TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

AAM	Active Appearance Model
AOF	Average Outward Flux
ASM	Active Shape Model
CT	Computed Tomography
DBN	Deep Belief Network
DBSCAN	Density-based Spatial Clustering of Applications with Noise
EMA	Electromagnetic Articulometer
ÉTS	École de technologie supérieure
LV	Left Ventricle
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
MSD	Mean Sum of Distances
OF	Optical Flow
PCA	Principle Component Analysis
ROI	Region of Interest
SSIM	Structural Similarity Index Measure
US	Ultrasound

LISTE OF SYMBOLS AND UNITS OF MEASUREMENTS

mm	millimeters
s	seconds
kHz	kilohertz

INTRODUCTION

0.1 Motivation

The study of tongue motion has a variety of applications. It can help understand how the tongue moves in articulation and can inform different related research areas including disordered speech affected by disease, second language acquisition, speech processing, and biomechanical tongue modeling. Measuring tongue function is difficult because the tongue is positioned within the oral cavity and inaccessible to most instruments. In speech science, ultrasound (US) imaging is one of the most used techniques to measure tongue movements involved in articulation due to the fact it can capture real-time movements of the tongue surface as an image sequence and it is non-invasive.

US refers to sound waves with high frequencies that are not audible to human hearing (Fenster *et al.* (2001)). There are many technological applications of US in many different fields. US is particularly attractive for studying speech in children because it is non-invasive. There are many studies and guides regarding US and its applications in medical imaging of the tongue (Bressmann *et al.* (2007), Epstein & Stone (2005), Stone (2005)). (Stone (2005)) presents a very comprehensive survey explaining how US works, the image quality, the validity of the data collection procedures and stabilization systems.

With all the advantages of US technology, there have been many studies in the literature that explore a wide range of related subjects including tracking, modeling, analyzing the tongue contour and its surface using 2D or 3D US images.

This thesis focuses on fully automatically detecting tongue contours from US images without manual intervention by a human or use of any training data. Tracking the tongue contour in US image sequences is a difficult problem; thus, it is a popular topic in the literature and there have been many studies that suggest systems for this task (Akgul *et al.* (1998), Li *et al.*

(2005a), Roussos *et al.* (2009), Fasel & Berry (2010), Tang *et al.* (2012), Hageman *et al.* (2013), Xu *et al.* (2016a), and Laporte & Ménard (2018)). Most of these approaches depend on an initial set of tongue contour points that should be manually given in advance to the system and the tongue contour is then tracked over the remaining images. Therefore, we call these approaches "semi-automatic". This thesis addresses a different but related problem and that is to automatically segment the tongue contour points an US frame without prior information about its location, which is a challenging problem. While such a method can be used to extract tongue contours frame by frame in a video sequence, it is best exploited in combination with a tracking algorithm that exploits prior information. In this thesis, we show how this can be achieved.

The biggest advantage of the method proposed by this thesis is that it eliminates the need for manual intervention. This opens the possibility of designing and implementing software that could detect the tongue contour on a US machine in real-time and provide immediate visual biofeedback to the patient, e.g., during a speech therapy session (Bernhardt *et al.* (2005)). Moreover, the detection approach proposed here could be extended and applied to other related applications in medical imaging tasks that involve US.

0.2 Problem statement

This thesis addresses the problem of automatically detecting contours of tongue in 2-dimensional US images. To solve this problem, we divide our ultimate goal into a number of major problems that are dealt with at different levels:

- Automatically segmenting tongue contours from 2D US images: this means that the input is a single US frame and the output is an approximate locus of tongue contour points on this frame;

- Transforming a semi automatic tongue contour tracking approach to a fully automated one using the automated tongue segmentation module;
- Determining when it might be useful to re-initialize the automated tracking approach (re-initialization).

To set up a framework that addresses these problems, there are specific challenges that should be overcome. Some of these challenges include dealing with noisy US images, designing meaningful metrics to evaluate the quality of the points extracted, and deciding when to reset the system to improve the tracking accuracy.

0.3 Proposed solution

This section summarizes the proposed solution to the research problem discussed in the previous Section. We first try to address the problem of automatic tongue contour segmentation in US images. This is achieved through a number of steps described in detail in Chapter 2:

- We develop a system that captures and segments the shape of the tongue from the US image. This is done by masking the US frame, filtering it with a phase symmetry filter and adaptively binarizing the synthesized image.
- When the outline of the bright white shape arising from the echo of US off the tongue surface is captured, its skeleton, the set of points lying at an equal distance to all its boundary points, is generated. This set of points is pruned and some probable outliers are removed. Then, a spline is fit to the remaining points and these are shifted towards the tongue surface location by applying a snake fitting procedure.
- The segmented tongue contour points are evaluated by two novel quality measures. This evaluation makes it possible to select a reliable initialization set for tracking. The proposed

approach makes it possible to use any semi-automated tracking approach and make it a fully automated one.

- The final contribution of this thesis is its use of automatic resets to improve the accuracy of the tracking when the system is dealing with a long sequence of frames. This is where re-initialization would come to rescue the tracking approach and reduce the potential aggregated error. We define two criteria for the cases where the tracking is possibly lost. Whenever one of the two criteria is met the algorithm automatically resets the initialization step and tries to find a set of suitable candidate points for the tracking approach from scratch.

To validate each part of the proposed solution, we examine intermediate steps and final outcomes extensively through a number of different experiments in Chapter 3. Specifically, we focus on the accuracy of our automated segmentation algorithm and then we evaluate the outcome of using this module along with a semi-automated tracking approach.

0.4 Thesis structure

The remainder of this thesis is structured as follows: Chapter 1 reviews the essential articles in the literature related to the work presented in this thesis. These articles include background material on US of the vocal tract imaging, image segmentation using snake models, and state-of-the-art tongue contour tracking and segmentation approaches.

Chapter 2 discusses the details of the proposed approach for the problem of automatically detecting the tongue contour in US images.

Chapter 3 details the experimental framework used to test the proposed approach and the data acquired to do so. It goes over different quality and accuracy measures that were either borrowed from the literature or designed specifically to evaluate our work. The accuracy of the proposed segmentation results is presented, showing that the proposed approach can potentially be used in a semi-automated tracking system to make it a fully automated one. We then analyze the output of such an automated system in practice compared to a state-of-the-art semi-automated tracking one. Results indicate that the proposed automatic segmentation method can successfully convert a semi-automated approach to an automatic one.

Finally, Chapter 4 is dedicated to reviewing the contributions of this thesis, its shortcomings and the work that could be done in the future to improve the current framework.

CHAPTER 1

BACKGROUND AND LITERATURE REVIEW

One of the essentials tasks in the process of human tongue information analysis is the segmentation of tongue contours (Hageman *et al.* (2013), Tang *et al.* (2012), Peng *et al.* (2010),Stone (2005)). This chapter presents background material on this subject and is organized as follows: in Sections 1.1 and 1.2, we focus on the general knowledge regarding tongue , and US imaging. We then review some of the existing methods for tongue tracking in US images using snakes in Section 1.3. Section 1.4 reviews how active appearance models can help with some of the challenges that exist in snake based approaches. Section 1.5 focuses on motion retrieval from consecutive frames to track the tongue contour and Section 1.6 summarizes some of the learning based methods. In Section 1.7 we review work showing how adding temporal and spatial regularization can improve tongue segmentation. Section 1.8 reviews a recent approach to reset automatic tracking with the aim of improving performance. We introduce the main approach used in this thesis as the semi-automatic tracking system of the presented research work in Section 1.9. Finally, we go over a previous analysis of some of the tongue shape analysis techniques in Section 1.10, and we conclude the chapter by summarizing our strategy with regard to the challenges from each method introduced earlier in this chapter in Section 1.11.

1.1 Tongue anatomy

The tongue is a muscular organ in the mouth of most vertebrates that manipulates food for mastication, and is used in the act of swallowing. Besides these functionalities, the tongue enables human to produce speech and animals to vocalize (Hixon *et al.* (2014)).

The arrangement of muscle fibers within the tongue enable it to maneuver freely inside the mouth cavity. The tongue surface consists of three main parts: the tip, the body and the base or root (see Figure 1.1).

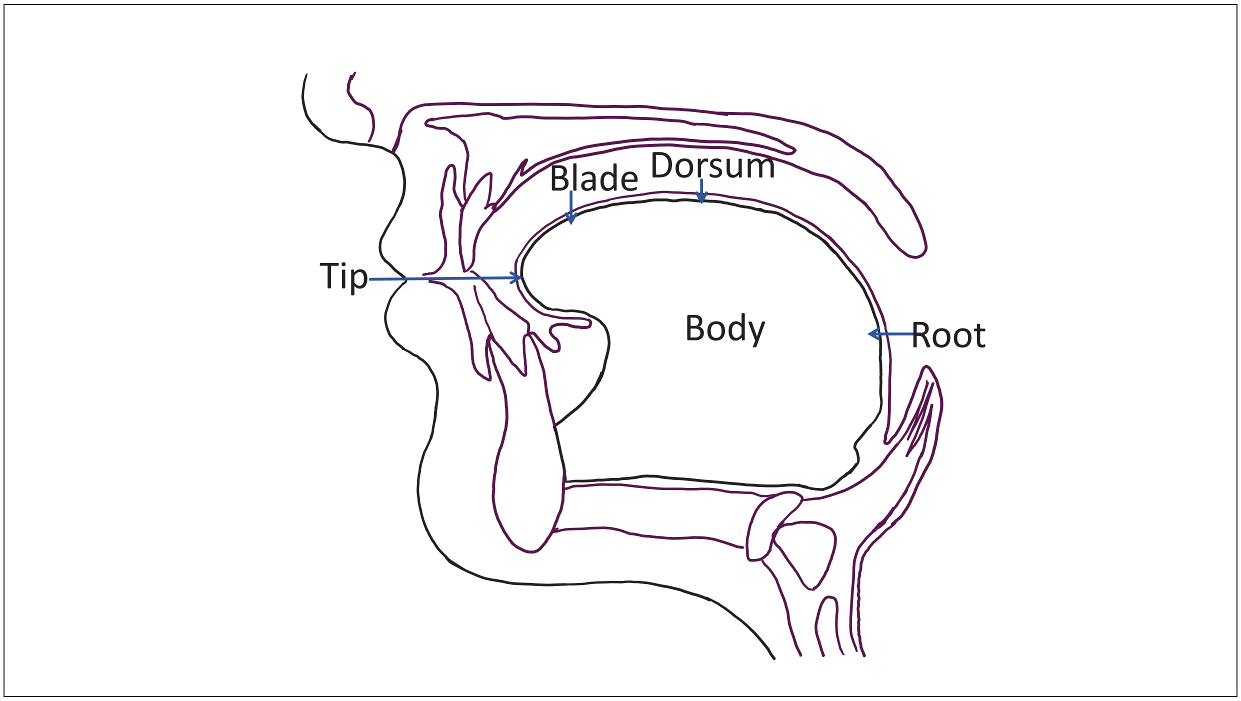


Figure 1.1 Diagram showing five components of the tongue.

The tongue muscles include eight different muscles in two groups: four intrinsic muscles and four extrinsic muscles. Those muscles that originate from the tongue body are called intrinsic muscles whereas extrinsic muscles are based outside the tongue body. Analyzing the tongue function with regard to its structure is a challenging problem since tongue anatomy is complex and interactions between muscles are not yet completely understood by the researchers in the field. In speech production, tongue shape changes and contact between the tongue and palate are used to change the flow of air more or less rapidly within the vocal tract and produce different consonants and vowels. In this thesis, we focus on the surface of the tongue and try to detect it in a 2D US midsagittal cross-section.

1.2 US imaging

Medical US imaging, also called sonography, is the act of producing pictures from the inside of body parts that are exposed to high frequency sound waves. US is basically sound waves with a higher frequency than what is audible to the human ear (higher than 20 kHz). US waves are produced by sending electric pulses to piezoelectric transducers. When an US wave passes through interfaces between different tissues, it partially reflects back to the transducers to be converted to electrical signals and the machine can determine the distance to the interface as a function of the pulse round-trip time from the time delay between the transmitted pulse and the detection of its echo and the speed of sound, assumed constant in biological tissues. The interference between the different echoes produces a “dotted” pattern best known as “speckle” in ultrasound images (Jensen (2007)). Figure 1.2 shows how the speckle appears in a sample US image and how this noisy pattern can pose a challenge to both human and machine image interpretation.

Typically, medical US imaging uses frequencies in the range of 1 to 18 MHz. As the frequency goes higher, the image obtained has a better spatial resolution, whereas lower frequencies make it possible for US to go deeper in the patient tissues. There is no use ionizing radiation in US imaging as used in X-rays. US imaging can show the movements within a particular body organ, thus it is categorized as a real-time imaging technique. Due to its various benefits such as producing no radiation, giving real-time images, being inexpensive, mobile and being widely accessible in hospitals and research centers, US imaging is one of the most frequently used medical imaging technologies worldwide.

There are a number of different techniques to assess tongue motion (Stone (1997)). From these techniques, X-ray imaging is the best known of the imaging systems. The X-ray beam captures structures such as teeth, jaw, and vertebrae in mouth besides the soft tissue and that makes it hard to use X-rays to measure the tongue. Computed Tomography (CT) is another technique that uses X-rays to image slices of the body as thin as 0.5 mm or less. Radiation exposure by CT removes this technique from the list of instruments of choice for speech research. Magnetic Resonance Imaging (MRI) produces high contrast resolution images of the tongue surface by placing electromagnets that surround the body and create a magnetic field. Though MR images show the entire vocal tract, the capturing is very slow, the sections are very wide, and it causes potential claustrophobia for subjects. Electromagnetic Articulometer (EMA) is another technique which is basically a point-tracking based method that tracks markers in the mouth using a magnetic field. EMA technique produces high temporal resolution and allows tracking well defined anatomical landmarks. EMA involves having sensors in the mouth, which is not convenient for many subjects, specifically for young patients.

Although US imaging of the tongue may not always result in high quality images, this technique is still a favorable imaging technology for speech therapy and speech science research. Using US imaging, a clinician can derive many pieces of information such as the approximate measurement of where the palate would be in relation to the tongue. Due to the fact US does not require inserting any device into the patient's mouth, it is also a prime choice for children. The biofeedback provided by real-time US visualization allows patients and clinicians to aim for a specific tongue shape they need to produce during speech therapy. These advantages of US facilitate articulatory gains in speech therapy for many patients (Bernhardt *et al.* (2005), Shawker & Sonies (1985),Bacsfalvi & Bernhardt (2011)). US imaging is one of the standard routines in the phonetics literature (Chi-Fishman (2005), Li *et al.* (2005a), Tang *et al.* (2011)), this thesis also focuses on detection of tongue contours from US images.

In speech studies, tongue US images are typically acquired as follows: a) to get a better quality image, first gel is applied on the tip on the US probe b) then the probe is fixed underneath of the subject's chin c) having a fixed probe, the US images are shown on a display (see Figure 1.2 for a sample US image).

The most significant structure of interest in US images of the tongue is the the upper surface of the tongue which is the lower edge of the bright white band that would be created by the reflection of the US at the interface between the tongue surface and air above it. Many types of noise are created in this process. Backscattered echoes from structures within the tongue (tendons, blood vessels) can cause high contrast edges. Waves being parallel to portions of tongue surface may cause the loss of tongue contour within the affected portion. Moreover, there are other possible scenarios that the US waves would not reflect properly and do not create high quality images. For example, in some people, US attenuates within the tissue located between the probe and the tongue, so the reflection off the tongue is weak. Another

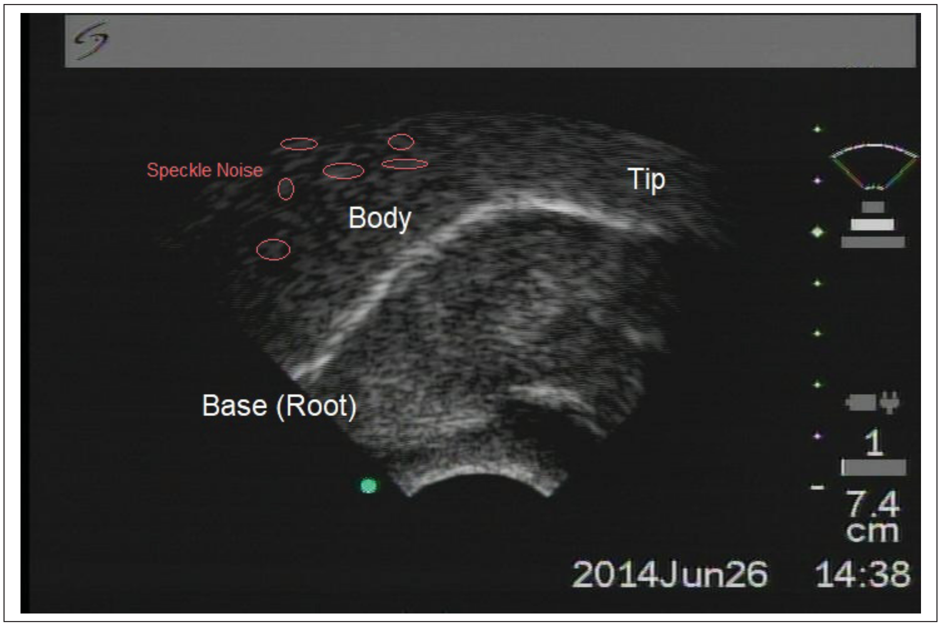


Figure 1.2 Different main parts of the tongue muscle are shown in this mid-sagittal US frame of a tongue. Note that since there are no anatomical landmarks on the US image, it is challenging to determine where the tongue tip and/or body are located at in the US image.

scenario is when the air gets trapped underneath the tongue (especially near the tip), so US does not penetrate all the way to the tongue, creating loss of signal (shadow).

1.3 Tongue tracking using snakes

1.3.1 Active contours and snakes

One of the popular classical approaches among linguistics researchers to address the problem of tongue contour segmentation from a US 2D image is to use snakes or active contour models. First introduced by Kass *et al.* (1988), snakes aim to find salient contours for delineating an object outline from a possibly noisy image. Snakes alone cannot necessarily find salient image contours. Rather, snake energy minimization methods push a suggested initial set of points toward an appropriate local minimum. Kass *et al.* (1988) suggest minimizing a functional that is composed of internal energy and external energy. Assuming the snake contour is a set of n points v_i, $i = 0 \ldots n-1$, the snake energy function can be stated as:

$$E^*_{snake} = \sum_{i=1}^{n} E_{snake}(v_i) = \sum_{i=1}^{n} \alpha E_{internal}(v_i) + \beta E_{external}(v_i) \tag{1.1}$$

where the internal energy $E_{internal}(v_i)$ is designed to impose a piecewise smoothness constraint on the snake contour, and the external energy $E_{external}(v_i)$ is composed of other forces that aim to control the fitting of the contour onto the image. Kass *et al.* (1988) suggest two external energy components: the first one is an image force which pushes the snake contour toward salient image features such as lines, edges or etc, the second one is the externally constrained forces that the user defines to interactively guide the snake towards or away from particular features.

Snakes, as a feature extraction technique, have a number of advantages over other traditional approaches in the literature such as being able to automatically adopt a minimum state and be used to track moving objects. However, snakes are sensitive to local minima and their accuracy is dependent on the convergence policy.

1.3.2 Tongue segmentation in US images using snakes

Akgul *et al.* (1998) proposed a method to segment and track the tongue surface contour in 2D US images using snakes introduced earlier in Section 1.3.1. As mentioned earlier, snake energy is formulated as a linear combination of internal energy and external energy terms. This formulation is detailed as follows: internal energy is a weighted combination of smoothness energy and similarity to the initial model energy, on the other hand, external energy is a fraction of the negative of the image gradient at each pixel (for more accurate details of the approach please see Section 2.2.1).

As US images are noisy, using the gradient of the image is not sufficient to segment the tongue surface. Therefore, the snake model of Akgul *et al.* (1998) uses manual initialization to improve tongue contour segmentation accuracy.

1.3.3 EdgeTrak

The snake formulation approach for tongue segmentation and tracking presented in Akgul *et al.* (1998) inspired many other approaches in the literature. Li *et al.* (2005a) extended this formulation to improve the segmentation and track the tongue contour more robustly in the presence of noise and spurious high-contrast edges in ultrasound images.

To understand the contribution of the method proposed by Li *et al.* (2005a), let us assume a 2D US image of a tongue contour (see Figure 1.3). The bright white band shown in this figure between the upper edge and lower edge shows the reflection at the interface with air above the tongue surface, and the lower edge (green edge) is the surface of the tongue sought for analysis by speech scientists. The external energy term of Akgul *et al.* (1998) is based purely on the image gradient information. Thus, it is a challenge to distinguish between the two edges (the upper edge and the lower edge). Li *et al.* (2005a) introduced a new energy called "band energy" to guide the snake towards the lower edge of the bright white band. With "band" energy, snake segments avoid attraction to the irrelevant high gradient above the air reflection.

Having $E_{\text{band}}(v_i, I)$ as an additional term, the authors extended the energy term in Akgul *et al.* (1998) by defining a new external energy term as:

$$E'_{ext} = E_{\text{band}}(v_i, I) \times E_{\text{ext}}(v_i, I). \tag{1.2}$$

Calling this approach "EdgeTrak", Li *et al.* (2005a) present their implemented approach as a publicly available system. Like Akgul *et al.* (1998)'s method, EdgeTrak demands the user input points near the tongue surface as an initial points in first frame of the sequence and by interpolating initial points by B-spline, the system has a contour near the tongue surface.

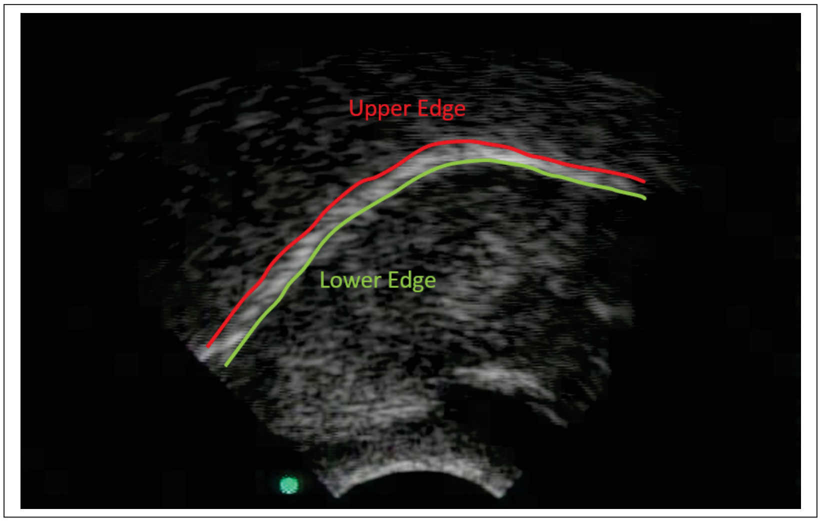


Figure 1.3 The bright white band above the tongue surface shown between the upper edge and lower edge.

1.4 Active appearance and shape models for tracking

One of the problems with EdgeTrak is that it possibly fails when some parts of the tongue from previous frames are not visible in a rapid tongue tracking task. In such cases, error can propagate and tracking cannot usually recover from that. Although preprocessing the US frames and applying boundary constraints on the snake can help the moving contour keep its size (Aron *et al.* (2008)), the constrained snake still needs manual refinements. To address this, Roussos *et al.* (2009) proposed a different tracking approach and that is to train a model with prior information about the shape variations of the tongue contour and its appearance in US images, known as active appearance models (AAMs). In this method, two models, one for shape variation of the tongue (obtained using annotated X-ray videos of the speaker's head) and one for texture model (based on the US image intensities around the tongue contour), are trained.

Besides active appearance models, active shape models (ASM) also can be used along with snakes for segmentation of structures such as the tongue. Hamarneh & Gustavsson (2000) proposed a method that combines ASM and snakes for segmenting the human left ventricle in cardiac US images. This is achieved by obtaining a shape variation model that is trained by averaging ventricle shapes and then the salient contours of ventricles are found by letting a snake that deforms to find the boundaries. This approach was successfully applied to tongue tracking by Ghrenassia *et al.* (2013).

1.5 Pairwise registration / optical-flow

Tracking the tongue contour is the same as retrieving motion under rotation and distortion conditions. Therefore, one of the simplest methods to address the problem of segmentation/-tracking is to estimate motion via a gradient based approach. Chien *et al.* (2017) present an approach to track tongue motion in ultrasound images for obstructive sleep apnea using an optical flow (OF) method by Lucas & Kanade (1984).

Chien *et al.* (2017) also suggest the strategy of iterative motion estimation, where an initial motion vector at the coarsest spatial scale is computed first and then those regions of interest are moved using that initial motion vector and after that another optical flow is calculated at a finer scale, and this is repeated until completion of all desired resolutions. Moving the ROI's at a coarser scale accelerates convergence in general and when a finer scale OF is applied the results are more accurate. Although this method is technically simple, it has two major limitations for the task of tracking a tongue contour motion. The first limitation is the heavy computations that need to be done per frame that make this approach very slow in comparison with other dynamic methods . Moreover, the errors in the method accumulate from one frame to the next.

1.6 Machine learning methods

Recent and very rapid developments in machine learning methods in the last decade have led to their equally rapid and successful application to image analysis tasks using deep neural networks. US tongue image analysis is no exception. Neural networks can work well if there are enough data they can learn from; which, in our problem, translates to having a database of segmented US images of tongue contours. There have been some works in the recent years that exploited the possibility of using deep neural networks to trace the tongue contour in US images.

Fasel & Berry (2010) presented a method based on deep belief networks (DBN) to extract tongue contours from US without any human supervision. Their approach works in a number of stages. First, a deep convolutional neural network is built and trained on concatenated sensor and label input vectors (US images and manually segmented contours). Second, the first layer of this network is modified to accept only sensor inputs (no contour information anymore). The second neural network can establish the relationship between the first neural network and the sensor-only (US) images so that the whole system can infer the labels (tongue segmentation). To minimize the reconstruction error of labels, the network is fined-tuned using a discriminative algorithm. The work by Fasel & Berry (2010) has resulted in a publicly available software called Autotrace.

The approach by Fasel & Berry (2010) makes a complex neural network model based on the tongue segmentations, which require the intensity of all pixels in the US images plus their contour segmentations as inputs. As this approach frames the tongue contour segmentation goal as a typical deep learning problem, it needs a large amount of training data to fine tune weights of 5514 neurons dispatched on 3 hidden layers. Fabre *et al.* proposed a similar methodology in line with the work presented by Fasel & Berry (2010) but with a simpler neural network. In their approach, they take advantage of a PCA-based decomposition technique called “Eigen-Tongues” which is a compact representation of raw pixels intensities of tongue US images (explained originally in Hueber *et al.* (2007)), and they also present a PCA-based model of the tongue contours which they call “EigenContours” along with a neural network that establishes a relationship between the two compact representations of the US image data and the segmented contour pixels. This method provides a simpler model than Autotrace, suggesting that fewer training data are needed for segmentation.

As manually labeling tongue contours in US images is a very time-consuming task, Jaumard-Hakoun *et al.* (2016) modified the Autotrace approach so that it works with labels extracted automatically from US images using simple image processing operations. Having an initial labeling, Jaumard-Hakoun *et al.* (2016)'s approach first pre-processes the US image with the aim of finding regions of interest (ROIs). To do the contour detection, the algorithm makes a set of candidate pixels as those ones that are white themselves and followed by a black pixel. To limit this set of candidate points, the algorithm looks back to the contour points from the previous frame and if the candidate point is in the one-pixel vicinity of ex-contour points then it is automatically labeled as a contour point. The entire set of all these candidate points are chosen as the automatically labeled image data input to the Autotrace deep neural network (in replacement of manually segmented contours). The idea of determining a contour point from a set of candidate points introduces the use of weak temporal consistency constraints in the application of training deep neural network for tongue contour detection. One of the potential weaknesses of machine learning based methods is being speaker dependent. In other words, a learned segmentation algorithm may not work on new speakers that the neural net architecture has not seen before. Another possibility is that the learned algorithm can be dependent on the imaging parameters and this makes the method incapable of generalizing to an US depth that the network has not been trained on.

1.7 Temporal consistency constraints for tongue tracking

The need for a good initialization is a must for snakes based approaches. Moreover, they might require manual refinements while the approach is performing. One possible way to reduce these types of problems is to use a database of segmented US images (when it is available) in a method that use AAMs and/or ASMs. More training data also can help designing tongue contour segmentation/tracking approaches that use deep learning neural networks. In this section, we review a semi-automatic graph based approach presented by Tang *et al.* (2012). This work reformulates the tongue contour-tracking as a graph-labeling problem where optimality of segmentations is tuned by both spatial and temporal regularizations.

As a semi-automatic approach, this method also requires that users select points in the initial frame, called control points. Tang *et al.* (2012) represent the set of segmentations for a sequence of N image frames as a graph $G(V,E)$ where each vertex $v_i \in V$ represents the control point $x_{t,i}$. t is the frame index and i is the point index in the initialization contour set of points (see Figure 1.4).

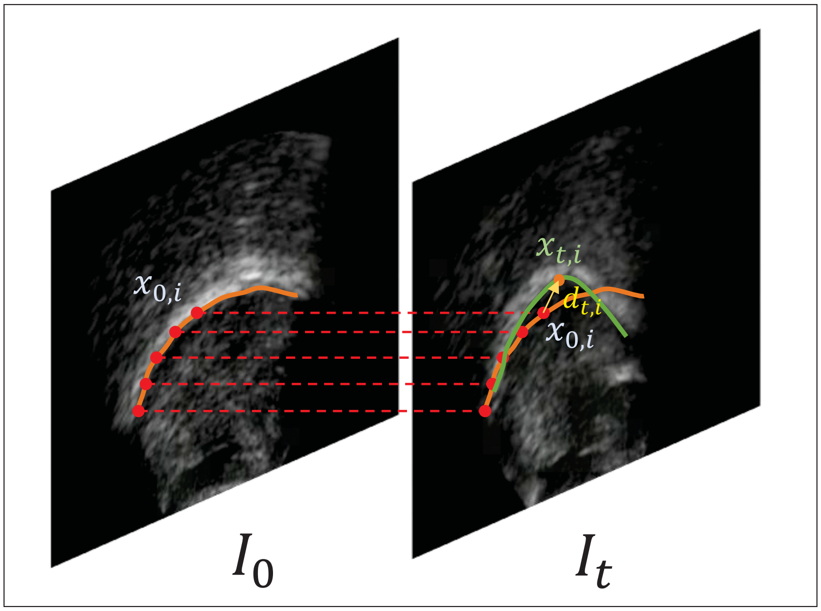


Figure 1.4 Displacement vector setup in Tang *et al.* (2012).

Given an initial set of points, the approach presented looks to find a set of displacement vectors $d_{t,i}$ for $x_{t,i}$, where

$$x_{t,i} = x_{0,i} + d_{t,i} \tag{1.3}$$

that minimize a global energy functional composed of a data energy term and two types of regularization energy terms that make sure that the algorithm tracks points that keep the entire contour smooth and continuous (spatial constraint), and also contours evolve smoothly over time (temporal constraint). Optimal segmentations computed for the set of displacements ($d_{t,i}$s) incorporating both spatial regularization and temporal regularization are obtained by the Markov Random Field (MRF) energy of the labeling on $v_i \in V$. Tang *et al.* (2012) used graph-cuts optimization algorithms to find a set of of optimal displacement vectors.

The approach presented by Tang *et al.* (2012) is presented a publicly available software called "TongueTrack".

1.8 Reinitializing trackers

One limitation to the tracking approaches is that they may drift from the correct answer and that could happen for a variety of reasons (e.g. the tongue moves too fast, it disappears or gets too blurry, etc.). Xu *et al.* (2016a) suggest a trick to reduce this effect, and that is the idea of re-initializing the tracking system from time to time.

In this approach, the authors suggest a re-initialization whenever the current image is sufficiently similar (according to the SSIM criterion) to that used for manual initialization. Thus, the tracking system is actually looking for images for which the user provided a reasonable manual segmentation, whether or not it is actually lost (see Appendix I for the description of SSIM).

In their work, they tried three tracking approaches: EdgeTrak (Li *et al.* (2005a)), TongueTrack (Tang *et al.* (2012)), and the method proposed by Xu *et al.* (2016b), and their experimental setups showed consistent improvement in terms of tracking performance. This led us to implement a similar but more flexible and less user-dependent re-initialization approach (explained in Section 2.2.3) for this thesis.

1.9 Particle filter-based tongue segmentation and tracking

Although snake based methods suffer from a number of limitations, they are still among popular tongue tracking approaches in the literature. Laporte & Ménard (2018) presented a novel approach that addresses three major limitations of the original work presented by Li *et al.* (2005a) known as EdgeTrak. These limitations include the limited capture range, producing contours that are not typically similar to contour of tongue during speech and not being able to recover successfully from errors. Laporte & Ménard (2018)'s approach introduces the use of a particle filtering algorithm as well as an ASM that enforces shape constraints to limit the search space dimensionality. A particle filter is used to enforce weak temporal consistency constraints allowing recovery from error, and this is intuitively related to (but quite different from) the re-initialization method proposed by Xu *et al.* (2016a). This method addresses the mentioned limitations of EdgeTrak and presents a concise tracking approach to segment the tongue surface in US video recordings.

As results presented by this method are quite promising and due to the immediate availability of implementation, we chose this system as our semi-automatic approach in this thesis. We discuss the details of this method in details in Section 2.2.1.

1.10 Error analysis of extracted tongue contours

A survey paper published by Csapó & Lulich (2015) discusses an experiment comparing publicly available tongue segmentation/tracking methods including EdgeTrak, Autotrace, and TongueTrack. A small set of 1,145 tongue US images were captured from four subjects (where the speakers repeat a short English sentence 8 times) and were manually segmented. In addition to comparing the automatic tracking method results to the manually segmented contours, Csapó & Lulich (2015) also considered a baseline algorithm, which is simply copying the first frame's manual segmentation to all other frames of the video sequence.

One of the findings in this experimental work was that the results generated by automatic tracking could be very dependent on what they are trained on. Particularly, Autotrace's results were dependent on whether the test images were similar to the trained images or not (results were very dependent on whether the training set and the test set contain images from the same speakers or sequences with the same phrase). Moreover, results obtained by this study show that all tracking methods have a similar pattern of error to the one produced by the baseline algorithm. This shows the difficulty of tracking the tongue contour in the case of rapid tongue movements and highlights the importance of having algorithms that could track tongue contour movements in longer sequences of US images. In the next section, we will discuss how the work presented in this thesis addresses this difficulty as well as other limitations identified in this literature review.

1.11 Conclusion

In this Chapter, we reviewed the most important approaches reported in the literature for the task of automatic tongue contour segmentation and tracking from US images. These methods can be categorized into the following two groups:

1. Detection approaches with manual initialization: this category includes approaches that need a set of initialization points to work. This is crucial for many approaches such as EdgeTrak (Li *et al.* (2005a)) or Laporte & Ménard (2018). These approaches may or may not use training information for segmentation/tracking purposes. We call these methods semi-automatic approaches in this thesis.
2. Detection approaches that do not need manual initialization: this category includes approaches that work with no need for human intervention while the algorithm is performing. The work presented by Fasel & Berry (2010) is an example of this category where large training sets play a crucial role for accuracy.

In this thesis, we propose an algorithm that addresses some of the most challenging limitations of all approaches reviewed in this chapter. We tackle the following problems :

- The need for a manual initialization
- The need for a huge training data set of manually segmented contours by fully automated methods
- The loss of tracking in rapid movement scenarios

We overcome or reduce these difficulties by proposing an approach that automatically segments tongue contours via intuitive image processing procedures, and can be combined with any semi-automatic approach to accurately track the tongue contours in US video sequences. The proposed approach also draws inspiration from the re-initialization idea proposed by Xu *et al.* (2016a) which improves results in rapid movements along image very long sequences.

CHAPTER 2

METHODOLOGY

In this chapter, we present a novel approach to automatically segment tongue contours from US images. This can be used to automatically initialize a fully automated tracker and improve the tracking using automated, timely, re-initialization. The chapter is organized as follows. Section 2.1 proposes a new approach that try to automatically segment tongue contour points from US images. In this section, we try to replace the manual initialization by automatically segmenting the tongue contour as accurately as possible. In Section 2.2, we use the automatic segmentation algorithm introduced in Section 2.1 and we suggest a new approach to make semi-automatic frameworks act like a fully-automatic tongue tracking systems that do not need any manual intervention.

2.1 Automatic tongue segmentation

In US images the echo from the tongue surface generally appears as a continuous bright region. Therefore, the core idea behind finding the tongue contour automatically is to first find that white region which we call Region of Interest (ROI) and then extract the tongue contour from that region.

Figure 2.1 illustrates the proposed approach for automatic tongue contour segmentation. First, a mask (Section 2.1.1) is applied to remove the irrelevant information that is present in the input US video sequence. Then, a phase symmetry filter (Section 2.1.2) is applied to enhance the regions that look like the tongue contour. The enhanced image is binarized (Section 2.1.3) and processed by the skeletonization module (Section 2.1.4) which produces a set of candidate points that are close to the actual tongue contour points that lie underneath the white region. To obtain a smooth connected contour for the tongue, we perform spline fitting (Section 2.1.5) using the skeletal points generated from the skeletonization module. The fitting process is fused with an outlier removal step to avoid including non-tongue contour points as much as possible.

The resulting points are processed by a snake fitting module with the aim of adjusting the contour in accordance with the actual tongue surface on the US image (Section 2.1.6).

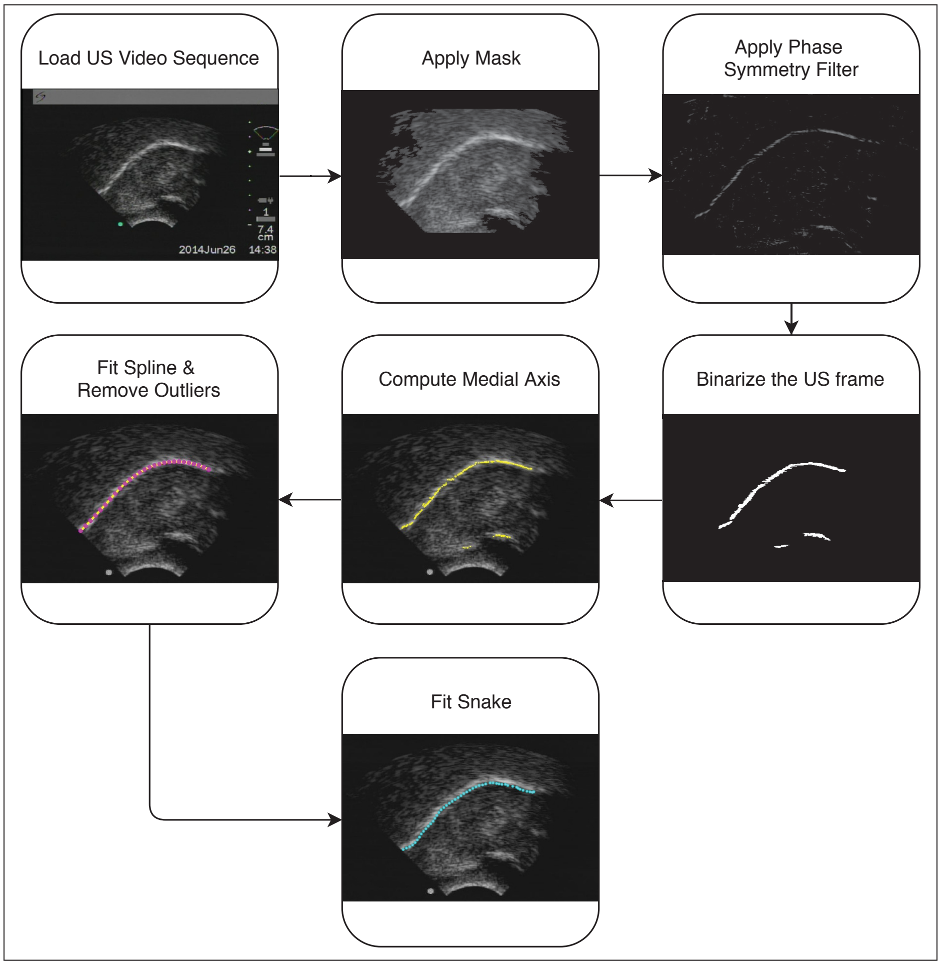


Figure 2.1 Block diagram of the automatic tongue contour segmentation method proposed by this thesis. For the remainder of this chapter, we will be using this US frame as the example image for tongue segmentation.

2.1.1 Masking

The first step towards automatic segmentation the tongue contour is to remove irrelevant information from the images by cropping the US video frame. To perform the cropping, we first find an image mask by looking at parts of the image plane where there is variation from one image to the next. By considering a small set of frames (the first 20 frames), it is easy to detect the background which should be almost the same between all images since they come from the same machine. In Figure 2.2, the middle shows a mask which is the result of considering a sequence of frames where the background consists of all pixels whose gray level intensity standard deviation over time is below 1% of the range from black to white intensities.

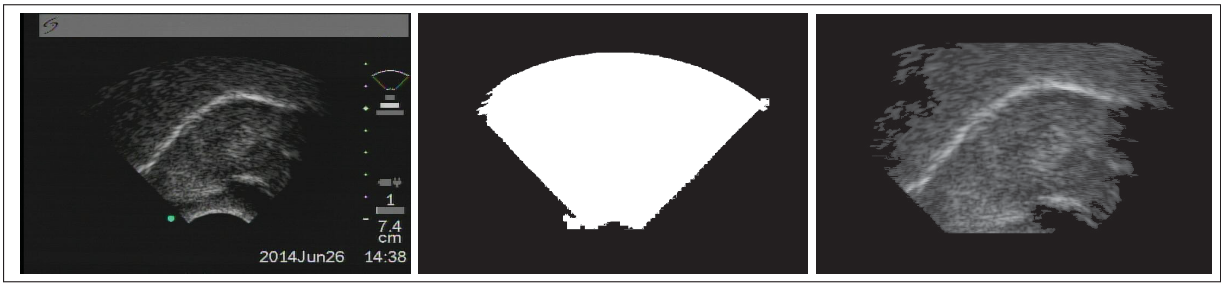


Figure 2.2 The left shows an example US frame. The middle shows the mask.

2.1.2 Phase Symmetry Filter

To enhance US images so that they emphasize the regions containing the tongue contour, we apply a ridge enhancement filter known as a phase symmetry filter (first introduced by Kovesi *et al.* (1997)) to each frame of the video sequence. Figure 2.3 shows the elevation map of the example US frame shown in Figure 2.2. The elevation map shows how the highly specular surfaces produce ridges and these are due US reflecting off the tongue. The tongue reflection in US images is generally described as a continuous narrow bright region. This type of ridge-like, thin bright strip on a dark background is precisely the kind of feature that is targeted by the phase symmetry measure, a dimensionless quantity that is invariant to changes in image brightness or contrast.

As we are interested in phase information as a method of accurate localization of the desired surfaces, Log Gabor filters are used to capture the local phase and symmetry features. In signal processing Log Gabor filters are used to describe the space and frequency characteristics of a signal simultaneously. Some of their applications in image processing are edge detection

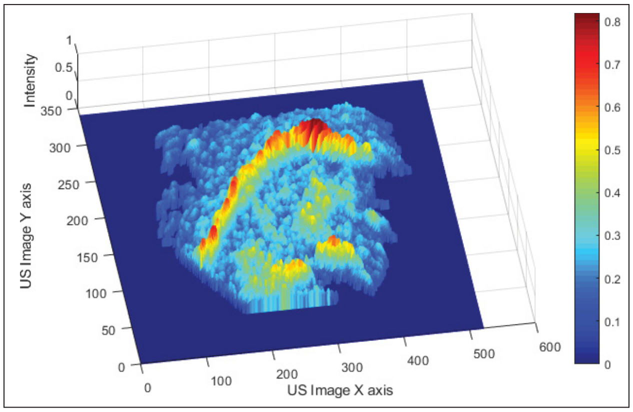


Figure 2.3 The elevation map of the masked US image from the Figure 2.1 where ridges and high peaks show the high intensity regions mostly corresponding to the tongue area.

where edges appear in the frequency domain as high frequencies and corner detection where they can be described in terms of localized frequency information by using a Log-Gabor filter. The one-dimensional Log Gabor filter introduced by Field (1987) has the frequency response:

$$G(f) = \exp\left(\frac{-\left(\log(f/f_0)\right)^2}{2\left(\log(\sigma_f/f_0)\right)^2}\right) \tag{2.1}$$

where f_0 represent the center frequency of the filter, and σ_f affects the bandwidth of the filter. In the two-dimensional case, the filter considers both a particular frequency and also a particular orientation:

$$G(f,\theta) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma_f/f_0))^2}\right)\exp\left(\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\right) \tag{2.2}$$

where θ_0 represents the center of orientation and σ_θ the width of the orientation (see Figure 2.4). Here, the orientation is a Gaussian distance function according to the angle in polar coordinates.

Log Gabor functions with sine and cosine waves each modulated by a Gaussian are good candidates to compute local frequency and, in particular, phase information in signals. Image features such as ridges can be characterized by high degree of order in frequency domain,

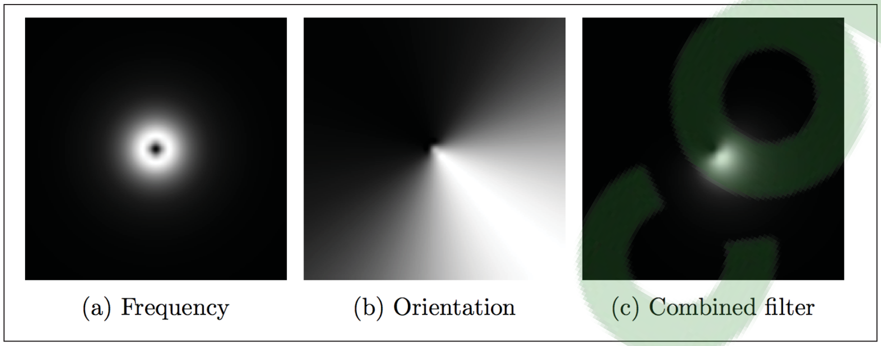


Figure 2.4 Construction of two-dimensional Log Gabor filter.

meaning that processing phase information is essential to capture such features. Image signals with even and odd symmetry will have real and imaginary Fourier transform respectively. In this work, we are interested in a ridge enhancement filter to localize our desired features from US images. To capture an axis of symmetry for a ridge like feature point, that point may result in a dominating even filter response over the odd filter response. The symmetry measure used in this work focuses purely on the local level (intensity signals in 2D) and is achieved by analyzing the local phase with values that vary linearly between 0 and π. The local phase of a given image I is computed by convolving the even (M_n^e) and odd (M_n^o) parts of the inverse Fourier transform of the frequency-based representation of the filter given in Equation 2.1 (here, we are assuming I is 1D):

$$[e_n(x), o_n(x)] = [I(x) \star M_n^e, I(x) \star M_n^o] \tag{2.3}$$

where n represents the scale(frequency) of the cosine and sine wavelets. Assuming that $e_n(x)$ and $o_n(x)$ represent the real and imaginary parts of the complex valued frequency components respectively, the amplitude of the log Gabor filter response is:

$$A_n(x) = \sqrt{e_n(x)^2 + o_n(x)^2} \tag{2.4}$$

and the phase is:

$$\phi_n(x) = atan2(e_n(x), o_n(x)). \tag{2.5}$$

The 1D phase symmetry measure is the difference between the even filter and odd filter responses. Kovesi *et al.* (1997) proposes the following equation as a local symmetry measure:

$$\begin{aligned} Sym(x) &= \frac{\sum_n \lfloor [|e_n(x)| - |o_n(x)|] - T \rfloor}{\sum_n A_n(x) + \varepsilon} \\ &= \frac{\sum_n \lfloor A_n(x)[|\cos(\Phi_n(x))| - |\sin(\Phi_n(x))|] - T \rfloor}{\sum_n A_n(x) + \varepsilon} \end{aligned} \quad . \qquad (2.6)$$

The factor T is a noise compensation term and ε is a small constant so that the denominator will not be equal to zero. The measure of symmetry introduced in Kovesi *et al.* (1997) is related to the phase congruency model of feature perception where one could interpret symmetry as a delta feature extractor (see Figure 2.5) meaning that this would provide us with a ridge enhancement filter. This 1D analysis can be extended to 2D by applying it in multiple orientations and forming a weighted sum of the results.

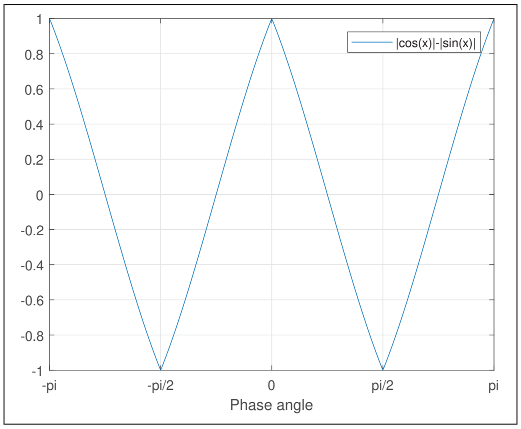


Figure 2.5 Plot of the symmetry measure $|\cos(x)| - |\sin(x)|$, where $x =$ Phase angle. A delta feature starts off having all frequency components aligned in phase and in symmetry.

In the work presented here, we use the Matlab implementation of phase symmetry developed by Peter Kovesi (http://www.peterkovesi.com/matlabfns/#phasecong). As for the parameters used in this package, we empirically tuned the number of wavelet scales (n) and the number of filter orientations (τ). In our implementation, we chose $n = 5$ and $\tau = 14$ empirically as well suited to our experiments.

By applying the phase symmetry filter to US images we see that it is a good candidate for the specific task of enhancing the ridges in the US image in comparison to other traditional image processing tools such as the Canny edge detector (see Figure 2.6).

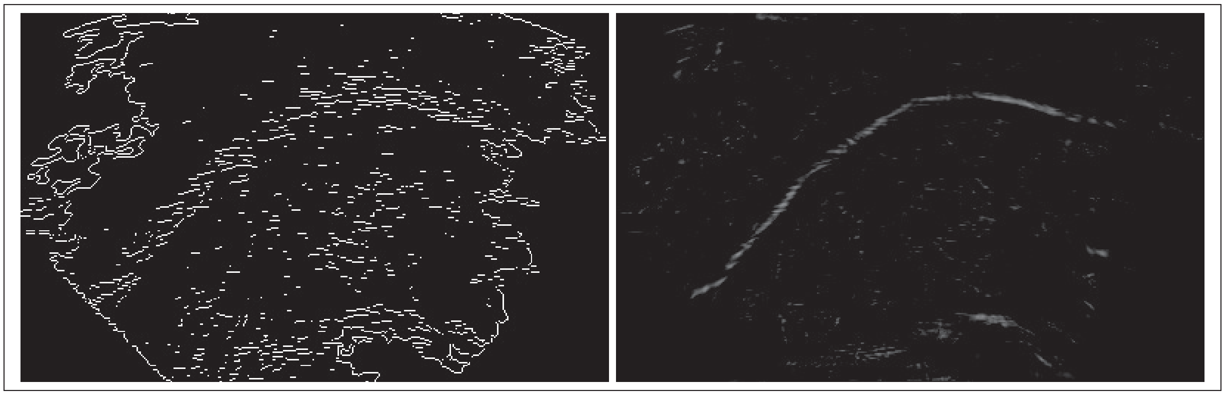


Figure 2.6 Left: shows the result of applying the Canny edge detector on the example US image. The green line shows where the surface of tongue is located. Right: shows the result of applying the phase symmetry filter on the same US image. It can be seen that phase symmetry filter is better at ignoring this speckle noise than the Canny edge detector.

2.1.3 Binarizing the Ultrasound Image

As we are interested in bright regions that include tongue contour points, we set our next step to produce a binary image of the frame in which a phase symmetry filter has been applied. This is done through an adaptive thresholding procedure and that aims to identify those white regions that either include or are close to the tongue contour points; we call them regions of interest (ROI) in this work.

To find ROIs, we first binarize the filtered image from previous step using a threshold that is chosen as the median of all intensity values in the filtered image (see Figure 2.7). To express this mathematically, let us assume that I represents the input US image (masked and cropped), I_f represents the phase symmetry filter output, and $\lambda = \text{median}(I_f)$. The binarized image ($I_b$)

in the Figure 2.7 (left) is obtained by a simple thresholding as it follows:

$$I_b(i,j) = \begin{cases} 0, & I_f(i,j) <= \lambda \\ 1, & I_f(i,j) > \lambda \end{cases} \tag{2.7}$$

We also consider another image (I_c), that is similar to I_b except that the white regions in I_b now get their pixel intensities from the original US image I

$$I_c(i,j) = 1 - I_b(i,j) + I(i,j)I_b(i,j) \tag{2.8}$$

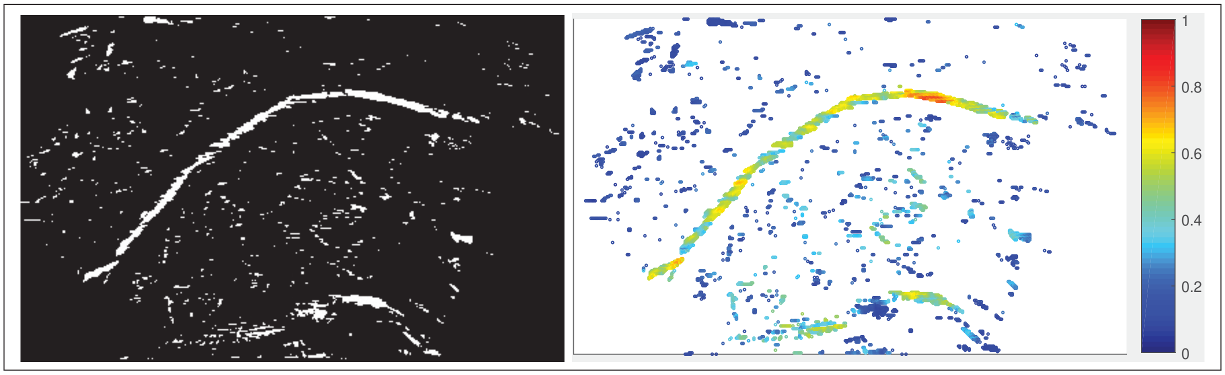


Figure 2.7 Left: The initial result of binarization with the threshold chosen as the median of all intensities in the filtered image obtained from the phase symmetry module (I_b). Right: The white region pixels of the obtained binary image (left) are colored based upon the intensity values of the same pixels in the original US image normalized between 0 to 1 (I_c).

Let W_k represent the k^{th} white connected component in I_b. An importance score is defined as:

$$\Psi(W_k) = \text{mean}(I_c(W_k)) \times \text{area}(W_k), \tag{2.9}$$

where $\text{mean}(I_c(W_k))$ represent the average intensities of all the pixels of I_c within W_k and $\text{area}(W_k)$ represents the area of the connected component W_k. Now, let us define a new image I_d as:

$$I_d(i,j) = \begin{cases} \Psi(W_t), & (i,j) \in W_t \\ 1, & (i,j) \notin W_k \text{ for } \forall k \end{cases} \tag{2.10}$$

I_d emphasizes the regions of the US image that have high average intensities as well as a bigger area. Combining ROI size with ROI average intensity makes it easier to eliminate small white regions that are produced by speckle noise. A color-coded version of I_d is shown in Figure 2.8 (left). This example shows that most noise regions are associated with low scores when in this scoring scheme.

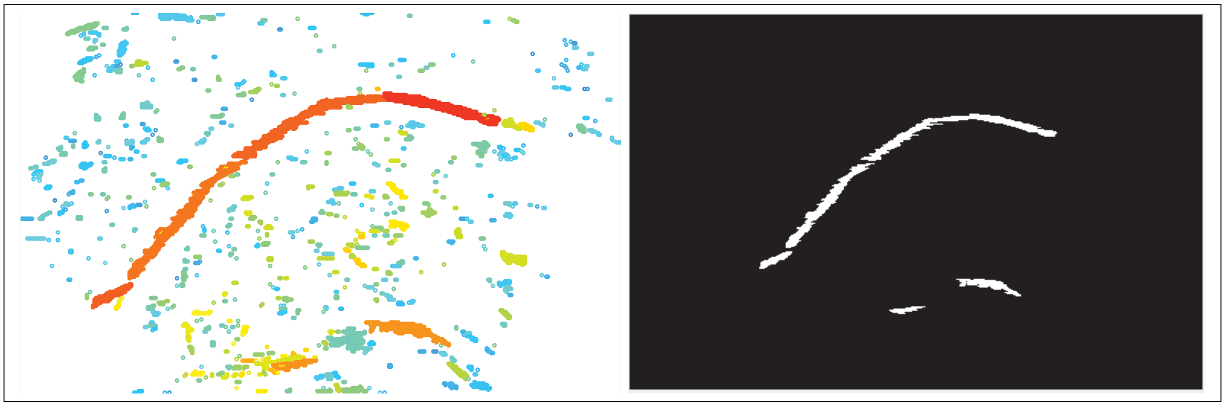


Figure 2.8 Left: shows the white regions rank ordered and colored based on their importance score $\Psi(W_k)$. Intensities are colored from blue (for low importance) to red (for high importance). Right: shows the result of binarizing the I_d image using Otsu's method.

Finally, we apply Otsu's thresholding method (Otsu (1979)) to binarize I_d, which applies the threshold that minimizes the intra-class intensity variance. We use the default Matlab implementation of Otsu's method and show the result of binarization in Figure 2.8 (right).

2.1.4 Computing the Medial Axis

After the binarization step is performed, we have some ROIs (see Figure 2.8 right) that are potentially close to the tongue surface. Our main goal is to extract a single curve representing the tongue contour. Therefore, we use skeletons (medial axes) in this thesis. In the following, we use the terms “medial axis” and “skeleton interchangeably”. The medial axis of a shape was first introduced by Blum (1967) as the locus of all points lying inside the shape and having more than one closest point to the boundary of that shape. The medial axis is a powerful shape descriptor and it is used in this thesis to simplify the representation of ROIs from regions with some width to scattered points that are close to the actual tongue contour points. In this work, we selected the flux skeleton approach since this medial representation is robust to noise in the shape boundary. Flux skeletons were introduced by Dimitrov *et al.* (2003) and have been improved in different applications (Rezanejad & Siddiqi (2013), Rezanejad *et al.* (2015)). In our implementation we used the package developed by Rezanejad *et al.* (2015). We will review the geometry of flux skeletons in the following.

To compute the medial axis within a bounded shape, Dimitrov *et al.* (2003) introduced a new measure called Average Outward Flux (**AOF**). **AOF** is defined as outward flux of the gradient of the Euclidean distance map to the boundary of a 2D shape through a shrinking disk normalized by the perimeter of that disk. To elaborate, assume an arbitrary region R with a closed boundary curve denoted ∂R. If the gradient of the Euclidean distance function to ∂R is given by $\dot{\mathbf{q}}$, the **AOF** through ∂R is then defined as

$$\mathbf{AOF} = \frac{\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds}{\int_{\partial R} ds}, \tag{2.11}$$

where s is the arc length along a branch of the medial axis and $\mathbf{N}$ represents the outward normal at each point on the boundary ∂R.

Using the divergence theorem, Dimitrov *et al.* (2003) show that the **AOF** takes non-zero values for skeletal points and zero values everywhere else, when it is computed on a shrinking disk whose radius tends towards zero. Knowing this, finding skeletal points can be simplified as finding non-zero values on an **AOF** map. Since the tongue ROIs are typically narrow, a jittering effect is present in the binarized pixels and could easily lead to inaccurate medial axes (Xie *et al.* (2010)). A major advantage of the flux-based method is that **AOF** is a region-based

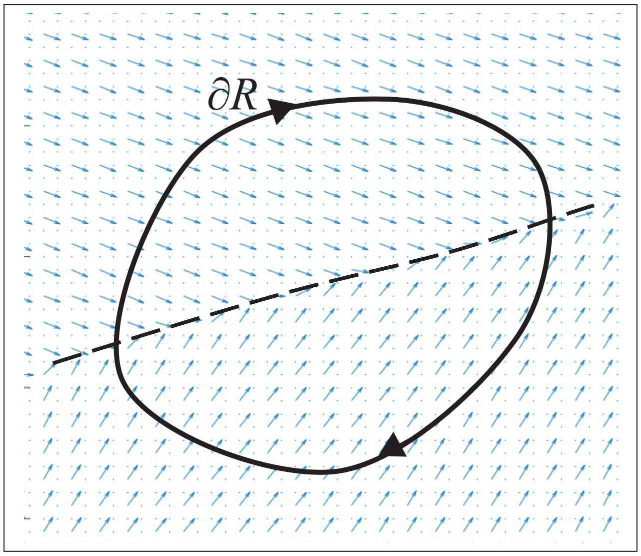


Figure 2.9 Arbitrary region R including a branch segment of the skeleton (shown in dashed lines). The boundary of the region is represented by ∂R and the blue quiver plot represents the gradient of the Euclidean distance function to the boundary of a 2D shape, represented as $\dot{\mathbf{q}}$.

measure (see Equation 2.11) and is very stable with respect to the noise or perturbations of the boundary of ROIs. Therefore, the computed skeleton is very robust to the aforementioned jittering effect. Figure 2.10 shows the average outward flux map (left image) and the skeletal points computed for the binarized region of interest from previous step (right image).

2.1.5 Spline Fitting and Outlier Removal

Skeletonization produces a set of skeletal points that can be used to fit a representative curve for the tongue contour. In this thesis, we use the formal B-Spline function which is a generalization of Bezier curves, and creates a smooth curve that goes through a set of 39 control points that are sub-sampled from the set of skeletal points. In the case where the skeleton has less than 39 points, the system will automatically up-sample the remaining number of points by linearly interpolating between skeletal points.

Unfortunately, not all points on the medial axes of ROIs are located near the tongue contour (see Figure 2.10 right), and we have to somehow take care of those outliers. To designate can-

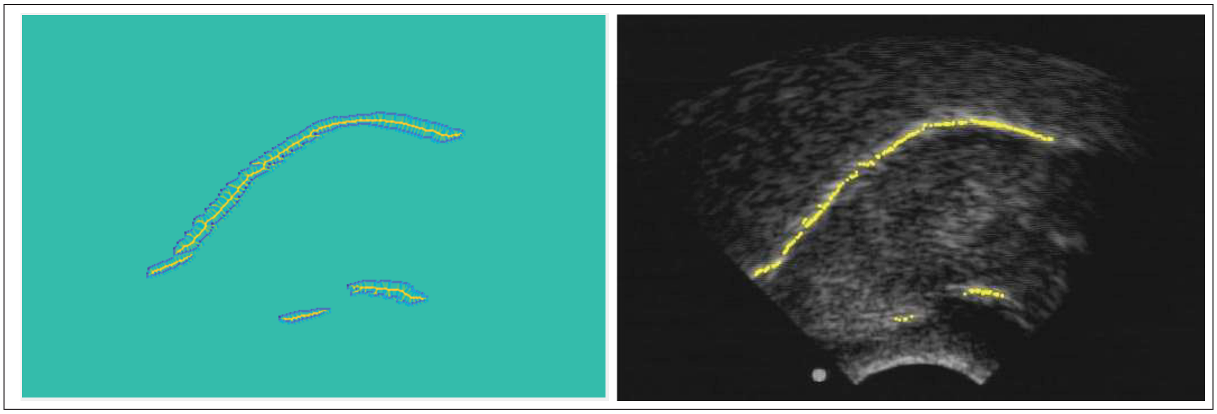


Figure 2.10 Left: the average outward flux map applied to our binarized example from previous step. Here, blue shows the boundary of the ROIs, yellow shows the high values of AOF. Right: the skeletal points obtained from the AOF map overlayed on the input US image. This figure shows an example where accidental white regions that appear in US are picked as candidate ROIs and have generated outlier skeletal points and how they differ from of the points are close to the legitimate tongue contour.

didate points as being close to the tongue contour we use a spline fitting algorithm that handles outliers. We use the Density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm, proposed by Ester *et al.* (1996), to handle outliers generated from the remaining small connected components (ROIs) that have not been removed by the thresholding step of Section 2.1.3. DBSCAN is a clustering algorithm that works with spatial data and rather than having a fixed number of classes it divides the data in different clusters based on their distance (ε - the maximum distance between points) from each other and a minimum number of points (**MinPts**) within each cluster. We set the $\varepsilon = 20$ pixels and **MinPts** = 10 in our implementation.

When the clustering is done (see Figure 2.11), the largest cluster is taken to contain the tongue's reflection and the remaining smaller clusters are assumed to contain outliers. The next step is to fit a b-spline curve (as mentioned above) to the main cluster to produce candidate points for initialization of the automatic tongue tracking system. In this work we used a Matlab B-spline fitting package freely available on-line (https://www.mathworks.com/matlabcentral/fileexchange/13812-splinefit) (see Figure 2.12).

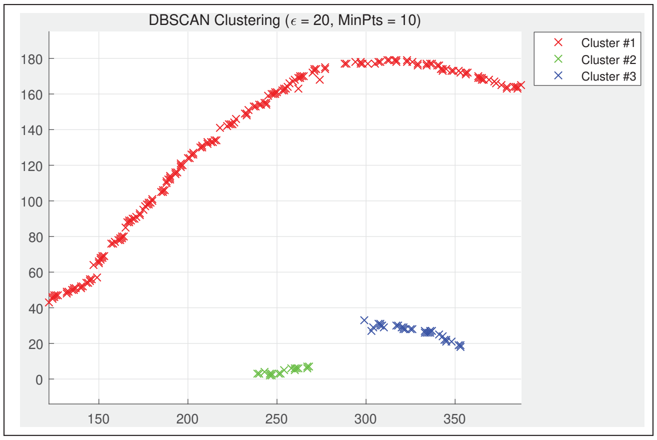


Figure 2.11 Example of how the DBSCAN clustering algorithm would apply to the generated skeletal points.

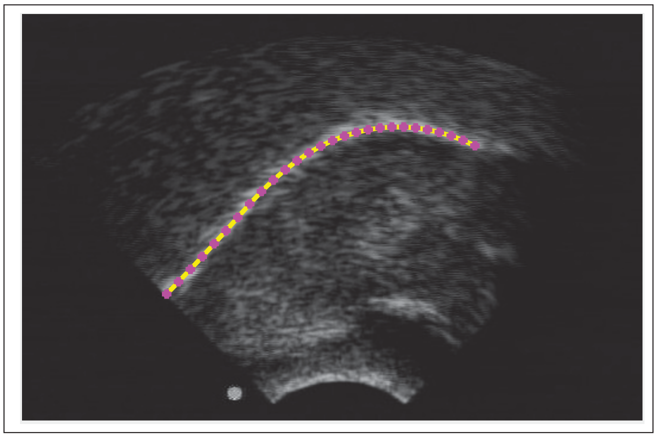


Figure 2.12 The result of spline fitting and outlier removal steps on the US example. The continuous yellow curve shows the resulting spline fit where outliers are removed, and the pink circle dots show the sampled points on the spline fit we use to fit a snake in the next step.

2.1.6 Snake Fitting

The final step of our proposed automatic segmentation of tongue contour from an US image is to fit an active contour model (snake) to the points obtained from the spline fitting/outlier removal module. This is done to allow the extracted points to adjust themselves according to the actual tongue contour points. The snake model is presented as an energy minimizing deformable spline constrained by two energy functions. The first function is an internal energy measure which characterizes the rigidity and complexity of the contour shape and the second one is external energy measure that describes how well the snake latches to structures present in the image. In our framework, we use the approach of Li *et al.* (2005a) to fit a snake to each of the splines obtained in the previous step. Given a contour $V = \{v_1, v_2, ..., v_n\}$ where the v_i, $i = 1, \ldots, n$ are the points generated by the spline fitting and outlier removal module, the total energy to be minimized is defined as:

$$E'_{\text{snake}} = \sum_{i=1}^{n} \alpha E_{\text{int}}(v_i) + \beta E_{\text{gradient}}(v_i) E_{\text{band}}(v_i). \tag{2.12}$$

Equation 2.12 can be minimized using the dynamic programming approach proposed by Amini *et al.* (1990).

The first energy function is E_{int} which represents internal energy functional that encodes local constraints on the curvature and stiffness of the snake:

$$E_{\text{int}}(v_i) = \lambda_1 \left(1 - \frac{\overrightarrow{v_{i-1}v_i}.\overrightarrow{v_i v_{i+1}}}{|\overrightarrow{v_{i-1}v_i}|.|\overrightarrow{v_i v_{i+1}}|}\right) + \lambda_2 \frac{||v_i - v_{i-1}| - d|}{d} \tag{2.13}$$

where λ_1 and λ_2 are weighting parameters and d is the average length between two consecutive snake points.

The second energy measure E_{gradient} helps move the contour towards regions of high image gradient in the US image:

$$E_{\text{gradient}}(v_i, I) = 1 - \frac{||\nabla I(v_i)||}{C} \tag{2.14}$$

where I is the input US image, and C is a normalization factor. Based on the implementation proposed by Laporte & Ménard (2018), $C = \max_{v_i} ||\nabla I(v_i)||$.

The third energy function is E_{band} which measures the contrast between the bright region above the contour and the region immediately below it:

$$E_{\text{band}}(v_i, I) = \begin{cases} E_{\text{penalty}}, & \text{if contrast}(v_i, I) < 0 \\ 1 - \text{contrast}(v_i, I), & \text{otherwise} \end{cases} \tag{2.15}$$

where E_{penalty} is a constant penalty factor and contrast(v_i, I) is the local image contrast at the boundary defined by the snake at vertices v_i and v_{i+1}. (see Figure 2.13).

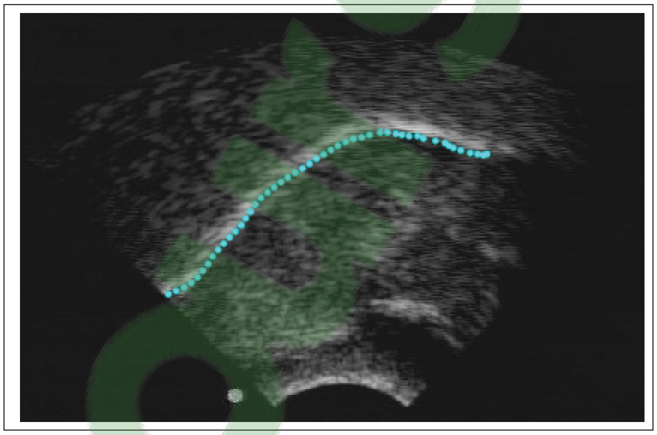


Figure 2.13 The result of the snake fitting step on the US example. The blue curve shows the result of snake fit on sampled points from the spline fit.

2.2 Applications to tongue tracking

In this section, we discuss how to use the automatic tongue detection method discussed in Section 2.1 to improve an existing tracking framework. First, an existing tracking framework (Laporte & Ménard (2018) is reviewed. Then, in the following two sections, this chapter explains how to use the proposed method within this framework to (a) initialize the tracker and (b) re-initialize it from time to time. A fully automated framework needs two parts: 1) a semi-automatic tongue tracking framework; in this work, we chose the approach proposed by Laporte & Ménard (2018); 2) a module that can automatically initialize a set of candidate points for the semi-automatic system; this set of points should be chosen strategically so the system would have the best initial points. In this work, we added a third part to improve accuracy and reduce the amount of manual intervention required to correct segmentations afterwards and that is a re-initialization strategy that automatically resets the tracker at strategically chosen moments to improve accuracy. The re-initialization should be done automatically as well and should not require any manual intervention. In the following, we will discuss how each of these three parts are implemented in this work.

2.2.1 Semi-Automatic Tongue Tracking Framework

To evaluate the usefulness of our approach, we apply it to the multi-hypothesis framework of Laporte & Ménard (2018) for tongue tracking. In this section, we explain how we track the tongue contour given a set of candidate points automatically initialized. The algorithm used here is based on the combination of Snakes (Kass *et al.* (1988)), Active Shape Models (Cootes *et al.* (1995)) and Particle Filtering (Arulampalam *et al.* (2002)).

Firstly, an active shape model (ASM) is built based on a data set of segmented tongue contours that each contain n vertices. Then, the coordinates of these contour vertices are normalized with respect to contour position and length. By applying principal component analysis (PCA) on normalized vertex sets, the contour shape can now be represented in each frame by a summarized compact vector of 6 variables (x, y, s, w_1, w_2, w_3), where (x, y) represents the location, s is the ratio of the current tongue contour length to that measured in the initial contour, and (w_1, w_2, w_3) represent the weights of the first three principal components of the active shape model built by PCA. Note that the first three modes of variation account for 98% of the observed variance.

Secondly, a multivariate Gaussian state transition model that can predict a variety of possible tongue states is built for the sampling procedure of the particle filtering algorithm. This is achieved by generating a 6×6 covariance matrix Σ which is based on differences between consecutive state vectors, which represent motion between two frames.

To track the tongue contour at each time step, first, each particle is fitted as a snake to the image by minimizing the simplified snake energy:

$$E_{\text{snake}} = \sum_{i=1}^{n} \alpha E_{\text{int}}(v_i) + \beta E_{\text{gradient}}(v_i) \tag{2.16}$$

Once this is done, the likelihood of each particle is established using:

$$E'_{\text{snake}} = \sum_{i=1}^{n} \alpha E_{\text{int}}(v_i) + \beta E_{\text{gradient}}(v_i) E_{\text{band}}(v_i) \tag{2.17}$$

which is a more robust energy functional and discounts high image gradients that are unrelated to tongue contour. Equation 2.16 is used for particle *optimization* since it is faster to compute, and Equation 2.17 is used for particle *weight computation*. The likelihood of a particle is used to select the best solution for the current frame and re-sample new particles from the current set with replacement. This measure is negatively related to snake energy, therefore, the likelihood of each particle is set as: $L = \exp(-E'_{\text{snake}})$, and at each step all likelihoods are normalized by the sum of all likelihoods so the sum of all particle weights is equal to 1.

At each step, an adaptive number of particles is generated (potential contours for the next frame). The approach proposed by Laporte & Ménard (2018) chooses the number of particles in a way that controls the trade-off between accuracy and computation time. The number of particles is chosen adaptively at every frame, and allows the cumulative likelihood of the evaluated particles reach a certain threshold T defined as:

$$T = 7 \times \exp(-E(V_{init}, I_{init})) \tag{2.18}$$

where $-E(V_{init}, I_{init})$ is the energy of the manually-segmented contour in the initialization frame. In this formulation, 7 is an empirically chosen factor and the number of particles is set to have a minimum limit of 10 and maximum limit of 1000.

2.2.2 Automatically Finding Candidate Initial Points Within a Window of X Frames

The automatic tongue segmentation process described in Section 2.1 provides a set of candidate tongue contour points for any input image. The set of candidate points can ultimately be used to initialize a tracking approach without manual intervention. When segmentation is needed after acquiring a recording, initialization does not need to take place in the first frame of the sequence; rather, any frame can be used as a starting point. Therefore, we developed two quality measures that are predictive of the reliability of our segmentation method so that we can automatically choose an initial frame where we are most confident that segmentation will work. First of all let us assume that we have performed the steps of segmentation from masking to computing medial axis points and we have a set of points represented as $V = (v_1, ..., v_n)$, where skeletal points v_i, $i = 1, \ldots, n$ are sorted by position from left to right on the US image. Spline fitting, outlier removal and snake fitting steps are all designed with the aim of perfecting the set of points found from skeletonization.

We suggest two assessment criteria that are inspired by prior knowledge about the general shape of a tongue and specifically what it looks like when captured by an US machine.

The first score reflects the fact that points that represent the tongue should not be from very disjoint groups of ROIs, that is simply due to the shape of tongue contour, which is a continuous, smooth curve. This leads to the first reliability measure as the inverse of total contour length:

$$\Gamma_1 = \left(\sum_{i=1}^{n-1} ||\overrightarrow{v_i v_{i+1}}|| \right)^{-1} \tag{2.19}$$

A decrease in the sum of the distances between pairs of consecutive points would lead to an increase in the consistency of points, therefore a higher Γ_1 means a better set of candidate points. Poor consistency means that points generated from our approach are more disjoint from each other (there are gaps in the contour) and therefore either they are not representing the entirety of the tongue or some of the points are generated from ROIs are close to tongue (outlying bright regions). In both cases, a lower score would imply the inadequacy of the considered set of points.

The first score would be quite high if only a small segment of the tongue (e.g. the middle) had been segmented. To address this, we use the help of a second score that would prevent selection of such a frame. The second score suggested here deals with another property of the shape of the tongue, and that has to do with the length of the tongue. In the perfect scenario,the sum of distances between contour points should be equal to the completeness of the tongue. The second reliability measure assesses the completeness of the tongue contour. As the tongue contour typically occupies a wider range of positions along the x axis, the second score was designed as the ratio of the coverage length of the candidate points on the x-axis to the image width:

$$\Gamma_2 = \sum_{i=1}^{n-1} d_{v_i v_{i+1}} \cos\angle(\overrightarrow{v_i v_{i+1}}, \overrightarrow{x}) \tag{2.20}$$

where:

$$d_{v_i v_{i+1}} = \begin{cases} ||\overrightarrow{v_i v_{i+1}}||, & \text{if } ||\overrightarrow{v_i v_{i+1}}|| <= 2\sqrt{2} \\ 0, & \text{otherwise} \end{cases} \tag{2.21}$$

and $\overrightarrow{x}$ represent the x-axis. In the second measure, we are considering the skeletal points found as a graph where each edge is connected if the two consecutive neighboring vertices have a distance less than $2\sqrt{2}$ pixels (here we are assuming there shall not be more than one pixel distance between connected points assuming the points are placed in discretized pixel world). We should note that this score is measured before the sub-sampling procedure which happens in spline fitting and outlier removal step. Γ_2 is defined as the sum of the projections of these edges on the x-axis.

The final score is computed based on the combination of these two scores:

$$\Gamma = \Gamma_1^{\eta_1} \Gamma_2^{\eta_2} \tag{2.22}$$

where η_1 and η_2 are chosen empirically.

Within a window of X frames from the starting frame, we compute Γ for each frame and we choose the frame with the highest segmentation reliability score as the initial frame and the candidate points extracted automatically from that frame as a replacement to manually segmented points used by the semi-automatic tongue tracking framework described in section 2.2.1. As the number (X) of frames within the time window increases, we might end up extracting a better starting frame, but this would happen at the cost of time. In our experiments we set $X = 10$.

2.2.3 Re-Initialization

Any tongue contour tracker may temporarily or permanently lose the trajectory and fail due to a variety of reasons. Low signal to noise ratio is a normal drawback of US imaging and segmenting a specific structure from an US frame cluttered with speckle noise is difficult. Sometimes the tongue just moves too fast for the tracking algorithm to find it. These are examples of cases where a tracked tongue contour may drift away from the actual surface of the tongue.

In a typical US video of a tongue, we can see frames where the tongue trajectory is not even easily detectable by a trained professional and such a case could result in a loss of tracking of the tongue contour in any tracking framework. Inspired by Xu *et al.* (2016a), to overcome the challenge of loss in tracking, our framework should be able to automatically re-initialize tracking from time to time. Thus, we designed our system such that it can find where a reset could be helpful.

The criteria used to do these resets look for two types of situations. The basic principle behind the first criterion is to look for image similarity and as suggested by Xu *et al.* (2016a), we used Structural Similarity index measure (SSIM). To implement the re-initialization step (see Algorithm 2.1), we use structural similarity (SSIM) algorithm to compare the current frame and last chosen initial frame (see Appendix I). By considering the similarity value from this comparison, we can make a choice between re-initialization or continuing with tracking. As for implementation, the ready-to use SSIM function from Matlab was used (https://www.mathworks.com/help/images/ref/ssim.html). The second criterion is to do the automatic reset when the number of particles from the semi-automatic tracker (Laporte & Ménard (2018)) gets bigger than a particular threshold (which has been chosen empirically - see Algorithm 2.1). The number of particles, on the other hand, tells us about how hard the particle filter is working, and how uncertain it is about its own conclusions. This is when we are looking for situations where we are lost.

Algorithm 2.1 Re-Initialization algorithm

```
Input: Unlabeled ultrasound frames $F = \{f_1, \dots f_N\}$, reset window size $W$, total
  number of frame in the video sequence $N$
Output: Tongue contour segmentation labels $L = \{l_1, \dots l_N\}$
Mode ← Re-initialization
$i \leftarrow 1$
while $i \leq N$ do
    if $Mode(i) = Re\text{-}initialization$ then
        best-init-frame ← Do-Automatic-initialization($f_i$,...,$f_{i+W-1}$)
        $l_i$ = Do-Tracking(best-init-frame)
        Mode ← Particle-filter-tracking
        $i \leftarrow i + W - 1$
    end
    else
        $l_{i+1}$ = track-one-frame-ahead($f_{i+1}$)
        $i \leftarrow i + 1$
        if SSIM($f_t$,$f_{i+1}$) $\leq \tau_1$ or $nOfParticles \geq \tau_2$ then
            Mode ← Re-initialization
        end
    end
end
In our experiments, we set thresholds $\tau_1 = 0.9$ and $\tau_2 = 400$, and they are empirically
  chosen.
```

Every time that either of these criteria are met, the semi-automatic module is paused and the automatic tongue segmentation is performed anew according to the procedure we discussed in Section 2.2.2. Algorithm 2.1 shows how to combine the thresholds for each of these two criteria.

2.3 Summary

The automatic tongue segmentation method proposed in Section 2.1 is a novel approach to find potential regions of tongue contour in US images and then extract a set of suitable tongue contour points from that without any manual intervention or training data. We showed how this automatic segmentation could help us to turn any semi-automatic tracking approach that needs manual initialization into a fully automated method in Section 2.2. Finally, in Section 2.2.3, we proposed a novel re-initialization approach to improve the accuracy of tracking. In the next chapter, we report results of experiments using these novel methods on real speech US video sequences and how they can be used to build an effective automated tongue detection system.

CHAPTER 3

EXPERIMENTS

This chapter discusses the experimental setups proposed in this thesis along with the results of the segmentation method described in chapter 2 tested on real US data.

Section 3.1 explains how data acquisition is performed. Section 3.2 then presents evaluation measures. Using these measures, we then evaluate the segmentation approach proposed in this thesis and its usefulness in a tracking context in section 3.3. Section 3.4 shows some sample tongue detection results on US frames. In section 3.5, we analyze the reliability scores we defined in section 2.2.2. Finally, in section 3.6, we show how the re-initialization module would affect the tracking results.

Note that all the experiments presented here were done on a PC with the Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz processor, 8.00GB of installed memory (RAM) on Microsoft Windows 10 Pro, and Matlab 2018a.

3.1 Data acquisition

For our experiments, we used the same US video sequences that were presented in Laporte & Ménard (2018). The machine used for recording is a Sonosite 180 plus US scanner with a micro-convex 8-5 MHz transducer set at a 84 degree field of view. To stabilize the probe, an elastic band was attached to the probe and to a helmet on the subject's head. After recording, the US video sequences were manually segmented by a trained operator using an interface developed by Fasel & Berry (2010) (Source code is available for download: https://github.com/jjberry/Autotrace). In our experiments, we used 16 free speech US video segments from Laporte & Ménard (2018). Each segment was between 20 seconds and 84 seconds long.

The subjects were 12 adolescent speakers of Canadian French aged from 10 to 14 years old. Out of these 12 subjects, 7 suffered from Steinert's disease (denoted SX or SX_Y where X represent the video segment number) and 5 were healthy subjects (denoted CX - or Control group). Subjects were given time to talk freely about their favorite movies or their personal experience at school. The setup was tuned and adjusted before each recording, to ensure optimal image quality for the videos. This led to different imaging depths depending on the subject.

Table 3.1 summarizes the characteristics of each recording.

Recording	**Status**	**Depth**	**#Frames**	**Duration**
C1	Healthy	15 cm	2591	84 s
C2	Healthy	12 cm	1343	44 s
C4	Healthy	9.8 cm	1827	60 s
C5	Healthy	9.8 cm	1271	42 s
C6	Healthy	7.4 cm	1039	34 s
S1	Steinert	7.4 cm	1552	51 s
S2	Steinert	7.4 cm	1973	65 s
S2_2	Steinert	7.4 cm	2269	75 s
S2_3	Steinert	7.4 cm	2464	82 s
S4	Steinert	7.4 cm	2160	72 s
S5	Steinert	7.4 cm	1272	42 s
S7	Steinert	7.4 cm	878	29 s
S7_2	Steinert	7.4 cm	778	25 s
S8	Steinert	9.8 cm	983	32 s
S8_2	Steinert	9.8 cm	821	27 s
S9_2	Steinert	7.4 cm	627	20 s

Table 3.1 Composition of the test data set.

3.2 Error measures

To compare automatically extracted tongue contour points with the ground truth data (manually segmented tongue contour points), we used a number of error measures reported in the literature. This section discusses these measures in detail and explains how these were used to test the proposed segmentation approach and compare it to existing methods.

3.2.1 Mean sum of distances

The first error measure we used in our experiments is called the mean sum of distances (MSD) and was initially proposed by Li *et al.* (2005a). MSD is a measure that quantifies the distance between two contours. Let $U = \{u_1, u_2, ..., u_n\}$ and $V = \{v_1, v_2, ..., v_n\}$ be two sets of tongue contour points, where u_i and v_j are the i^{th} and j^{th} points on contours U and V respectively, and n is the number of points on each contour. Then, the MSD is defined as the normalized sum of distances from each contour point u_i to its closest counterpart v_j and vice-versa:

$$MSD(U,V) = \frac{\sum_{j=1}^{n} \min_{i} ||v_j - u_i|| + \sum_{i=1}^{n} \min_{j} ||u_i - v_j||}{2n} \tag{3.1}$$

MSD is a symmetric measure since $MSD(U,V) = MSD(V,U)$.

3.2.2 Tongue curvature & tongue asymmetry

The next measures we take into consideration are related to the shape of the tongue contour. We call these measures the tongue curvature and tongue asymmetry, inspired by Ménard *et al.* (2012), and these measures capture linguistically relevant shape features. The analysis by Ménard *et al.* (2012) considers a triangle defined by three vertices A, B, and C lying on the tongue contour. Points A and B are the points of intersection of pre-defined polar grid lines of the US image with the contour that are closest to the traced tongue root and tip, and point C is the point of the tongue contour that is farthest away from the line joining A and B. By projecting point C on line that joins A to B we get D . We apply a similar procedure except that instead of using a pre-defined polar grid we consider the mask computed in section 2.1.1. In this procedure, points A and B are the leftmost and rightmost points of the computed tongue contour if they are located inside the mask. Otherwise, they are defined as the intersection of tongue contours with the mask on either side (see Figure 3.1).

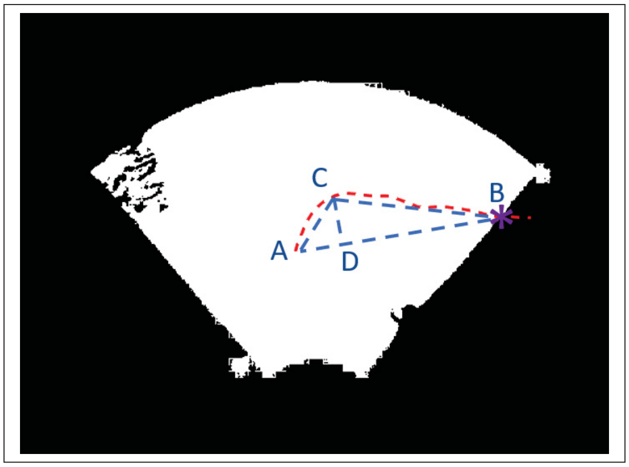


Figure 3.1 Assuming the dashed red line is representing the computed tongue contour, this figure shows how our approach computes the three points A, B, and C. The purple star shows the intersection of tongue contours with the mask on either side.

Now the shape measures are defined for curvature and asymmetry respectively:

$$\kappa = \frac{||CD||}{||AB||} \tag{3.2}$$

where κ represents the curvature score, and:

$$\gamma = \frac{||AD||}{||DB||} \tag{3.3}$$

where γ represents the asymmetry score. To find the point C with the maximum distance to the line $\overleftrightarrow{AB}$, we iterate over all contour points and find the maximum distance using the following formula:

$$\text{distance}(\overleftrightarrow{AB}, C) = \frac{|(B_y - A_y)C_x - (B_x - A_x)C_y + B_x A_y - A_x B_y|}{\sqrt{(B_y - A_y)^2 + (B_x - A_x)^2}} \tag{3.4}$$

where the coordinates of each of the points in the equation represented as $A = (A_x, A_y)$, $B = (B_x, B_y)$, and $C = (C_x, C_y)$. Once C is found, it is easy to find the point D which is the intersection of the line $\overleftrightarrow{AB}$ and the line that is perpendicular to $\overleftrightarrow{AB}$ and goes through C.

To compute how similar the obtained curvatures for each of the methods are to the ground truth data, we considered the following score as a curvature similarity measure:

$$\mathrm{acc}_{\kappa} = 1 - \frac{|\kappa_{\mathrm{met}} - \kappa_{\mathrm{gt}}|}{\kappa_{\mathrm{gt}}} \tag{3.5}$$

where κ_{met} is the curvature score of a contour computed by a specific method, and κ_{gt} is the curvature score computed for the ground truth data. Similar tongue contour curvature score to ground truth results in higher similarity scores (close to one).

Similar to curvature similarity, we consider the following measure as the asymmetry similarity:

$$\mathrm{acc}_{\gamma} = 1 - \frac{|\gamma_{\mathrm{met}} - \gamma_{\mathrm{gt}}|}{\gamma_{\mathrm{gt}}} \tag{3.6}$$

where γ_{met} is the asymmetry computed for a contour extracted using a specific method, and γ_{gt} is the asymmetry computed for the ground truth data.

3.3 Comparing the proposed segmentation method to semi- and fully- automated tracking approaches

In this section, we evaluate our proposed segmentation method (labeled "skel") that works frame-by-frame and then compare it to two tracking approaches. One is our fully automated approach (labeled "auto") detailed in section 2.2.2, and the other is the semi-automated method (labeled "semi") of Laporte & Ménard (2018), manually initialized at the same frame as the fully automated method. Since there is a random component in the particle filtering module in the tracking approaches, for both fully and semi-automated approaches, we repeated the same process 10 times and averaged error measures over the repetitions in the remainder of this chapter. Figure 3.2 compares the MSD across these three methods. Skeletal points extracted frame by frame and not tracked from one frame to the next have the highest MSD values compared to the other two approaches (skel: 2.8480 ± 1.5897, semi:1.0534 ± 0.6356, auto:1.0156 ± 0.5701). The frame by frame segmentation method does not perform as well as the other two tracking algorithms, simply due to the fact it is neither using any trained information nor tracking data. We examine some of the failure cases of frame-by-frame segmentation in comparison with the other two approaches in section 3.4. Note that the same segmentation method, when used to automatically initialize a fully-automatic tracking approach from a carefully selected frame, yields MSD scores quite similar to the semi-automatic approach where the initial points are captured manually. This means that our approach can be used for automatic initialization without loss of accuracy.

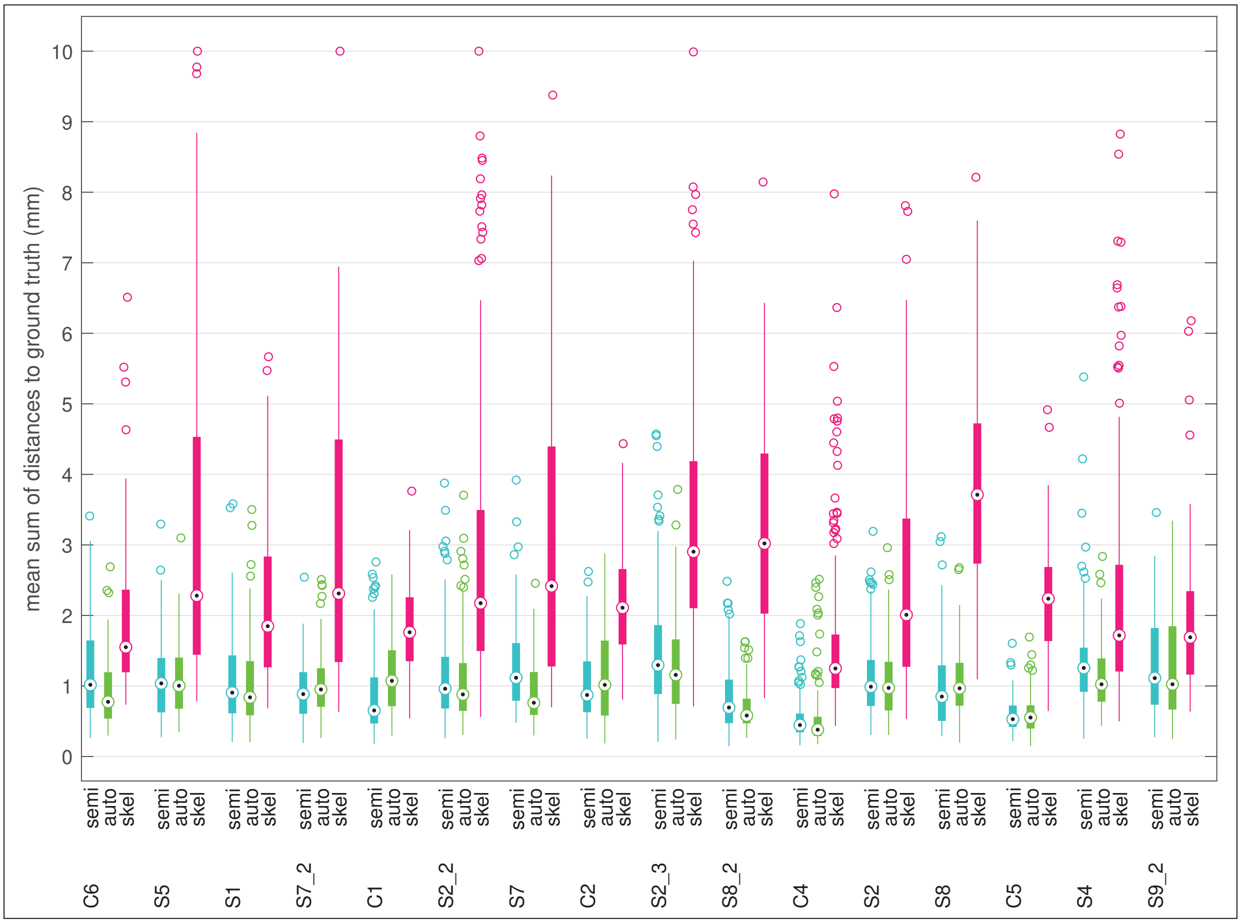


Figure 3.2 This figure compares the MSD values of tongue contour points computed from three approaches: our automatic segmentation approach before snake fitting (**skel**), our fully automatic tracking approach (**auto**), and the semi-automatic approach of Laporte & Ménard (2018) (**semi**), where all three are compared with ground truth manually segmented contour points. Since, the particle filter algorithm has a random component, and it does not always give the same result, for the two tracking approaches (**auto**, **semi**), we repeat the experiment 10 times and we are presenting the averaged result.

In addition to MSD, we measured tongue curvature and asymmetry similarity scores (see Figures 3.3 and 3.4), and the results show that the two tracking methods (fully automated, and semi-automated) have higher shape similarity scores (closer to one) than the automated segmentation method (skeletonization) used frame by frame. The fully automated approach performs similarly to the semi-automated approach where initial points are selected manually.

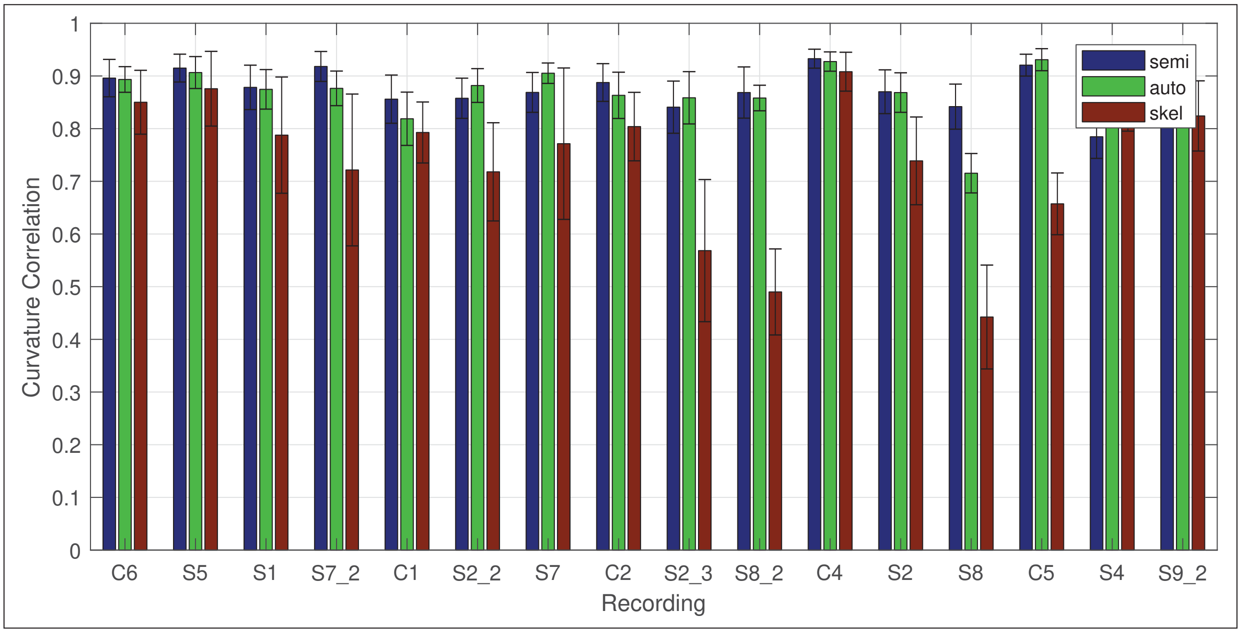


Figure 3.3 This figure shows box plot of curvature similarity between contours extracted using each of the three methods of Figure 3.2 and the ground truth data.

3.4 Sample results and challenges

This section demonstrates some tracking results including different examples of successes and failures for the various methods tested in this thesis. The approaches tried include the B-spline contour points extracted from skeletonization, the semi-automatic contour tracking approach by Laporte & Ménard (2018) that uses particle filter tracking, and the fully automatic contour tracking approach initialized with the initial frame selected and segmented automatically. These are all compared to ground truth data. The examples provided here give a more qualitative idea of how well the approaches are working, and what are some of the difficulties in the

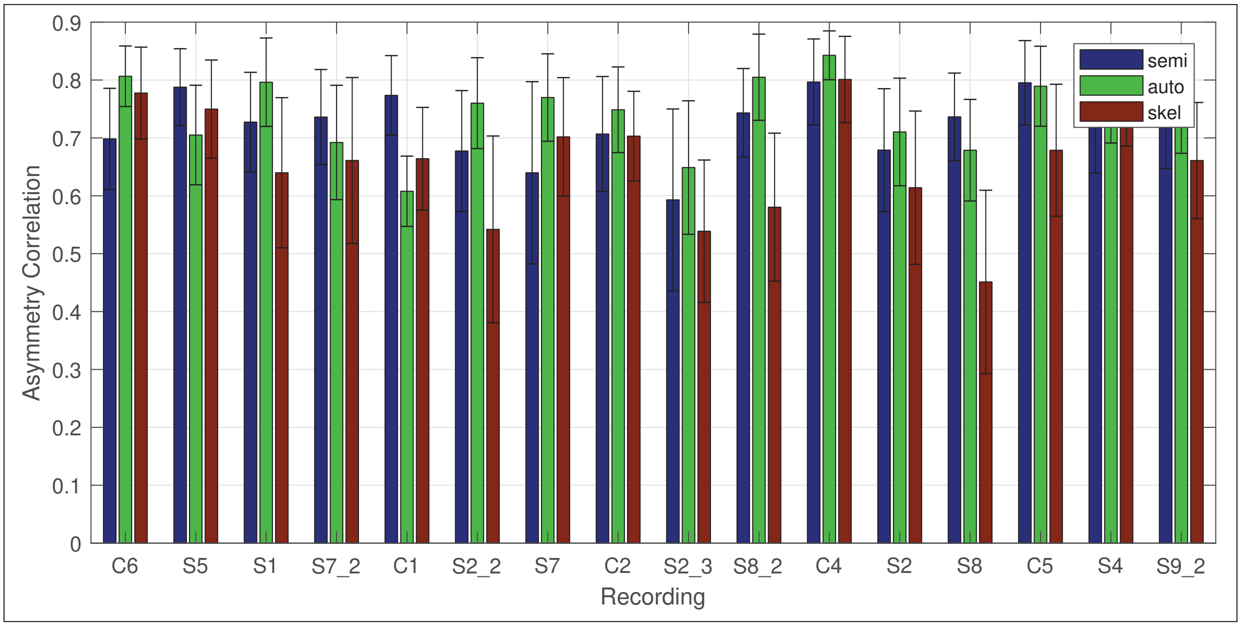


Figure 3.4 This figure shows box plot of asymmetry similarity between the contours extracted by each of the three methods of Figure 3.2 and the ground truth data.

segmentation and tracking tasks. We start with cases where all approaches are finding accurate tongue contours. Figure 3.5 shows that the three approaches achieve very similar results to ground truth data for many US frames. These examples could be called easier to detect/track as the all three algorithms were able to find proper sets of points that were close to the ground truth data.

Figure 3.6 show cases where automatic segmentation fails. This is not catastrophic since the goal is not to segment each frame individually without prior information. Rather, it is to find suitable set of initial points to initialize or re-initialize the tracker. Therefore, in many images, the actual tracked points would differ considerably from the automated segmentation result obtained without tracking information.

Besides failure cases that could happen in the skeletonization phase, the semi-automatic and fully-automatic tracking approaches can also fail due to a number of reasons. There are frames where the tracking gets lost and cannot recover a proper set of candidate points. Low signal to noise ratio in many frames could make the tracking task hard and not optimally solved.

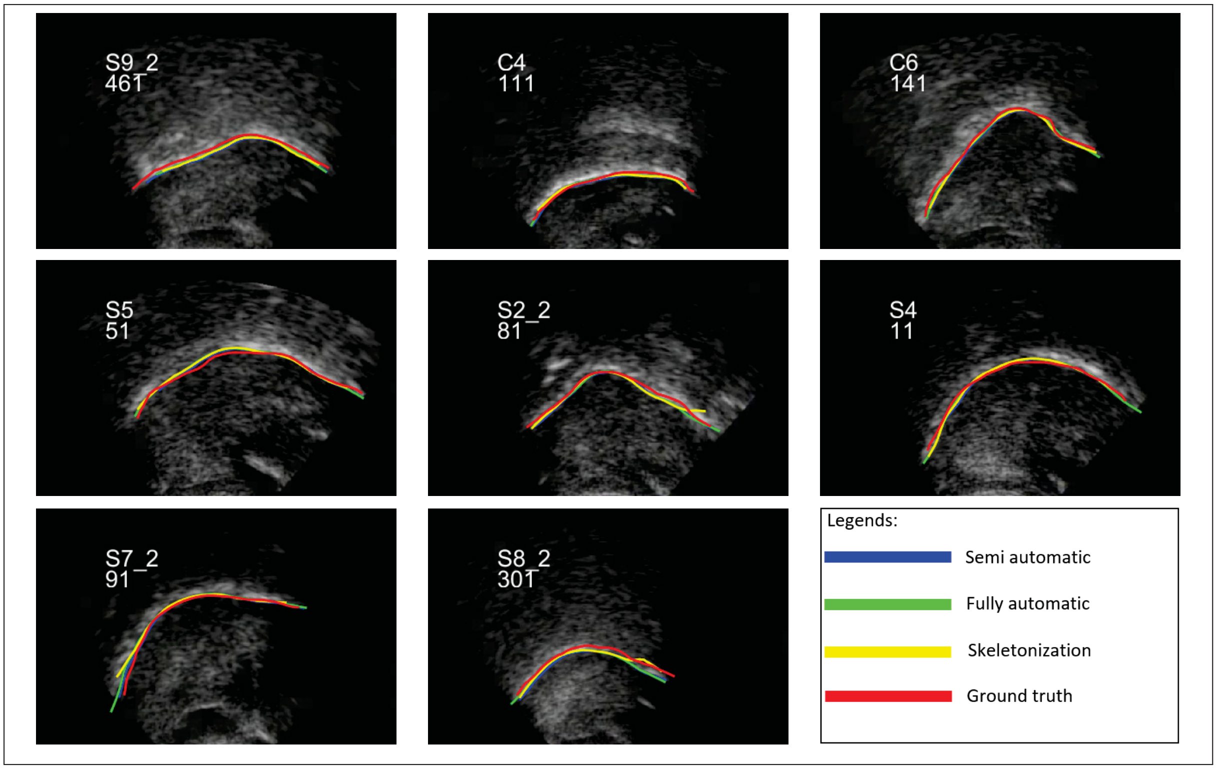


Figure 3.5 (Best viewed by zooming in on the PDF.) Sample contour points obtained from different approaches where the computed points are similar to ground truth data.

Moreover, the length of snakes can grow beyond the actual tongue in US images. Altogether, there are many cases where either of the semi- and/or fully automatic approaches can fail (see Figures 3.7 & 3.8). To address such cases in this thesis, we implemented a re-initialization module where the system performs a reset when it detects a possible loss or a good opportunity to re-initialize the tracking system (a frame that is easy to segment). Section 3.6 includes examples illustrating how re-initialization would resolve some of these cases.

Figure 3.9 presents detailed results for a video sequence which posed a particular challenge, and that is when the subject swallows. The last row in this figure shows the mean sum of distances of the tongue contour points from frame 400 to frame 500 to the ground truth data. The bright regions in the first column of this row show that the frame-by-frame method is not detecting tongue contours very well in the sequence during swallowing. However, the semi and fully automated tracking methods were able to track the tongue and generate better

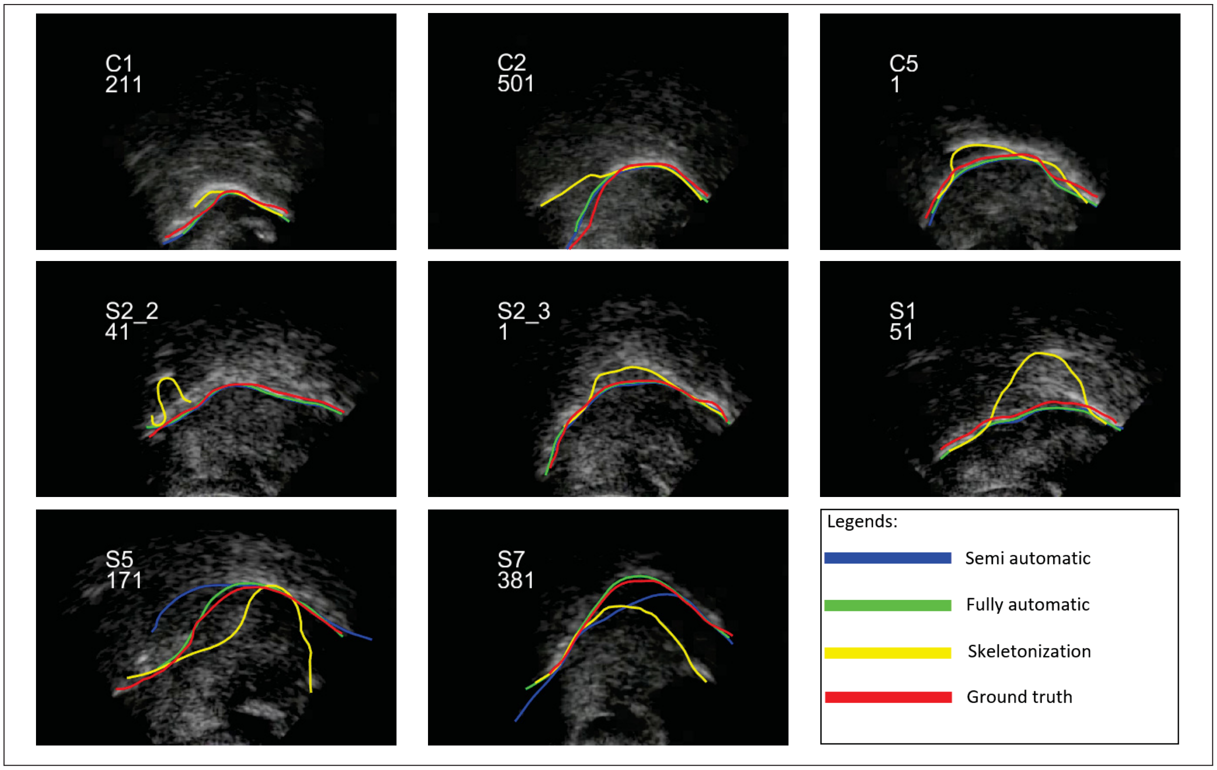


Figure 3.6 (Best viewed by zooming in on the PDF.) Example cases where skeletal points fail in segmentation but the fully automatic approach generates points close to ground truth manually segmented points.

contour points in the vicinity of the true tongue state. Although the frame-by-frame automated tongue contour segmentation method ('skeletonization') fails to detect the tongue contour in such difficult cases initialization or re-initialization using these poor segmentation results can be avoided using the reliability scores introduced in section 2.2.2. These scores are analyzed in the next section.

3.5 Analyzing reliability scores

Section 2.2.2 introduced two reliability measures, to evaluate the suitability of the automatically segmented contour points for tracker initialization. These reliability measures are later used to select a best set of candidate points to automatically initialize the semi-automatic tracking system and convert it to a fully automatic one. To validate the reliability measures, we examine their relationship to the MSD between the automatically segmented skeletal points and

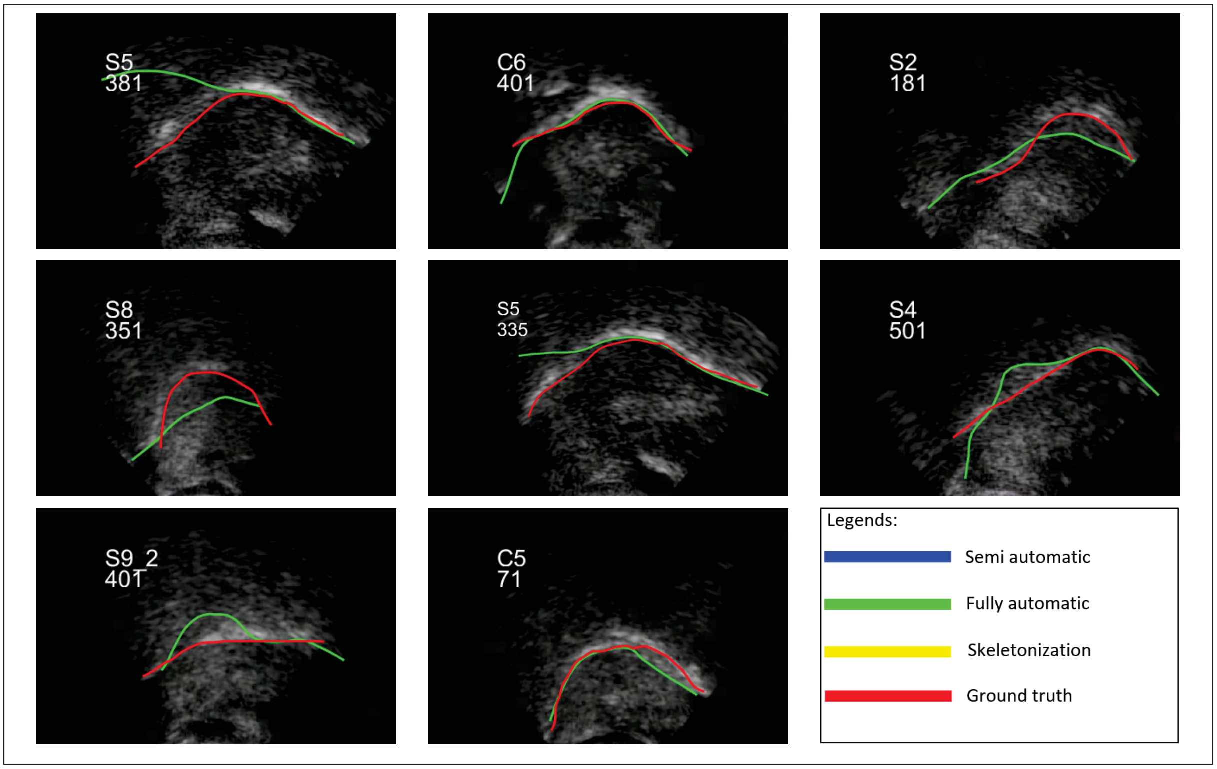


Figure 3.7 (Best viewed by zooming in on the PDF.) Example cases where the fully-automatic approach fails in tracking.

the manually segmented ground truth tongue contours. A contour is assumed to be segmented well if the skeletal points generated by the automated segmentation system are close enough to the ground truth data and vice versa. Here, we consider the following goodness measure:

$$g(V_i^{\text{sk}}, V_i^{\text{gt}}) = \frac{1}{MSD(V_i^{\text{sk}}, V_i^{\text{gt}})} \tag{3.7}$$

where V_i^{sk} and V_i^{gt} represent the tongue contour points obtained by the automated segmentation method and the manually segmented ground truth data in frame i respectively. To obtain results that reflect the relative difficulty of segmenting one frame over another within a given video sequence using the proposed segmentation method, we normalize the g function by its maximum value over all contours of all US frames in our experiments, yielding scores between

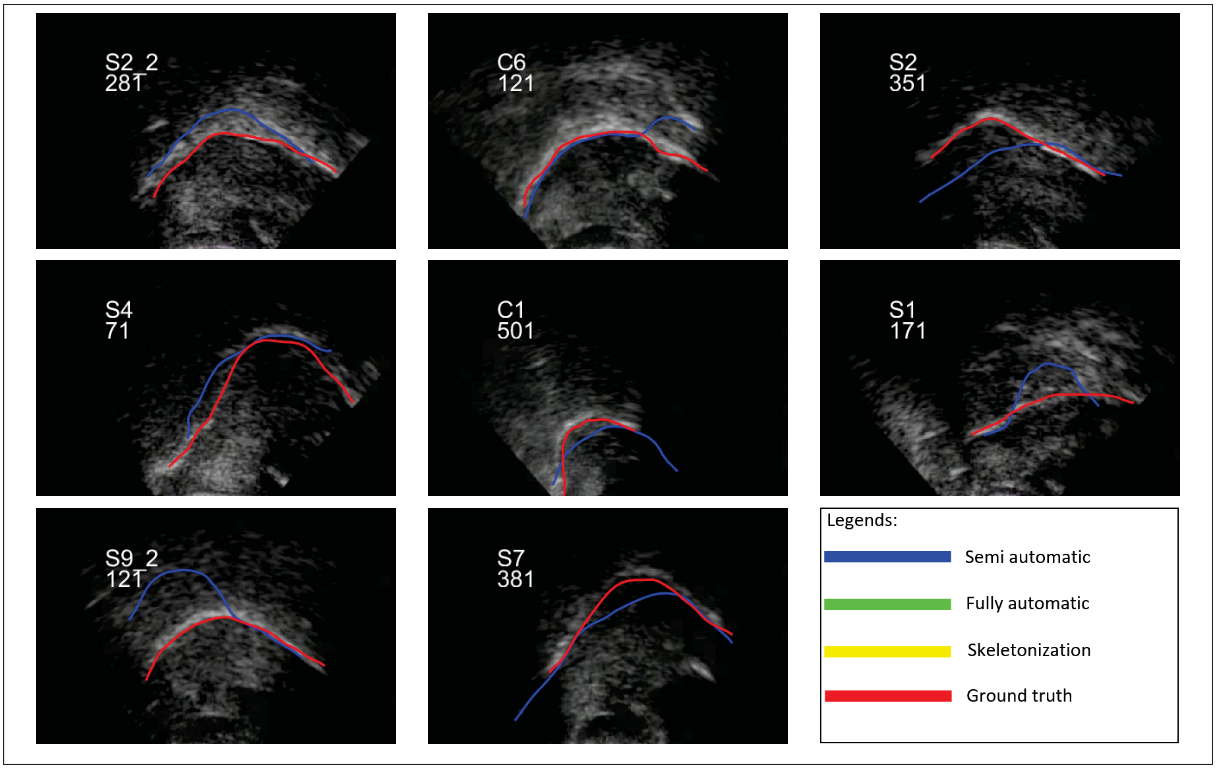


Figure 3.8 (Best viewed by zooming in on the PDF.) Example cases where the semi-automatic approach fails in tracking.

0 and 1.

$$f(V_i^{\text{sk}}, V_i^{\text{gt}}) = \frac{g(V_i^{\text{sk}}, V_i^{\text{gt}})}{\max_i g(V_i^{\text{sk}}, V_i^{\text{gt}})} \tag{3.8}$$

We examine the relationship of $f(V_i^{\text{sk}}, V_i^{\text{gt}})$ to the combination of the two scores $(\Gamma_1^{\lambda_1}\Gamma_2^{\lambda_2})$, to determine whether they share a similar trend.

The 16 videos used in our experiments contain a total of 23776 frames. We segmented all these frames using the automated segmentation module and then compared the resulting contours to the ground truth (manual segmentation) using the *MSD* measure. We then sorted all these frames in ascending order based on their f score (Equation 3.8). Figure 3.10a shows these scores sorted in ascending order as a dashed red line. As 23776 frames are sorted based on their f score, the Γ_1 and Γ_2 scores for each of the associated automatically segmented contours are computed presented as a point cloud in Figures 3.10a, and 3.10b. Logarithmic scales are

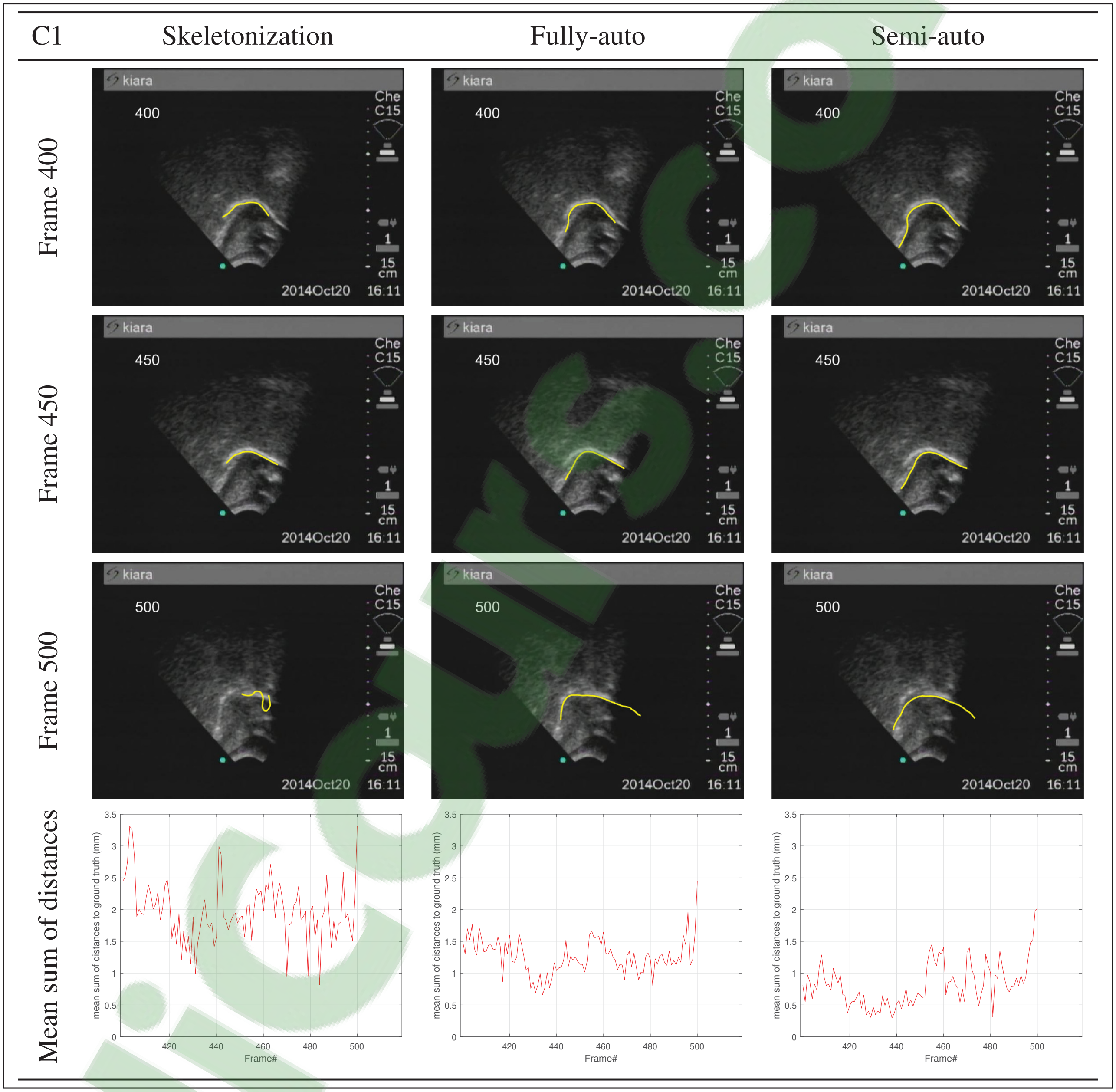
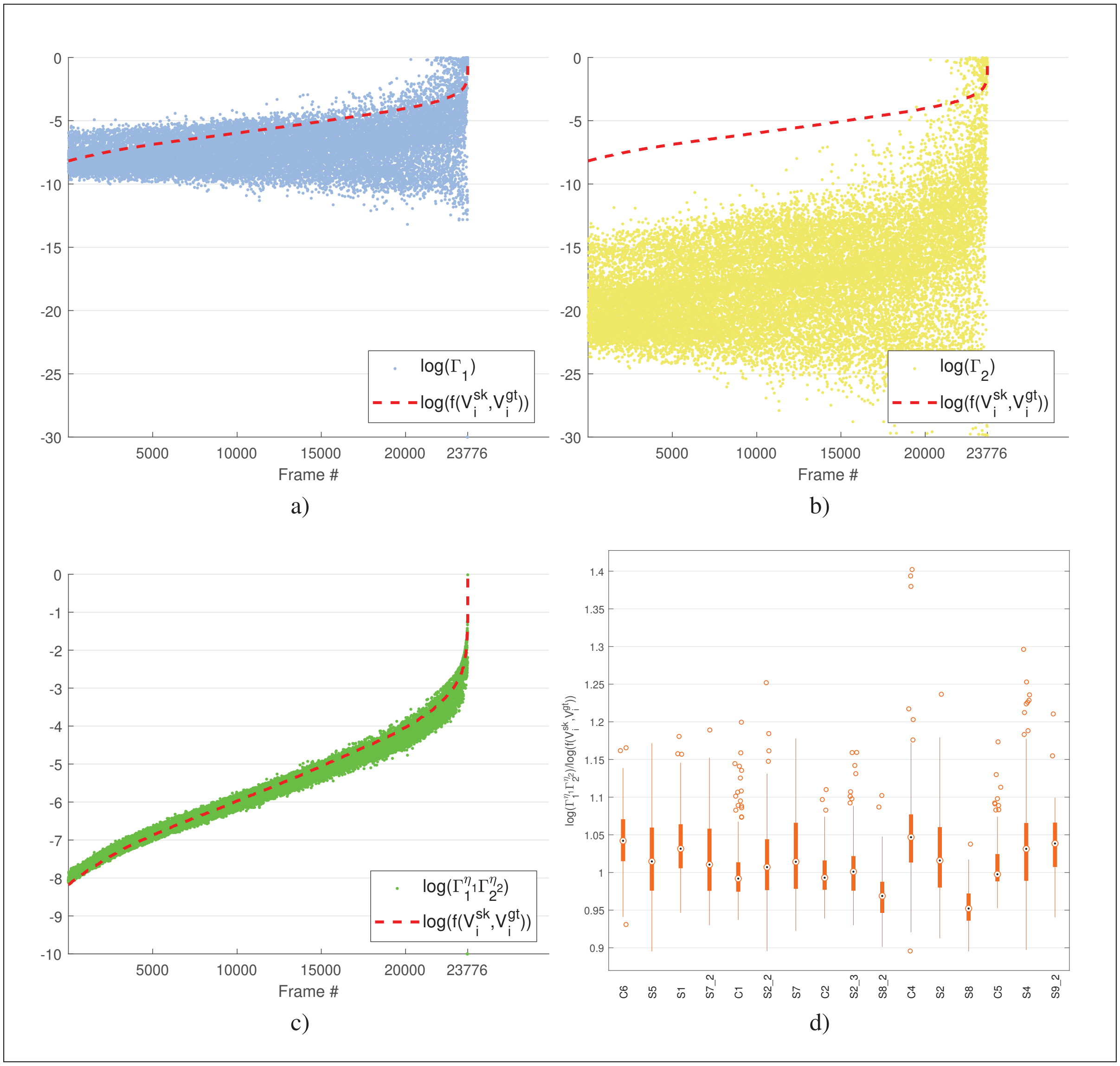


Figure 3.9 (Best viewed by zooming in on the PDF.) This Figure shows the automated segmentation and the two tracking approaches result on a subject during swallow action.

used for improved visualization. Though reliability scores are fairly broadly distributed in Figures 3.10a and 3.10b, both of these plots show a similar trend to the inverse MSD. Figure 3.10c shows the result of combining the Γ_1 and Γ_2 scores according to Equation 2.22 in relation to sorted f scores. Figure 3.10d shows the boxplot of the ratio of the Γ score to the normalized



a)

b)

c)

d)

Figure 3.10 Analysis of the Γ score introduced in section 2.2.2 (see the text for more information).

inverse MSD (f score) along each video separately. This ratio is clearly quite close to 1 in all cases, indicating that there is a strong relationship between the Γ score and the inverse MSD, which suggests that the proposed score can safely be used as a reliability measure to select a candidate set of points for tracker initialization. This is confirmed by the results already presented in section 3.3, which showed little difference in error between the fully automated and semi-automated tracking methods.

3.6 Re-initialization

This section, discusses experiments done using the tracker re-initialization approach described in section 2.2.3. As discussed earlier in section 3.4, the semi-automatic and fully automatic tracking methods can lose track of the tongue contour from time to time. When there is a possible loss of tracking or the opportunity to reset in a reliable fashion, re-setting the approach and forcing it to automatically re-initialize points may help with the overall quality of the tracked tongue contour points.

Our system was fine-tuned empirically to suggest cases where the system should reset itself, and we compared the goodness of tracked contour points for three approaches: 1) the semi-automatic approach of Laporte & Ménard (2018), 2) the fully automatic approach proposed in this thesis and 3) the fully automatic approach proposed in this thesis with an added re-initialization module. Putting the three approaches in the experimental setup as before, and comparing results (see Figure 3.11), we see that the re-initialization approach decreases the MSD in all 16 videos consistently compared to both the approaches without re-initialization. Table 3.2 shows the performance boost that re-intialization process brings to the tracking. This indicates that the re-initialization helps keep track of the tongue contours in videos with large numbers of frames (where there is a chance that the tracking could get lost).

Measure	**Fully-auto**	**Semi-auto**	**Re-initialization**
Mean	1.0156	1.0534	0.6372
Standard Deviation	0.5701	0.6356	0.3589

Table 3.2 Comparison of the mean and standard deviation of all MSD values across 16 videos for the three methods of **semi**, **auto**, and **re-init**.

We also recorded the average number of times that a reset happens in each video sequence in a window of 1000 frames (see Figure 3.12). Here, we see that this number varies from one video to the next, and is very dependent on the quality of the US data and the shape of the tongue. Here, we represent the resets caused by high SSIM and also those due to a large number of particles separately. Results show that large numbers of particles cause more resets than high

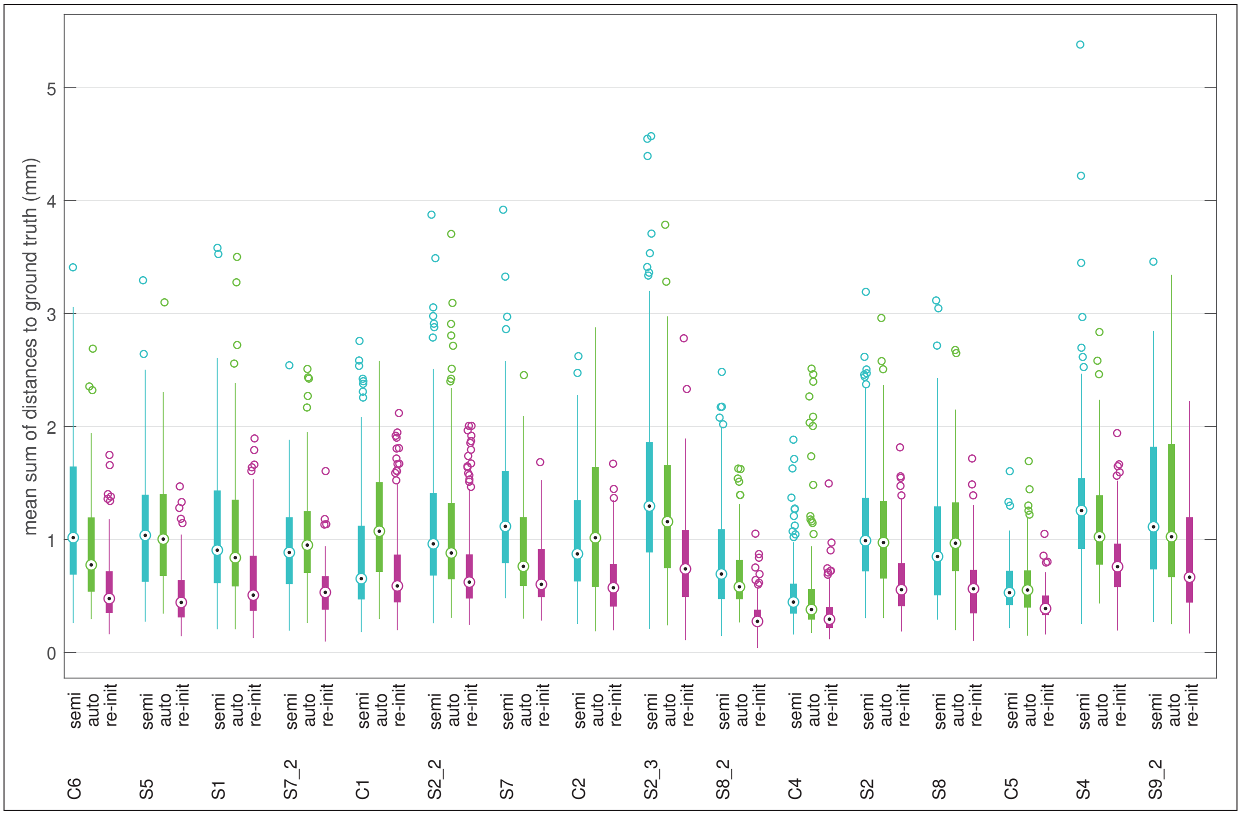


Figure 3.11 This Figure shows the MSD comparison between three methods of **semi**, **auto**, and **re-init** where the first two have been extensively used and discussed in the previous Figures of this chapter. The **re-init** method is the same as the **auto** with added module of re-initialization.

SSIM in most cases. We also analyzed the relative computation time required when using re-initialization compared the baseline to semi and fully automated approaches (see Figure 3.13). Clearly, re-initialization increases computation time; therefore, deciding whether to use re-initialization in our problem is a choice between accuracy and time complexity.

3.7 Summary

In this chapter, we analyzed the approaches outlined in the methodology chapter. We discussed the data acquisition, reviewed some error measures that were used and showed the experimental results of applying our automatic tongue detection methods. Our results show the strength of the proposed automatic tongue detection method by this thesis as well as the validity of the

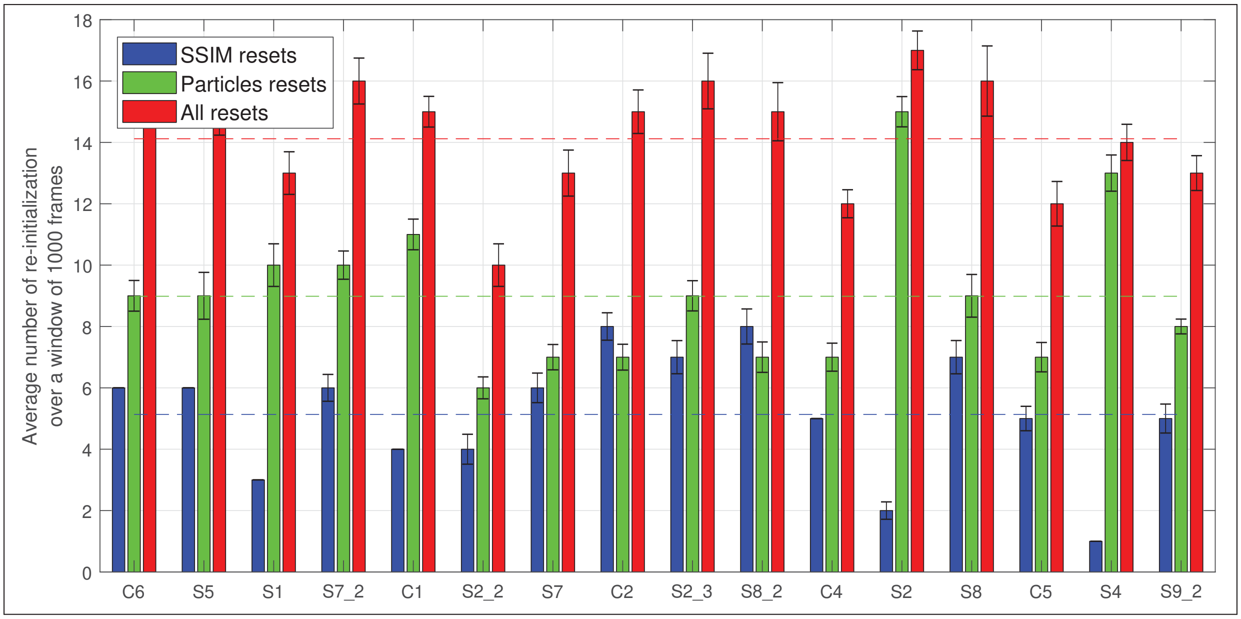


Figure 3.12 The average number of re-initializations occurring in a window of 1000 frames for each video used in our experiments. Note that since there is a random factor in the tracking approach these numbers are also averaged over 10 repetitions. The blue, green, and red dashed lines represent the mean number of resets caused due to SSIM criteria, number of particles criterial and the sum of these two over all video sequences resepectively.

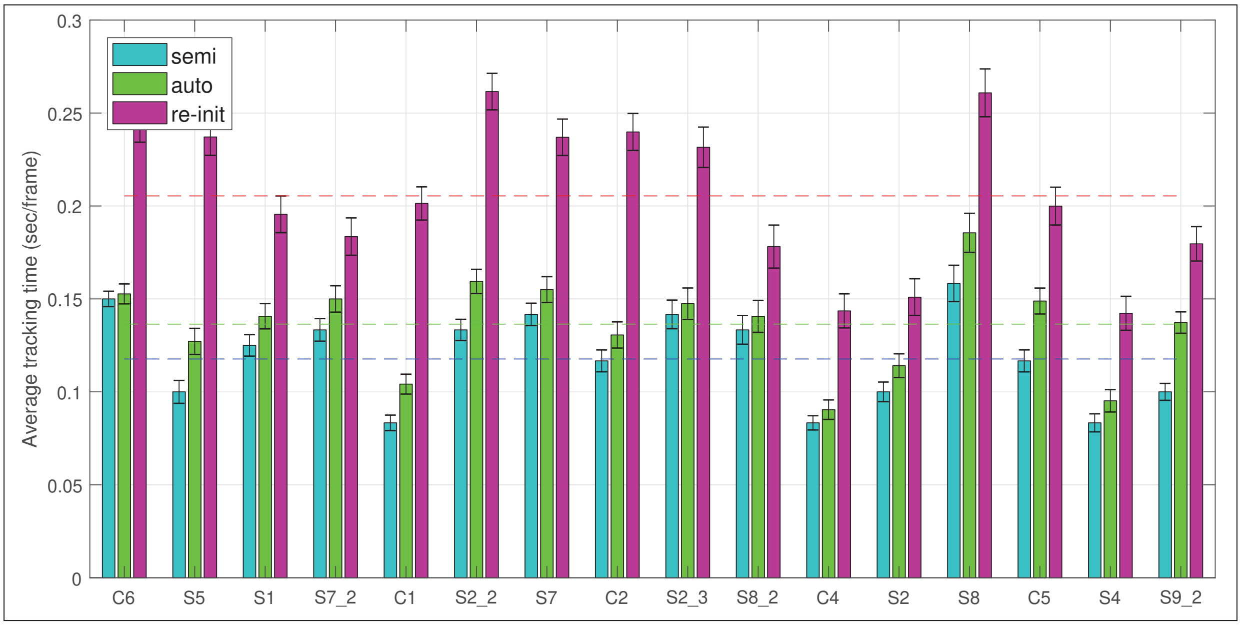


Figure 3.13 The average computation time of each method for each recording. This figure shows that using re-initialization during tracking is slower than the other two tracking methods.

choices for reliability scores. Finally, we concluded that our novel approach of re-initialization would improve the mean and standard deviation of MSD error of the automatic tongue tracking approach by almost 40%.

CHAPTER 4

CONCLUSION AND FUTURE WORK

The goal of this work was to design a system that detects and tracks tongue contour points in US images with no need for manual initialization. This involved the development of a novel algorithm to automatically detect a tongue contour from an US image and self-evaluate its reliability. In addition to detection of the tongue contour from US images, the system proposed in this thesis can turn any semi-automatic tracking approach into a fully automatic one by selecting a suitable set of initialization contour points that are segmented automatically. The final system is equipped with a reset module which instructs the tracking algorithm to re-initialize whenever the system detects that a reset would help achieve more accurate tracking.

4.1 Contributions

The contributions of this thesis can be summarized as follows:

1. The first and most important contribution of this thesis is a new approach to automatically segment tongue contours from US images. This approach is original compared to other existing ones due to the fact that it is not based on prior information on the shape of the tongue surface nor does it need manual initialization or refinements. Although the segmentation method is used in the application of tongue contour detection/tracking in this thesis, one could possibly utilize the same mechanism in other application domains (i.e. other organs). Our experiments show that the segmentation system works well when used in combination with an existing semi-automated tracking approach and results show either a similar or better performance when we apply the automatic initialization procedure.
2. A second major contribution of this thesis lies in the reliability scores introduced in Section 2.2.2. These scores help extend the segmentation approach to be utilized within any semi-automatic tongue tracking method, thereby making it a fully-automatic tracking

method. Being able to automatically evaluate the segmented tongue contours enables the resulting system to select a candidate set of initial points that are extracted completely automatically and can be used in place of manual initialization. The analysis provided in Section 3.5 demonstrates that these scores are proper choices to evaluate the quality of tongue contour segmented for the purpose of automatically initializing a semi-automatic tracking approach.

3. A third major contribution of this thesis relates to the fully-automatic tracking approach that was built by combining the automatic tongue detection approach and a semi-automatic tracking method. The fully automatic tracking method is a combination of automatic initialization of tongue contour points using the automatic segmentation method and the semi-automatic tracking system developed by Laporte & Ménard (2018). This novel system offers a variety of benefits over the state-of-the-art existing methods including being more accurate, and not depending on human intervention, refinement or manual initialization. Moreover, using automatically generated contour points for initialization has never been reported in past work. We also improved the entire system by adding a reset module so whenever certain criteria are met the system re-initializes based on the tracker-independent automatic tongue detection method to improve the final result of tracking (see Section 2.2.3). The use of these criteria is novel in the sense that it is incorporating an existing measure (Xu *et al.* (2016a)) alongside a measure that is based on the number of particles and can potentially play the role of indicator for the cases where tracking is uncertain. We analyzed this method by observing how this could benefit the overall tracking performance in Section 3.6, and we noticed that both mean and standard deviation measures of MSD accuracy are improved by about 40%.

4.2 Future work

There are many directions that could be explored to improve the proposed method. The methodology described in this work utilized 2D US images of the tongue muscle, and this could easily be extended to work in 3D US images. The masking, phase symmetry filtering, binarization and medial skeletonization steps are all transferable to 3D. The final outcome of such a system would be a set of tongue surface points.

Moreover, as mentioned earlier, we could easily modify and extend the system proposed in this thesis and use it in other similar applications involving segmentation and tracking in US images. This could be extended to many medical applications of US imaging, including echocardiography.

Finally, with the recent advances in the domain of neural networks (Goodfellow *et al.* (2016)), one could setup a convolutional neural network that could generate tongue contour points from US image frames. Since this requires a huge dataset of annotated US frames, we could use our automatic segmentation method along with the reliability scores and generate a large amount of anotated images for this purpose.

APPENDIX I

THE STRUCTURAL SIMILARITY INDEX

The structural similarity (SSIM) index (Wang & Bovik (2002)) is a quantity that measures the similarity between two images. In this index, three structural measures from an image are considered: luminance, contrast, and structural term the which is defined as those attributes that represent the structure of objects in the scene and is independent of the average luminance and contrast.

$$SSIM(A,B) = lum(A,B)^{\alpha} \times con(A,B)^{\beta} \times st(A,B)^{\gamma} \tag{A I-1}$$

Let us assume m_A, m_B, σ_A, σ_B, and σ_{AB} respectively represent the local means, standard deviations, and co-variance of images A and B. With this consideration:

$$lum(A,B) = \frac{2m_A m_B + \theta_1}{m_A^2 + m_B^2 + \theta_1} \tag{A I-2}$$

$$con(A,B) = \frac{2\sigma_A \sigma_B + \theta_2}{\sigma_A^2 + \sigma_B^2 + \theta_2} \tag{A I-3}$$

$$st(A,B) = \frac{2\sigma_{AB} + \theta_3}{\sigma_A \sigma_B + \theta_3} \tag{A I-4}$$

θ_1, θ_2, θ_3 are variables to stabilize the division with weak denominator.

BIBLIOGRAPHY

[Accessed: 2018-09-19]. Muscles that move the tongue. Consulted at https://upload.wikimedia.org/wikipedia/commons/9/97/1109_Muscles_that_Move_the_Tongue.jpg.

Akgul, Y. S., Kambhamettu, C. & Stone, M. (1998). Extraction and tracking of the tongue surface from ultrasound image sequences. *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pp. 298–303.

Amini, A. A., Weymouth, T. E. & Jain, R. C. (1990). Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9), 855–867.

Aron, M., Roussos, A., Berger, M.-O., Kerrien, E. & Maragos, P. (2008). Multimodality acquisition of articulatory data and processing. *Signal Processing Conference, 2008 16th European*, pp. 1–5.

Arulampalam, M. S., Maskell, S., Gordon, N. & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188.

Bacsfalvi, P. & Bernhardt, B. M. (2011). Long-term outcomes of speech therapy for seven adolescents with visual feedback technologies: Ultrasound and electropalatography. *Clinical Linguistics & Phonetics*, 25(11-12), 1034–1043.

Bernhardt, B., Gick, B., Bacsfalvi, P. & Adler-Bock, M. (2005). Ultrasound in speech therapy with adolescents and adults. *Clinical Linguistics & Phonetics*, 19(6-7), 605–617.

Blum, H. (1967). A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, (5), 362-380. Consulted at www.scopus.com. Cited By :37.

Bressmann, T., Ackloo, E., Heng, C.-L. & Irish, J. C. (2007). Quantitative three-dimensional ultrasound imaging of partially resected tongues. *Otolaryngology—Head and Neck Surgery*, 136(5), 799–805.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 679–698.

Chi-Fishman, G. (2005). Quantitative lingual, pharyngeal and laryngeal ultrasonography in swallowing research: a technical review. *Clinical Linguistics & Phonetics*, 19(6-7), 589–604.

Chien, C.-Y., Chen, J.-W., Chang, C.-H. & Huang, C.-C. (2017). Tracking dynamic tongue motion in ultrasound images for obstructive sleep apnea. *Ultrasound in Medicine & Biology*, 43(12), 2791–2805.

Cootes, T. F., Taylor, C. J., Cooper, D. H. & Graham, J. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1), 38–59.

Csapó, T. G. & Lulich, S. M. (2015). Error analysis of extracted tongue contours from 2d ultrasound images. *Sixteenth Annual Conference of the International Speech Communication Association*.

Davidson, L. (2006a). Comparing tongue shapes from ultrasound imaging using smoothing spline analysis of variance. *The Journal of the Acoustical Society of America*, 120 1, 407-15.

Davidson, L. (2006b). Comparing tongue shapes from ultrasound imaging using smoothing spline analysis of variance a. *The Journal of the Acoustical Society of America*, 120(1), 407–415.

Dimitrov, P., Damon, J. N. & Siddiqi, K. (2003). Flux invariants for shape. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1.

Epstein, M. A. & Stone, M. (2005). The tongue stops here: Ultrasound imaging of the palate. *The Journal of the Acoustical Society of America*, 118(4), 2128–2131.

Ester, M., Kriegel, H. P., Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*, 226–231.

Fabre, D., Hueber, T., Bocquelet, F. & Badin, P. Tongue tracking in ultrasound images using eigentongue decomposition and artificial neural networks. *Sixteenth Annual Conference of the International Speech Communication Association*, pp. 2410–2414.

Fasel, I. & Berry, J. (2010). Deep belief networks for real-time extraction of tongue contours from ultrasound during speech. *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 1493–1496.

Fenster, A., Downey, D. B. & Cardinal, H. N. (2001). Three-dimensional ultrasound imaging. *Physics in Medicine and Biology*, 46(5), R67.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, 4(12), 2379–2394.

Ghrenassia, S., Laporte, C. & Ménard, L. (2013). Statistical shape analysis in ultrasound video sequences: tongue tracking and population analysis. VI, 53–55.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press.

Hageman, T., Slump, I. C., van der Heijden, I. F., Balm, A. & Salm, I. C. (2013). Tracking of the tongue in three dimensions using a visual recording system. *University of Twente*.

Hamarneh, G. & Gustavsson, T. (2000). Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images. *Computers in Cardiology*, 2000, 115–118.

Hixon, T. J., Weismer, G. & Hoit, J. D. (2014). *Preclinical Speech Science: Anatomy, Physiology, Acoustics, Perception*. Plural Pub.

Hueber, T., Aversano, G., Cholle, G., Denby, B., Dreyfus, G., Oussar, Y., Roussel, P. & Stone, M. (2007). Eigentongue feature extraction for an ultrasound-based silent speech interface. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, 1, I–1245.

Jaumard-Hakoun, A., Xu, K., Roussel-Ragot, P., Dreyfus, G. & Denby, B. (2016). Tongue contour extraction from ultrasound images based on deep neural network. *arXiv preprint arXiv:1605.05912*.

Jensen, J. A. (2007). Medical ultrasound imaging. *Progress in biophysics and molecular biology*, 93(1-3), 153–165.

Kambhamettu, C. & Goldgof, D. B. (1994). Curvature-based approach to point correspondence recovery in conformal nonrigid motion. *CVGIP: Image Understanding*, 60(1), 26–43.

Kass, M., Witkin, A. & Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4), 321–331.

Kovesi, P. et al. (1997). Symmetry and asymmetry from local phase. 190, 2–4.

Lai, K. F. & Chin, R. T. (1995). Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11), 1084–1090.

Laporte, C. & Ménard, L. (2015). Robust tongue tracking in ultrasound images: a multi-hypothesis approach. *Sixteenth Annual Conference of the International Speech Communication Association*, pp. 633–637.

Laporte, C. & Ménard, L. (2018). Multi-hypothesis tracking of the tongue surface in ultrasound video recordings of normal and impaired speech. *Medical Image Analysis*, 44, 98–114.

Li, M., Kambhamettu, C. & Stone, M. (2005a). Automatic contour tracking in ultrasound images. *Clinical linguistics & phonetics*, 19(6-7), 545–554.

Li, M., Kambhamettu, C. & Stone, M. (2005b). Tongue motion averaging from contour sequences. *Clinical Linguistics & Phonetics*, 19(6-7), 515–528.

Lucas, B. D. & Kanade, T. (1984). An iterative image registration technique with an application to stereo vision. *Proceedings of the DARPA Image Understanding Workshop*, 121-130.

Maeda, S. (1979). An articulatory model of the tongue based on a statistical analysis. *The Journal of the Acoustical Society of America*, 65(S1), S22–S22.

Marr, D. & Hildreth, E. (1980). Theory of edge detection. *Proc. R. Soc. Lond. B*, 207(1167), 187–217.

Ménard, L., Aubin, J., Thibeault, M. & Richard, G. (2012). Measuring tongue shapes and positions with ultrasound imaging: A validation experiment using an articulatory model. *Folia Phoniatrica et Logopaedica*, 64(2), 64–72.

Metz, C., Klein, S., Schaap, M., van Walsum, T. & Niessen, W. J. (2011). Nonrigid registration of dynamic medical imaging data using nd+ t b-splines and a groupwise optimization approach. *Medical Image Analysis*, 15(2), 238–249.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.

Peng, T., Kerrien, E. & Berger, M.-O. (2010). A shape-based framework to segmentation of tongue contours from mri data. *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 662–665.

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

Rezanejad, M. & Siddiqi, K. (2013). Flux graphs for 2d shape analysis. In *Shape Perception in Human and Computer Vision* (pp. 41–54). Springer.

Rezanejad, M., Samari, B., Rekleitis, I., Siddiqi, K. & Dudek, G. (2015). Robust environment mapping using flux skeletons. *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 5700–5705.

Roussos, A., Katsamanis, A. & Maragos, P. (2009). Tongue tracking in ultrasound images with active appearance models. *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 1733–1736.

Shawker, T. H. & Sonies, B. C. (1985). Ultrasound biofeedback for speech training. instrumentation and preliminary results. *Investigative Radiology*, 20(1), 90–93.

Song, J. Y., Demuth, K., Shattuck-Hufnagel, S. & Ménard, L. (2013). The effects of coarticulation and morphological complexity on the production of english coda clusters: Acoustic and articulatory evidence from 2-year-olds and adults using ultrasound. *Journal of Phonetics*, 41(3), 281 - 295. doi: https://doi.org/10.1016/j.wocn.2013.03.004.

Stone, M. (1997). Laboratory techniques for investigating speech articulation. *The handbook of phonetic sciences*, 1, 1–32.

Stone, M. (2005). A guide to analysing tongue motion from ultrasound images. *Clinical Linguistics & Phonetics*, 19(6-7), 455–501.

Stone, M., Shawker, T. H., Talbot, T. L. & Rich, A. H. (1988). Cross-sectional tongue shape during the production of vowels. *The Journal of the Acoustical Society of America*, 83(4), 1586–1596.

Stone, M., Davis, E. P., Douglas, A. S., Aiver, M. N., Gullapalli, R., Levine, W. S. & Lundberg, A. J. (2001). Modeling tongue surface contours from cine-mri images. *Journal of Speech, Language, and Hearing Research*, 44(5), 1026–1040.

Stone, M., Langguth, J. M., Woo, J., Chen, H. & Prince, J. L. (2014). Tongue motion patterns in post-glossectomy and typical speakers: A principal components analysis. *Journal of Speech, Language, and Hearing Research*, 57(3), 707–717.

Tang, L., Hamarneh, G. & Bressmann, T. (2011). A machine learning approach to tongue motion analysis in 2d ultrasound image sequences. *International Workshop on Machine Learning in Medical Imaging*, pp. 151–158.

Tang, L., Bressmann, T. & Hamarneh, G. (2012). Tongue contour tracking in dynamic ultrasound via higher-order mrfs and efficient fusion moves. *Medical Image Analysis*, 16(8), 1503–1520.

Turetsky, R. J. & Ellis, D. P. (2003). Ground-truth transcriptions of real music from force-aligned midi syntheses.

Wang, Z. & Bovik, A. C. (2002). A universal image quality index. *IEEE signal processing letters*, 9(3), 81–84.

Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612.

Xie, N., Laga, H., Saito, S. & Nakajima, M. (2010). Ir2s: interactive real photo to sumi-e. *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 63–71.

Xu, K., Gábor Csapó, T., Roussel, P. & Denby, B. (2016a). A comparative study on the contour tracking algorithms in ultrasound tongue images with automatic re-initialization. *The Journal of the Acoustical Society of America*, 139(5), EL154–EL160.

Xu, K., Yang, Y., Stone, M., Jaumard-Hakoun, A., Leboullenger, C., Dreyfus, G., Roussel, P. & Denby, B. (2016b). Robust contour tracking in ultrasound tongue image sequences. *Clinical Linguistics & Phonetics*, 30(3-5), 313–327.

Yang, C. & Stone, M. (2002). Dynamic programming method for temporal registration of three-dimensional tongue surface motion from multiple utterances. *Speech Communication*, 38(1), 201–209.