly differ. Because of this variation in intensity levels, standard intensity motivated image segmentation algorithms such as thresholding and region growing alone will not be able in general to segment the LV

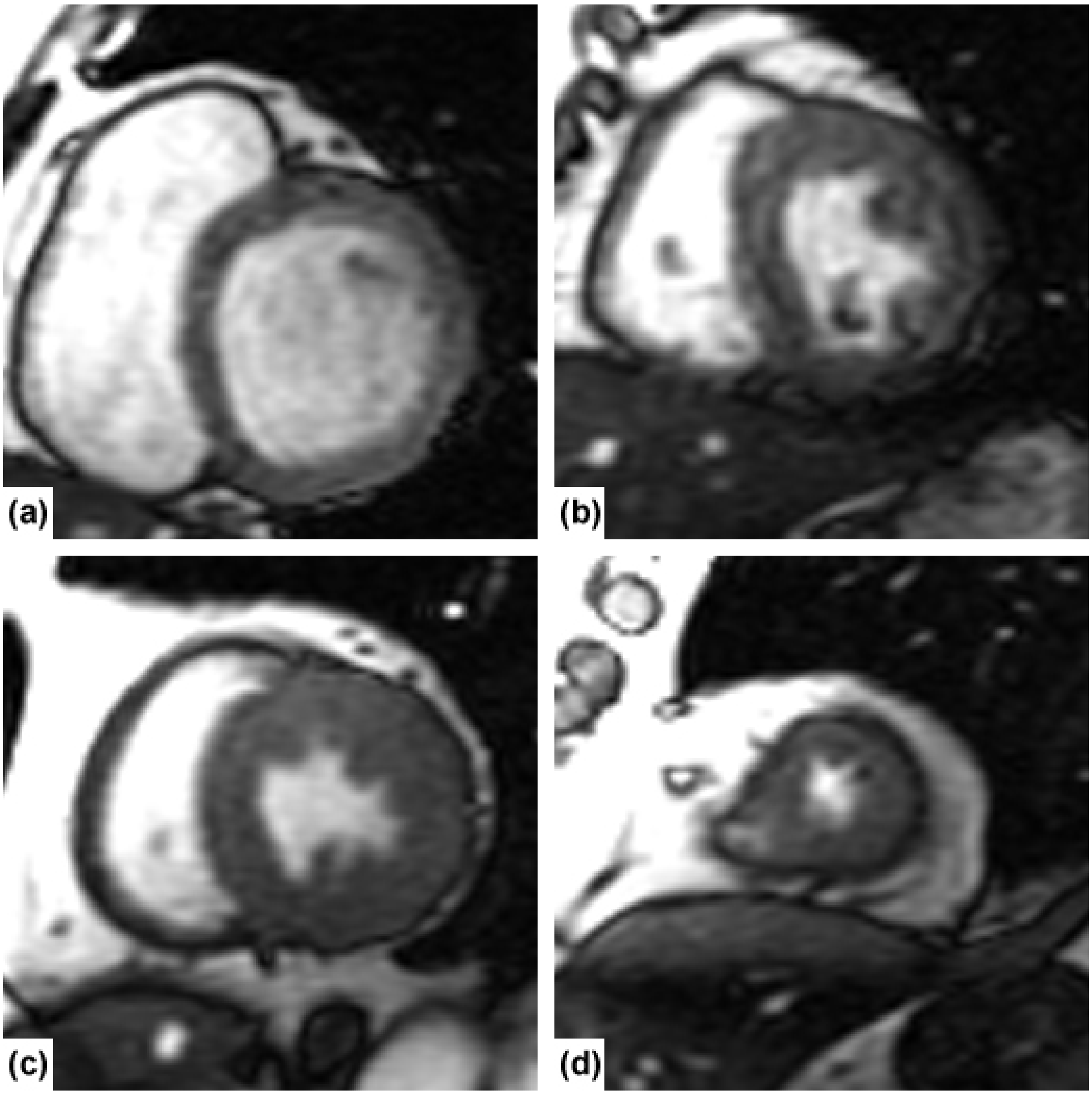


Figure 5.8: The high variance in the left ventricle shape due to slice location and anatomical differences.

throughout the dataset. Moreover, care needs to be taken when utilising intensity information between neighbouring slices or when using image intensity level as a restriction.

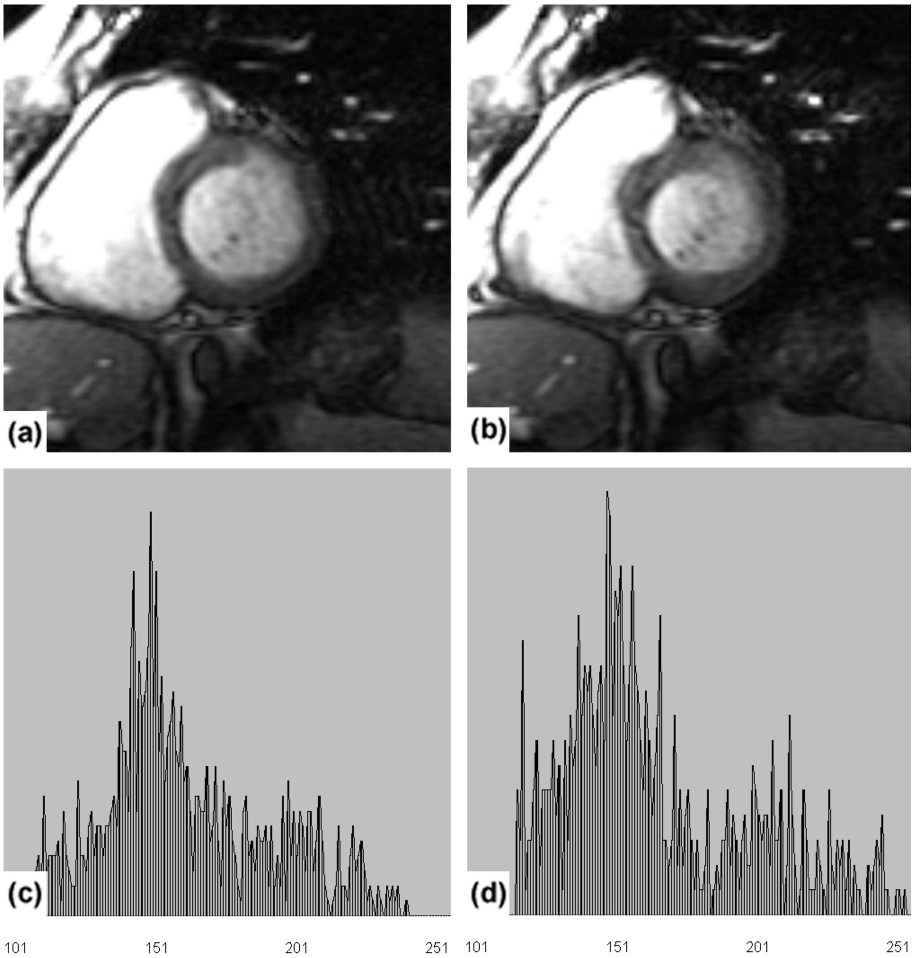


Figure 5.9: Endocardium histograms for two adjacent slices, observe the difference in grey-level values between them. (a) Slice from the medical dataset, (b) neighbouring slice for “a”, (c) endocardium histogram for “a”, (d) endocardium histogram for “b”.

3. The existence of papillary muscles inside and attached to the boundary

of the endocardium in some of the slices, see Figure 5.7. These muscles appear as small dark structures with intensity levels comparable to that of the myocardium. As a result, differentiating them from the myocardium becomes a difficult task. Nevertheless, the existence of papillary muscles should be taken into consideration when segmenting the endocardium as ignoring them can significantly degrade the segmentation results.

4. The low intensity difference between the epicardium and some of the surrounding structures particularly to the bottom of the epicardium. Edges in digital images correspond to a noticeable change of intensity levels. The first derivative of the image function has a maximum at the position corresponding to the edge in the image, and the second derivative have a zero at that position. Figure 5.10 demonstrates the result of applying the Canny edge detector to a slice in the dataset. Note the lack of edges to the right and bottom of the epicardium because of the low intensity difference. The use of boundary based algorithms alone such as Snakes and edge based algorithms to segment the epicardium is not feasible as it will yield poor results.

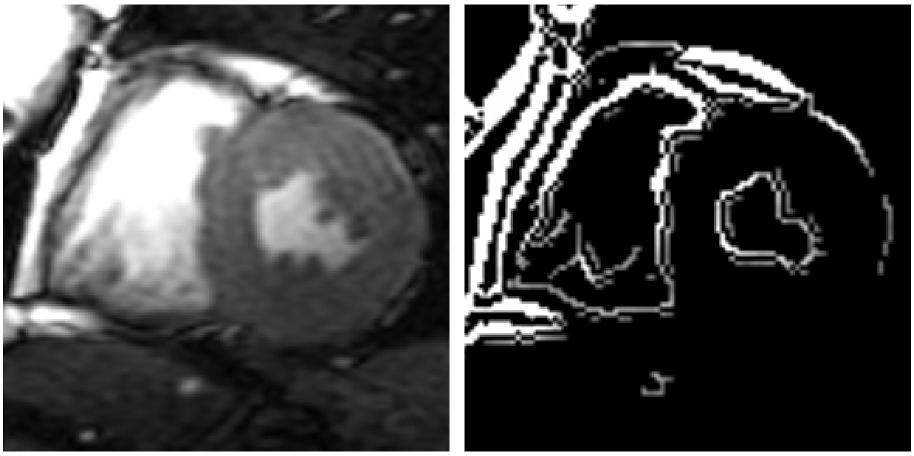


Figure 5.10: Canny edge detector ($\sigma = 1.4$) applied to a slice in the dataset. No edges were detected in the bottom of epicardium because of the low intensity difference.

5.3 Related Work on the Left Ventricle

The segmentation of the left ventricle from cardiac images has been achieved by others using three main approaches. The first approach is manual segmentation where the cardiologist or an expert user manually delineates the myocardium with the aid of a medical imaging toolkit such as DICOMWorks [99] or Amira [113]. The manual segmentation process is time consuming, tedious and prone to user error. It also suffers from inter-observer (between observers) and intra-observer (within observers) variability of the results. The second approach developed to aid the cardiologist in delineating the myocardium is semi-automatic segmentation techniques [44] [122]. In these interactive approaches the user draws an initial contour around the LV and interacts with the contour while it attaches itself to high gradient points. Although this approach considerably reduces the time needed to segment the left ventricle it is still time consuming bearing in mind the large number of slices in 4D datasets. In addition, it is still subject to inter and intra observer variability.

The third approach is a fully automated system for the segmentation of the left ventricle. This approach is the most desired method of medical segmentation as it requires no (or minimal) user interaction during the segmentation process. In addition, it is the fastest method when compared with the manual and semi-automatic techniques. In recent years, several systems have been proposed in the literature to automatically segment the LV [15] [97] [65] [86] [92] [69] [98] [77]. These automatic segmentation systems can be classified into two main groups. The first group are systems which depend on prior statistical and shape knowledge learned from a manually segmented training set [86] [92] [69] [98]. For example, Kaus et al. [69] integrated a multi-model prior knowledge that consists of shape, spatial relationship and intensity values into a deformable model. The second group of methods do not depend on any previously learned knowledge, instead it combines the results obtained from applying different image segmentation techniques [77] [65] [97] [15]. As a way of example, Lynch et al. [77] combined adaptive smoothing, edge detection and a clustering technique together to segment the LV.

Obviously, each group of the automatic techniques has its own advantages and pitfalls. The first group yields good results while segmenting slices similar to that of the training set. However, it cannot capture noticeable variability outside the learning set, which is likely to occur when there is high variance in the medical data. Also, creating a knowledge base from a training set is a tedious task and is often performed manually. On the other hand, the second group does not require any prior knowledge and thus no training is needed. Nevertheless, linking different image segmentation techniques together depends on a high level of parameterization in the system. In addition, adapting these parameters to different datasets and medical applications is not a straight forward operation and it might include a great deal of modifications to the system.

5.4 A Proposal for Populating the Generic Framework

In Chapter 4, a customisable generic MultiAgent framework was introduced with the purpose of handling various types of segmentation tasks. The framework was designed in a hierarchical manner, where the bottom layers consists of a society of rational reactive MicroAgents that adapt their behaviour according to changes in the world combined with their knowledge about the environment. On top of these layers are the knowledge and shape agents responsible for creating the virtual environment and setting up the logical rules and restrictions for the MicroAgents.

At the top layer is the cognitive agent, in charge of plan handling and user interaction. The framework is comparable to an enhanced active contour model (body) with a higher intelligent force (mind) controlling the active contour. In addition to the added control layers the framework possesses several other advantages over the active contour. For instance, the active contour energy minimisation process relies entirely on its potential energy function E(x,y). However, it is not always possible to represent attraction/repulsive forces as functions. In addition, region-based forces such as texture or colour cannot be easily integrated into a potential function. On the other hand, by designing the Agent based Snake as a society of cooperative MicroAgents a much wider range of forces can be used. Furthermore, attraction/repulsive forces can be imposed at different levels in the society, the whole society (macro), parts of it (segments) or a single agent (micro).

Several approaches were considered in order to customise the framework for the left ventricle case study. One of the initial propositions was to create a database of statistical region knowledge for the endo/epicardium. Afterwards, the prior knowledge is combined with gradient based image features and applied as attraction/repulsive forces for the MicroAgents. There are two main problems associated with this approach. Firstly, the difficulty associated with the creation of the knowledge database as it required a manual segmentation of a large number of slices. Secondly, the variability of intensity levels between patient's CMRI slices. While this method will work on slices with intensity levels similar to that of the training set, it is not guaranteed to work when applied to a dataset with a noticeable variance in intensity levels. Another proposed method is to divide the MicroAgents society into snake segments computed from a prior shape model. With each snake segment having its own properties and shape restrictions based on its spatial position in the image. Again, this method was also discarded because of the visible anatomical shape variability in the endocardium.

After concluding that depending on pre-computed statistical and shape knowledge is restrictive, the direction of the research shifted into utilising high level a priori knowledge about the left ventricle. A level of knowledge that exists in all slices and will not change from a patient to another. Examples of a priori knowledge include, the endocardium is inside the epicardium, endocardium has a lighter region than epicardium, both are closed structures and epicardium shares a boundary with the right ventricle. In the next section, a detailed description of the cognitive plan proposed to segment the left ventricle is introduced.

5.5 The Segmentation Plan

5.5.1 Definitions

The proposed plan for segmenting the LV is based on a multi-stage multi-behaviour approach. Each structure in the LV segmentation process goes through two main phases, a region phase followed by a boundary phase. In the region phase MicroAgents are deployed to segment the inside of the object. Afterwards, the segmented region is analysed by the shape agent to create the “initial contour MicroAgents”, and also by the knowledge agent to create the “region attraction force”. In the second phase, “contour MicroAgents” are attracted to the boundary and influenced by internal shape forces designed to keep the resulting curve smooth. Figure 5.11 displays the process flowchart for the endocardium segmentation. The epicardium process differs only in the attraction/repulsive forces used but mainly it follows the same steps. Before discussing the plan further, it is necessary to define some of the key terms used.

Definition 5.5.1 (Homogenous Region) A region is homogenous if the pixels inside it share some form of a similarity measure. The criteria of homogeneity can be based on grey-level, colour or texture. In this research a uniformity criterion based on the comparison of the max/min difference between the value of a pixel and the average over a region is utilised.

For a region R of size N let

$$m = \frac{1}{N} \sum_{P \in R} f(P) \tag{5.6}$$

Then a region is called homogenous if

$$max|f(P) - m| < T, P \in R \tag{5.7}$$

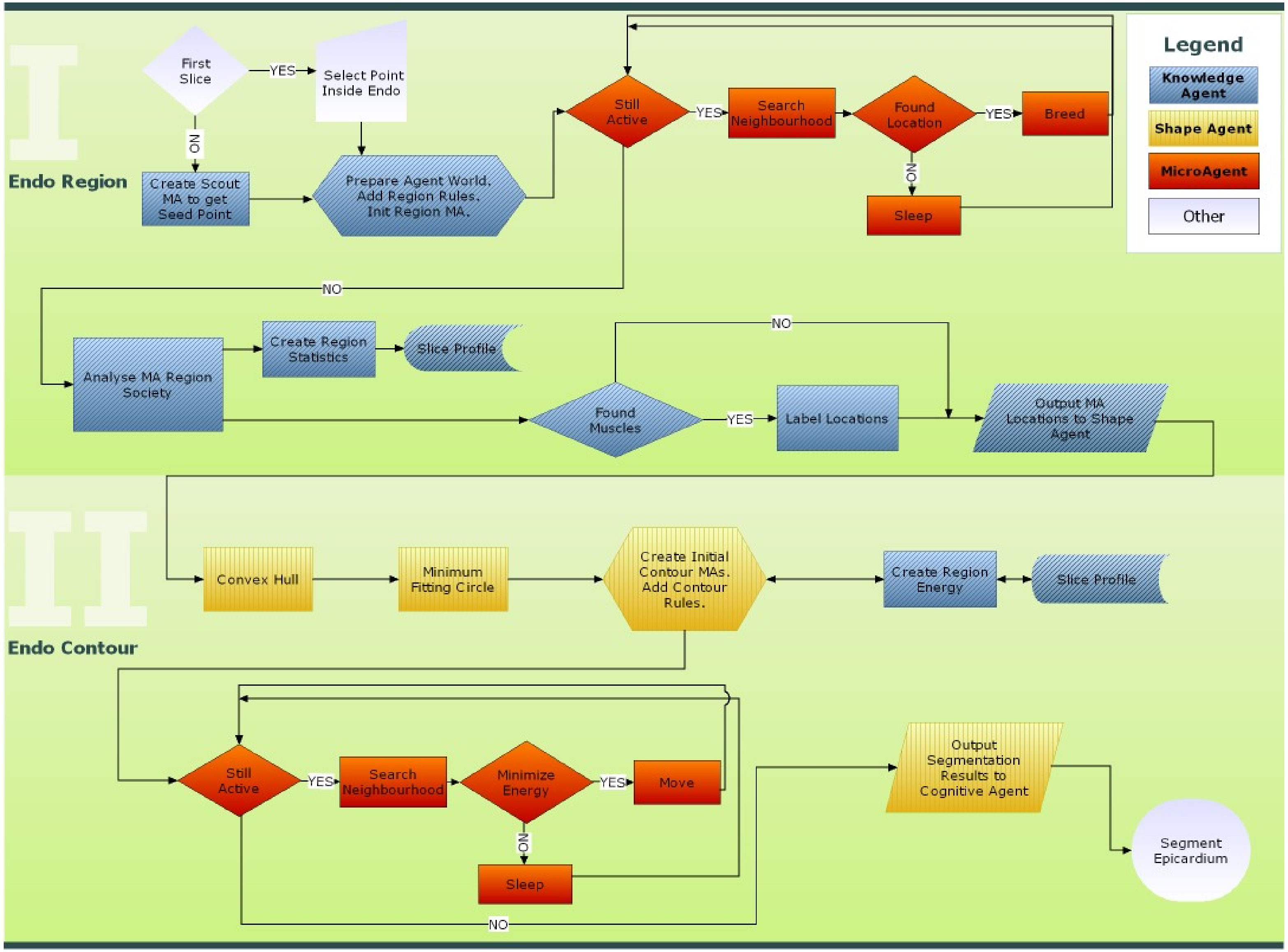


Figure 5.11: The multistage process for segmentation the endocardium using the proposed framework. The plan contains two main phases, a region phase and a contour phase.

for some threshold T.

Definition 5.5.2 (MA Internal Forces) The internal forces are utilised to restrict the MicroAgents movement in the world and encourage them to maintain a smooth shape in areas that lack strong image features. The internal forces used in this research are based on Williams and Shah's [125] continuity and curvature terms.

$E_{continuity}$: this energy encourages even spacing between MicroAgents. It ensures that MicroAgents do not cluster up on strong image areas. The continuity force is computed as the difference between the average distance between MicroAgents and the distance between the current MA and the previous MA.

$$E_{continuity} = |AVG_{distance} - Current_{distance}| \tag{5.8}$$

$E_{curvature}$: This energy penalises high curvature values and pushes the MicroAgents into forming a smooth circular curve. The curvature is estimated between the current MA, previous and next MA.

$$E_{curvature} = |MA_{n+1} - 2MA_n + MA_{n-1}|^2 \tag{5.9}$$

Definition 5.5.3 (MA Image Forces) The image forces are used to attract MicroAgents into pre-computed features such as edges and image gradient. Although, several authors have proposed high attraction forces such as gradient vector flow [128] and distance potentials [23] they were not used for two reasons. First, these forces perform poorly in noisy environments such as medical datasets. Secondly, the shape agent is responsible for placing the MicroAgents close to the target boundary therefore there should be no need for a high attraction force. The currently employed image force is based on a Gaussian smoothed Sobel edge detector, this force has a good attraction range and performs well under noisy environments.

Let $I(x, y)$ be the intensity function of the image, the image energy function designed to attract the MicroAgents is

$$E_{ext}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2 \tag{5.10}$$

where $G_\sigma(x, y)$ is a 2D gaussian function with standard deviation σ, ∇ is the gradient operator, and * is the edge detector.

Definition 5.5.4 (Region MA) The region MA is a MicroAgent attracted to homogenous locations inside the world. The region MA behaviour is influenced by the knowledge about the world, sensory inputs and the behaviour of other MicroAgents in the region.

Definition 5.5.5 (Contour MA) The contour MicroAgent is similar to the region MA but with two main differences. First, it is primarily attracted to objects boundaries. The second, its behaviour is restricted by internal shape forces.

Definition 5.5.6 (Scout MA) The scout MA can either search for regions or boundaries. The main difference between it and the region and contour MicroAgents is that it works individually and is not affected by other MicroAgents. The primary use of a scout MA is to find starting points for structures using a set of logical rules.

Definition 5.5.7 (Endo/epi region) The left ventricle as illustrated in Figure 5.7 comprises of three layers, endocardium (inner layer), myocardium (middle layer) and epicardium (outer layer). To simplify the naming conventions in the remainder of this chapter, it will be assumed that the left ventricle consists of two objects (structures), one inside the other. The inner object will be referred to as endo and the outer object will be referred to as epi. The term "endo region" will be used to refer to the region inside of the endocardium layer excluding the papillary muscles. Likewise, the term "epi region" will be used to refer to the region between the epicardium and endocardium.

5.5.2 Region Phase

The first phase in the cognitive plan is the region segmentation of the endocardium. This phase is essential in the process as it provides valuable knowledge about the region. This knowledge is used in phase two to guide the contour MicroAgents into the boundary. The phase can be considered as a substitute of the manual extraction of prior statistical knowledge performed by expert users in some systems. Note that, the region and contour phases are similar for both the endocardium and epicardium, the main difference is the attraction/repulsive forces imposed on MicroAgents. Whenever a difference exists it will be referred to, otherwise it is implied that the same applies for both the endocardium and epicardium. The region phase has three inputs.

1. Seed location: The location of the first region MA inside the object. For the first slice of the endocardium, the location is manually inputted by the user using the developed software. For the epicardium and subsequent slices the seed location is automatically located using a scout MA.

2. Homogenous criteria. The similarity measure that will attract MicroAgents to locations in the world (see Definition 5.5.1).

3. Region MA logical rules. These rules triggers different MicroAgents behaviours such as searching, breeding and terminating. The rules are stored in the blackboard system by the knowledge and shape agents and are read by the MicroAgents on their creation. For the "endo region" a location in

the world is part of the region if it satisfies the following rules

$$L(x,y) \in Region_{Endo} \equiv \sim edge \wedge \ \sim inhabited \ \wedge \ homogenous$$

As for the "epi region" the rules are

$$L(x,y) \in Region_{Epi} \equiv \sim edge \wedge \ \sim inhabited \ \wedge \ homogenous \ \wedge \ I(x,y) \\ < MIN(Endo_{Intensity}) \ \wedge \ distance < (Myocardium_{width} + Endo_{radius})$$

After preparing the virtual world and storing the rules in the blackboard, the first region MA starts searching its neighbourhood for locations that satisfy the logical rules. Based on the local search findings the following behaviours are triggered.

Move: The MA moves to a new location if only one suitable location is found.

Breed: The MA breeds to new locations if multiple suitable locations are found.

Sleep: The MA sleeps if no more suitable locations are found and current location is suitable.

Terminate: The MA terminates (dies) if no locations are found and the current location is not suitable.

Report Edge: The MA avoids locations that contain edges, but reports edge points to the blackboard as it will be used later on in muscle/noise detection. This behaviour is only triggered in the "endo region" MA.

MicroAgents keep searching the world until all of them are inactive (sleeping), their combined behaviours resulting in the segmentation of a homogeneous region. This region is then analysed by the knowledge agent to create statistical knowledge such as, intensity mean, standard deviation and variance. The knowledge agent also detects and labels muscle locations in the environment so that endo contour MicroAgents do not get attracted to them. A structure is a muscle if it meets the following conditions.

$$Muscle \equiv \ edge \wedge \ \sim inhabited \wedge I(x,y) < MIN(Endo_{Intensity}) \wedge \ \in Region_{Endo}$$

5.5.3 Contour Phase

In this phase in the segmentation plan, contour MicroAgents collaborate to locate the boundary of a medical structure. The behaviour of the contour MicroAgents society is comparable to an enhanced active contour model. In addition to the energy minimisation goal, the MicroAgents must also comply with the logical rules stored in the blackboard by the shape and knowledge agents. The main attraction force for endo contour MicroAgents are edges since they are clearly defined in the endocardium case. As for the epi contour MicroAgents they are attracted to a region force because of the difficulty of obtaining strong edges resulting from the low contrast difference between epicardium and its surrounding structures.

The starting positions for the contour MicroAgents are computed from the region segmented in the previous phase. Since the endocardium and epicardium have a circular shape in general, a good starting estimate would be finding the Minimal Enclosing Circle (MEC) of the segmented region. Afterwards, the initial contour MicroAgents are placed on the circle boundary. MEC algorithms are used to solve the problem of finding the smallest circle that completely contains a set of points. In this research, a MEC algorithm based on Elzinga and Hearn [34] is used, the algorithm description is provided in Algorithm 5.1. Elzinga and Hearn's algorithm has a time complexity of $O(n^2)$, to reduce the high computational requirements the convex hull of the region is first approximated using Andrew's algorithm [3]. Observe that the MEC is entirely determined by the convex hull of a given set of point. For this reason, points of the set residing on the minimal enclosing circle are always on the convex hull of the set.

After the shape agent obtains the initial locations of contour MicroAgents and stores internal shape restrictions in the blackboard, the knowledge agent then creates a region attraction force from the statistics obtained in the previous phase. The region force attracts/repulses MicroAgents to a region with a homogenous criterion similar/different to that of the previously segmented region. For ex-

```
Input: A set of points on a plane, produced from the region MicroAgents
       segmentation.
Output: Minimal enclosing circle, represented by radius R and centre
        point (Cx,Cy).
1. Choose any two points, Pi and Pj
2. Construct the circle whose diameter is l2(Pi, Pj)
if circle contains all points then
    centre of the circle is the optimal solution
else
    choose a point Pk outside the circle
end
3. if triangle determined by Pi, Pj and Pk is a right/obtuse triangle then
    rename the two points opposite the right angle as Pi and Pj
    goto step 2
else
    construct the circle passing through the three points
    if circle contains all the points then
        stop
    else
        goto 4
    end
end
4. Choose point Pl not in circle
Choose Q from Pi, Pj, Pk with greatest distance to Pl
Extend diameter through Q to a line that divides the plane
Choose R from Pi, Pj, Pk that is in half plane opposite Pl
With points Q, R, and Pl, goto step 3
```

Algorithm 5.1: Minimal Enclosing Circle (MEC) algorithm used to find the minimum circle covering a set of points on a plane, based on [34].

ample, endo region force will attract contour MicroAgents into the endocardium region and away from the epicardium region. The region forces are important as the image forces alone might be stronger in surrounding structures than the target structure. This will result in the MicroAgents being wrongly guided into surrounding structures. Then again, employing the region forces ensures that MicroAgents are guided to the correct structure even if the image forces are stronger in elsewhere structures.

The following rules are stored in the blackboard to influence the endo contour MA behaviours.

$$L(x,y) \in Contour_{Endo} \equiv \; Minimise(E_{int} + E_{ext}) \wedge \; \sim inhabited \; \wedge \\ \sim muscle \; \wedge \; Inside \in Region_{Endo} \; \wedge \; outside \ni Region_{Endo}$$

As for the epi contour MicroAgents the rules are

$$L(x,y) \in Contour_{Epi} \equiv \; Minimise(E_{int} + E_{region}) \wedge \; \sim inhabited \; \wedge \\ Inside \in Region_{Epi} \; \wedge \; outside \ni Region_{Epi}$$

In a similar fashion to the region MicroAgents, contour MicroAgents search the local neighbourhood for locations that minimises the energy and comply with the rules. If such a location exists then the MA moves to it otherwise it sleeps. After the contour MicroAgents are all inactive (sleeping), their combined behaviours results into locating the boundary of the target object and thus concluding the segmentation task.

5.5.4 Moving Between Structures and Slices

After segmentation the endocardium from a single slice, a scout MA is deployed to find a location $L_{Epi \leftrightarrow RV}$ on the shared boundary between the epicardium and the right ventricle. The $myocardium_{width}$ is then approximated as the distance between the endocardium boundary and $L_{Epi \leftrightarrow RV}$. Estimating the $myocardium_{width}$ is of great importance as it is used to restrict the epi region MicroAgents search area, $distance < (Myocardium_{width} + Endo_{radius})$. After adding this distance to the logical rules, four region MicroAgents are then initialised between the endocardium boundary and $L_{Epi \leftrightarrow RV}$. The epicardium segmentation then follows the same steps as the endocardium segmentation with differences only in the attraction/repulsive forces used.

There exist two options into moving the MicroAgents to another slice in the dataset. The first option is to change the agent environment by populating it with the new slice data. Followed by, waking up the existing endo/epi contour MicroAgents whom in turn will respond to changes in the environment and attach themselves to the boundaries. The second option involves finding the region of the endocardium in the next slice using a scout MA. After that, repeating the same steps from the start until finding the endocardium and epicardium boundaries. Although the first option is faster, the second option of repeating the whole process is selected for two reasons. The first reason has to do with the noticeable intensity difference between slices for the same structure. Creating the region force from a single slice then applying it to the whole dataset will limit the usability of the region force and might cause erroneous results. The second reason for repeating the process is not making the segmentation error propagate between slices.

The segmentation output of each slice is stored in a separate XML file. These files are read by the framework in subsequent slices in order to initialise the endo region scout MA. For example, when segmentation slice number "two" the XML results file for slice number "one" is read and so forth. The use of XML as a storage medium is to allow straightforward integrating of the segmentation results with other imaging systems such as computer-guided surgery, diagnosis and treatment planning systems.

5.6 Key Issues in the Software Realisation of the Framework

Several Agent-based programming languages were proposed in the literature to implement software Agents such as Agent0 [109] and APL [64]. The languages authors argue that a specialised Agent language is needed to implement Agents instead of using current object oriented languages. Essentially these languages focus on dealing with the algorithmic part of the Agent, knowledge representation and communication/coordination between Agents. Nevertheless, there is doubt if these languages are genuinely needed. After all, the most fundamental difference between MultiAgent systems and distributed objects is more at the design and philosophical level than the programming level. Another direction in Agent languages propose building the language on top of existing object oriented languages. As an example, Grosso et al. [49] [48] [117] suggested an Agent programming framework based on the C# language and the Common Language Infrastructure (CLI). This direction is more promising since it does not re-invent the wheel and is suitable for a wide range of Agent applications. Further debate on the need of specialised Agent-based languages can be found in [85].

At the time of writing, the mentioned Agent languages are still in the design stages with no software development tools publicly available. This resulted in limiting the choice of a programming language to an existing object oriented language such as C++, C# or Java. While little differences exist between them C# was chosen to implement Agents because of its strong object oriented features and performance advantage over Java.

Another issue related to the software development is the level of execution required for Agents. They can be implemented as separate processes, threads or using a quasi-parallel approach. Agents implemented as processes have more autonomy as each agent process is totally independent from other processes in the system. Nevertheless, the computation requirements are high and process management is difficult as it is only possible at the operating system level. Agents implemented as threads are not independent, but they are able to share system resources. Threads have medium computation requirements, managing them are easier and this can be done at the application level. The quasi-parallel approach has the lowest computational requirement, but it has the lowest level of autonomy between Agents as they all share the same thread.

The level of Agents execution in the system is also related to the targeted segmentation application. It is recommended to implement Agents as multi-threads or multi-processes running on several machines when dealing with large datasets such as remote sensing images. As this case study is intended for a medical application running on a single machine a quasi-parallel approach was employed.

The DICOM reader implemented in the system in order to handle medical images is based on a modified ezDICOM ActiveX control [103]. ezDICOM is an open source medical viewer licensed to use under the BSD license.

The reader is refered to Appendix A for a more detailed description of the software implementation, which also includes screenshots of the software application.

5.7 Experiments and Discussion

The cognitive plan proposed for segmenting the left ventricle using the framework was applied to five different patients' datasets. The collaboration between agents in the system led to successful segmentation of the endocardium and epicardium structures. In Figure 5.12, the plan is applied to a single slice in order to demonstrate the outcome of the major steps. Figure 5.12(a) displays the location of the seed region MA. The starting location of the first MA has little impact on the final result as long as it initialised inside the endocardium and outside of a muscle. The user selects the initial location for the first slice, as for the subsequent slices it is automatically located using a scout MA. The process of selecting the initial location in the first slice can be eliminated if knowledge about the estimated structure location in the image was known in advance. As it is only performed once by the user for the whole dataset there is little significance in automating it.

A factor with a significant effect on the endo region MicroAgents behaviour is the value of threshold T in Equation 5.7. A large value of T will result in the endocardium region being over-segmented, while a small value of T will result in the region being under-segmented. Threshold T value has less effect on epi region MicroAgents behaviour since they are bounded with more rules. Two rules noticeably decrease the significance of T in epi region MicroAgents. The first is search-

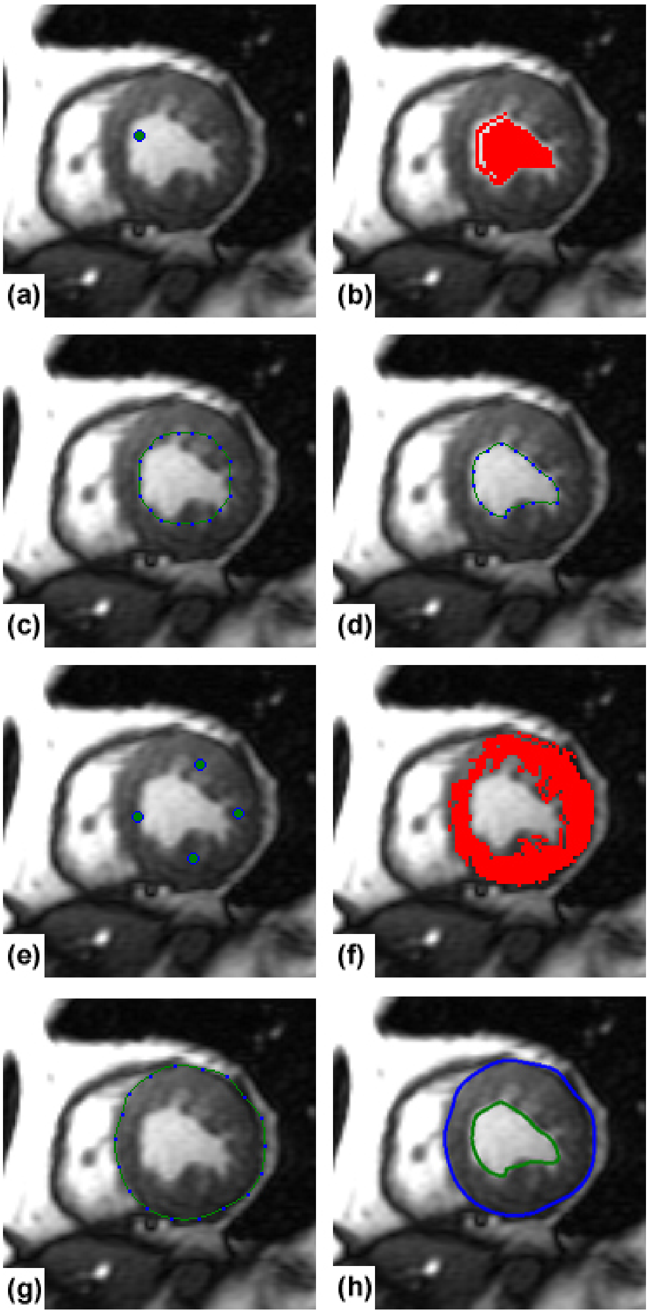


Figure 5.12: The outcome of MicroAgents behaviour at different stages in the plan. (a) initial endo region MA, (b) the segmented endo region, (c) initial contour MicroAgents, (d) endo boundary segmentation, (e) initial epi region MicroAgents, (f) the segmented epi region, (g) epi boundary segmentation, (h) final segmentation result.

ing for locations darker than the endocardium region, $I < MIN(Endo_{Intensity})$. The second is the restricted range of the search area where it should be inside $Myocardium_{width} + Endo_{radius}$. Observations from the endocardium histogram combined with empirical studies of the patients' datasets resulted in finding a good estimate of T to be around 40 ± 2.

Figure 5.12(b) demonstrates the result of the MicroAgents after segmenting the endo region. The quality of the endo region segmentation need not be very accurate, since it is only a preparation step for the contour MicroAgents as discussed earlier. In Figure 5.12(c) the initial endo contour MicroAgents are displayed. Although, the starting locations are computed from a MEC algorithm there exists no circular constraints on the shape of the boundary. The final shape of the boundary only depends on the reactive behaviours of the contour MicroAgents. The outcome of the boundary segmentation of the endocardium is displayed in Figure 5.12(d). A circular constraint utilised in some LV segmentation algorithms would have limited the MicroAgents ability to locate this specific endocardium accurately.

The start of the epicardium segmentation region phase is displayed in Figure 5.12(e). The location of the four starting region MicroAgents is automatically found by scout MicroAgents. Initialisation of one region MA inside the epicardium might be sufficient. Nevertheless, four MicroAgents were initialised on either side of the epicardium in order to start with a better distribution of the average intensity levels. This results in enhanced segmentation of the epicardium region in noisy and inhomogeneous environments. Figures 5.12(f-g) demonstrates the outcomes of the epi region and contour phases respectively. The final outcome of the endocardium and epicardium segmentation is displayed in Figure 5.12(h).

5.7.1 Experiments with the Framework

To study the effect of the region force on the behaviour of contour MicroAgents experiments with and without the region force were conducted. In the first experiment endo contour MicroAgents are initialised without the region force. As the result show in Figure 5.13(c) the MicroAgents got attracted to parts of the epicardium boundary since it has higher image energy than that of the endocardium boundary. In the second experiment MicroAgents are initialised under the same conditions as the first experiment except with adding the region force to the logical rules. The region force attracts the MicroAgents towards the endocardium region and away from the epicardium, see Figure 5.13(d).

In the epicardium boundary segmentation the region force is the main attraction force, since it is difficult to obtain edges in part of epicardium boundary as previously discussed. Without the region force the only forces influencing the MicroAgents behaviour are the internal energies which eventually force them to shrink into forming a smooth shape.

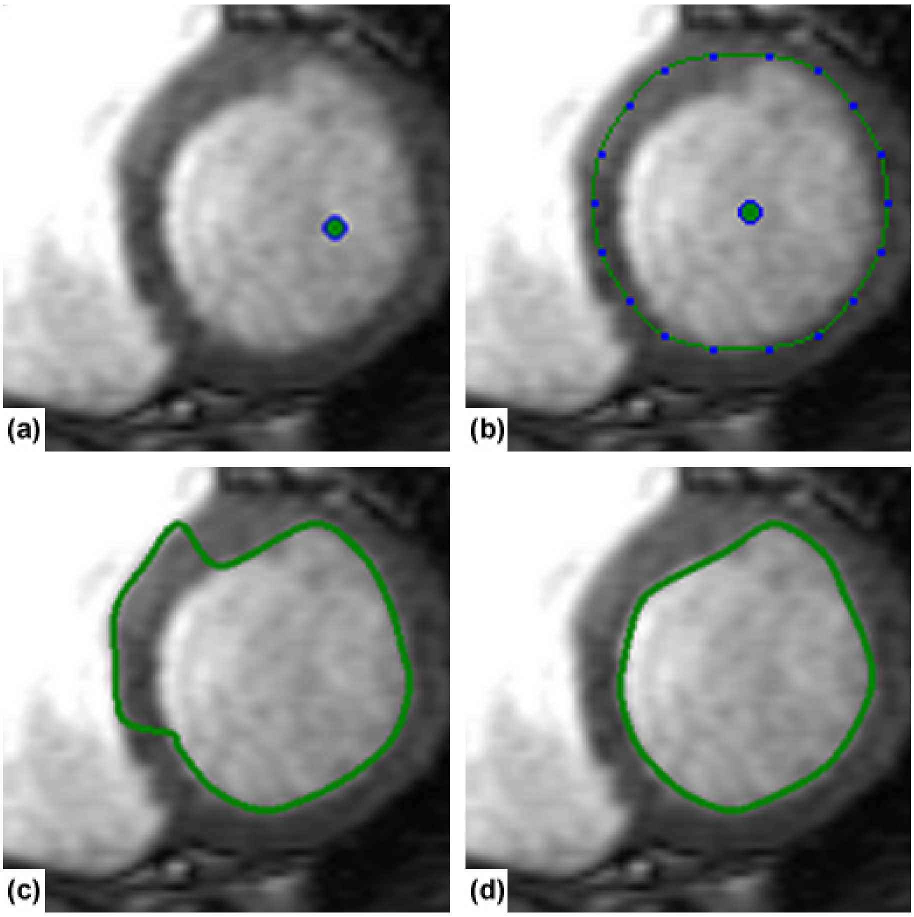


Figure 5.13: MicroAgents behaviour with and without the region force. (a) Initial region MA, (b) Initial contour MicroAgents, (c) Result without the region force. Some MicroAgents are attracted to the epicardium because it has higher image energy, (d) Result with the region force. The region force pulls the MicroAgents towards the endocardium region.

Figure 5.14 demonstrates the results of applying the cognitive plan to different slices from various spatial locations and different patients' datasets. These slices are selected at different locations from the base to the apex of the heart in order to demonstrate how the MicroAgents adapt to noticeable variation in slices. One of the downsides of depending on intensity levels to create region energy forces is that endocardium and epicardium intensity levels do overlap on some slices. The overlap usually occurs just at the end of the epicardium region and start of the right ventricle. This area can not be captured by the MicroAgents as part of the epicardium, therefore some epi contour MicroAgents will be to some extent outside of the real boundary and inside the region as shown in Figure 5.14 (a-b).

The muscle detection and labelling technique performed by the knowledge agent searches for structures with dark intensity levels inside the endocardium region and labels their pixel locations as muscle locations. These locations act as repulsive forces for contour MicroAgents later on. To reduce the complexity of the muscle detection procedure, muscle segments attached to the boundary are not dealt with. The reason for this is that it is difficult to differentiate them from the actual boundary. Figure 5.14(c) demonstrate an example of this situation, a muscle is attached to the endocardium boundary and surrounded by what seems to be epicardium region. It is even hard to decide for a trained human expert if this is a muscle or if it is part of the epicardium region. Therefore in this plan muscles attached to the boundary as the referred to case are left unhandled. The

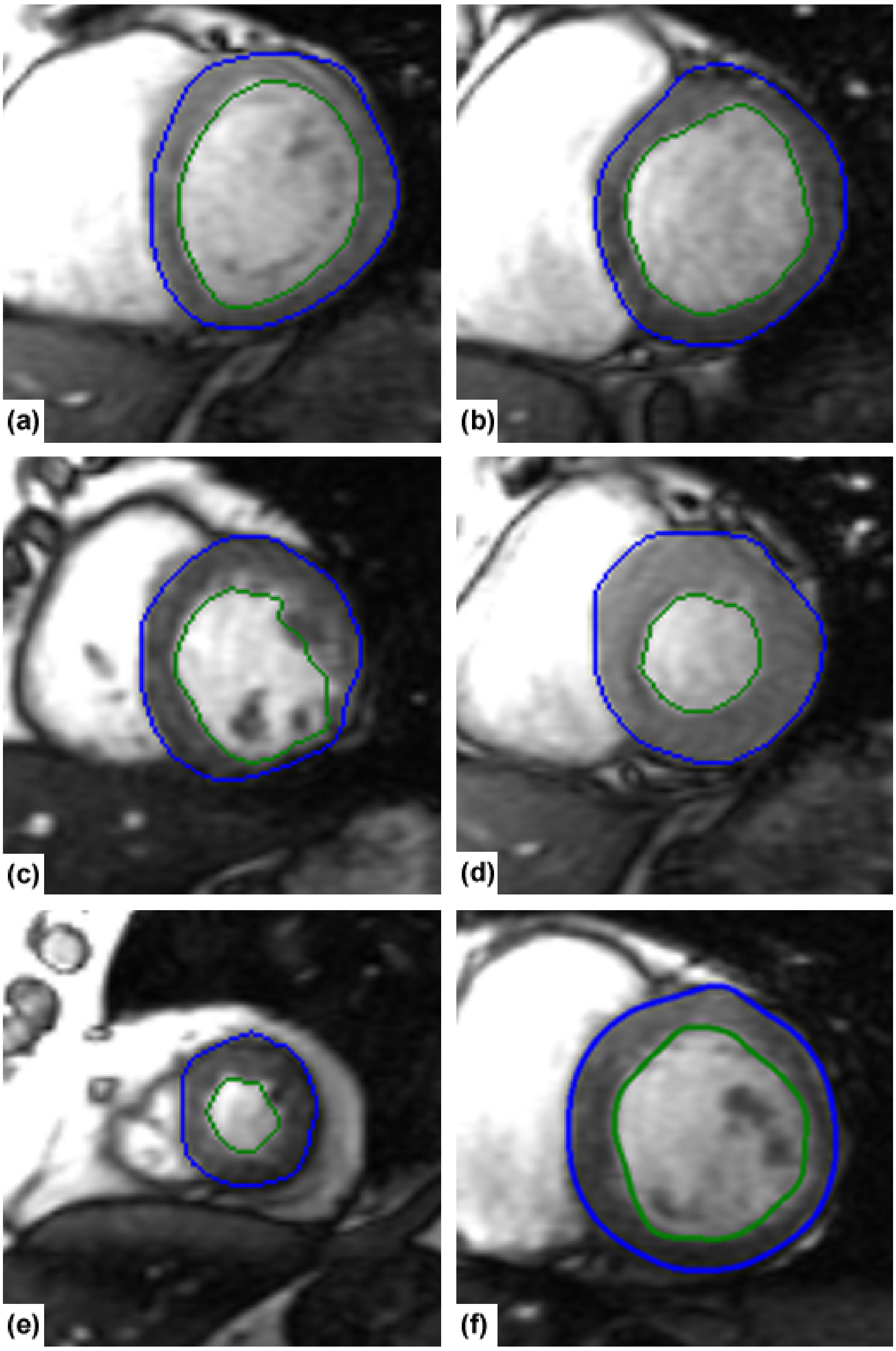


Figure 5.14: Results of applying the cognitive plan to different slices from various spatial locations and patients' datasets.

reader is refered to Appendix B for more visual results from the left ventricle segmentation problem.

5.7.2 Comparing the Results with other Algorithms

Three popular snake based algorithms were applied to the same medical dataset in order to compare their results with that of the agent's framework. The assessed algorithms are the greedy snake [125], gradient vector flow (GVF) [128] and a level set based snake [18]. None of these algorithms has an automatic initialisation process as they all depend on the user or a higher mechanism to place them near the target object. Therefore, the initial contour was manually drawn around the endocardium using the algorithms provided software. Figure 5.15 shows the initial placed contours along with the results of applying these algorithms to a slice from the medical dataset. Note that several combinations of parameters were tried on but no combination gave any noticeable improved result.

The greedy snake external energy force is based on the image gradient. Since the endocardium gradient is lower than the epicardium is some parts of the boundary, the snake is falsely attracted to parts of the epi boundary, as seen in Figure 5.15(b). The GVF snake has a long attraction range to minimise the importance of the initial contour placement. Nevertheless, because of the long attraction force it is highly unstable and performs poorly in noisy images. In essence the

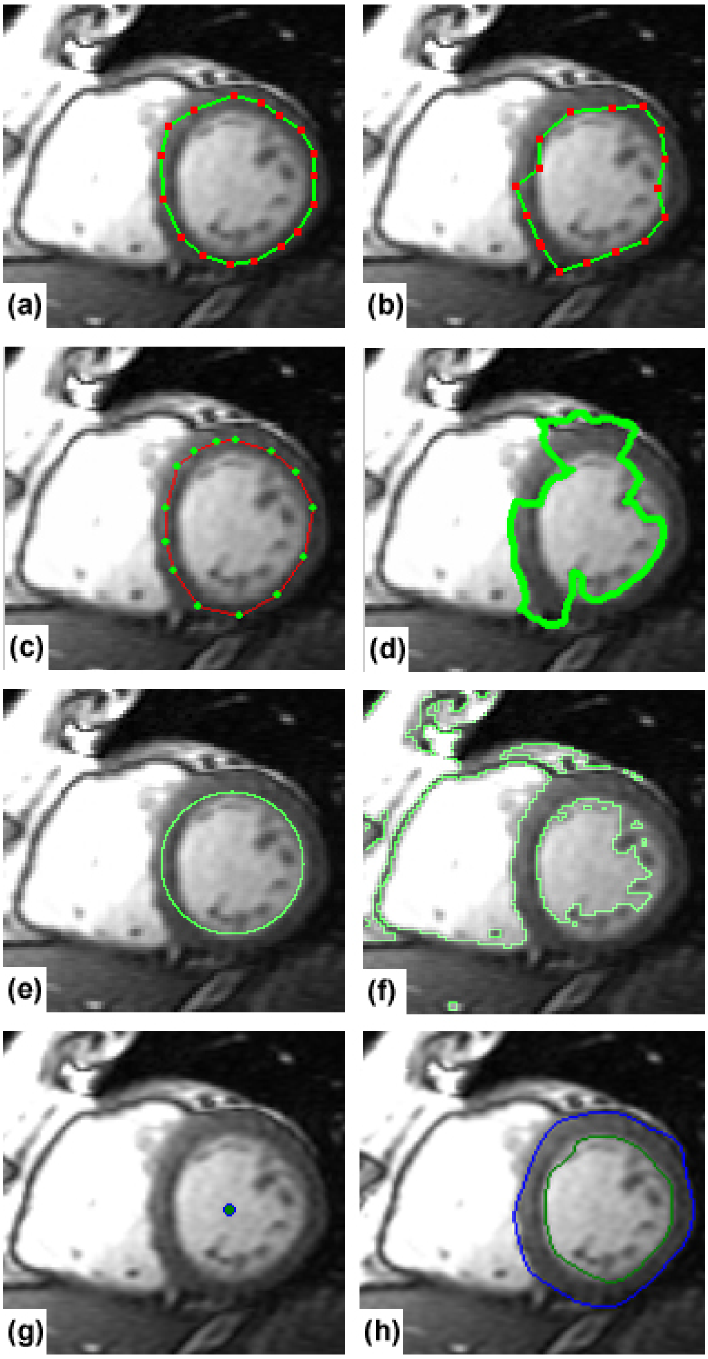


Figure 5.15: Segmentation results from other algorithms. (a) Greedy snake initial contour, (b) Greedy result, (c) GVF initial contour, (d) GVF result, (e) Level set initial contour, (f) Level set result, (g) Proposed framework initial region MA, (h) Proposed framework result.

GVF snake is attracted to high image gradient, and in the same way as the greedy snake, parts of it falsely rest on the epicardium, see Figure 5.15(d). The level set snake has the ability of adapting its topology based on the existing image features. This ability causes the level set snake to split and merge without any interaction. The ability is desirable in some scenarios when the number of objects in the image is not known in advance. Nevertheless, level sets are slow, unstable and sensitive to noise. In addition, the final result can contain several fragmented boundaries under noisy environments. This is a highly undesirable feature under controlled segmentation environments, see Figure 5.15(f).

In general, all of the assessed algorithms are not aware of the segmentation task in contrast to the proposed framework. Awareness and knowledge about the task in hand is crucial to successful segmentation. It allows the segmentation algorithms to effectively handle complex, missing and noisy image features situations. In addition, it allows the MicroAgents to decide when to trust an image feature and when to ignore it. For example, by instructing the MicroAgents to find the endocardium boundary they employ the region force to get attracted to the endo region and away from the epicardium.

5.7.3 Discussion

Observe that, through utilising the proposed intelligent framework the complexity of the image segmentation problem is reduced. By breaking up the left ventricle segmentation task into small steps, specialised agents can focus on solving sub-problems separately. This leads to good results being achieved without the reliance on previously computed knowledge or on a high level of parameterization in the system. The only employed parameter in the plan is the value of threshold T in Equation 5.7. Additionally, only the following a priori knowledge about the left ventricle are utilised in the plan.

- Endocardium is a closed structure that might contain smaller darker structures, i.e. muscles.
- Epicardium is a closed structure darker than the endocardium.
- Epicardium and the right ventricle share a boundary to the left of the epicardium.

5.8 Guidelines for Populating the Framework for other Medical Applications

In general, the combined region and boundary phases employed to segment the left ventricle are suitable for solving several medical segmentation problems. In either phase, higher agents prepare the virtual world for MicroAgents by pre-processing the image and by storing logical rules and knowledge in the blackboard. Upon initialising in the environment, the MicroAgents reactive behaviour enables them to locate the required image features.

While customising the framework for a new medical problem, the attraction/repulsive forces used might be changed in order to better suit the available image features. The homogeneousness criteria used for both the endo/epicardium regions is based on the average image intensity. However, this criterion is not suitable for all medical structures regions. Depending on the nature of the target region the use of other uniformity criterion such as texture or colour might be needed.

As for the use of MicroAgents in locating boundaries, the relation of the target object with surrounding structures can be used as a guide on the type of forces to employee. Edge based forces such as the one used with the endocardium are suitable to use for objects with noticeable intensity difference with surrounding structures. In general, edge maps obtained from these structures are suitable to use as a base for image energies. On the other hand, region forces such as the one deployed for the epicardium are better suited to boundaries with little difference in intensity levels from surrounding structures.

The shape of the target object is an important factor to consider while customising the framework. For structures with high variance in shape such as the endo/epicardium it is not recommended to use any shape constraints. However, for structures with relatively constant shape, the use of constraints enhances the segmentation in noisy and missing image feature environments. In the proposed framework shape constraints can be enforced using two approaches. The first is adding shape restrictions, such size and circularity, in forms of logical rules into the blackboard. The second approach involves the use of the snake segments concept introduced in Chapter 4. In this approach a shape template of the target object is divided into connected snake segments. Each segment contains a group of MicroAgents sharing similar shape attributes. The segmentation using this approach starts with the predefined snake segments. Later on, MicroAgents in the snake segments are attracted to the available image features while maintaining the imposed shape constraints. The snake segments approach is more complex than the first approach. Nevertheless, it provides better flexibility in imposing shape constraints. Given that, different constraints can be imposed per snake segment, compared to imposing constraints on the whole MicroAgents society.

In the two phase segmentation plan deployed to locate the endo/epicardium, obtaining the initial starting locations for contour MicroAgents is achieved by getting the MEC of the segmented region then initialising the contour MicroAgents around the circle boundary. Although initialising the contour MicroAgents in a circular manner does not restrict the final shape to be circular. It is not always suitable to start the contour society with a circular shape.

In general, for convex and slightly concave shapes it is recommended to initialise the MA contour society from a convex shape. The convex shape is obtained from applying a convex hull algorithm as in [3] to the result of the region MicroAgents. As for highly concave structures, it is recommended to extract a concave shape from the segmented region, afterwards utilise it as a starting figure for the contour MicroAgents. Extracting a concave hull from a set of points is more difficult to accomplish than obtaining the convex hull. While there is a unique solution for the convex hull of a point set, the same does not apply for the concave hull. Since several concave hulls exist for the same set, obtaining what can be referred to as the best hull depends on the final application. Several algorithms are proposed in literature in order to locate the concave hull, for example refer to [31] and [87].

The two phase plan deployed to segment the endocardium is applied to the right ventricle with minor differences. First, since there exist a high variation in the intensity levels of the RV region the selected value of T is 80. Although the value of T appears high, keep in mind that the case study medical dataset has as 16-bit grey scale intensity levels. Second, the concave hull algorithm described in [87] is applied to the segmented region instead of the MEC algorithm. In Figure 5.16, the result of the contour MicroAgents after finishing the segmentation is displayed. Notice since the bottom right area of the RV is darker than the rest of RV, the region force attracts the MicroAgents at that location into the region and away from the real boundary. Omitting the region force would result in some contour MicroAgents being attracted to surrounding structures since the image force is stronger there. For structures with similar complex regions other similarity criteria other than the one applied to endo/epicardium is advised to be used.

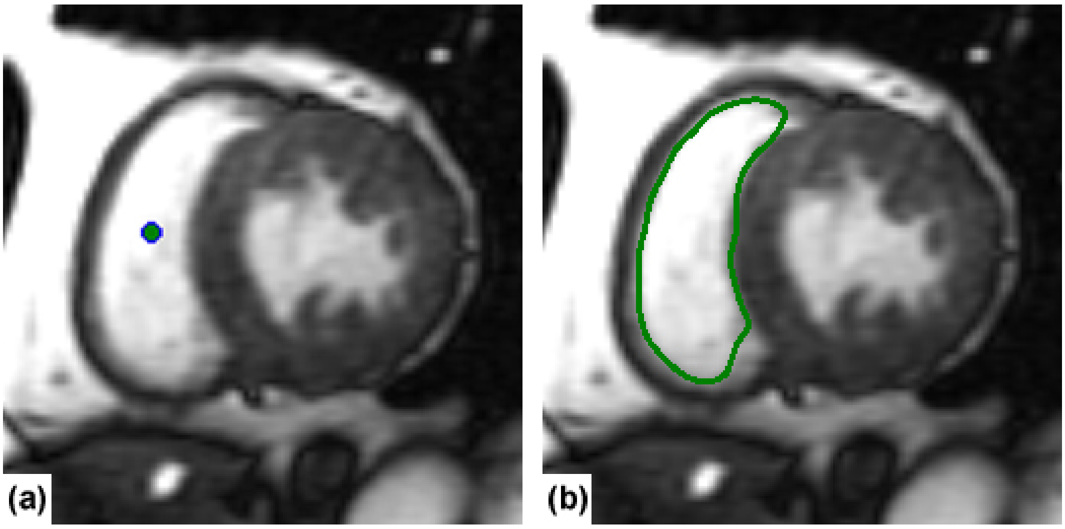


Figure 5.16: Result from segmenting the right ventricle using the two phase approach. (a) Initial region MA with $T = 80$, (b) the contour MicroAgents segmentation result.

In summary, the following are the recommended steps needed in order to populate the framework for other medical segmentation problems.

1. Analysis the medical problem carefully. A great deal of the solution depends on understanding the problem.

2. Choose the attraction/repulsive forces based on strong image features and

relation to surrounding structures.

3. Use available a priori knowledge about the target object in forms of logical rules.

4. Start with simple rules and behaviours. Simple reactive behaviour can produce good results.

5. For structures with relatively fixed shapes, use snake segments or apply shape constraints.

6. When combining region and contour MicroAgents, obtaining a good starting locations for contour MicroAgents depends on the estimated target shape.

7. Image segmentation is an open-ended problem. Experimenting with several combinations of rules and behaviours might be needed before obtaining the best results.

5.9 Summary

In this chapter, the proposed agent framework is customised in order to segment the LV from a time series MRI dataset. Segmenting real images provides a greater challenge than segmenting synthetic images. Artificial images even if corrupted with noise are still of greater quality compared to medical images. The cognitive plan for locating the LV is based on dividing the segmentation into several steps.

This division allows specialised agents to focus on solving sub-problems separately. Three types of MicroAgents are utilised, region MicroAgents, they serve as a preparing step since no prior statistical knowledge is used. Contour MicroAgents to locate the boundaries of the endo/epicardium. In addition to scout MicroAgents to move between structures and slices in the dataset. Although no pre-computed knowledge is utilised in the framework, good results are obtained from segmenting several patients' datasets.

In addition to segmenting the LV, this chapter demonstrates that the use of AI techniques such as the rational agent approach is needed in order to manage the complexity of the image segmentation problem. The author believes that dividing the segmentation problem into smaller sub-problem each dealt with in a rational way is the key to successful segmentation.

Chapter 6

Evaluation of the Framework

6.1 Introduction

Hundreds of image segmentation algorithms have been proposed in the literature. Since almost none of the proposed algorithms are generally applicable to all types of images and applications, the evaluation of the segmentation algorithms is crucial and accordingly an important subject in the study of segmentation. Nonetheless, the research area of segmentation evaluation has attracted far less attention than that of the development and research of new segmentation algorithms. Moreover, many of the newly developed algorithms are most often subjectively compared with some particular algorithms and a few hand-selected images [133].

Pal and Pal [91] are strong supporters of the reliance on qualitative human judgement of results for evaluation, as they state that “a human being is the best judge to evaluate the output of any segmentation algorithm”. While the author does not question the general superiority of the human visual system over any currently available machine vision system, the author disagrees with the Pal and Pal statement. The human visual perception cannot in general be the best judge to evaluate the output of segmentation algorithms as it is subjective, qualitative, time consuming for large datasets and might require a large panel of human observers for better validity.

Several methods are proposed in the literature for image segmentation evaluation. A popular classification for these methods is provided by Zhang [133]. Zhang classified evaluation methods into three main groups: analytical, empirical goodness and empirical discrepancy. The analytical methods examine the segmentation algorithm itself by considering the principles, requirements, complexity, etc. of the algorithms. In empirical goodness methods, desirable properties of the segmented images are measured using a goodness parameter such as uniform grey-level within regions, or contrast of grey-level between regions. Whereas, the empirical discrepancy methods compare the segmentation results with ideally segmented images usually referred to as reference images, gold standards or ground truth. Although the previous keywords are often used interchangeably throughout the literature, real life boundaries outlined by expert users cannot be considered as definitive gold standards or ground truth since they suffer from observer bias and inter and intra observer variability. Therefore, in the remainder of this chapter only the term "reference images" is used to refer to manually segmented test data.

The aim of this chapter is to evaluate the proposed MultiAgent framework with regards to the medical case study. Assessing the quality of the segmentation output is achieved by applying several empirical discrepancy measurements on the medical dataset introduced in Chapter 5. The "reference images" employed in this evaluation were created by manually delineating the endocardium boundary using Amira's [113] segmentation toolkit. Note that, the manual segmented images cannot be considered to be an "error free" representation of the real boundary. Nonetheless, they provide a suitable good quality reference which can be used in the evaluation of the MicroAgent's automatically segmented images.

6.2 Evaluation Methods for Image Segmentation

In general, image segmentation is an open ended problem, where there exists no single unique solution for locating the boundary of an object. With the availability of hundreds of segmentation algorithms, it is important to compare and measure the performance of these algorithms. However, choosing the most appropriate measures for evaluating segmentation algorithms should not be considered as an effortless task. Although, for many applications the accuracy of the segmentation output is the most sought after measure, other applications might require a fast output. For example, in applications which requires real time segmentation, the speed of finding a solution is often more important than the accuracy of output. Therefore, it is impractical to evaluate all segmentation algorithms using the same measure(s). Generally speaking, choosing the best measurements to evaluate a particular segmentation algorithm should depend heavily on the final application.

As mentioned previously, Zhang [133, 134] categorised evaluation methods into analytical, empirical goodness and empirical discrepancy. Figure 6.17 demonstrates a simple overview of Zhang classification groups with relation to the image segmentation process. The first group of evaluation methods consists of analytical methods which examine the segmentation algorithm itself. Several aspects of the segmentation algorithm can be considered by the analytical methods such as, the computational requirements, principles and the amount of prior knowledge used. Perhaps the most favourable feature of analytical methods is that it avoids the actual implementation of the algorithms and does not require setting up experiments as the empirical methods do. However, analytical studies cannot obtain all the attributes of segmentation algorithms due to the lack of a unified segmentation theory. In addition, unless the compared algorithms are similar in nature, it is problematic to obtain useful results from analytical evaluation methods. Also, as indicated by Zhang [134], both analysis and practice concluded that the analytical methods alone can only provide a small amount of additional information to that of empirical methods. The author believes that it is extremely difficult to analytically evaluate the proposed MultiAgent framework without being subjective. Therefore, no analytical evaluation of the MultiAgent framework will be performed in this chapter.

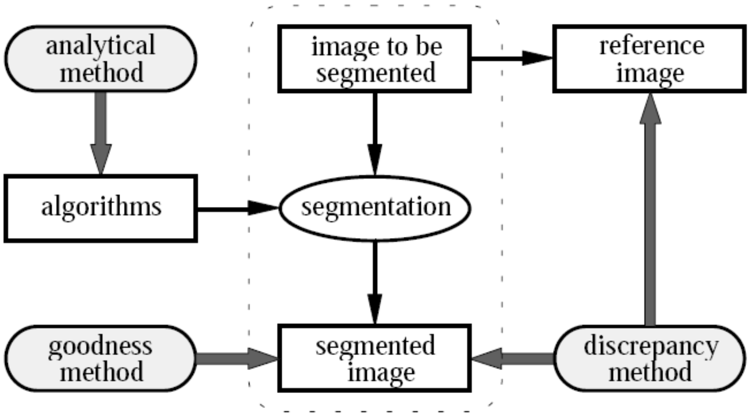


Figure 6.17: Zhang classification of the evaluation methods into three main groups: analytical, goodness and discrepancy methods, adapted from [134].

The empirical goodness methods measure desirable properties of the segmented images using a goodness parameter. These goodness (quality) measures are established according to human intuition in order to specify an ideal or a good segmentation situation. Examples of goodness methods include: uniform grey-level within regions, contrast of grey-level between regions, region shape and size. The previous measures are helpful when a strong pattern exists in the segmented images. For instance, the grey-level uniformity measure is only useful if the segmented region is reasonably homogeneous. Likewise the size measure is only useful if the estimated target size of the segmented object is known in advance. Since the objects regions in the cardiac case study presented in Chapter 5 are not uniform and are highly variable both in size and shape using the previous goodness measures will not yield meaningful results.

The third group of evaluation methods, according to Zhang's classification, are the empirical discrepancy methods. Discrepancy methods measure the difference between the segmented images and ideally segmented images usually referred to as reference images. Various researchers [54, 82, 133] agree that the discrepancy methods are in general more effective than the goodness methods in assessing the performance of segmentation algorithms. However, a major difficulty involving the use of discrepancy methods is obtaining the reference images. Naturally, obtaining the reference images from synthetic images is a simple task achieved by an image generation procedure [135]. Nonetheless, the same does not apply to real images, acquiring reference images from real images normally involves the use of manual segmentation. Therefore, reference images extracted from real images are debatable as they inherit all the limitations of manual segmentation. A practical but rather expensive method to enhance the credibility of these reference images is to merge the segmentation of several expert human observers into one dataset. As the number of human experts participating in creating the unified reference dataset increase, the quality and creditability of the dataset increases as well.

Another approach to classify evaluation methods is grouping them into two main groups based on whether or not they require external input. The first group, which can be referred to as internal evaluation, does not require extra input and only depends on the same inputs as the segmentation process. The second group, which can be referred to as external methods, require the use of additional input resources (i.e. reference images) in order to perform the evaluation. The internal methods correspond to the analytical and empirical goodness methods, while the external methods correspond to the empirical discrepancy methods. The next section utilises several external (discrepancy) methods in order to assess the performance of the MultiAgent framework in regards to the medical case study.

6.3 Evaluating the Framework

6.3.1 Discrepancy Methods

In real life segmentation applications, a small error percentage in the segmentation process can usually be tolerated. In the case of automatic segmentation algorithms applied to complex and noisy datasets the error is unavoidable [28]. Several discrepancy methods have been proposed to measure the error (difference) between the actual segmented image and the reference image. A perfect segmentation (if such thing exists) will have a discrepancy measure value of zero.

A high value corresponds to a big error in the segmented image which indicates a poor performance from the segmentation algorithm. There exists no global fixed percentages which clearly defines small, medium or big errors, such classifications is up to the final application to define. Table 6.3 summarizes some of the commonly used discrepancy methods. As with other evaluation methods the choice of a suitable discrepancy method mainly depends on the segmentation application.

#	***Measurement***	***Description***
1	Mis-segmented pixels [131]	The percentage of pixels from a certain class mis-classified as a different class.
2	Average distance [16]	Sum of distances from each point in curve 'A' to nearest corresponding point in curve 'B' divided by the number of points.
3	Hausdorff distance [55]	Furthest distance from any point in curve 'A' to the nearest corresponding point in curve 'B'.
4	Area accuracy [133]	The relative area error between the segmented object area and the actual object area.
5	Number of objects [130]	The difference between the true number of objects in the reference image and the actual object number in the segmented image.
6	Merging errors [71]	Under and over merging errors, the amount by which the regions in the segmented image overlap the regions in the reference image.
7	Figure of certainty [112]	Grey-level difference between the segmented image and the reference image.

Table 6.3: Examples of discrepancy methods commonly used to asses the segmentation algorithms performance.

Measures 1-4 in Table 6.3 were chosen to evaluate the MultiAgent framework. This choice was made since these measures are more suitable to evaluate boundary based segmentation and are applicable to the cardiac case study. Other measurements such as the number of objects [130], merging errors [71] or figure of certainty [112] were not chosen since they are not relevant to neither the framework nor the medical application. The number of mis-segmented pixels measure computes the percentage of pixels from a certain class that are mis-classified as a different class. Relating to the cardiac case study, it is assumed that each slice contains two classes, the endocardium (class ’A’) and the background (class ’B’). Any pixel which belongs to class ’A’ in the reference images but does not exist in the actual segmented image is counted as mis-segmented. To compute the mis-segmented error for the endocardium, Algorithm 6.2 is employed. Note that, the error computed is with relation to the endocardium (class ’A’) and not the background (class ’B’). Since, the endocardium accounts for a small part of the whole image ($\cong 2\%$), computing the error for class ’B’ will always yield a negligible value.

Hausdorff distance [55] and average distance [16] are both location based measurements. They evaluate the segmentation algorithm based on the distance between the mis-segmented pixel and the nearest pixel that belongs to the same boundary. The previous measurement, i.e. the number of mis-segmented pixels, does not take into account the spatial information of mis-segmented pixels. Therefore, it is possible that objects segmented considerably different have the

```
Input: A1, A2: arrays of pixels which contains all the points in the
       reference object and the actual segmented object respectively.
Output: The percentage of mis-segmented pixels.
1. $iNumberMisSegment$ = 0
2. foreach $Point\ p\ in\ A1$ do
    if $!(A2.Contains(p))$ then
        $iNumberMisSegment = iNumberMisSegment + 1$
    end
end
3. $iTotalNumPixels$ = A1.count()
4. $Error_{MisSeg} = (iNumberMisSegment/iTotalNumPixels) * 100\%$
```

Algorithm 6.2: Mis-segmented pixels algorithm used to compute the error between two set of points representing the reference image and segmented image.

exact discrepancy measure value if the measures only counts the number of mis-segmented pixels. In general, distance based measures provides a more accurate representation of the actual segmentation error when compared to count based measurements. Also, distance measures are well suited for boundary based segmentations as the curves themselves are directly compared without the need to derive any parameters.

The average distance measure is computed by averaging the distances between each pixel on the segmented curve and the nearest pixel on the reference curve. Representing the reference and segmented curves as a set of points, $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ..., b_n\}$, where each a_i and b_i is an ordered pair of the x and y coordinates of a point on the curve, the distance to the closest point (DCP) for a_i to the curve B is defined as [16].

$$d(a_i, B) = \underset{j}{min}||b_j - a_i|| \tag{6.11}$$

The Hausdorff distance between the two curves is defined as the maximum of the DCP's between the two curves [55].

$$h(A, B) = max(\underset{i}{max}\{d(a_i, B)\}, \underset{j}{max}\{d(b_j, A)\}) \tag{6.12}$$

The average distance is defined as.

$$AVG_{distance} = (\sum_{i=1}^{n} d(a_i, B))/n \tag{6.13}$$

The Hausdorff distance measurement only computes the maximum error in the boundary, it does not provide error information on the segmentation as a whole. Therefore, it is possible to obtain the same Hausdorff measurement value for two distinct segmentations. This concept is illustrated in Figure 6.18, where two considerably different segmentations share a similar Hausdorff distance of around 35 pixels. Although, it is sometimes beneficial to obtain the maximum error, Hausdorff distance combined with the average distance should provide a better representation of the error.

The last applied discrepancy method on the segmented images is the area measure. According to this measure, a segmented image has the highest quality

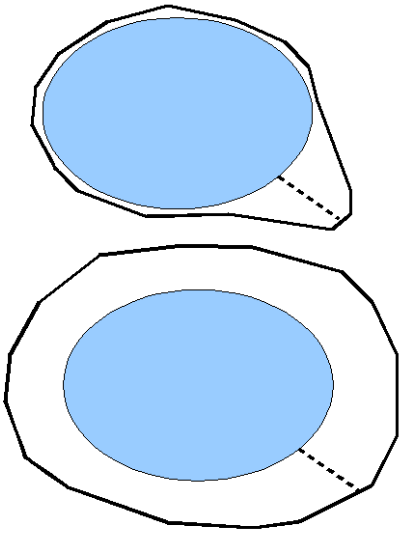


Figure 6.18: Two examples illustrating the Hausdorff distance between the two curves. The Hausdorff distance in both examples is about 35 pixels, whereas the average distance in the first example is about 10 pixels and in the second example is about 30 pixels. adapted from [16].

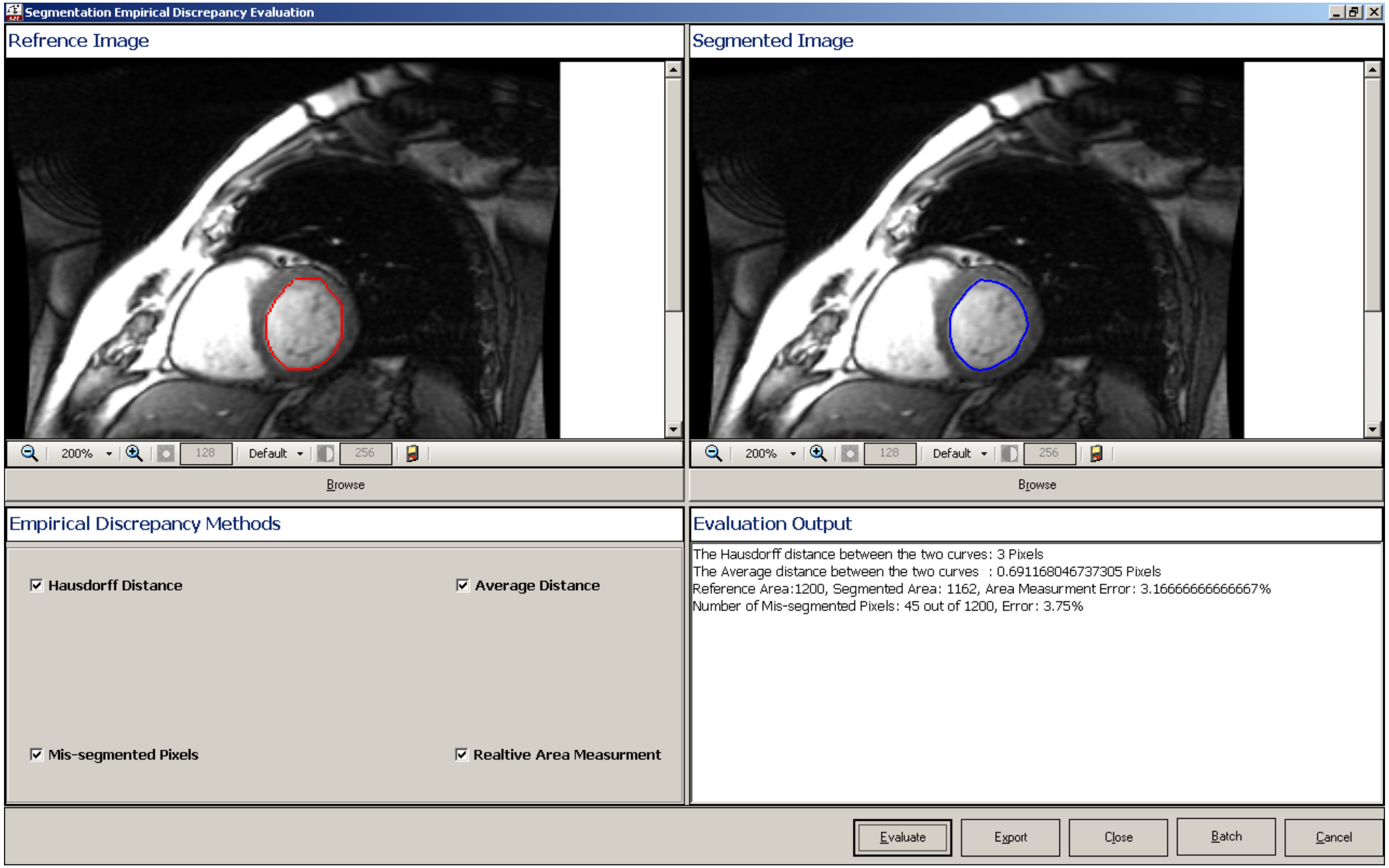
if the segmented area precisely matches the area in the reference image. One of the drawbacks of the area measure that it is possible for the reference object and segmented object to have the same area measure even if both of them are quite different. For this reason, it is advisable to use the area measure along with other discrepancy measures. Let R_a denote the area obtained from the reference image and S_a denote the area measured from the segmented image, the relative area error is defined as.

$$Error_{area} = \frac{|R_a - S_a|}{R_a} * 100\% \tag{6.14}$$

6.3.2 Reference Images

The reference images utilized in the evaluation were created by manually delineating the endocardium boundary using Amira's segmentation toolkit. The hand segmentation was performed on 100 slices representing a patient dataset, with the output of each slice stored in a separate file. The slices have a low resolution with each slice having an image resolution of 256*256 pixels and a pixel spacing of (1.406234,1.406234), in other words, every pixel maps to 1.406234 mm (x) and 1.406234 mm (y). Delineating the boundary with a mouse is tedious and time consuming operation, particularly when segmenting low resolution images. Although, the reference images created for the evaluation are of relatively "good quality" they cannot be considered to be an "error free" representation of the real boundary. Appendix B contains a complete listing of the reference images along with the segmented images.

In order to perform the experiments an extension utility for the MultiAgent software application had to be developed. Although, the employed discrepancy methods are established measures, there exist no publicly available suitable tool for performing image segmentation evaluation. The evaluation utility basic functionality, as shown in Figure 6.19, is to allow the user to input two files, one representing the reference image and the other representing the segmented curve. Afterwards, the user chooses the required measure(s) needed to perform the eval-



☑ Hausdorff Distance
☑ Average Distance
☑ Mis-segmented Pixels
☑ Realtive Area Measurment

```
The Hausdorff distance between the two curves: 3 Pixels
The Average distance between the two curves  : 0.691168046737305 Pixels
Reference Area:1200, Segmented Area: 1162, Area Measurment Error: 3.16666666666667%
Number of Mis-segmented Pixels: 45 out of 1200, Error: 3.75%
```

Figure 6.19: Screenshot of the discrepancy evaluation tool. The utility evaluates the segmented images using the several empirical discrepancy methods.

uation. The evaluation results are then shown in the utility window with the possibility to export them to a XML file. Other uses include displaying both the reference and segmented images side by side to facilitate the human visual inspection and comparison of the images. Also, the utility contains a useful batch command which performs the evaluation on multiple evaluation datasets contained within a directory.

6.3.3 Results

The mean processing time for the LV segmentation application running under Windows XP/.NET Framework 2.0 on an Intel 1.46 GHZ dual core CPU laptop (with 1 GB RAM) per dataset is around 4.85 min. Table 6.4 contains statistical results from the image segmentation evaluation. The results were obtained by computing the discrepancy between 100 segmented images and their equivalent reference images. The mean error for the average distance measure is 1.194 pixels, which translates to 1.6713 mm in physical coordinates. As for the Hausdorff distance, the mean error is 3.343 pixels, which in turn translates to 4.6786 mm. Both the area and mis-segmented errors are percentage values relative to the reference images. The average difference between the segmented image areas and the reference image areas is 10.05%. Also, the mean error for the number of mis-segmented pixels is 7.154%. Overall, the results indicate that the segmented images have a good resemblance to the reference images.

	Average Distance	***Hausdorff***	***Area***	***Mis-segmented***
$Mean$	1.194	3.343	10.05	7.154
$Median$	1.175	3.000	8.651	6.508
σ	0.404	1.021	6.742	3.142
σ^2	0.163	1.043	45.46	9.875
Min	0.479	1.414	0.401	2.321
Max	2.375	6.708	29.05	15.00

Table 6.4: Statistical results from the evaluation of the MultiAgent framework using four discrepancy methods. The error measurement unit is pixels for both the average and Hausdorff distance methods. As for the area and mis-segmented pixels measurements, the error is a percentage value relative to the reference image.

Figure 6.20 demonstrates area charts for the results of the discrepancy methods. The y-axis for each chart represents the error, and the horizontal line represents the statistical mean, while the x-axis represents the slice number. The slice numbering followed the same numbering as in the DICOM tags, which is, from the base to the apex of the heart for a single cardiac cycle. The four curves in Figure 6.20 share some similarities around the peaks and valleys. The existence of these similarities indicates that the different evaluation measurements reported similar results from the slices. That is to say, segmented images similar to that of the reference images had in general low error values from all of the evaluation measurements, and vice versa

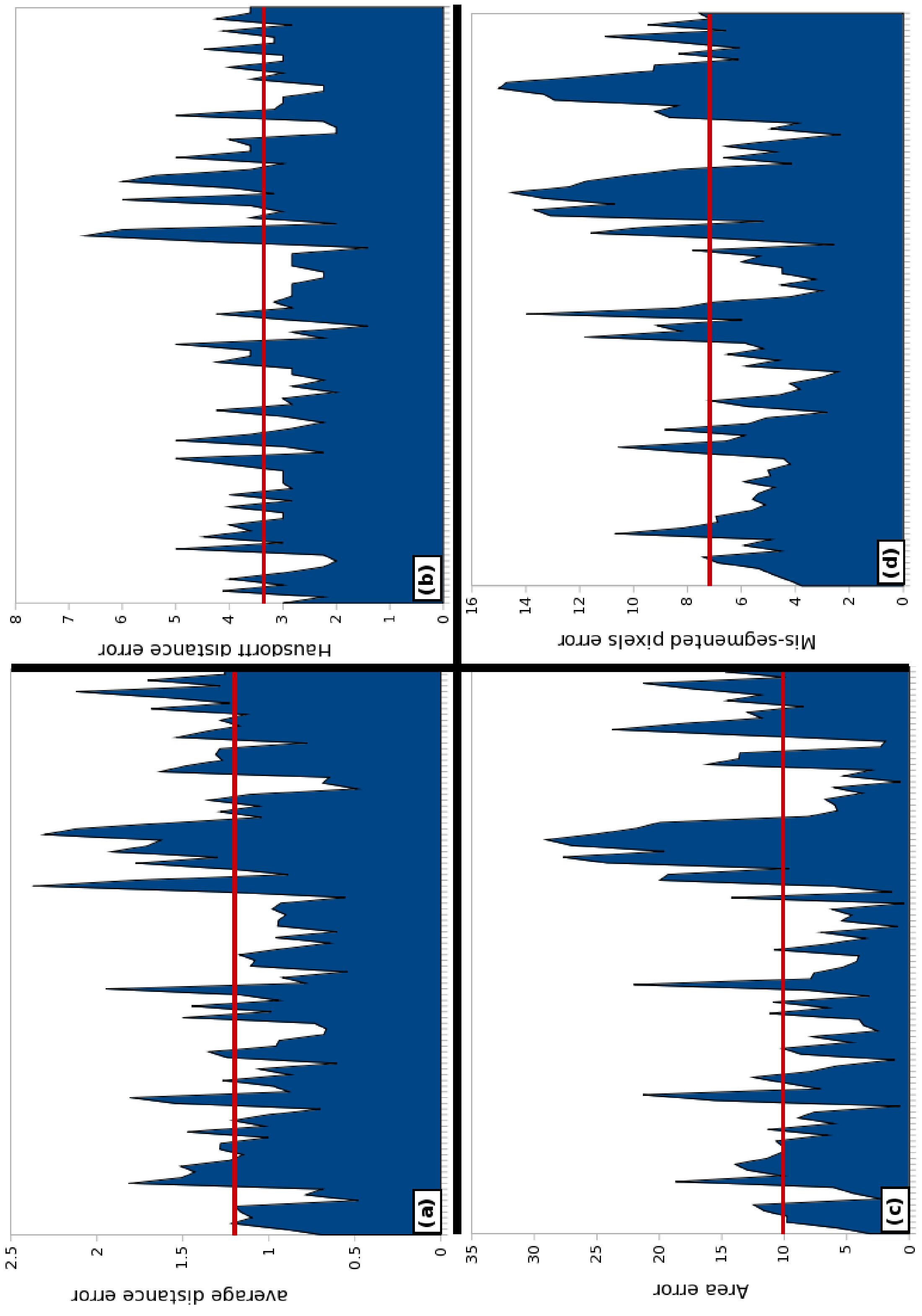


Figure 6.20: Area chart representation of the evaluation results for the MultiAgent framework. For the four charts the x-axis represent the slice number. 100 slices were evaluated.

6.3.4 Discussion

Examining the area measurement evaluation results more closely reveals a general under-segmentation error. The total area sum for the reference images is 75915, and for the segmented images it is 69640. The line chart in Figure 6.21 provides a clear indication of the under-segmentation error. The minimums and maximums in the line chart relate to the expanding and contracting of the heart during the cardiac cycle. It is worth noting that for most slices the reference object area is larger than that of the segmented object. The following are the major factors contributing to the under-segmentation error.

- **The logical rules influencing the MicroAgents behaviour.** The region forces introduced in Chapter 5 attract the contour MicroAgents towards the region of the target object. Since boundaries are usually thicker than one pixel width this will result in locating the inner boundary of the object.

- **The existence of muscles attached to the boundary.** In some of the slices, there exist papillary muscles attached to the boundary of the endocardium. These muscles appear as small dark structures with intensity levels comparable to that of the myocardium. Since differentiating them from the myocardium is a difficult task, the MicroAgents will be attracted to these false boundaries which results in smaller areas.

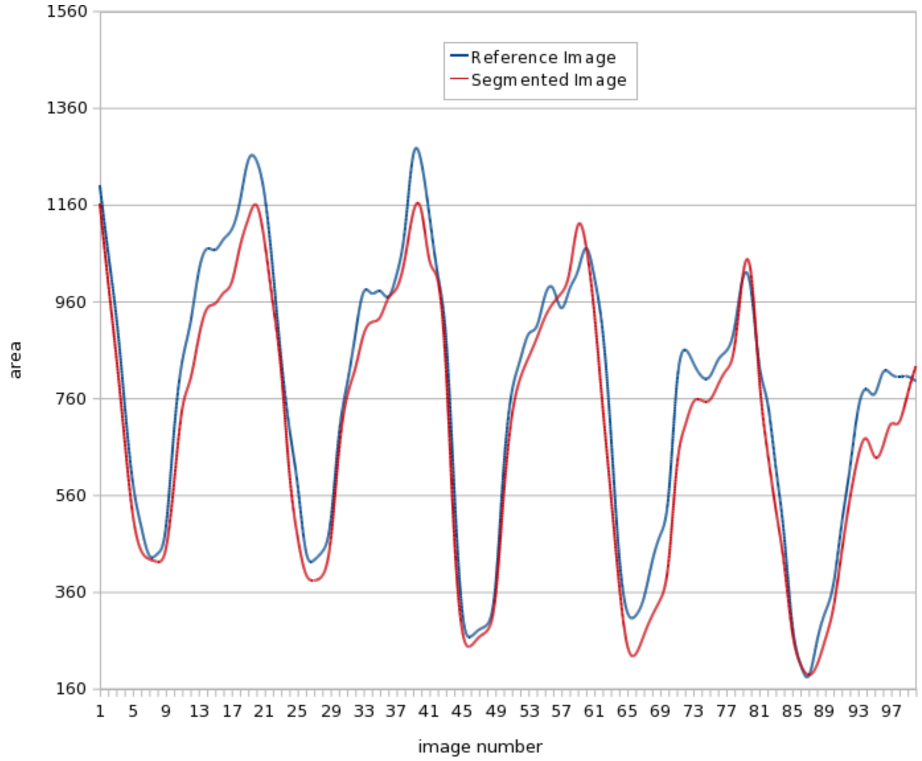


Figure 6.21: Line chart comparing the area of the reference and segmented images. A noticeable trend in this chart is that the segmented images have smaller areas due to an under-segmentation error.

- **The human perception of the boundary.** A boundary of an object inside a digital image can be either an interior boundary or an exterior boundary [19]. Interior boundaries consists of pixels belonging to the object itself. While exterior boundaries contains pixels which belongs to the background. The area inside the exterior boundary is always larger than the area inside the interior boundary. The human expert outlining the reference images, as shown in Appendix B, had a general tendency to segment the exterior boundary of the endocardium. Other human observers might have segmented the interior boundary of the endocardium. Given that the perception of the boundary is subjective and differs from an observer to another. It is recommended that several human observers prepare the reference images in order to better approximate the real boundary.

There are some possible methods to improve the left ventricle segmentation in order to minimise the under-segmentation error. One of the methods is refining the logical rules in order to encourage the MicroAgents to find the outer boundary. Also, applying some sort of inflating post-processing on the final result where the boundary is expanded by some factor. Finally, by improving the muscle detection and labelling technique performed by the knowledge agent to handle muscles attached to the boundary more efficiently.

Some of the problems encountered while evaluating the MultiAgent framework is the lack of unified “freely available” datasets and tools to assist in the evaluation. This problem is not only associated with a particular segmentation algorithm such as the MultiAgent framework. In general, it affects all the empirical evaluation of segmentation algorithms in the medical domain. Currently, any discrepancy empirical evaluation of medical image segmentation algorithms requires the creation of a reference image dataset. Therefore, the availability of free datasets and tools will help ease the evaluation process by eliminating the time consuming and tedious task of preparing the reference dataset. Also, it will greatly participate in standardizing the evaluating procedure and makes it easier to objectively compare different algorithms under the same environment. In essence, creating standardised benchmarking tools for the evaluation of image segmentation algorithms in the medical domain requires the following.

- Several freely available segmented datasets offering different levels of segmentation difficulty. These reference images should be created by multiple human observers in order to minimise the subjectivity.

- Well documented open source software application to perform the evaluation. The software application should include the commonly used evaluation algorithms in addition to a flexible design that permits adding new algorithms with ease. Also, the software should be accompanied with clear guidelines on how to perform the evaluation.

6.4 Comparing with Commercial Products

To the author's knowledge there exists no commercial software that is capable (out of the box) of automatically segmenting the LV from a 4D dataset. However, there are several commercial products, such as Amira [113], EIKONA3D [74], Visiopharm Segment [121], MEDx [56], 3D-DOCTOR [26] and Simpleware ScanIP [75], which can be utilised in the manual/semi-automatic segmentation of medical structures. For example, most of the previously mentioned products provide seeded region growing, thresholding, edge detection and interactive contour segmentation. Since a commercial specialised LV segmentation software does not currently exist, it is not possible to compare the agent framework with any commercial product. Moreover, comparing the output of the framework with the semi-automatic segmentation output of the available general purpose image analysis software is not practical.

In general, many of the commercially available medical imaging software are aimed towards general image segmentation problems. However, few specialised segmentation software for particular medical problems do exist. For instance, both BrainVoyager [57] and Xinapse [76] provide focused brain image analysis functionality. An extensive list of the available commercial medical image analysis software packages can be found at the Internet Analysis Tools Registry (IATR) website [40].

6.5 Summary

This chapter has evaluated the MultiAgent framework segmentation performance with regards to the medical case study. Four discrepancy evaluation methods were chosen for the evaluation. The results indicate that the segmented images closely resemble the reference images with an average distance error of 1.194 pixels (1.6713 mm). Also, the results implied a general under-segmentation error, mainly due to the logical rules themselves and the existence of muscles attached to the boundary. Possible improvements in order to decrease the under-segmentation error include the use of post-processing and/or refining the logical rules.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Digital image segmentation is a central problem in medical imaging, since accurate segmentation is essential for many image analysis applications and highly influences the quality of the final output. Hundreds of image segmentation algorithms have been proposed in the literature, ranging from low level algorithms such as thresholding to more complex model based solutions such as deformable models. Nevertheless, to the author's knowledge, no single algorithm can be considered suitable for all sorts of images and applications. Moreover, most of these algorithms are either derivatives of low-level algorithms or created in an ad-hoc manner in order to solve a particular segmentation problem. Active contours provide a promising model-based approach to medical image segmentation through the integration of bottom-up constraints (external forces) along with top-down a priori knowledge (internal forces). However, active contours do not solve the entire problem of image segmentation as they rely on other mechanisms, such as the user, to place them near the desired object. Several attempts were made by other researchers to improve and automate the original snake algorithm and enhance its ability to locate noisy and missing boundaries. Nevertheless, these attempts have not been successful in dealing with the vast variety of structures and image data. Since, most of these improvements are either application specific or they introduced new restrictive external and internal forces that require fine-tuning by users for optimal performance.

The aim of this research is to create a customisable automatic image segmentation framework based on MutliAgents and active contours. Several segmentation algorithms are initially created as application specific, then generalised in order to solve several segmentation problems. Nonetheless, this might not be a suitable approach to create customisable image segmentation frameworks. Customisability should not be regarded as an isolated feature that can be added at a later step, it should be an integral part and considered whilst designing the framework. A generic customisable image segmentation framework can be beneficial for the medical imaging research community. Since it permits more focused collaboration on the image segmentation problem instead of the divided efforts consumed by creating several ad-hoc algorithms. In addition, it should allow the researchers to focus on the image analysis application more than the image segmentation problem.

The research of this thesis proposes the Agent Society for Image Processing (ASIP), which is a generic framework for intelligent image segmentation motivated by the active contour model and MultiAgent systems. ASIP is presented in a hierarchical manner as a multilayer system consisting of several high-level agents (layers). The bottom layers contain a society of rational reactive MicroAgents that adapt their behaviour according to changes in the world combined with their knowledge about the environment. The use of the term micro has several purposes, it serves to illustrate that these agents have a simple architecture, they are part of a higher macro agent and to avoid confusing the MicroAgents with the other high-level agents in the framework. On top of these layers are the knowledge and shape agents responsible for creating the artificial environment and setting up the logical rules and restrictions for the MicroAgents. At the top layer is the cognitive agent, in charge of plan handling and user interaction. The framework as a whole is comparable to an enhanced active contour model (body) with a higher intelligent force (mind) initialising and controlling the active contour.

Dividing the segmentation process into smaller problems and having an expert agent focus on a single sub-problem at a time provides better control and flexibility over the system. The framework was divided into the five independent layers based on the different type of functionality each layer provides. Also, since a solution for an image segmentation problem might involve the use of technologies from different and diverse domains, the framework was designed to allow experts from various fields to contribute to the framework. For example, a mathematician can contribute to the geometrical constraints in the shape agent, a statistician to the knowledge agent, an AI expert to planning in the cognitive agent and so on. By having intelligent layers in the framework it possesses several other advantages over the traditional active contour. For instance, the active contour energy minimisation process relies entirely on its potential energy function E(x,y). However, it is not always possible to represent attraction/repulsive forces as functions. In addition, region-based forces such as texture or colour cannot be easily integrated into a potential function. On the other hand, by designing the Agent based Snake as a society of cooperative MicroAgents a much wider range of forces can be used. Furthermore, attraction/repulsive forces can be imposed at different levels in the society, the whole society (macro), parts of it (segments) or a single agent (micro).

The automatic segmentation of the Left Ventricle (LV) from a 4D MRI dataset was chosen to populate the ASIP framework. In general, real life medical applications provide a greater segmentation challenge when compared to synthetic images. Therefore, the research opted for a challenging medical instead of using artificial images. Several approaches were considered in order to customise the framework for the LV case study. One of the initial propositions was to create a database of statistical region knowledge for the epicardium (outer layer) and endocardium (inner layer). Afterwards, the prior knowledge is combined with gradient based image features and applied as attraction/repulsive forces for the MicroAgents. There are two main problems associated with this approach. Firstly, the difficulty associated with the creation of the knowledge database as it required a manual segmentation of a large number of slices. Secondly, the variability of intensity levels between patients' slices. While this method will work on slices with intensity levels similar to that of the training set, it is not guaranteed to work when applied to a dataset with a noticeable variance in intensity levels. Another proposed method is to divide the MicroAgents society into snake segments computed from a prior shape model. With each snake segment having its own properties and shape restrictions based on its spatial position in the image. Again, this method was also discarded because of the visible anatomical shape variability in the endocardium.

After concluding that depending on pre-computed statistical and shape knowledge for this medical case study is restrictive, the direction of the research shifted into utilising high level a priori knowledge about the LV. A level of knowledge that exists in all slices and will not change from one patient to another. Examples of a priori knowledge include, the endocardium is inside the epicardium, the endocardium has a lighter region than the epicardium, both are closed structures and the epicardium shares a boundary with the right ventricle.

Each structure in the LV cognitive plan goes through two main phases, a region phase followed by a boundary phase. In the region phase MicroAgents are deployed to segment the inside of the object. Afterwards, the segmented region is analysed by the shape agent to create the "initial contour MicroAgents", and also by the knowledge agent to create the "region attraction force". In the second phase, "contour MicroAgents" are attracted to the boundary and influenced by internal shape forces designed to keep the resulting curve smooth. Three types of MicroAgents were utilised. Firstly, region MicroAgents serve as a preparing step since no prior statistical knowledge is used. Secondly, contour MicroAgents locate the boundaries of the endo/epicardium. Finally, scout MicroAgents move between structures and slices in the dataset.

Although no pre-computed knowledge were utilised in the LV segmentation, good results were obtained from segmenting several patients' datasets. The results of the segmentation were compared with three snake based algorithms, the greedy snake, gradient vector flow and a level set based snake. The agent framework outperformed the three algorithms since it was aware of the segmentation task and benefited from the available high-level knowledge in contrast to the other snake algorithms. In addition, the segmented structures were evaluated against manually segmented "reference images" using several empirical discrepancy measurements. Although, the manual segmented images cannot be considered to be an "error free" representation of the real boundary. They provide a suitable good quality reference which can be used in the evaluation of the MicroAgent's automatically segmented images. The discrepancy methods results indicated that there is strong agreement between the segmented images and reference dataset with an average distance error of 1.194 pixels (1.6713 mm). Also, the results implied a general under-segmentation error, mainly due to the logical rules themselves and the existence of muscles attached to the boundary.

In conclusion, the overall aim as stated in the first chapter of creating a generic image segmentation framework by utilising concepts from the active contour model and MultiAgent systems was achieved. The customisability of the ASIP framework facilitated experimenting with several approaches while segmenting the left ventricle. Utilising the rational agent approach in image segmentation is recommended, as it provides an incremental path from very simple systems, such as the MicroAgents roaming in the virtual world, to complex autonomous intelligent systems, such as the ASIP framework. In addition, the complexity of the image segmentation can be reduced by dividing the segmentation problem into smaller sub-problems and applying specialised agents to focus on solving each sub-problem separately. Although low level segmentation algorithms are suitable for some applications, they lack awareness and knowledge about the segmentation task in hand. This awareness is important for robust segmentation, since it allows the segmentation algorithms to effectively handle complex, missing and noisy image features situations. Finally, the ASIP framework should not be perceived as an end product, it is should be regarded as a shell that facilitates the creation of customised image segmentation solutions. Also, further collaboration from researchers in several domains is needed to improve the framework capabilities. The current limitations and possible directions for future research are provided in Section 7.3, the future work section.

7.2 Major Contributions

In summary, the major contributions of the thesis are the following.

The Agent Society for Image Processing (ASIP), a customisable generic framework for image segmentation. The framework is novel because of the autonomous multilayered design that integrates MultiAgent systems with active contour models. The ASIP framework contains the following agents. Cognitive agent responsible for planning and user interaction, knowledge agent responsible for statistical based knowledge, shape agent responsible for geometric shape constraints and a society of collaborative MicroAgents reacting to changes in the environment which leads to the actual segmentation. The autonomous agent based design of the framework allows researchers in diverse domains to add, modify or maintain layers separately without requiring knowledge outside of their expertise.

Agent Based Snake (ABS), a society of collaborative MicroAgents inhabiting a virtual world. The society itself is comparable to an enhanced ACM, where the MicroAgents resemble semi-independent snaxels or control points. The MicroAgent possesses reactive behaviours which are stimulated by a semi-dynamic environment containing repulsive/attraction forces similar to the ACM external forces. Implementing a generic active contour model as a society of MicroAgents has a major benefit of allowing a wider range of energy forces to be used within active contours, such as colour or texture. In general, this would allow further improvement and utilisation of active contours.

Cognitive plan for the segmentation of the LV, a novel approach for segmenting the endo/epicardium from a 4D MRI was employed. The proposed plan for segmenting the LV utilises the ASIP framework and is based on a multi-stage multi-behaviour approach. The novelty in that cognitive plan allowed the segmentation of the left ventricle without the reliance on previously computed knowledge or on a high level of parameterisation in the system.

Software contributions, The ASIP framework along with a GUI for interacting with the medical application was developed using the C# language and the .NET framework. Since agent languages are still in the design stages with no software development tools publicly available, the agents in the framework were represented using objects and implemented using a quasi-parallel approach. The use of an object oriented methodology in the software realisation facilitates creating new types of agents, logical rules and MicroAgents behaviours. In addition, a software utility for evaluating segmentation algorithms using several empirical discrepancy measurements was created.

An application for MultiAgent systems, the thesis as a whole could be also assessed from an agent point of view as a case study for deploying MultiAgent systems in low-level machine vision. The use of agent technologies in image segmentation is still fairly uncommon; therefore, the thesis provides insights into the usefulness of the agent approach which could encourage other researchers to employ it in the future.

7.3 Limitations and Future Work

There is always room for improvement in scientific research. The author believes that the novel MultiAgent framework for image segmentation proposed in Chapter 4 opens the way for new and exciting prospects of future research in medical imaging. The following are some of the remaining outstanding issues and potential future work for the framework.

- The cognitive agent utilises a case-based strategy for planning knowledge in order to solve a specific segmentation problem. Another approach worth investigation is the use of a search-based planning procedure described in [27].

- The development of a GUI-based toolkit to ease the creation of segmentation plans for new medical structures. The toolkit can be designed in a similar manner to a workflow process generator. In which the user, aware of the existing agents (tools) along with their capabilities, selects the appropriate agent at each step of the segmentation process.

- The customisation of the framework in order to solve several challenging medical case studies. The medical case study presented in Chapter 5 helped shape the framework, hence, various challenging applications introducing different types of difficulties will develop the framework further. For example, segmenting colour images, the use of texture information or the segmentation of relatively rigid shapes in order to apply the concept of snake segments.

Appendices

Appendix A.

The Software Realisation of the Framework

This appendix complements Chapter 5 by presenting additional information concerning the software realisation of the framework. As mentioned earlier in Chapter 5, several agent-based programming languages were proposed in the literature to implement software agents, such as Agent0 [109] and Agent Programming Language (APL) [64]. The languages authors argue that a specialised agent language is needed to implement agents instead of using current object oriented languages. Essentially these languages focus on dealing with the algorithmic part of the agent, knowledge representation and communication/coordination between agents. Although, there are some perceived benefits for these languages, it is debatable if new agent-based languages are genuinely needed. Since, the most fundamental difference between MultiAgent systems and distributed objects is more at the design level than the programming level.

A more promising approach in agent programming languages proposes building a layer on top of existing high-level object oriented languages. For example, Grosso et al. [49] [48] [117] suggested an agent programming framework based on the C# language and the Common Language Infrastructure (CLI). This direction should facilitate developing agent-based systems without re-inventing the wheel. Since creating a new language requires considerable efforts which involve designing the language, creating a new compiler/interpreter, publishing books and training. At the time of the writing, the mentioned agent languages are still in the design stages with no software development tools publicly available. This resulted in limiting the choice of a programming language to an existing high-level object oriented language such as C++, C# or Java. While little differences exist between them, C# was chosen to implement agents because of its strong object oriented features and performance advantage over Java. However, while developing the software solution, the author concluded that C# lacks essential image processing libraries when compared to C++ or Java.

The software realisation of the framework along with the Left Ventricle (LV) solution contains the following main components.

ASIP framework classes: agents in the framework were implemented as objects and a quasi-parallel approach was employed for agent execution. Figure A.22 demonstrates the class diagram of the ASIP framework.

Image Processing (IP) classes: several classes were developed in order to perform low level image processing operations on the images and basic geometric calculations. The DICOM reader implemented in the system in order to read DICOM images is based on ezDICOM [103], which is an open source DICOM reader licensed under the BSD license.

GUI classes: the UI mainly focused on the medical application. It consists of a plan builder and a main window which allows the user to interact with the framework while segmenting a medical dataset.

The main window of the software application demonstrated in Figure A.23 allows the user to interact with the framework. To segment a dataset the user clicks on "Open Image Set" which allows the user to select a medical dataset along with a previously created cognitive plan. The first slice in the medical dataset is displayed in the image viewer on the left side. Also, other information views such as the DICOM tags, a view of the agent world and the histogram (available after the segmentation) are accessible by clicking on the related tabs. For the first slice in the dataset, the seed location is manually inputted by clicking on "Initialise Society" and choosing any point inside the endocardium. For the epicardium and subsequent slices the seed location is automatically located using a scout

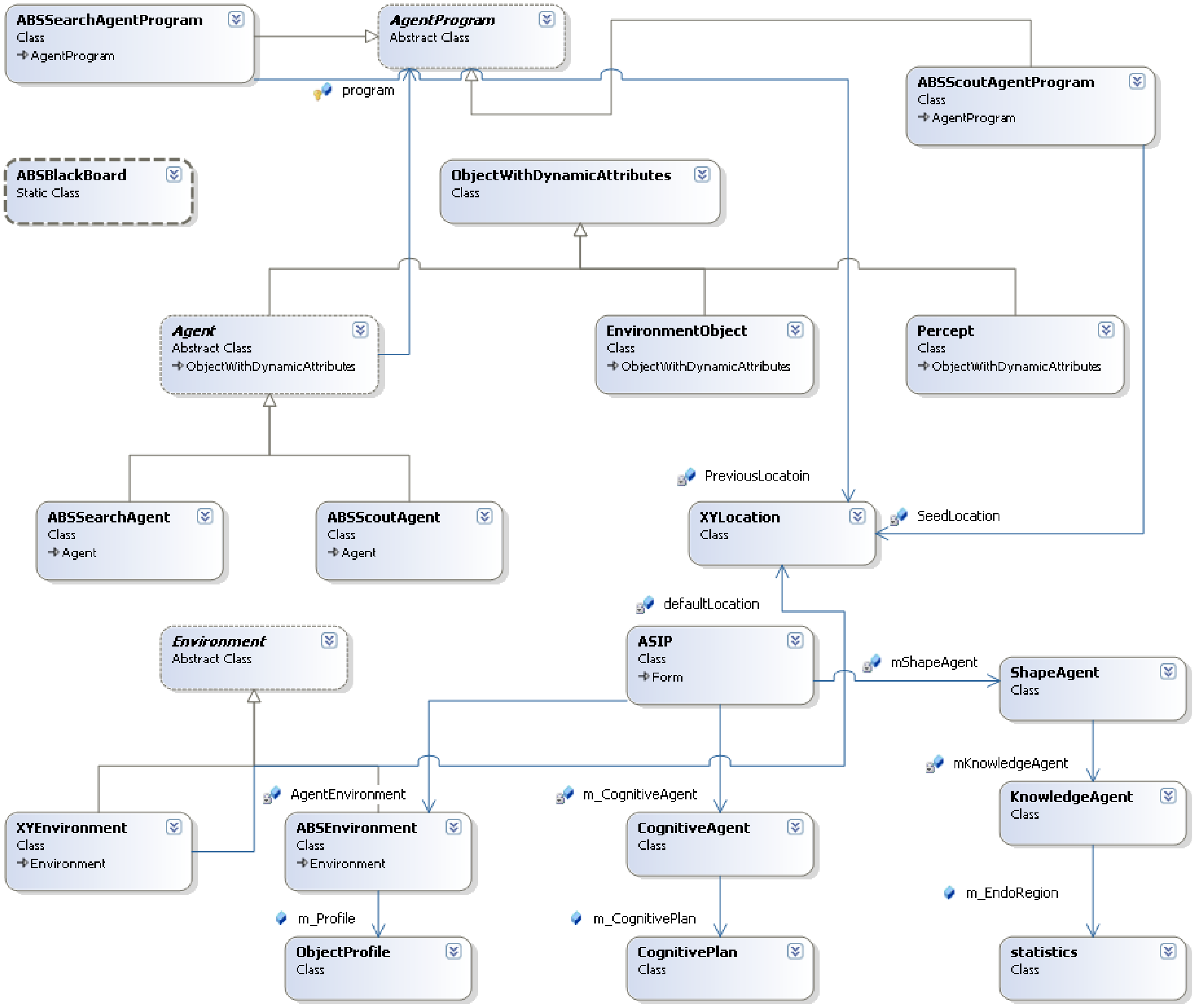
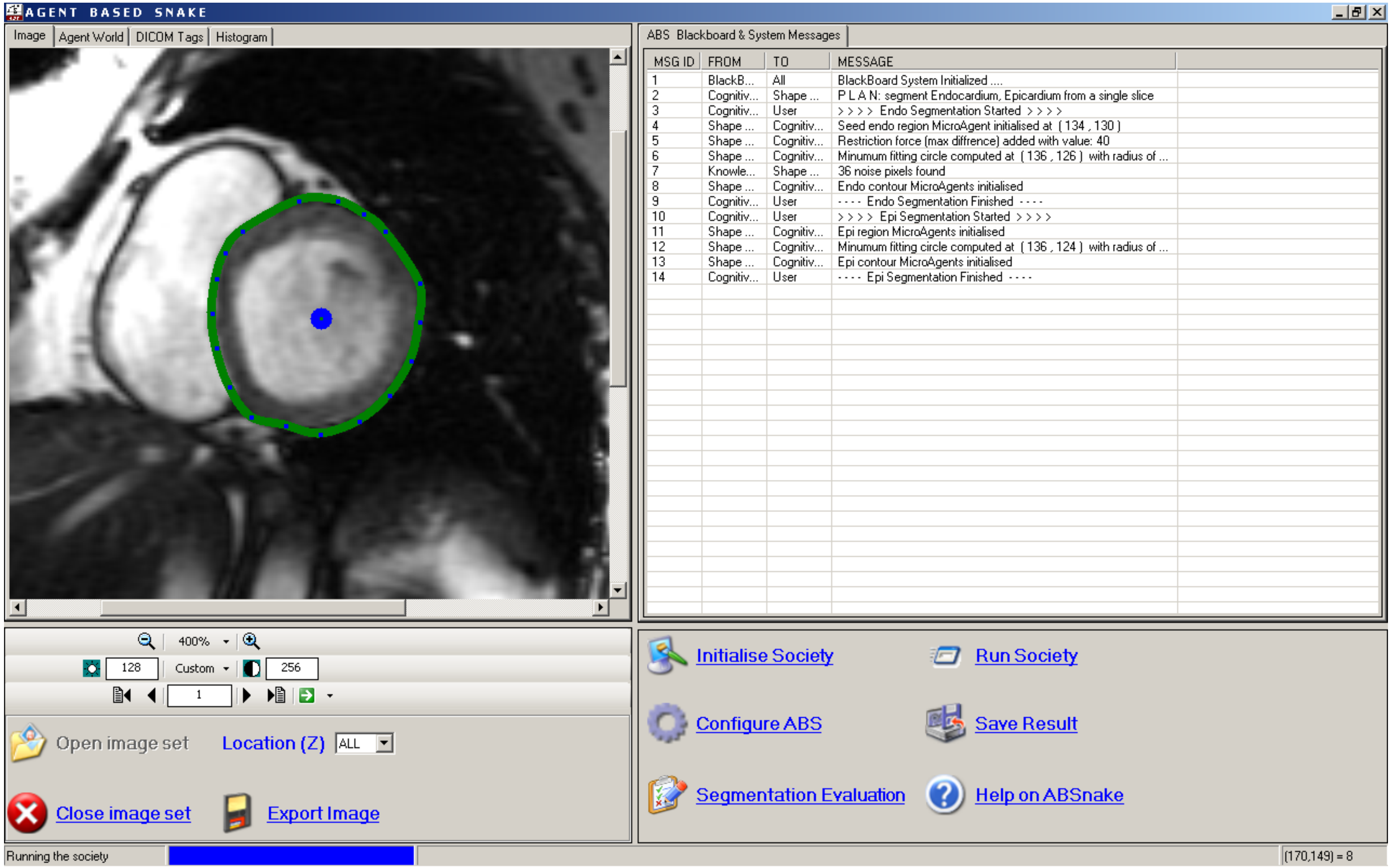


Figure A.22: Class diagram demonstrating the ASIP framework main classes.

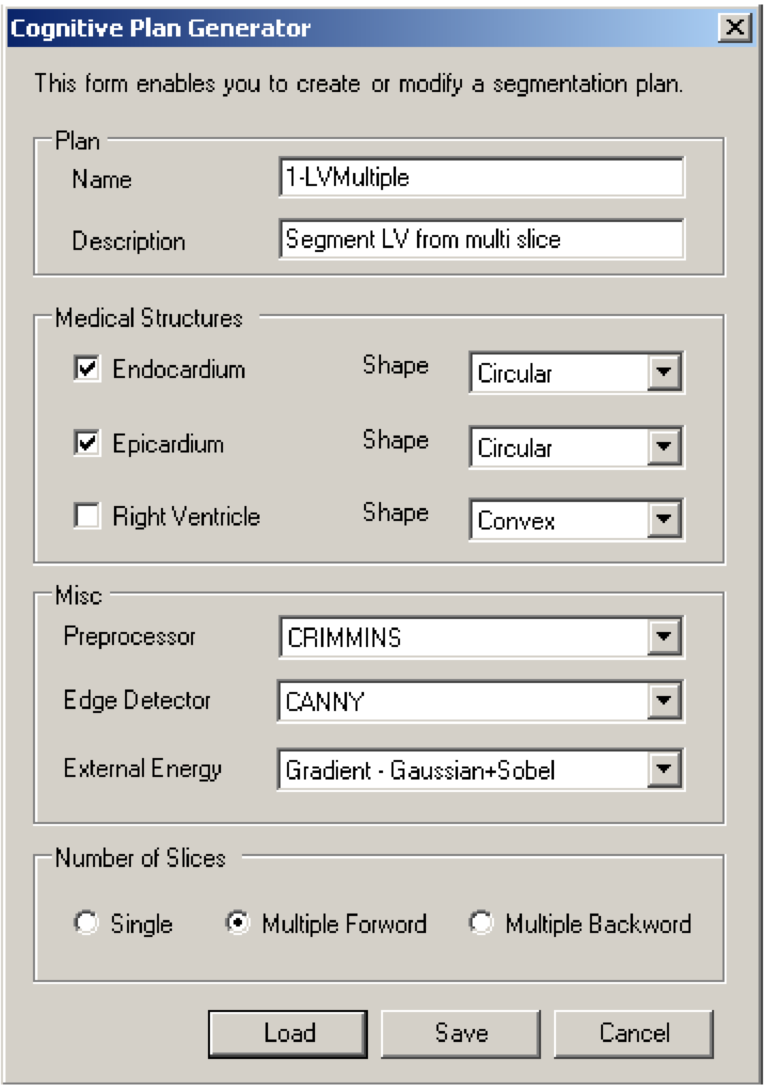
MA. After initialising the society, the user then clicks on “Run Society” to start segmenting the dataset. While segmenting the dataset, high-level communication messages are displayed in the listview on the right side. Finally, the user can save the output of the segmentation in XML format by clicking on “Save Result”.

A basic cognitive plan builder tailored for the ventricle segmentation is displayed in Figure A.24. The plan builder allows the user to select the target medical structure(s) along with their shapes. Currently, three shapes are supported, circular, convex and concave. In addition, the user can choose the preprocessors (if any), edge detector and external energy to be used in the segmentation. The builder also allows the creation of a plan to segment either a single slice or a complete dataset.



MSG ID	FROM	TO	MESSAGE
1	BlackB...	All	BlackBoard System Initialized ....
2	Cognitiv...	Shape ...	P L A N: segment Endocardium, Epicardium from a single slice
3	Cognitiv...	User	> > > > Endo Segmentation Started > > > >
4	Shape ...	Cognitiv...	Seed endo region MicroAgent initialised at (134 , 130)
5	Shape ...	Cognitiv...	Restriction force (max diffrence) added with value: 40
6	Shape ...	Cognitiv...	Minumum fitting circle computed at (136 , 126) with radius of ...
7	Knowle...	Shape ...	36 noise pixels found
8	Shape ...	Cognitiv...	Endo contour MicroAgents initialised
9	Cognitiv...	User	- - - - Endo Segmentation Finished - - - -
10	Cognitiv...	User	> > > > Epi Segmentation Started > > > >
11	Shape ...	Cognitiv...	Epi region MicroAgents initialised
12	Shape ...	Cognitiv...	Minumum fitting circle computed at (136 , 124) with radius of ...
13	Shape ...	Cognitiv...	Epi contour MicroAgents initialised
14	Cognitiv...	User	- - - - Epi Segmentation Finished - - - -

Figure A.23: Screenshot of the developed software main window.



This form enables you to create or modify a segmentation plan.

Plan

Name: 1-LVMultiple

Description: Segment LV from multi slice

Medical Structures

☑ Endocardium — Shape: Circular

☑ Epicardium — Shape: Circular

□ Right Ventricle — Shape: Convex

Misc

Preprocessor: CRIMMINS

Edge Detector: CANNY

External Energy: Gradient - Gaussian+Sobel

Number of Slices

○ Single ● Multiple Forword ○ Multiple Backword

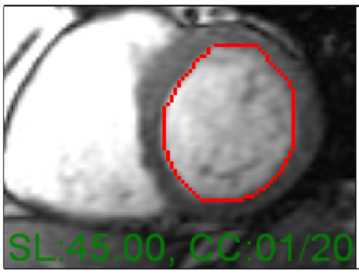
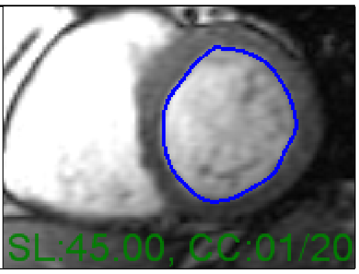
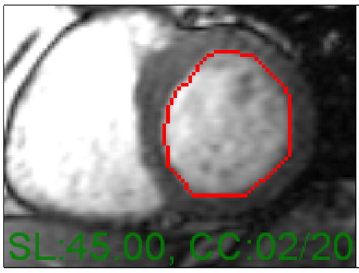
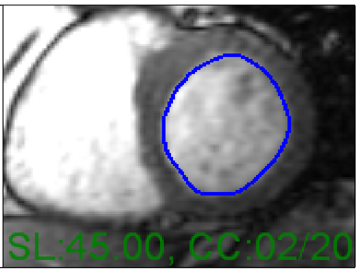
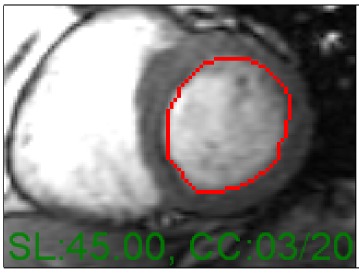
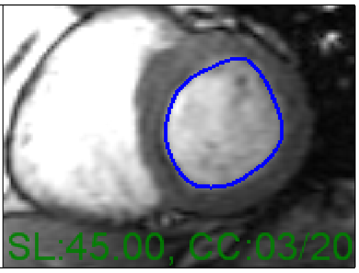
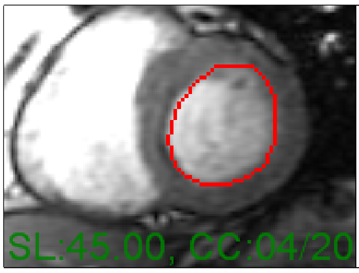
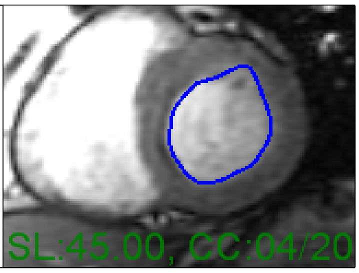
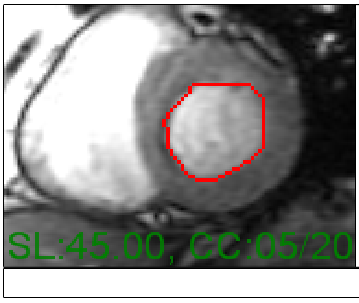
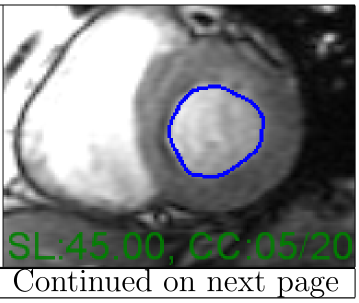
Figure A.24: Screenshot of the plan generator used in the LV segmentation.

Appendix B.

The Left Ventricle Evaluation Datasets

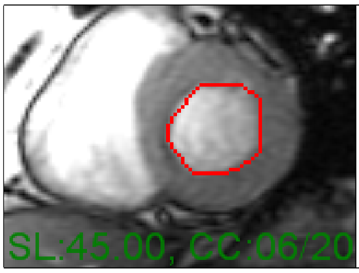
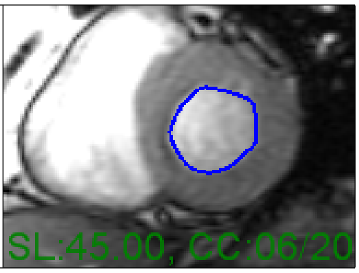
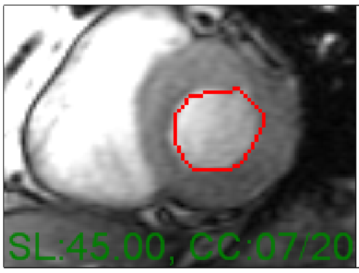
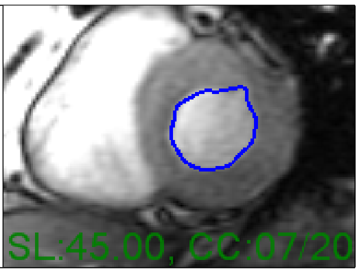
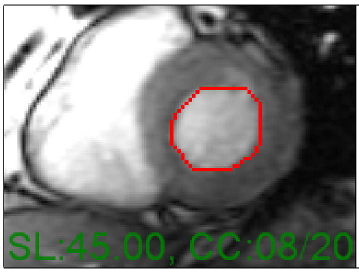
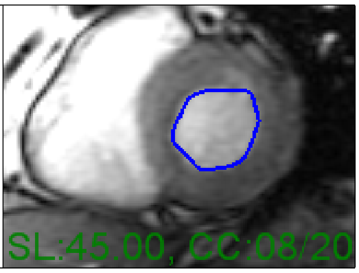
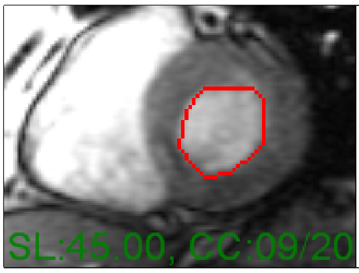
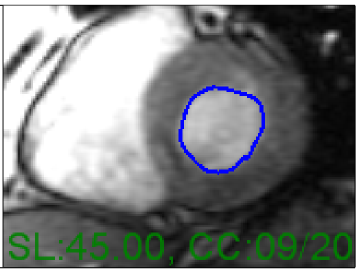
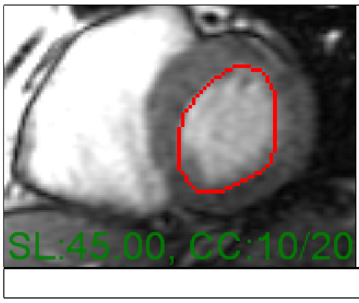
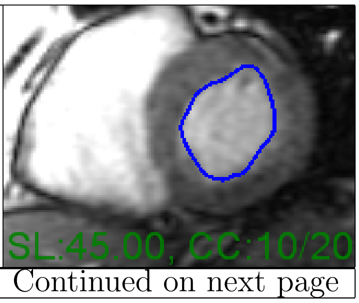
This appendix contains the complete list of the reference and segmented images used in the image evaluation. Each row contains a cropped view of the reference image along with its corresponding segmented image. The reference images were created by manually delineating the endocardium boundary. While, the segmented images are the output of MultiAgent framework for the medical case study presented in Chapter 5.

Table A.5: Reference image set employed in the empirical evaluation.

Reference Image	Segmented Image
 	
 	
 	
 	
 	

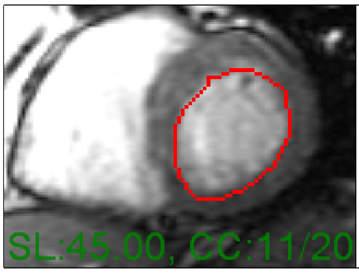
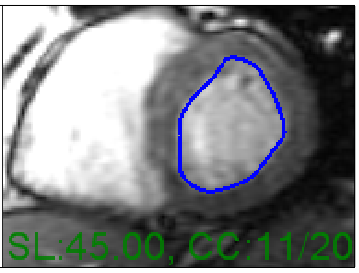
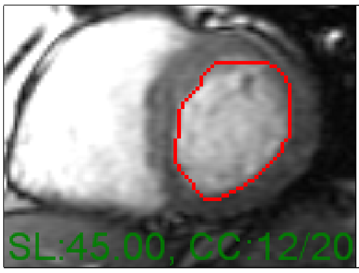
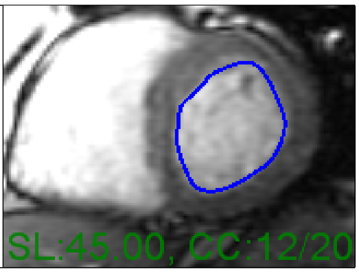
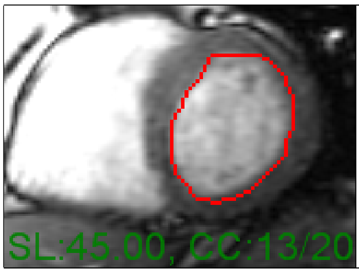
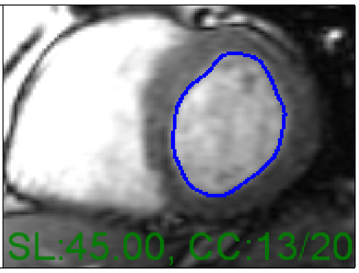
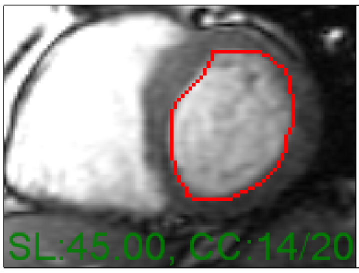
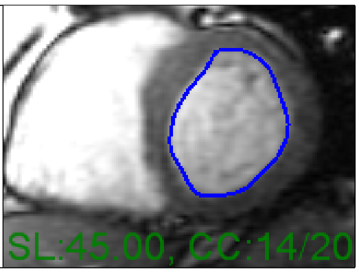
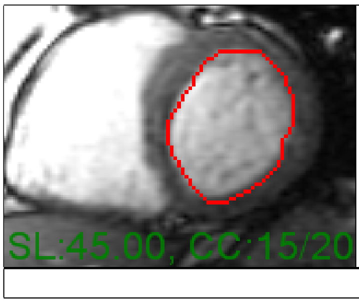
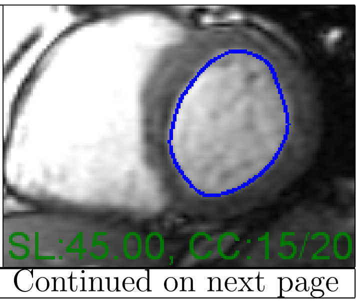
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

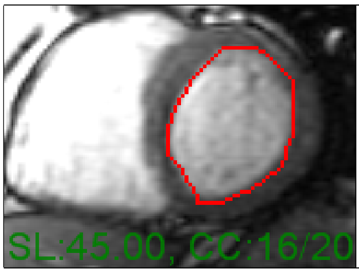
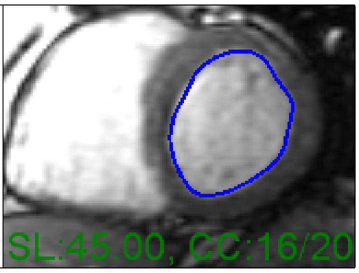
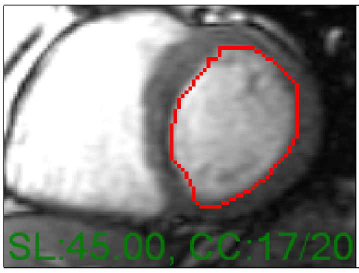
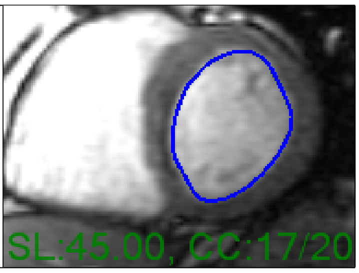
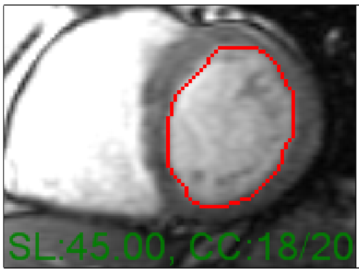
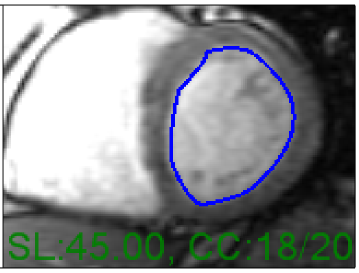
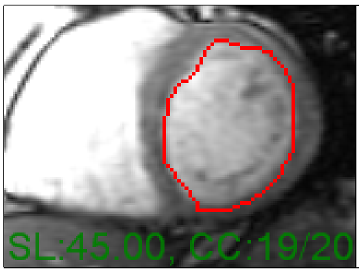
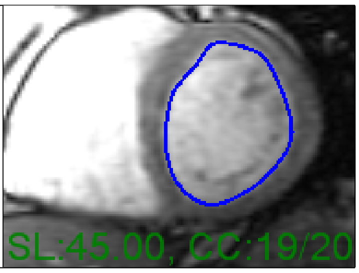
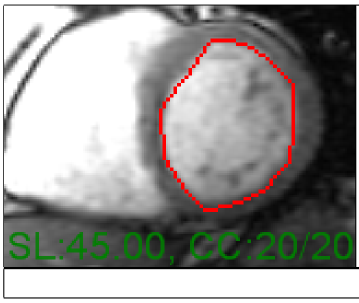
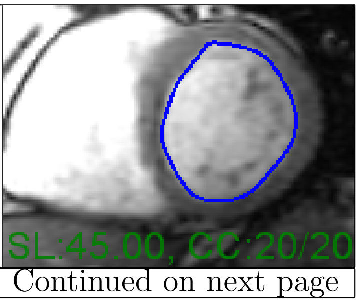
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

Continued on next page

Table A.5 – continued from previous page

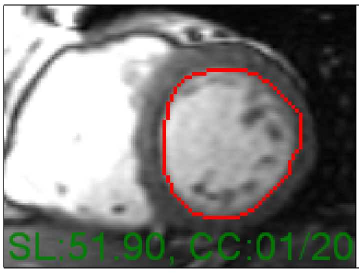
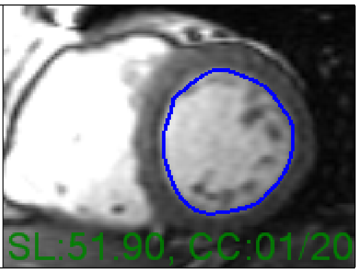
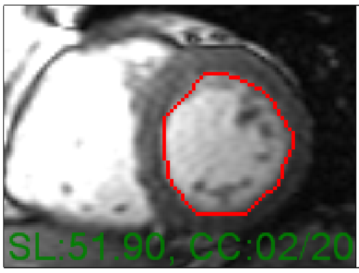
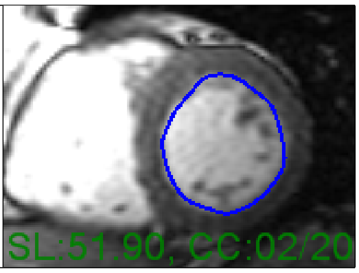
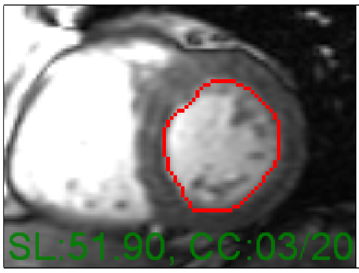
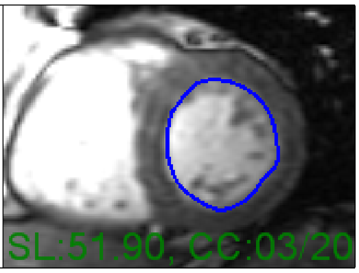
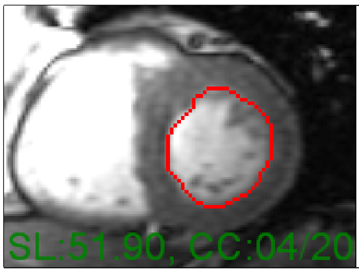
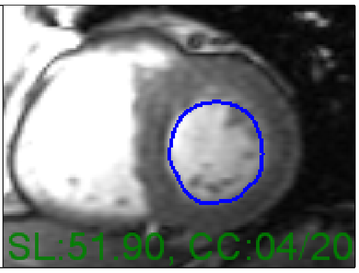
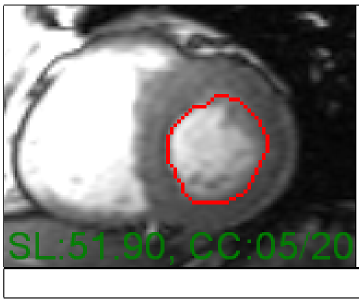
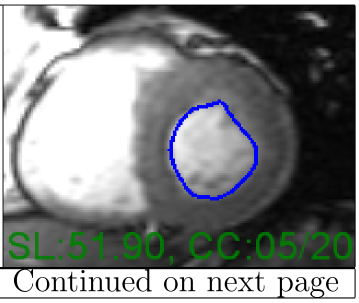
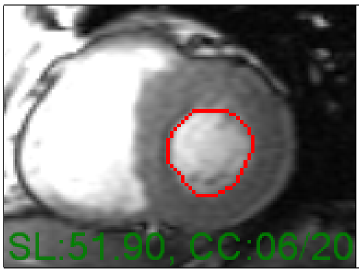
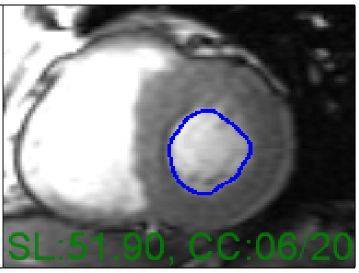
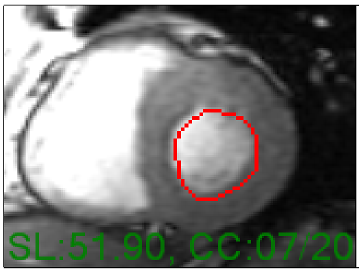
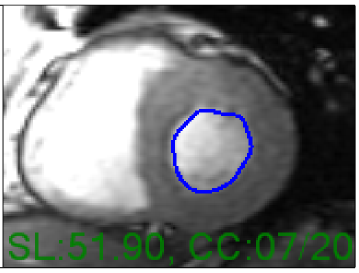
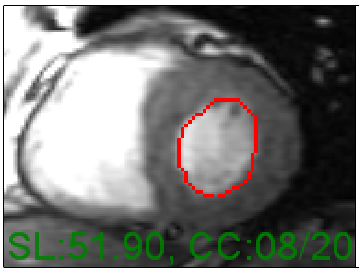
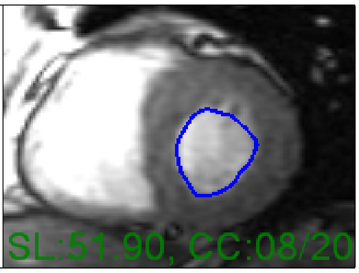
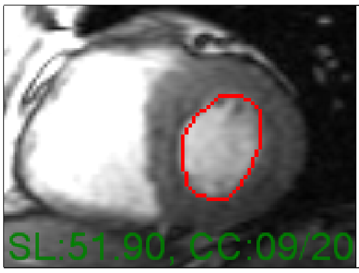
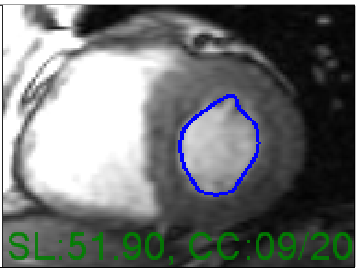
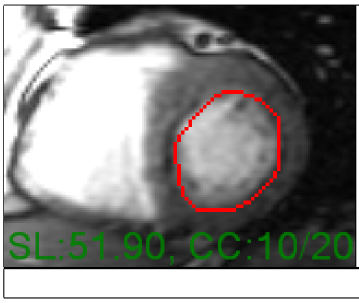
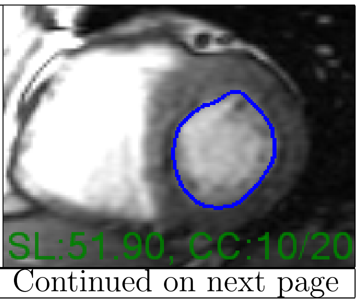
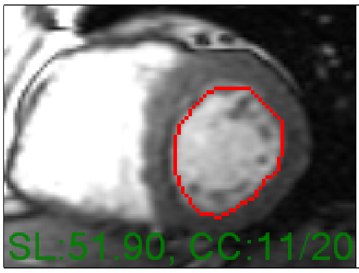
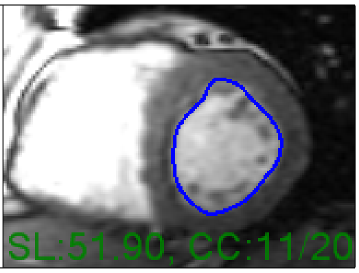
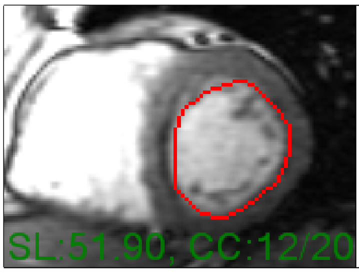
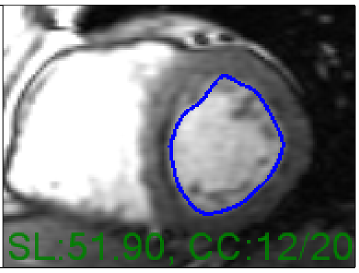
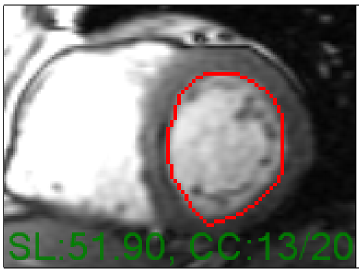
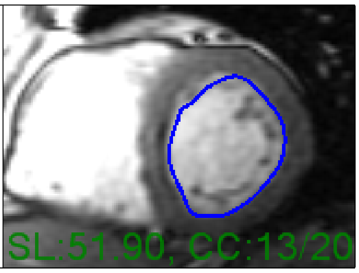
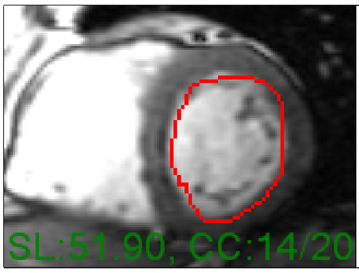
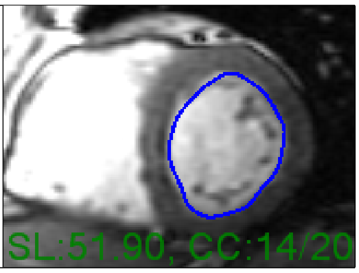
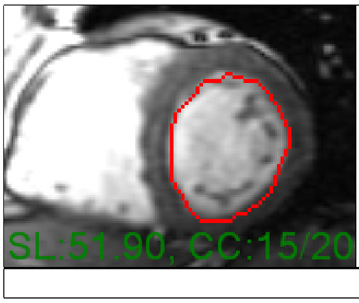
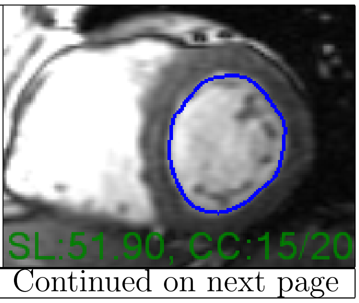
Reference Image	**Segmented Image**
	
	
	
	
	
	Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

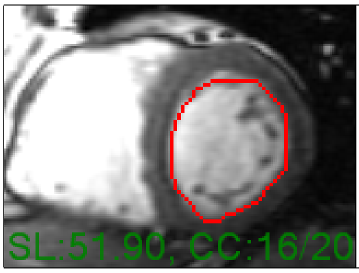
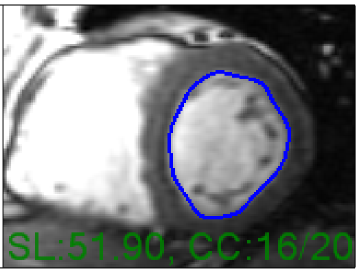
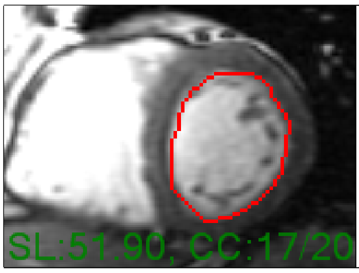
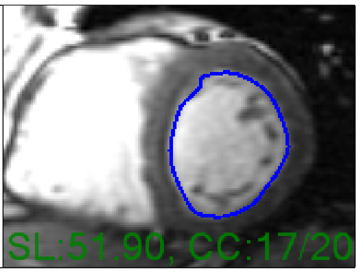
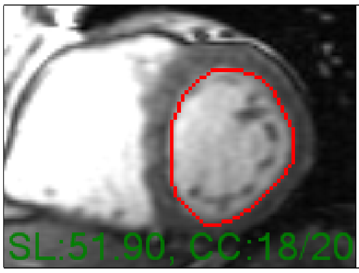
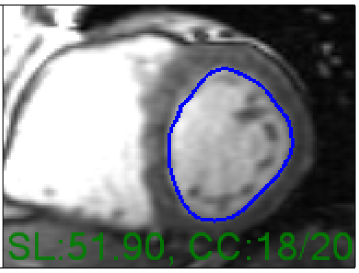
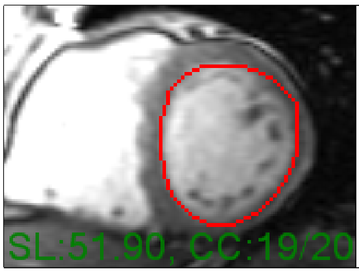
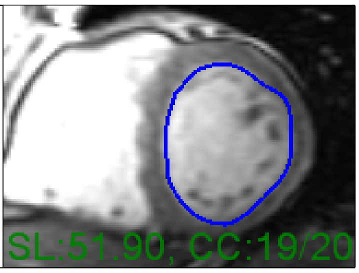
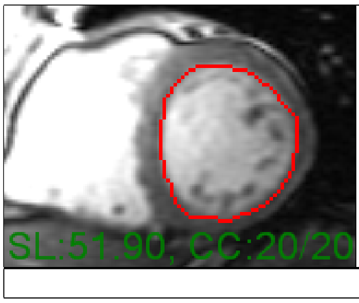
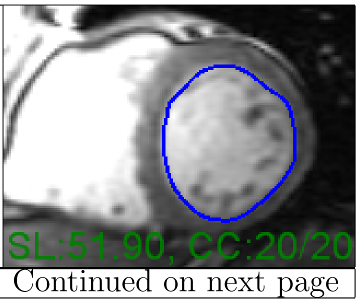
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

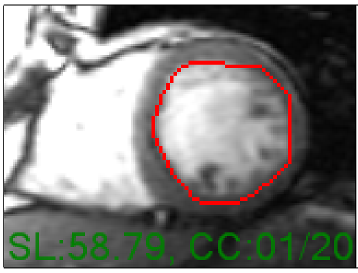
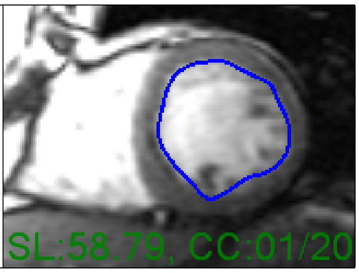
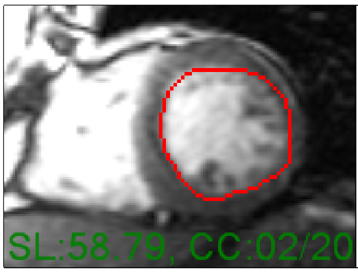
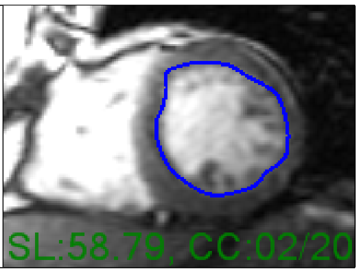
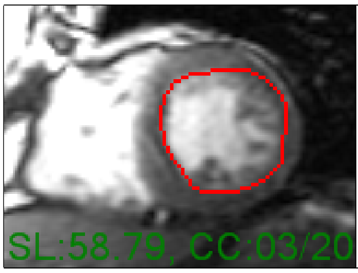
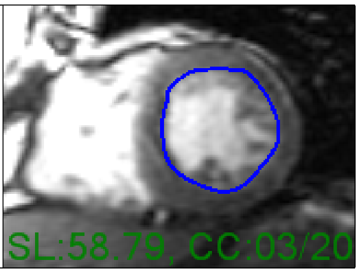
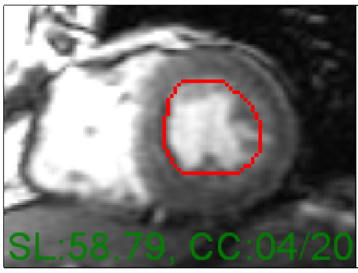
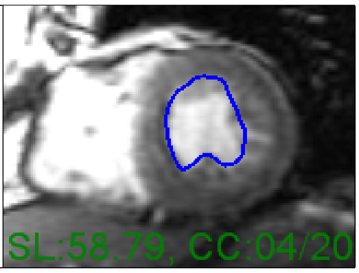
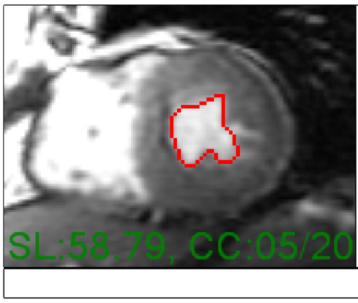
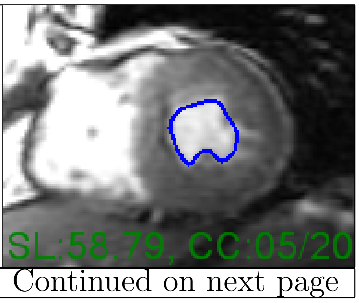
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

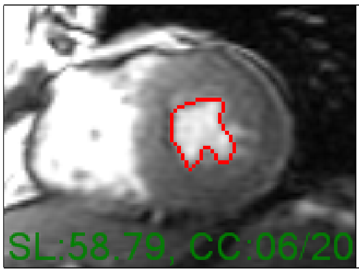
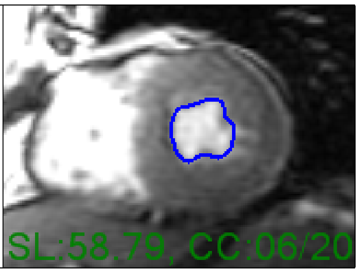
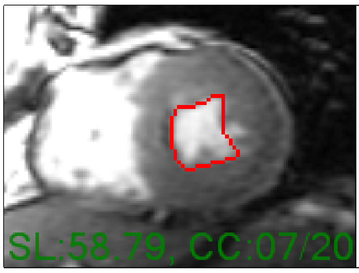
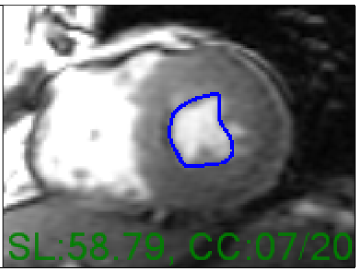
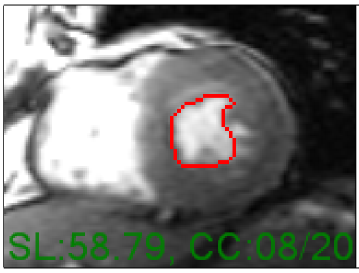
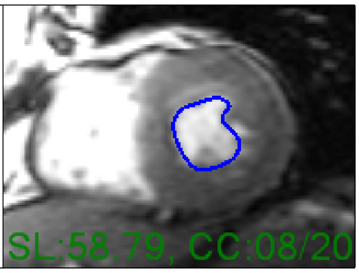
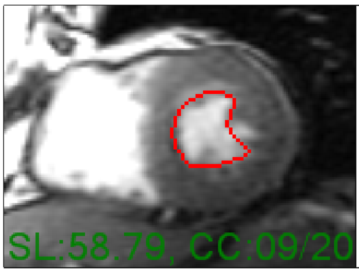
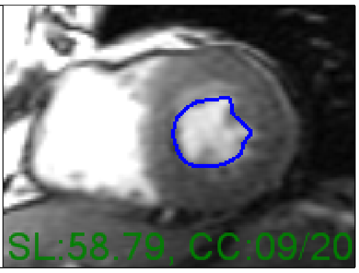
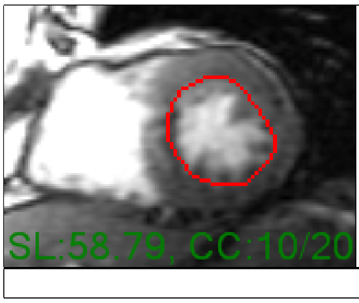
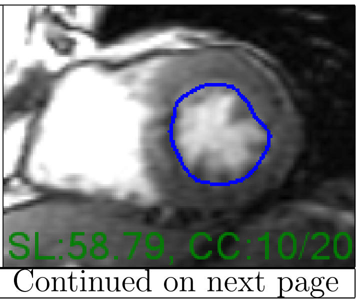
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

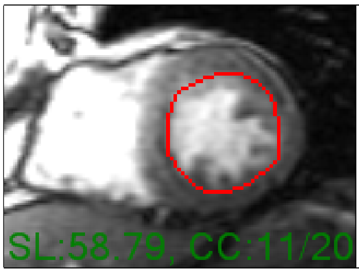
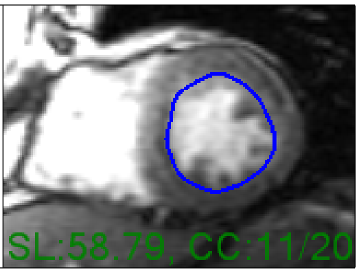
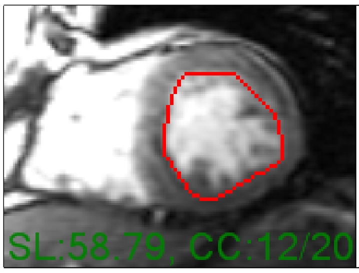
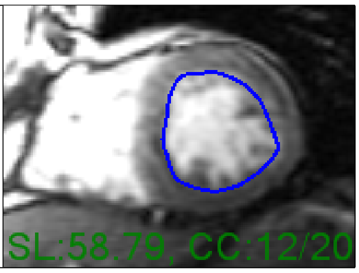
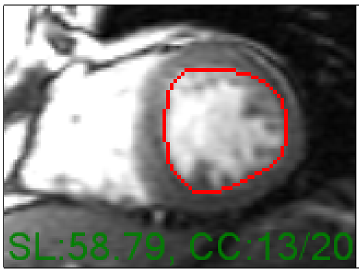
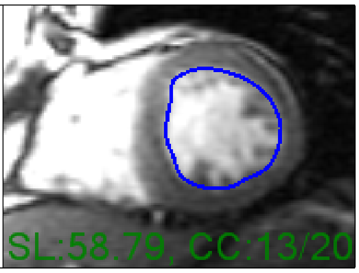
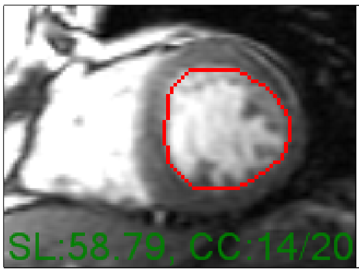
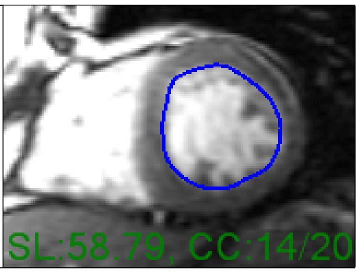
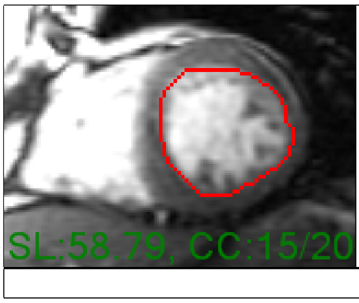
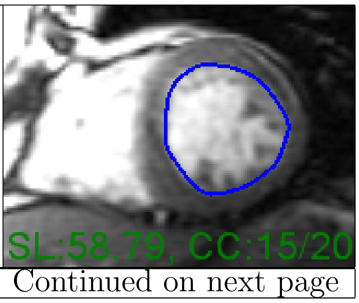
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

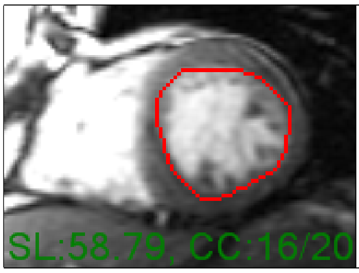
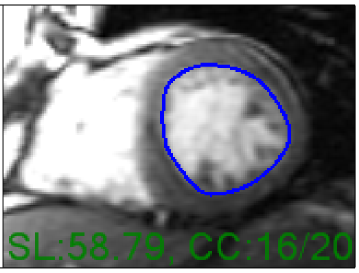
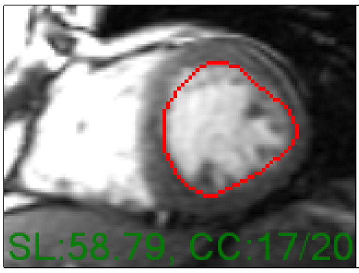
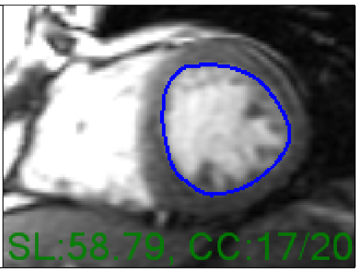
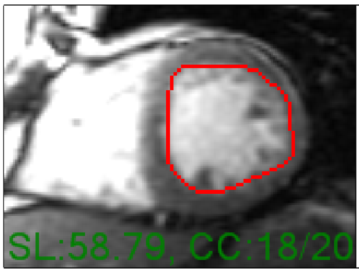
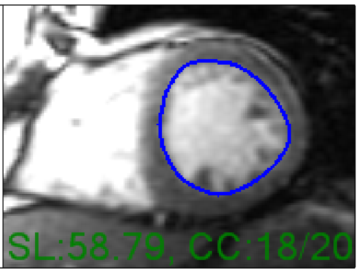
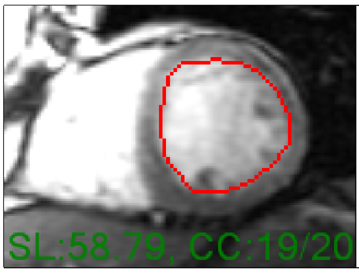
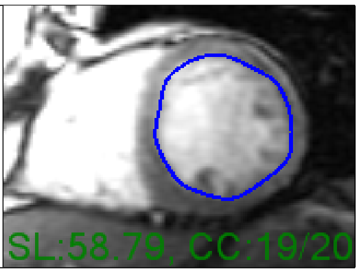
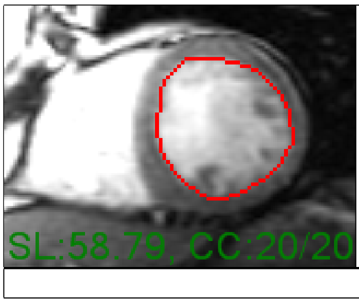
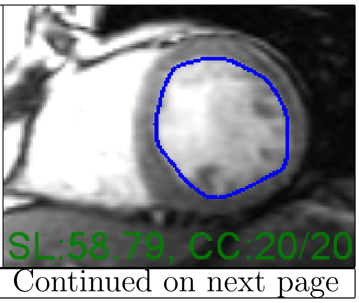
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

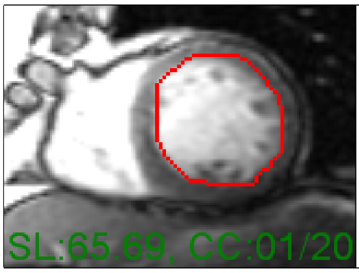
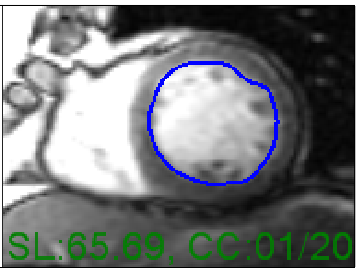
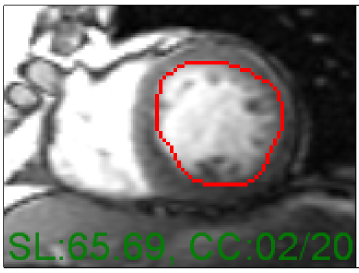
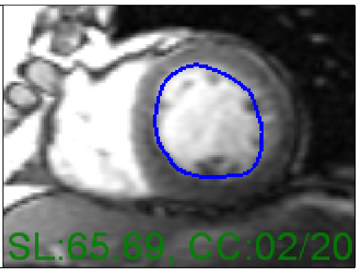
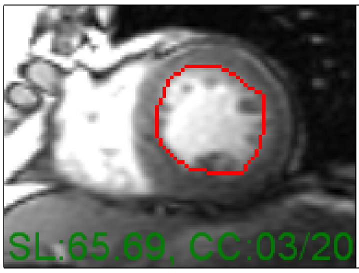
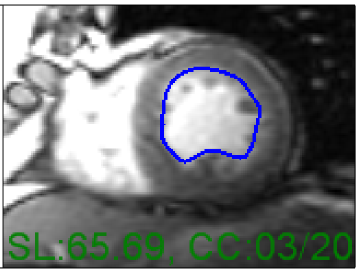
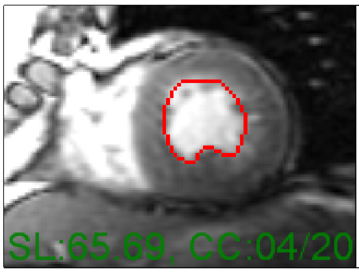
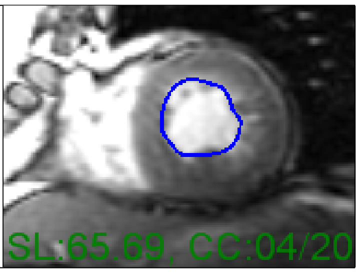
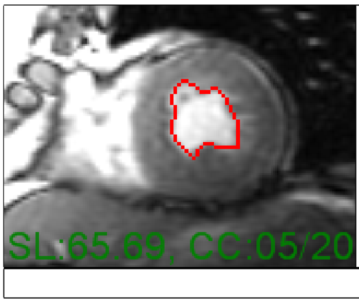
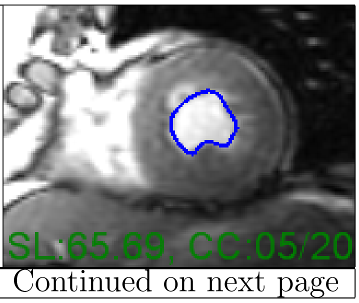
Continued on next page

Table A.5 – continued from previous page

Reference Image	Segmented Image
	
	
	
	
	

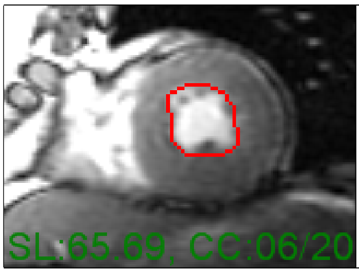
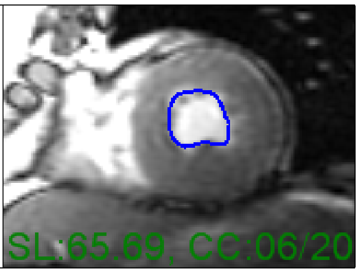
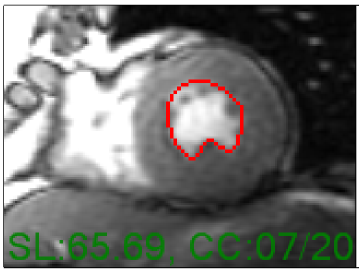
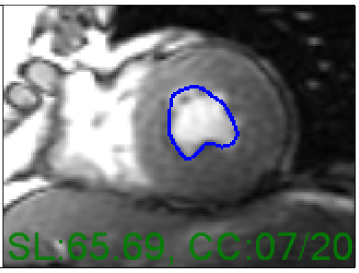
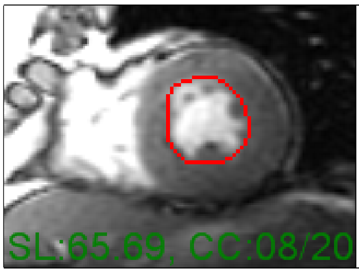
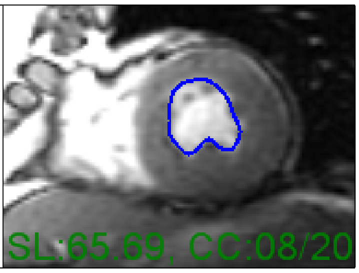
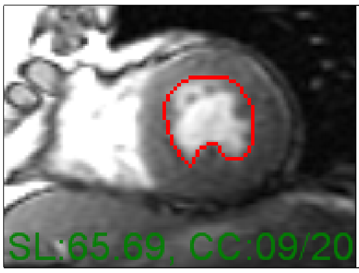
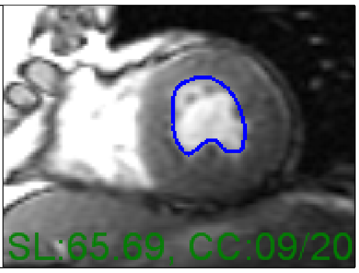
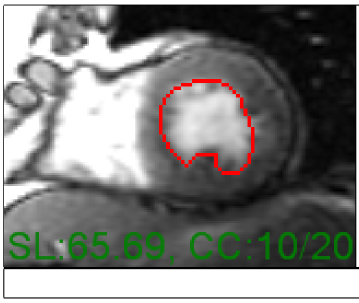
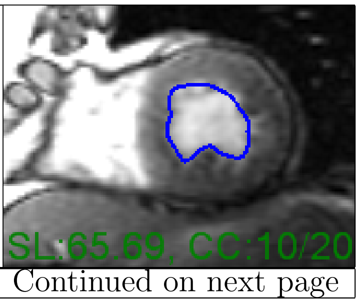
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

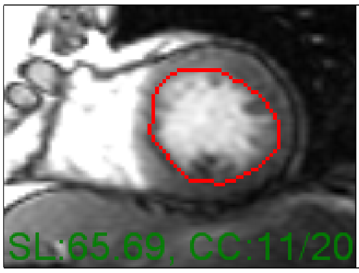
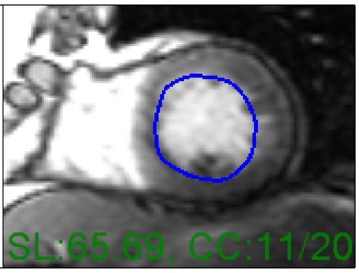
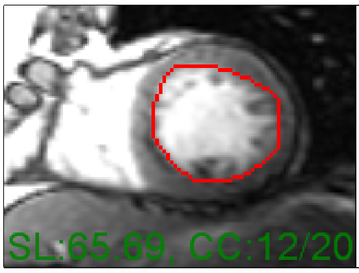
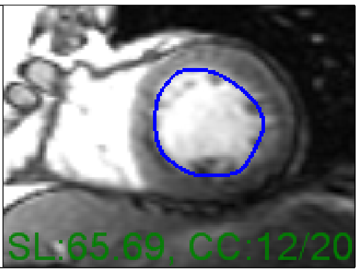
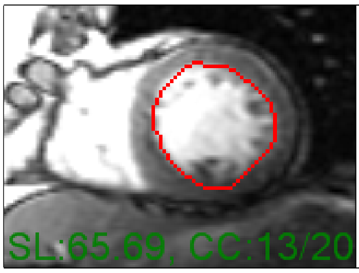
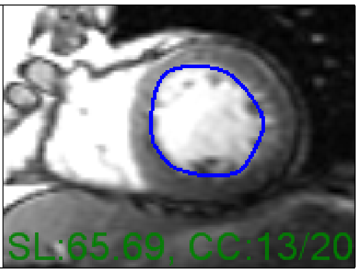
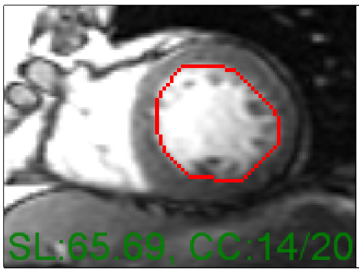
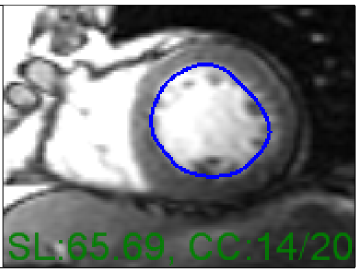
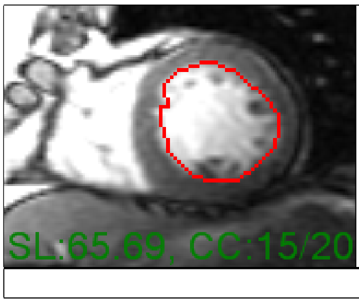
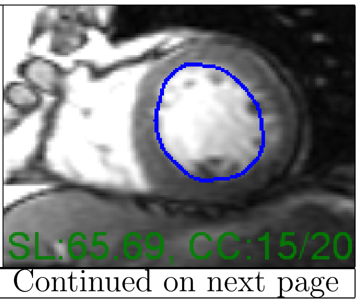
Continued on next page

Table A.5 – continued from previous page

Reference Image	Segmented Image
	
	
	
	
	

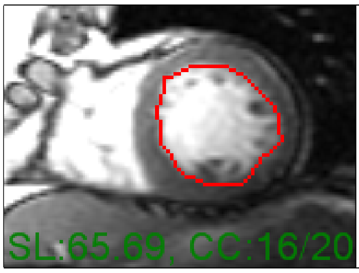
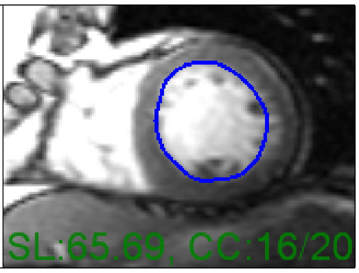
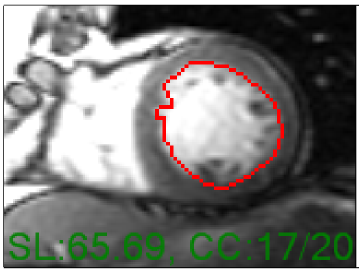
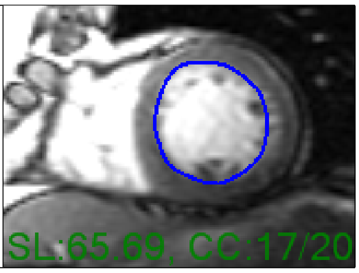
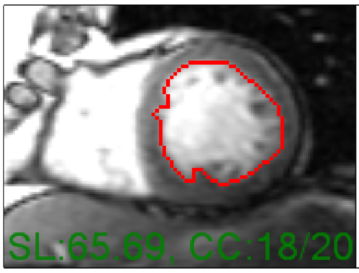
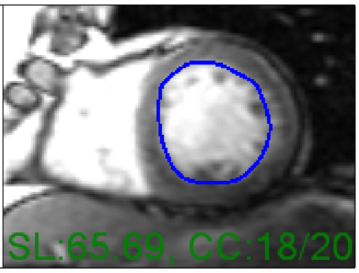
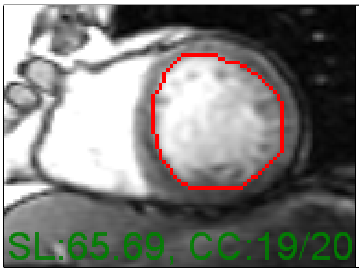
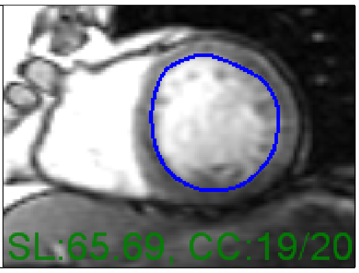
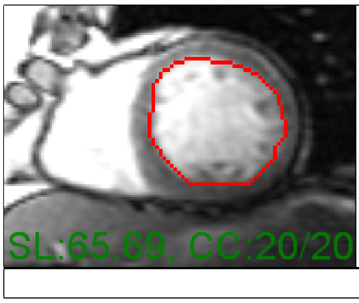
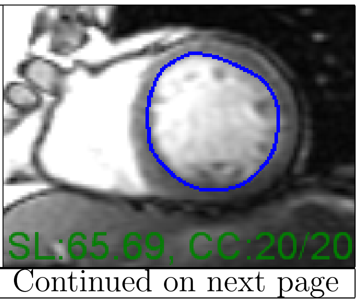
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

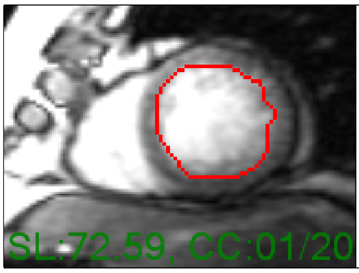
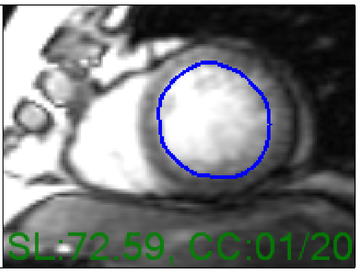
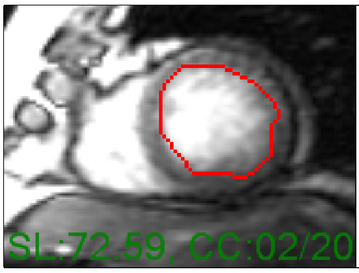
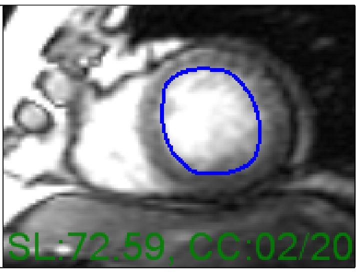
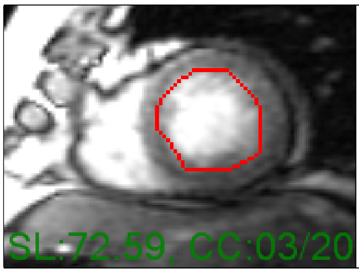
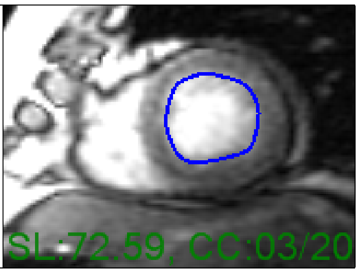
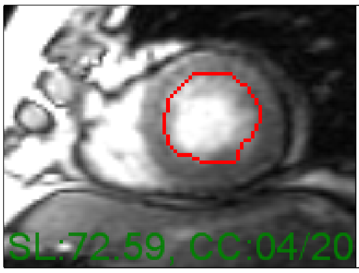
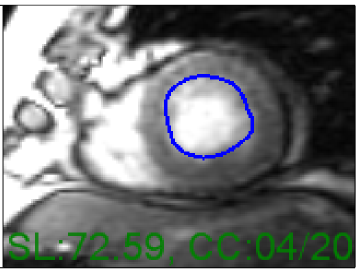
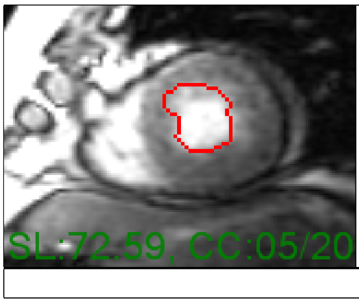
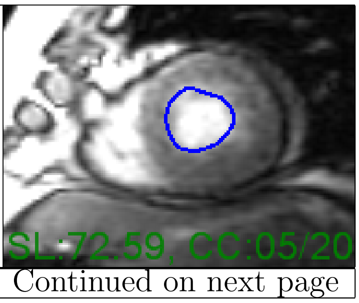
Continued on next page

Table A.5 – continued from previous page

Reference Image	Segmented Image
	
	
	
	
	

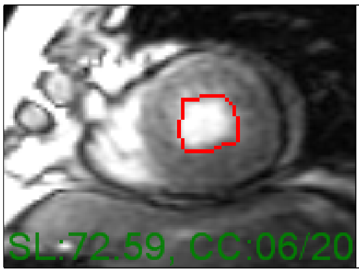
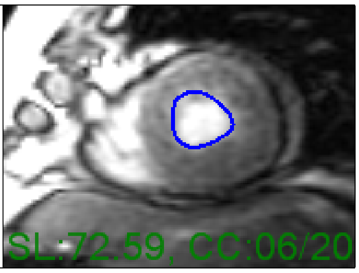
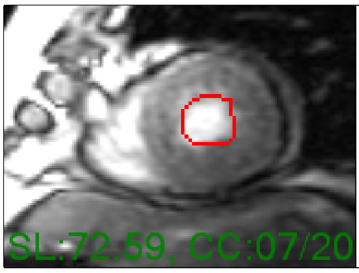
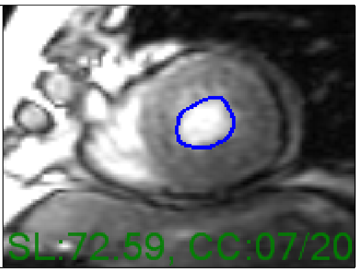
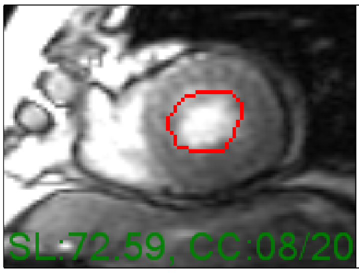
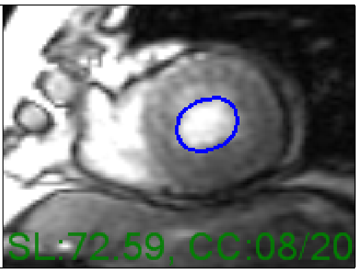
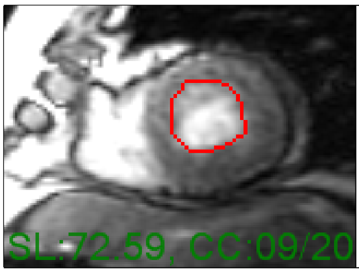
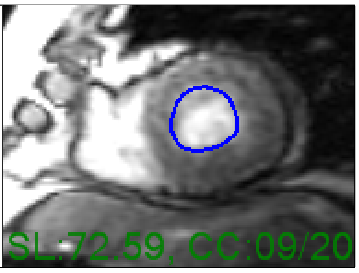
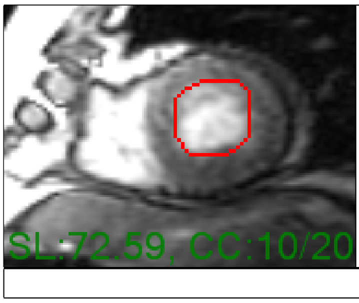
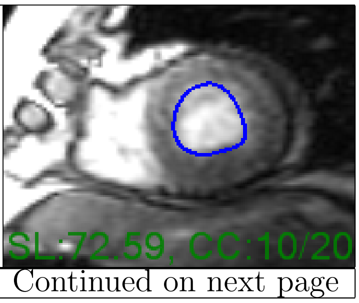
Continued on next page

Table A.5 – continued from previous page

Reference Image	Segmented Image
	
	
	
	
	

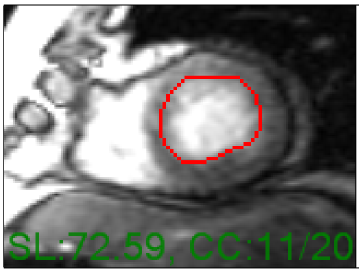
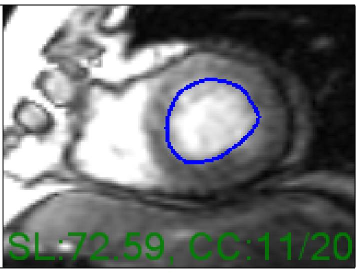
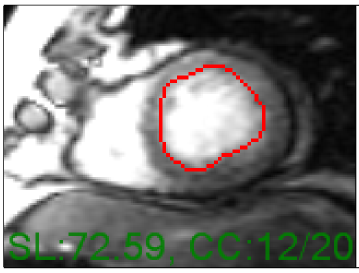
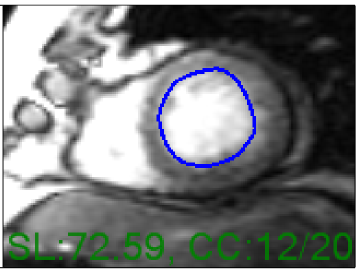
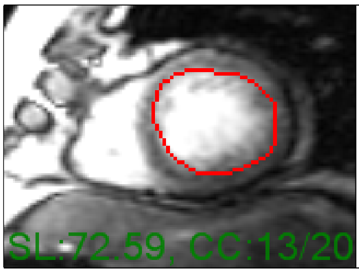
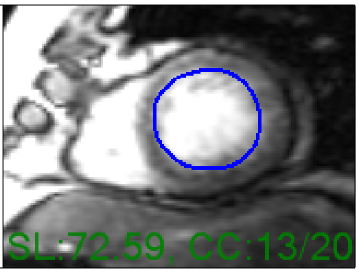
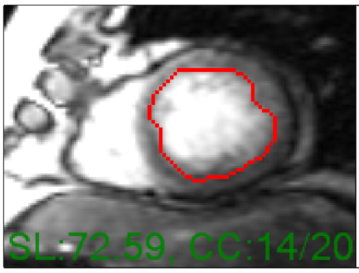
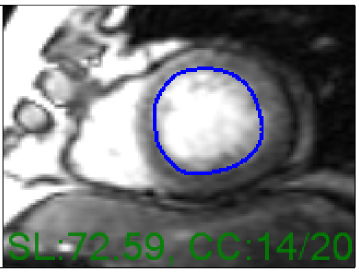
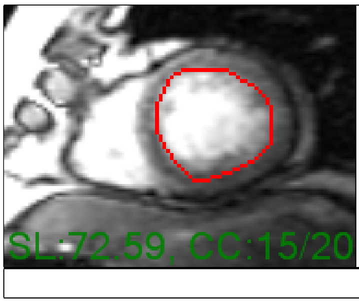
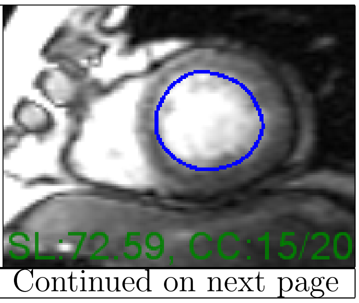
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

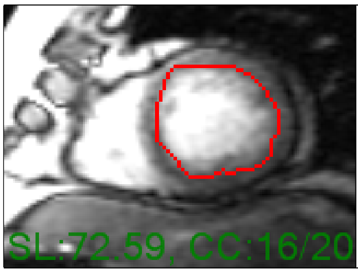
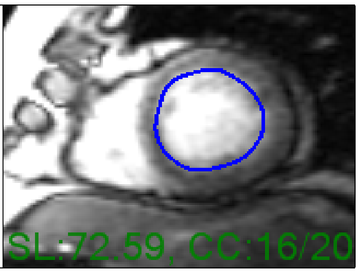
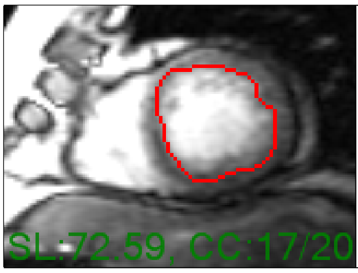
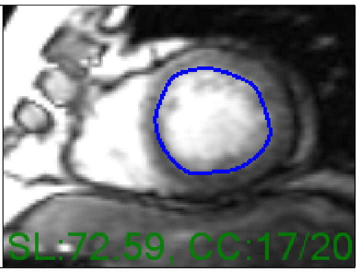
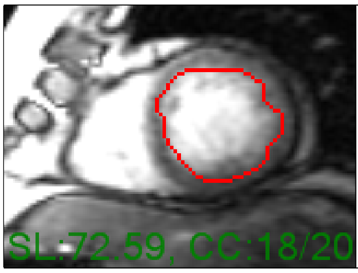
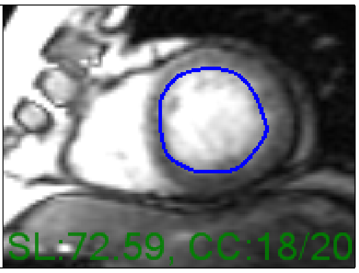
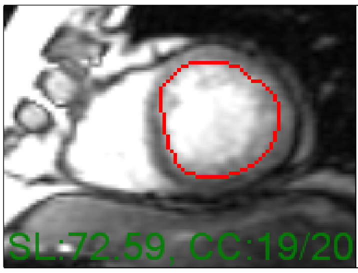
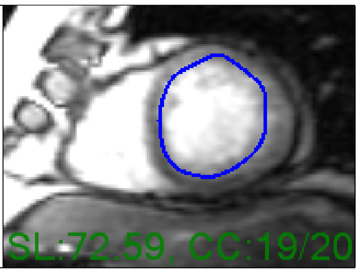
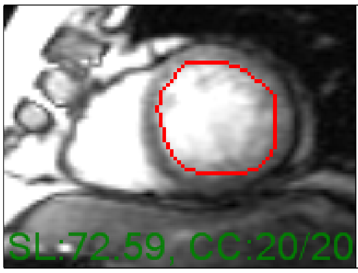
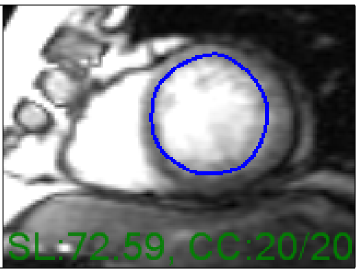
Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

Continued on next page

Table A.5 – continued from previous page

Reference Image	**Segmented Image**
	
	
	
	
	

References

[1] M. Acharyya and M. Kundu. Wavelet-based texture segmentation of remotely sensed images. In *International Conference on Image Analysis and Processing*, pages 69–74, 2001. (Cited on pages 2 and 11.)

[2] A. A. amd S. Tehrani and T. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *IEEE International Conference on Computer Vision*, pages 95–99, 1988. (Cited on page 25.)

[3] A. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, 1979. (Cited on pages 100 and 120.)

[4] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer, 1999. (Cited on page 20.)

[5] R. Boscolo, M. Brown, and M. McNitt-Gray. Medical image segmentation with knowledge-guided robust active contours. *RADIOGRAPHICS*, 22:437–448, 2002. (Cited on pages 20, 30 and 73.)

[6] A. Boucher, A. Doisy, X. Ronot, and C. Garbay. A society of goal-oriented agents for the analysis of living cells. *Artificial Intelligence in Medicine*, 14(1-2):183–199, 1998. (Cited on page 48.)

[7] J. Bradshaw, editor. *Software Agents*. AAAI Press/MIT Press, 1997. (Cited on page 41.)

[8] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1&2):3–15, June 1990. (Cited on pages 9, 32, 44 and 58.)

[9] R. A. Brooks. Intelligence without reason. In J. Myopoulos and R. Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 569–595, Sydney, Australia, 1991. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA. (Cited on pages 9, 32, 44 and 58.)

[10] R. A. Brooks. Intelligence without representation. *Artif. Intell.*, 47(1-3):139–159, 1991. (Cited on pages 9, 32, 44 and 58.)

[11] R. A. Brooks. A robust layered control system for a mobile robot. In S. S. Iyengar and A. Elfes, editors, *Autonomous Mobile Robots: Control, Plan-*

ning, and Architecture (Vol. 2), pages 152–161. IEEE Computer Society Press, Los Alamitos, CA, 1991. (Cited on page 44.)

[12] M. Campbell, A. Hoane, and F. Hsu. Deep Blue. *Artificial Intelligence*, 134(1-2):57–83, 2002. (Cited on page 34.)

[13] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. (Cited on page 17.)

[14] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, February 1997. (Cited on page 20.)

[15] A. Chakraborty, L. Staib, and J. Duncan. Deformable boundary finding in medical images by integrating gradient and region information. *Medical Imaging, IEEE Transactions on*, 15(6):859–870, 1996. (Cited on page 88.)

[16] V. Chalana and Y. Kim. A methodology for evaluation of boundary detection algorithms onmedical images. *Medical Imaging, IEEE Transactions on*, 16(5):642–652, 1997. (Cited on pages 131, 132, 133 and 135.)

[17] F. Chan, F. Lam, P. Poon, H. Zhu, and K. Chan. Object boundary location by region and contour deformation. *VISP*, 143(6):353–360, December 1996. (Cited on page 18.)

[18] T. Chan and L. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, February 2001. (Cited on page 114.)

[19] B. Chanda and D. Majumder. *Digital Image Processing and Analysis*. Prentice Hall of India, 2003. (Cited on page 143.)

[20] P. Chen and T. Pavlidis. Image segmentation as an estimation problem. *Decision and Control including the Symposium on Adaptive Processes, 1979 18th IEEE Conference on*, 18, 1979. (Cited on page 16.)

[21] C. Chesnaud, P. Refregier, and V. Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1145–1157, 1999. (Cited on page 73.)

[22] L. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 53(2):211–218, 1991. (Cited on pages 6, 27 and 30.)

[23] L. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1131–1147, 1993. (Cited on pages 6, 25, 27, 30 and 95.)

[24] G. Coleman and H. Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979. (Cited on page 18.)

[25] D. Corkill. Collaborating software: Blackboard and multi-agent systems & the future. In *In Proceedings of the International Lisp Conference*, New York, 2003. (Cited on pages 66 and 67.)

[26] A. S. Corp. 3d-doctor, medical modeling, 3d medical imaging. `http://www.ablesw.com//3d-doctor//index.html//`, last accessed on :04/12/2008, 1994. (Cited on page 145.)

[27] D. Crevier and R. Lepage. Knowledge-based image understanding systems: a survey. *Comput. Vis. Image Underst.*, 67(2):160–185, 1997. (Cited on pages 48, 71, 75 and 157.)

[28] C. N. de Graaf, A. S. E. Koster, K. L. Vincken, and M. A. Viergever. Validation of the interleaved pyramid for the segmentation of 3d vector images. *Pattern Recogn. Lett.*, 15(5):469–475, 1994. (Cited on page 130.)

[29] H. Delingette and J. Montagnat. Shape and topology constraints on parametric active contours. *CVIU*, 83(2):140–171, August 2001. (Cited on page 22.)

[30] J. C. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):85–106, 2000. (Cited on page 56.)

[31] H. Edelsbrunner. Weighted alpha shapes. Technical report, Champaign, IL, USA, 1992. (Cited on page 120.)

[32] G. Edwards, C. Taylor, and T. Cootes. Interpreting face images using active appearance models. *Third International Conference on Automatic Face and Gesture Recognition*, pages 300–305, 1998. (Cited on page 83.)

[33] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks: A review. *Pattern Recognition*, 35(10):2279–2301, October 2002. (Cited on pages 7 and 55.)

[34] J. Elzinga and D. Hearn. Geometrical solutions for some minimax location problems. *Transportation Science*, 6(4):379–394, 1972. (Cited on pages 100 and 101.)

[35] R. Engelmore and A. Morgan, editors. *Blackboard Systems*. Addison-Wesley, 1988. (Cited on page 66.)

[36] I. A. Ferguson. Towards an architecture for adaptive, rational, mobile agents. In *Proceedings of the Third European Workshop on Modeling Autonomous Agents and Multiagent Worlds*, pages 249–262, New York, USA, 1992. Elsevier Science. (Cited on page 45.)

[37] F. Fernandes, M. Alam, S. Smith, and F. Khaja. The role of transesophageal

echocardiography in identifying anomalous coronary arteries. *Circulation*, 88(6):2532–2540, 1993. (Cited on page 79.)

[38] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In N. Adam, B. Bhargava, and Y. Yesha, editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, Gaithersburg, MD, USA, 1994. ACM Press. (Cited on page 47.)

[39] FIPA. Agent communication language message structure specification. http://www.fipa.org, 2000. (Cited on page 47.)

[40] C. for Morphometric Analysis. Internet analysis tools registry: Home. `http://www.cma.mgh.harvard.edu//iatr//index.php//`, last accessed on :04/12/2008, 2008. (Cited on page 146.)

[41] J. Fox, M. Beveridge, and D. Glasspool. Understanding intelligent agents: analysis and synthesis. *AI Communications*, 16(3):139–152, 2003. (Cited on page 37.)

[42] S. Franklin. *Artificial Minds*. MIT Press, Cambridge, Massachusetts, first edition, 1995. (Cited on page 34.)

[43] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi. Yet another survey on image segmentation: Region and boundary information integration. In

Proceedings of the 7th European Conference on Computer Vision-Part III, pages 408–422. Springer-Verlag, 2002. (Cited on pages 18 and 31.)

[44] O. Gerard, T. Deschamps, M. Greff, and L. Cohen. Real-time interactive path extraction with on-the-fly adaptation of the external forces. *European Conference on Computer Vision*, 2002. (Cited on pages 6 and 87.)

[45] L. Germond, M. Dojat, C. Taylor, and C. Garbay. A cooperative framework for segmentation of MRI brain scans. *Artificial Intelligence In Medicine*, 20(1):77–93, 2000. (Cited on page 48.)

[46] D. Gil and P. Radeva. Curvature vector flow to assure convergent deformable models for shape modelling. In *Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 357–372, Lisboa, Portugal, 2003. (Cited on page 20.)

[47] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Prentice Hall, Upper Saddle River, NJ, second edition, 2001. (Cited on page 5.)

[48] A. Gozzi, M. Coccoli, and A. Boccalatte. Communication and interaction protocols for multi-agent systems in a framework based on c# and the common language infrastructure. In *In Proc. of The 10th ISPE INTERNATIONAL CONFERENCE ON CONCURRENT ENGINEERING:*

RESEARCH AND APPLICATIONS, pages 83–90, Madeira Island - Portugal, 2003. (Cited on pages 105 and 160.)

[49] A. Grosso, A. Gozzi, M. Coccoli, and A. Boccalatte. An Agent Programming Framework Based on the C# Language and the CLI. In *The 1st International Workshop on C# and .NET Technologies on Algorithms, Computer Graphics, Visualization, Distributed, and Web Computing*, pages 13–20, Plzen, Czech Republic, 2003. (Cited on pages 105 and 160.)

[50] S. Gunn and M. Nixon. Improving snake performance via a dual active contour. In V. Hlavac and R. Sara, editors, *Proceedings of Computer Analysis of Images and Patterns*, pages 600–605, Prague, Czech Republic, 1995. (Cited on page 30.)

[51] G. Hamarneh. *Towards Intelligent Deformable Models for Medical Image Analysis*. PhD thesis, Chalmers University of Technology, Sweden, 2001. (Cited on page 58.)

[52] G. Hamarneh, T. McInerney, and D. Terzopoulos. Deformable organisms for automatic medical image analysis. *Lecture Notes in Computer Science*, 2208:66–76, 2001. (Cited on page 58.)

[53] G. Hamarneh, T. McInerney, and D. Terzopoulos. Intelligent deformable or-

ganisms: An artificial life approach to medical image analysis, 2001. (Cited on page 58.)

[54] A. Hoover, G. Jean-Baptiste, X. Jiang, P. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996. (Cited on page 129.)

[55] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. (Cited on pages 131, 132 and 134.)

[56] M. N. Inc. Medx - multimodality image processing and analysis. `http://medx.sensor.com//products//medx//features.html//`, last accessed on :04/12/2008, 2004. (Cited on page 145.)

[57] B. Innovation. The brainvoyager product family powered by brain innovation b.v. `http://www.brainvoyager.com//`, last accessed on :04/12/2008, 2008. (Cited on page 145.)

[58] J. Ivins and J. Porrill. EVERYTHING YOU ALWAYS WANTED TO

KNOW ABOUT SNAKES (BUT WERE AFRAID TO ASK). *Artificial Intelligence*, 2000, 1995. (Cited on page 31.)

[59] A. Jain and R. Dubes. *Algorithms for clustering data*. 1988. (Cited on page 18.)

[60] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998. (Cited on pages 38 and 40.)

[61] N. Jennings and M. Wooldridge. Software agents. *IEE Review*, pages 17–20, 1996. (Cited on page 38.)

[62] L. Ji and H. Yan. Attractable snakes based on the greedy algorithm for contour extraction. *Pattern Recognition*, 35(4):791–806, April 2002. (Cited on page 30.)

[63] X. P. Ji, Z. Q. Wei, and Y. W. Feng. Effective vehicle detection technique for traffic surveillance systems. *Journal of Visual Communication and Image Representation*, 17(3):647–658, 2006. (Cited on pages 2 and 11.)

[64] C.-H. Jo and A. J. Arnold. The agent-based programming language: Apl. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 27–31, New York, NY, USA, 2002. ACM Press. (Cited on pages 104 and 159.)

[65] M. Jolly, N. Duta, and G. Funka-Lea. Segmentation of the left ventricle in cardiac MR images. *Proc of the IEEE Intl Conf on Computer Vision*, 1:501–508, 2001. (Cited on page 88.)

[66] T. Jones and D. Metaxas. Image segmentation based on the integration of pixel affinity and deformable models. In *In Proc. Conf. Computer Vision and Pattern Recognition*, pages 330–337, 1998. (Cited on pages 2, 11 and 51.)

[67] A. Kak, M. Slaney, I. E. in Medicine, and B. Society. *Principles of Computerized Tomographic Imaging*. IEEE Press New York, 1988. (Cited on page 80.)

[68] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. (Cited on pages 6, 19, 20, 23, 25 and 26.)

[69] M. Kaus, J. von Berg, J. Weese, W. Niessen, and V. Pekar. Automated segmentation of the left ventricle in cardiac MRI. *Medical Image Analysis*, 8(3):245–254, 2004. (Cited on pages 80 and 88.)

[70] R. A. Lange and L. D. Hillis. Diagnostic Cardiac Catheterization. *Circulation*, 107(17):e111–113, 2003. (Cited on page 79.)

[71] M. Levine and A. Nazif. An experimental rule-based system for testing low

level segmentation strategies. *Multicomputers and Image Processing Algorithms and Programs*, pages 149–160, 1982. (Cited on pages 131 and 132.)

[72] J. Liu and Y. Y. Tang. Adaptive image segmentation with distributed behavior-based agents. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(6):544–551, 1999. (Cited on page 48.)

[73] J. Liu, J. Udupa, D. Odhner, D. Hackney, and G. Moonis. A system for brain tumor volume estimation via MR imaging and fuzzy connectedness. *Computerized Medical Imaging and Graphics*, 29(1):21–34, 2005. (Cited on pages 2 and 11.)

[74] A. T. Ltd. Digital image processing software - alpha tec ltd. http://www.alphatecltd.com//eikona3d//features.html//, last accessed on :04/12/2008, 2005. (Cited on page 145.)

[75] S. Ltd. Scanip software. http://www.simpleware.com//software//scanip.php//, last accessed on :04/12/2008, 2000. (Cited on page 145.)

[76] X. S. Ltd. Xinapse systems. http://www.xinapse.com//software.html//, last accessed on :04/12/2008, 2008. (Cited on page 145.)

[77] M. Lynch, O. Ghita, and P. F. Whelan. Automatic segmentation of the left ventricle cavity and myocardium in mri data. *Elsevier Computers in*

Biology and Medicine, 34(4):389–407, March 2006. (Cited on pages 80, 83 and 88.)

[78] P. Maes. The agent network architecture (ana). *SIGART Bull.*, 2(4):115–120, 1991. (Cited on page 45.)

[79] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(2):158–175, 1995. (Cited on page 20.)

[80] D. Marr and E. Hildreth. Theory of edge detection. *RoyalP*, B-207:187–217, 1980. (Cited on page 5.)

[81] D. Marr and H. Nishihara. Visual information processing: Artificial intelligence and the sensorium of sight. In *RCV87*, pages 616–637, 1987. (Cited on pages 19 and 52.)

[82] B. McCane. On the evaluation of image segmentation algorithms. *Digital Image Computing: Techniques and Applications*, pages 455–461, 1997. (Cited on page 129.)

[83] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *MIA*, 1(2):91–108, 1996. (Cited on pages 6, 22 and 31.)

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

Conference Proceedings., IEEE International Conference on, pages 194–199, 1990. (Cited on page 31.)

[85] J.-J. C. Meyer. Agent languages and their relationship to other programming paradigms. In *ATAL '98: Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages*, pages 309–316, London, UK, 1999. Springer-Verlag. (Cited on page 105.)

[86] S. Mitchell, B. Lelieveldt, R. van der Geest, H. Bosch, J. Reiver, and M. Sonka. Multistage hybrid active appearance model matching: segmentation ofleft and right ventricles in cardiac MR images. *Medical Imaging, IEEE Transactions on*, 20(5):415–423, 2001. (Cited on page 88.)

[87] A. Moreira and M. Y. Santos. Concave hull : a k-nearest neighbours approach for the computation of the region occupied by a set of points. In *INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS THEORY AND APPLICATIONS*, pages 61–68, Barcelona, Spain, 2007. INSTICC Press. (Cited on page 120.)

[88] H. Nishida. Curve description based on directional features and quasi-convexity concavity. *Pattern Recognition*, 28(7):1045–1051, July 1995. (Cited on page 71.)

[89] H. Nishida. Structural shape indexing with feature generation models. *Comput. Vis. Image Underst.*, 73(1):121–136, 1999. (Cited on page 71.)

[90] J. Odell. Objects and agents compared. *Journal of Object Technology*, 1(1):41–53, 2002. (Cited on page 41.)

[91] N. Pal and S. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. (Cited on pages 31 and 124.)

[92] N. Paragios, M. Rousson, and V. Ramesh. Knowledge-based registration & segmentation of the left ventricle: a level set approach. *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pages 37–42, 2002. (Cited on page 88.)

[93] D. Pasquier, T. Lacornerie, M. Vermandel, J. Rousseau, E. Lartigau, and N. Betrouni. Automatic Segmentation of Pelvic Structures From Magnetic Resonance Images for Prostate Cancer Radiotherapy. *International Journal of Radiation Oncology, Biology, Physics*, 68(2):592–600, 2007. (Cited on pages 2 and 12.)

[94] T. Pavlidis. Image Analysis. *Annual Reviews in Computer Science*, 3(1):121–146, 1988. (Cited on page 2.)

[95] S. Petersen, V. Peto, P. Scarborough, and M. Rayner. *Coronary Heart*

disease Statistics. British Heart Foundation, London, United Kingdom, 2005 edition, June 2005. (Cited on page 79.)

[96] D. Pham, C. Xu, and J. Prince. A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000. (Cited on pages 14, 31 and 54.)

[97] C. Poon and M. Braun. Image segmentation by a deformable contour model incorporating region analysis. *Phys. Med. Biol*, 42(9):1833–1841, 1997. (Cited on page 88.)

[98] C. Prest, R. Phillips, N. Nikitin, C. Langton, and J. Cleland. Hierarchical cardiac MRI segmentation using shaped region-interface descriptors. *International Congress Series*, 1281:363–368, 2005. (Cited on page 88.)

[99] P. Puech and L. Boussel. Dicomworks - a free dicom utility and viewer. http://dicom.online.fr//, last accessed on :1/5/2007, 2000. (Cited on page 87.)

[100] W. Reddick, J. Glass, E. Cook, T. Elkin, and R. Deaton. Automated segmentation and classification of multispectral magneticresonance images of brain using artificial neural networks. *Medical Imaging, IEEE Transactions on*, 16(6):911–918, 1997. (Cited on page 18.)

[101] M. Robiony, I. Salvo, F. Costa, N. Zerman, M. Bazzocchi, F. Toso, C. Ban-

dera, S. Filippi, M. Felice, and M. Politi. Virtual Reality Surgical Planning for Maxillofacial Distraction Osteogenesis: The Role of Reverse Engineering Rapid Prototyping and Cooperative Work. *Journal of Oral and Maxillofacial Surgery*, 65(6):1198–1208, 2007. (Cited on pages 2 and 12.)

[102] M. Rogers and J. Graham. Robust active shape model search. In *Proceedings of the 7th European Conference on Computer Vision*, number IV, pages 517–530, May 2002. (Cited on page 83.)

[103] C. Rorden. ezdicom - activex and clx dicom components. `http://www.sph.sc.edu//comd//rorden//dicomcom.html//`, last accessed on :1/2/2006, 2004. (Cited on pages 106 and 161.)

[104] S. Rosenschein and L. Kaelbling. The synthesis of digital machines with provable epistemic properties. In *Proceedings of the 1986 Conference on Theoretical aspects of reasoning about knowledge*, pages 83–98. Morgan Kaufmann Publishers Inc., 1986. (Cited on page 45.)

[105] J. Rousselle, N. Vincent, and N. Verbeke. Genetic algorithm to set active contour. In *CAIP03*, pages 345–352, 2003. (Cited on page 7.)

[106] S. J. Russell. Rationality and intelligence. *Artif. Intell.*, 94(1-2):57–77, 1997. (Cited on page 61.)

[107] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. (Cited on pages 35, 61 and 63.)

[108] P. Sahoo, S. Soltani, A. Wong, and Y. Chen. A survey of thresholding techniques. *CVGIP*, 41(2):233–260, February 1988. (Cited on pages 5 and 18.)

[109] Y. Shoham. Agent0: An agent-oriented programming language and its interpreter. *Journal of Object-Oriented Programming*, 8(4):19–24, 1991. (Cited on pages 47, 104 and 159.)

[110] Y. Shoham. Agent-oriented programming. *Artif. Intell.*, 60(1):51–92, 1993. (Cited on page 46.)

[111] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *Never published but presented at a talk at the Stanford Artificial Project*, 1968. (Cited on page 17.)

[112] K. Strasters and J. Gerbands. Three-dimensional image segmentation using a split, merge and group approach. *Pattern Recognition Letters*, 12(5):307–325, 1991. (Cited on pages 131 and 132.)

[113] M. C. Systems. Amira homepage: 3d data visualization. `http://www.amiravis.com//`, last accessed on :20/2/2007, 1999. (Cited on pages 87, 126 and 145.)

[114] D. Terzopoulos. Artificial life for computer graphics. *Commun. ACM*, 42(8):32–42, 1999. (Cited on page 58.)

[115] A. Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950. (Cited on page 35.)

[116] M. Vaidyanathan, L. Clarke, L. Hall, C. Heidtman, R. Velthuizen, K. Gosche, S. Phuphanich, H. Wagner, H. Greenberg, and M. Silbiger. Monitoring brain tumor response to therapy using MRI segmentation. *Magn Reson Imaging*, 15(3):323–34, 1997. (Cited on pages 2 and 12.)

[117] C. Vecchiola, A. Gozzi, M. Coccoli, and A. Boccalatte. An Agent Oriented Programming Language Targeting the Microsoft Common Language Runtime. In *The 1st International Workshop on C# and .NET Technologies on Algorithms, Computer Graphics, Visualization, Distributed, and Web Computing*, pages 93–98, Plzen, Czech Republic, 2003. (Cited on pages 105 and 160.)

[118] C. Veenman, M. Reinders, and E. Backer. A cellular coevolutionary algorithm for image segmentation. *Image Processing, IEEE Transactions on*, 12(3):304–316, 2003. (Cited on page 48.)

[119] F. Velasco and J. Marroquin. Robust parametric active contours: the sandwich snakes. *MVA*, 12(5):238–242, 2001. (Cited on page 30.)

[120] R. C. Veltkamp. Shape matching: Similarity measures and algorithms. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, page 188, Washington, DC, USA, 2001. IEEE Computer Society. (Cited on page 69.)

[121] Visiopharm. Visiopharm: Turning images into knowledge. `http://www.visiopharm.com//default.html//`, last accessed on :04/12/2008, 2008. (Cited on page 145.)

[122] G. Waiter, F. McKiddie, T. Redpath, S. Semple, and R. Trent. Determination of normal regional left ventricular function from cine-MR images using a semi-automated edge detection method. *Magn Reson Imaging*, 17(1):99–107, 1999. (Cited on page 87.)

[123] B. Wall and D. Hart. Revised radiation doses for typical X-ray examinations. *Br J Radiol*, 70:437–9, 1997. (Cited on page 80.)

[124] N. Wilke, M. Jerosch-Herold, Y. Wang, Y. Huang, B. Christensen, A. Stillman, K. Ugurbil, K. McDonald, and R. Wilson. Myocardial perfusion reserve: assessment with multisection, quantitative, first-pass MR imaging. *Radiology*, 204(2):373–384, 1997. (Cited on page 80.)

[125] D. Williams and M. Shah. A fast algorithm for active contours and cur-

vature estimation. *CVGIP*, 55(1 January):14–26, January 1992. (Cited on pages 20, 25, 94 and 114.)

[126] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995. (Cited on pages 42 and 44.)

[127] C. Xu, D. Pham, and J. Prince. Image segmentation using deformable models. In M. Sonka and J. M. Fitzpatrick, editors, *Handbook of Medical Imaging*, volume 2, chapter 3. SPIE PRESS, 2000. (Cited on pages 20, 21, 24, 26 and 31.)

[128] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998. (Cited on pages 6, 20, 26, 30, 95 and 114.)

[129] B. Yanikoglu and P. Sandon. Segmentation of off-line cursive handwriting using linear programming. *Pattern Recognition*, 31(12):1825–1833, 1998. (Cited on pages 2 and 11.)

[130] W. Yasnoff and J. Bacus. Scene-segmentation algorithm development using error measures. *Anal Quant Cytol*, 6(1):45–58, 1984. (Cited on pages 131 and 132.)

[131] W. Yasnoff, J. Mui, and J. Bacus. Error measures for scene segmentation. *Pattern Recognition*, 9(4):217–231, 1977. (Cited on page 131.)

[132] H. Zaidi. Medical image segmentation: Quo Vadis. *Computer Methods and Programs in Biomedicine*, 84(2-3):63–65, 2006. (Cited on page 4.)

[133] Y. ZHANG. A survey on evaluation methods for image segmentation. *Pattern recognition*, 29(8):1335–1346, 1996. (Cited on pages 10, 124, 125, 127, 129 and 131.)

[134] Y. Zhang. A review of recent evaluation methods for image segmentation. *Signal Processing and its Applications, Sixth International, Symposium on. 2001*, 1, 2001. (Cited on pages 127 and 128.)

[135] Y. Zhang and J. Gerbrands. On the design of test images for segmentation evaluation. *Proc. European research conferences*, 1:551–554, 1992. (Cited on page 129.)

[136] S. Zucker et al. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, 1976. (Cited on page 16.)