Computers in Biology and Medicine 111 (2019) 103335

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Fully-automated tongue detection in ultrasound images

Elham Karimi[a], Lucie Ménard[b,d,c], Catherine Laporte[a,c,*]

[a] Department of Electrical Engineering, École de technologie supérieure, 1100 Rue Notre-Dame O, Montréal, QC, H3C 1K3, Canada
[b] Department of Linguistics, Université du Québec à Montréal, Montreal, QC, H2X 1L7, Canada
[c] Sainte-Justine University Hospital Research Centre, 3175 Chemin de la Côte-Sainte-Catherine, Montréal, QC, H3T 1C4, Canada
[d] Center for Research on Brain, Language, and Music, Montreal, H3G 2A8, Canada

ARTICLE INFO

Keywords:
Tongue detection
Image segmentation
Ultrasound
Fully-automated tracking
Speech

ABSTRACT

Tracking the tongue in ultrasound images provides information about its shape and kinematics during speech. Current methods for detecting/tracking the tongue require manual initialization or training using large amounts of labeled images. In this article, we propose a solution to convert a semi-automatic tongue contour tracking system to a fully-automatic one. This work introduces a new method for extracting tongue contours in ultrasound images that requires no training nor manual intervention. The method consists in an image enhancement step based on phase symmetry, followed by skeletonization and clustering steps, leading to a set of candidate points that can be used to fit an active contour to the image and subsequently initialize a tracking algorithm. Two novel quality measures were also developed that predict the reliability of the segmentation result so that an image with a reliable contour can be chosen to confidently initialize fully automated tongue tracking. This is achieved by automatically generating and choosing a set of points that can replace the manually segmented points for a semi-automated tracking approach. This paper also improves the accuracy of tracking by incorporating two criteria to reset the tracking algorithm from time to time. Experiments show that fully automated and semi-automated methods result in very similar mean sum of distances errors, respectively, indicating that the proposed automatic initialization does not significantly alter accuracy. Moreover, further results show that tracking accuracy is improved when using the new segmentation technique within the proposed re-initialization scheme.

1. Introduction

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 bio-mechanical 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. In speech studies, tongue US images are typically acquired with a micro-convex array probe placed under the subject's chin and oriented such as to image the mid-sagittal section of the tongue [1].

Analyzing such US data typically involves comparing tongue shapes between different individuals or populations, and this in turn usually requires delineating the mid-sagittal tongue contour in the US images. This paper focuses on fully automatically detecting tongue contours from US images without manual intervention by a human or use of any training data.

There have been many studies that suggest systems for tracking the tongue contour in US image sequences. One of the popular classical approaches among speech 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. [2], snakes aim to find salient contours for delineating an object outline from a possibly noisy image. Akgul et al. [3] proposed a method to segment and track the tongue surface contour in 2D US images using snakes. Snake energy is formulated as a linear combination of internal energy and external energy terms, respectively encoding contour smoothness (both spatial and temporal) and saliency, as measured by the image gradient. Li et al. [4] extended this formulation to make segmentation and tracking more robust in the presence of noise and spurious high-contrast edges in ultrasound images. Li et al. [4] introduced a new energy functional called “band energy” to guide the snake towards the lower edge of the

* Corresponding author. Department of Electrical Engineering, École de technologie supérieure, 1100 Rue Notre-Dame O, Montréal, QC, H3C 1K3, Canada.
E-mail addresses: elham.karimi.1@etsmtl.net (E. Karimi), menard.lucie@uqam.ca (L. Ménard), catherine.laporte@etsmtl.ca (C. Laporte).

https://doi.org/10.1016/j.compbiomed.2019.103335
Received 14 February 2019; Received in revised form 24 May 2019; Accepted 17 June 2019

bright white band formed by the echo of US off the surface of the tongue. With band energy, snake segments avoid attraction to the irrelevant regions of high gradient caused by speckle noise. A software implementation of this method, called “EdgeTrak”, was made publicly available by the authors. Like Akgul et al. [3]'s method, EdgeTrak demands the user input points near the tongue surface as initial points in first frame of the sequence and by interpolating initial points by B-spline, the system finds a contour near the tongue surface to fit a snake to the first image, thereby initializing tracking.

One of the problems with EdgeTrak is that it can fail 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. To address this, Roussos et al. [5] 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 AAMs, active shape models (ASM) also can be used along with snakes for segmentation of structures such as the tongue. Hamarneh et al. [6] 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. [7].

A simple, yet different solution to address the problem of tongue motion tracking is to estimate motion via a gradient based approach. Chien et al. [8] present an approach to track tongue motion in ultrasound images for obstructive sleep apnea using an optical flow (OF) method by Lucas and Kanade [9] within a multi-scale framework. The most important limitation of this approach is its computational cost, which makes it very slow in comparison with other dynamic 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. Neural networks can work well if there are enough data they can learn from. In our problem, this translates to having a database of segmented US images of tongue contours. Fasel and Berry [10] 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 and Berry [10] has resulted in a publicly available software called “Autotrace”.

The approach by Fasel and Berry [10] 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. [11] proposed a similar methodology in line with the work presented by Fasel and Berry [10] but with a simpler neural network. In their approach, they take advantage of a PCA-based decomposition technique called “EigenTongues” which is a compact representation of raw pixels intensities of tongue US images (explained originally by Hueber et al. [12]), 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. [13] 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. [13]'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 potential weakness of machine learning-based methods is to be speaker dependent [14]. In other words, a learned segmentation algorithm may not work on new speakers that the neural net has not seen before.

The literature presented so far makes limited use of temporal consistency constraints to guide tracking. However, Tang et al. [15] presented a semi-automatic graph based approach that reformulates tongue contour tracking as a graph-labeling problem where optimality of segmentations is tuned by both spatial and temporal regularizations. Given an initial set of points, the method finds a set of displacement vectors 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). Their implementation is publicly available [15] as software called “TongueTrack”.

Laporte and Ménard [16] introduced the use of a particle filtering algorithm for tongue tracking combined with an ASM that enforces shape constraints to limit the search space dimensionality. The particle filter enforces strong, yet flexible temporal consistency constraints, allowing swift recovery from error. As the 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 paper.

One limitation to the tracking approaches is that they may drift from the correct answer for a variety of reasons (e.g. the tongue moves too fast, it disappears or gets too blurry, etc.). Xu et al. [17] suggest a trick to reduce this effect, and that is the idea of re-initializing the tracking system from time to time. The authors suggest a re-initialization whenever the current image is sufficiently similar (according to the SSIM criterion [18]) to that used for manual initialization, where the manual segmentation provided by the user can reasonably be re-used. They 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 3.3) for this paper.

Most of the approaches currently documented in the literature and described above, with the exception of machine-learning based methods like Autotrace [10], depend on an initial set of tongue contour points that should be manually given in advance to the system so that the tongue contour can then be tracked over the remaining images. We call these approaches “semi-automatic”.

This study addresses a different but related problem and that is to automatically extract the tongue contour points from an US frame

without prior information about its location, which is a challenging problem. Tongue contour extraction methods based on deep learning, such as Autotrace and its variants [10,11,13] achieve this using large training sets of labeled data. In contrast, we propose to rely solely on the content of a single US image. 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 paper, we also show how this can be achieved.

The biggest advantage of the proposed method is that it eliminates the need for manual intervention. This opens the possibility of real-time tongue detection and tracking on an US machine, which could, for instance, help provide immediate visual biofeedback to the patient during a speech therapy session [19]. Moreover, the detection approach proposed here could be extended and applied to other related applications in medical imaging tasks that involve US.

We divide our ultimate goal into a number of major problems that are dealt with at different levels: 1) automatically segmenting tongue contours from 2D US images which means that the input is a single US frame and the output is an approximate locus of tongue contour points on this frame 2) transforming a semi-automatic tongue contour tracking approach to a fully automated one using the automated tongue segmentation module 3) determining when it might be useful to re-initialize the automated tracking approach (re-initialization).

The remainder of this article is structured as follows: Sections 2 and 3 discuss the details of the proposed approach for the problem of automatically detecting the tongue contour in US images. Section 4 details the experimental framework used to test the proposed approach and the data acquired to do so. Finally, Section 5 is dedicated to reviewing the contributions of this paper, its shortcomings and the work that could be done in the future to improve the current framework.

2. Automatic tongue segmentation

In US images, the echo from the tongue surface generally appears as a continuous bright region. Our 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.

Fig. 1 illustrates the proposed approach for automatic tongue contour segmentation. First, a mask (Section 2.1) is applied to remove the irrelevant information that is present in the input US video sequence. Then, a phase symmetry filter (Section 2.2) is applied to enhance the regions that look like the tongue contour. The enhanced image is binarized (Section 2.3) and processed by a skeletonization module (Section 2.4) 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 2.5) 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 2.6).

2.1. Masking

The first step towards automatic segmentation of 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 for 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. 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.

2.2. 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. [20]) to each frame of the video sequence. Fig. 2 shows the elevation map of the example US frame. The elevation map shows how the highly specular surfaces produce ridges and these are due to US reflecting off the tongue.

Image signals with even and odd symmetry will have real and imaginary Fourier transforms, respectively. Ridge-like features with an axis of symmetry (such as the bright echo caused by the tongue surface in US images) result in the even filter response dominating over the odd filter response. The local 1D phase symmetry measure proposed by Kovesi et al. [20] is precisely the difference between the even filter and odd filter responses:

$$Sym(x) = \frac{\sum_n \lfloor [\,|\, e_n(x) \,|\, - |\, o_n(x) \,|\,] - T \rfloor}{\sum_n A_n(x) + \varepsilon} = \frac{\sum_n \lfloor A_n(x) [\,|\, \cos(\Phi_n(x)) \,|\, - |\, \sin(\Phi_n(x)) \,|\,] - T \rfloor}{\sum_n A_n(x) + \varepsilon} \quad (1)$$

The factor T is a noise compensation term and ε is a small constant so that the denominator will not be equal to zero. This 1D analysis can be extended to 2D by applying it in multiple orientations and forming a weighted sum of the results.

We empirically tuned the number of wavelet scales ($n = 5$) and the number of filter orientations ($\tau = 14$) for our experiments.

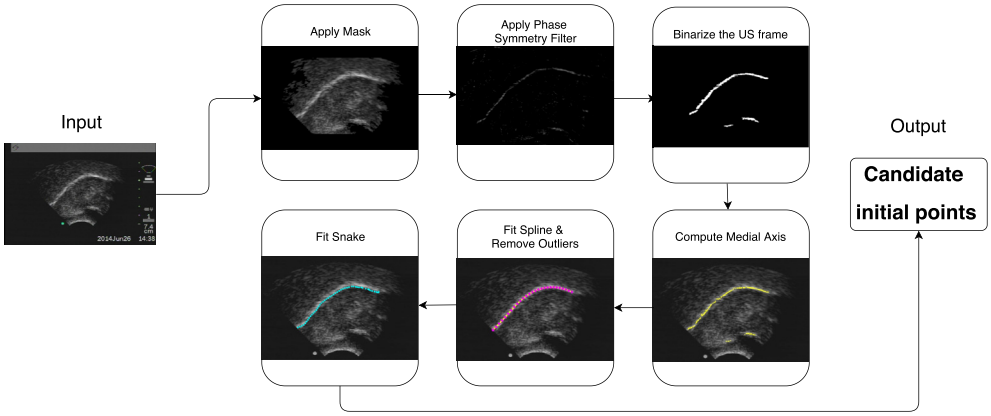

Fig. 1. Block diagram of the automatic tongue contour segmentation method proposed by this paper. For the remainder of this Section, we will consistently be using this US image as an illustrative example for tongue segmentation.

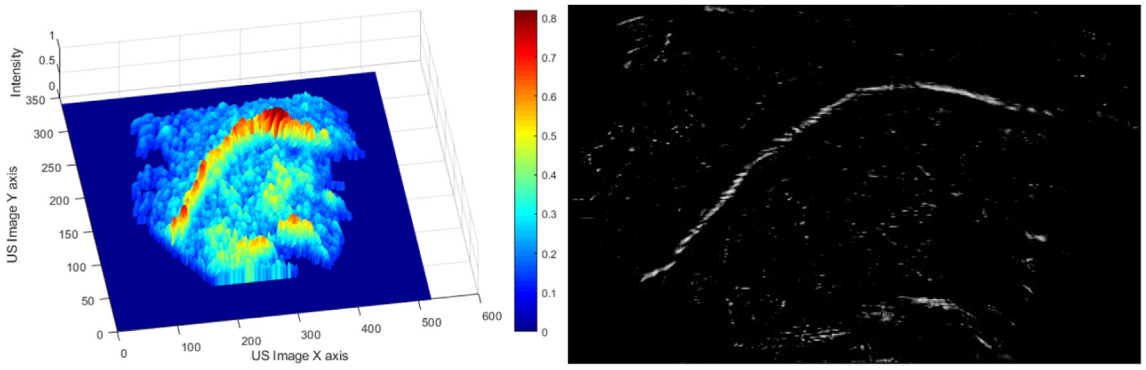

Fig. 2. Left: the elevation map of the masked US image from Fig. 1 where ridges and high peaks show the high intensity regions mostly corresponding to the tongue area. Right: shows the result of applying the phase symmetry filter on the same US image. It can be seen that phase symmetry filter is better at ignoring this speckle noise than the Canny edge detector.

2.3. Binarizing the ultrasound image

To find ROIs, we first binarize the phase symmetry image from previous step using a threshold that is chosen as the median of its intensity values (see Fig. 3). To express this mathematically, let I be the input US image (masked and cropped), I_f be the phase symmetry image, and $\lambda = \text{median}(I_f)$. The binarized image ($I_b$) is obtained by thresholding I_f with the threshold λ. 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: $I_c(i,j) = 1 - I_b(i,j) + I(i,j)I_b(i,j)$.

Let W_k represent the k^{th} white connected component in I_b. An importance score is defined as: $\Psi(W_k) = \text{mean}(I_c(W_k)) \times \text{area}(W_k)$, where $\text{mean}(I_c(W_k))$ represents 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:

$$I_d(i,j) = \begin{cases} \Psi(W_k), & (i,j) \in W_k \ \ \forall\, k \\ 1, & (i,j) \notin W_k \ \ \forall\, k \end{cases}.$$

I_d emphasizes the regions of the US image that have high average intensities as well as a large area. Combining ROI size with ROI average intensity makes it easier to eliminate small white regions that are produced by speckle noise. Finally, we apply Otsu's thresholding method [21] to binarize I_d.

2.4. Computing the medial axis

After the binarization step is performed, our main goal is to extract a single curve representing the tongue contour. For this purpose, we use skeletons (medial axes). The skeleton of a shape is the locus of all points lying inside the shape and having more than one closest point to the boundary of that shape [22]. 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. [23] and have been improved in different applications ([24,25]). To compute the medial axis within a bounded shape, Dimitrov et al. [23] introduced a new measure called Average Outward Flux (**AOF**). **AOF** is defined as the 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 ∂RIf the gradient of the Euclidean distance function to ∂R is given by $\dot{\mathbf{q}}$the **AOF** through ∂R is then defined as: $\textbf{AOF} = \frac{\int_{\partial R}\langle \dot{\mathbf{q}}, \mathbf{N}\rangle ds}{\int_{\partial R} ds}$, 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.

It can be shown 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 [23]. A major advantage of the flux-based method is that **AOF** is a region-based measure 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 jittering effect present in the binarized pixels of narrow tongue ROIs (see Fig. 4).

2.5. Spline fitting and outlier removal

Not all points on the medial axes of ROIs are located near the tongue

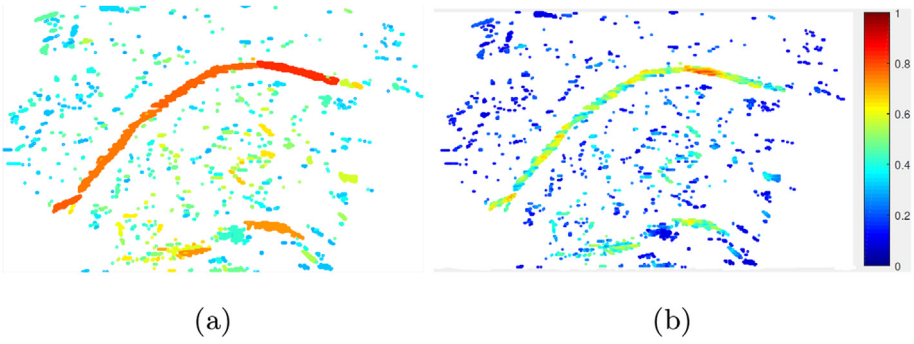

Fig. 3. Left: shows the white regions rank ordered and colored based on their importance score $\Psi(W_k)$. Intensities are colored from blue (for low importance) to red (for high importance). Right: The white region pixels of the obtained binary image (left) are colored based upon the intensity values of the same pixels in the original US image normalized between 0 and 1 (I_c).

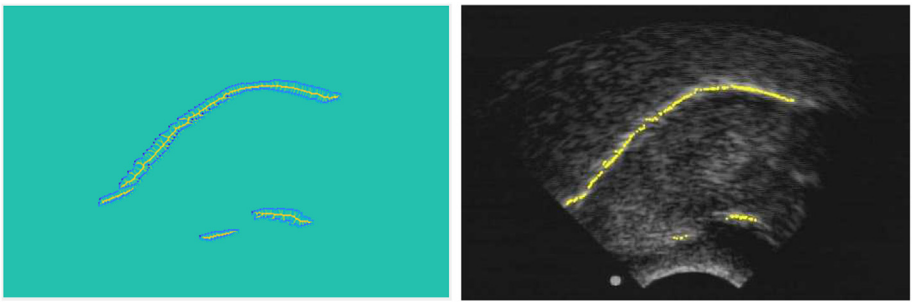

Fig. 4. Left: the average outward flux map applied to our binarized example from previous step. Here, blue shows the boundary of the ROIs, yellow shows the high values of AOF. Right: the skeletal points obtained from the AOF map overlayed on the input US image.

contour (see Fig. 4 right), and we have to somehow remove outliers. To designate candidate points as being close to the tongue contour we use the Density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm, proposed by Ester et al. [26]. 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 Fig. 5), 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 to the main cluster to produce candidate points for initialization of the automatic tongue tracking system.

2.6. Snake fitting

The final step of the proposed automatic tongue segmentation method is to fit an active contour model (snake) to the points obtained from the spline fitting/outlier removal module. This allows the extracted points to adjust to the actual tongue contour points. In our framework, we use the approach of Li et al. [4]. Given a contour $V = \{v_1, v_2, ..., v_n\}$ where the $v_i, i = 1, ..., n$ are the points generated by the spline fitting and outlier removal module, the total snake energy to be minimized is defined as:

$$E'_{\text{snake}} = \sum_{i=1}^{n} \alpha E_{\text{int}}(v_i) + \beta E_{\text{gradient}}(v_i) E_{\text{band}}(v_i). \tag{2}$$

In Eq. (2), E_{int} is a classical internal energy term which encodes soft shape constraints with respect to the amount of stretching and bending of the snake, and E_{gradient} is a classical external energy term encoding

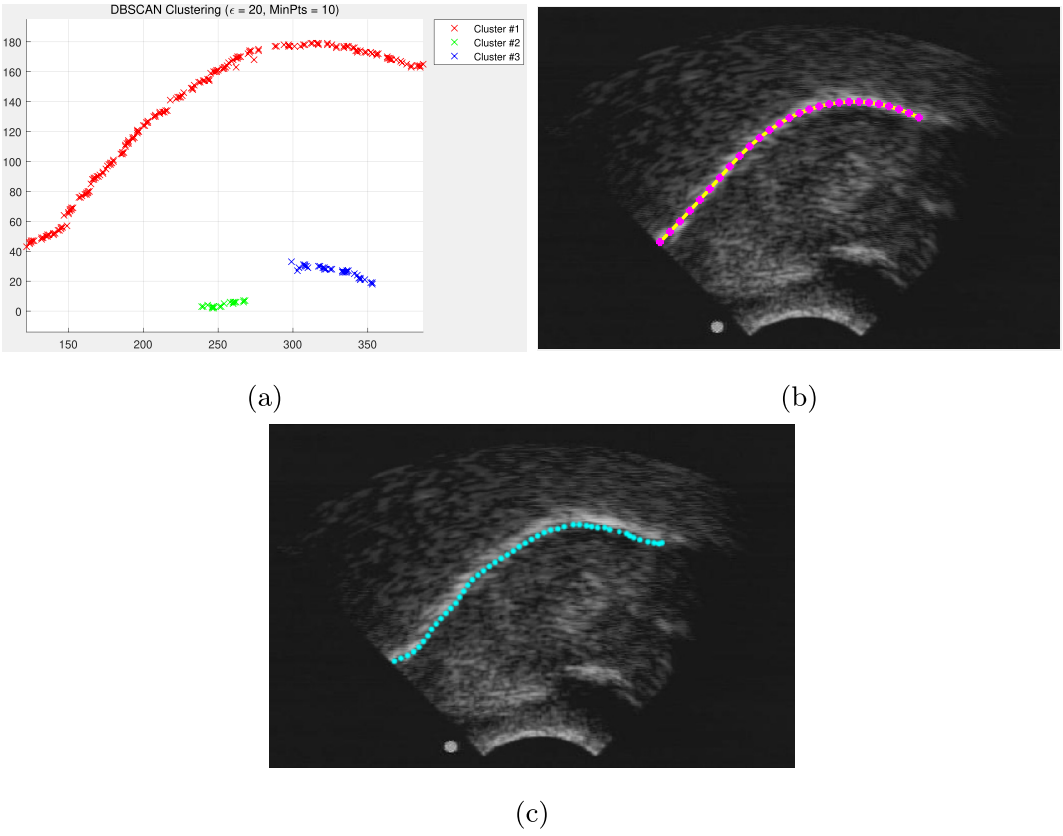

Fig. 5. a):example of how the DBSCAN clustering algorithm applies to the generated skeletal points. b):The result of spline fitting and outlier removal steps on the US example. The continuous yellow curve shows the resulting spline fit where outliers are removed, and the pink circle dots show the sampled points on the spline fit we use to fit a snake in the next step. c):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.

the strength of the image intensity gradient in the vicinity of a vertex. As their precise definitions are in no way unusual, the reader is referred to Li et al.'s paper [27] and Laporte and Ménard's paper [16] for mathematical and implementation details, respectively. On the other hand, the E_{band} factor that modulates the external (gradient) energy term is called band energy and is a unique feature of Li et al.'s model that was designed specifically for application to tongue contour extraction in US images. The band energy factor E_{band} measures the contrast between the bright region above the contour and the region immediately below it:

$$E_{\text{band}}(v_i, I) = \begin{cases} E_{\text{penalty}}, & \text{if contrast}(v_i, I) < 0 \\ 1 - \text{contrast}(v_i, I), & \text{otherwise} \end{cases}, \tag{3}$$

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

To understand the role of E_{band}, we shall consider the bright white band resulting from the reflection of US at the interface with air above the tongue surface. The lower edge of this white band is the surface of the tongue sought for analysis by speech scientists. The classical external energy term E_{gradient} used in many snake formulations is based purely on the image gradient information. Thus, it is a challenge to distinguish between the upper and lower edges of the white band. [4] introduced the band energy factor to guide the snake preferentially towards the lower edge of the bright white band. In the absence of image contrast of the correct polarity in the vicinity of the evaluated vertex, E_{band} takes on a penalty value whose multiplicative action strongly discourages attraction to irrelevant high gradient features (typically US speckle artifacts, or the upper edge of the white band). The strength of this action increases with the strength of the irrelevant gradient.

Based on the implementation proposed by Laporte and Ménard [16], $C = \max_{v_i} ||\nabla I(v_i)||$. In this work, we set $\alpha = 0.8$, $\beta = 0.2$ and penalty = 2, as suggested by Laporte and Ménard [16]. The result of snake fitting step is shown in Fig. 5c.

3. Applications to tongue tracking

In this section, we discuss how to use the automatic tongue detection method discussed in Section 2 to improve the semi-automatic tracking framework proposed by Laporte and Ménard [16]. The improvements are twofold: (1) automatic tongue detection is used to initialize the tracking framework, thereby making it fully automated, and (2) it is also used within a periodic re-initialization strategy that improves tracking accuracy and reduces the amount of manual intervention required to correct bad segmentations after processing.

3.1. Semi-automatic tongue tracking framework

To evaluate the usefulness of our automatic segmentation approach, we apply it to the multi-hypothesis framework of Laporte and Ménard [16] for tongue tracking. In this approach, firstly, an ASM is built based on a dataset of segmented tongue contours where each contour is represented by a compact vector of 6 variables containing location, a scale, and first three principal components of vertices obtained from principal component analysis (PCA). 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.

Finally to track the tongue contour at each time step, a particle filter tracking approach is implemented, where each particle is fitted as a snake to the image by minimizing the simplified snake energy: $E_{\text{snake}} = \sum_{i=1}^{n} \alpha E_{\text{int}}(v_i) + \beta E_{\text{gradient}}(v_i)$, once this is done, the likelihood of each particle is established using: $E'_{\text{snake}} = \sum_{i=1}^{n} \alpha E_{\text{int}}(v_i) + \beta E_{\text{gradient}}(v_i) E_{\text{band}}(v_i)$. 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. In this step, 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. The number of particles is chosen adaptively at every frame, and allows the cumulative likelihood of the evaluated particles reach a certain threshold: $T = 7\exp(-E(V_{init}, I_{init}))$ where $-E(V_{init}, I_{init})$ is the energy of the manually-segmented contour in the initialization frame. The reader is referred to Laporte and Ménard's original paper [16] for more details.

3.2. Automatically finding candidate initial points within a window of X frames

The proposed automatic tongue segmentation method is not perfect and can sometimes produce erroneous results. However, it can be applied to more than one image in any given US video sequence, thus increasing the likelihood of obtaining one correct result that can be used to automatically initialize tracking.

This section describes two quality measures that are predictive of the reliability of our segmentation results so that a suitable image can be selected to initialize tracking with high confidence. Let the skeletal points from the segmentation process be denoted by $V = (v_1, ..., v_n)$, where the points v_i, $i = 1, ..., n$ are sorted by position from left to right on the US image. We suggest two assessment criteria, the first one reflects the fact that points that represent the tongue should not be from very disjoint groups of ROIs. This leads to the first reliability measure as the inverse of total contour length:

$$\Gamma_1 = \left(\sum_{i=1}^{n-1} ||v_i \vec{v}_{i+1}|| \right)^{-1}. \tag{4}$$

Low Γ_1 means that points generated from our approach are more disjoint from each other (there are gaps in the contour). However, Γ_1 would be quite high if only a small segment of the tongue (e.g. the middle) had been segmented. To address this, we a second score, which assesses the completeness of the tongue contour. As a correctly segmented tongue contour typically occupies a wider range of positions along the x axis than an incomplete one, the second score was designed as the ratio of the coverage length of connected candidate points on the x-axis to the image width:

$$\Gamma_2 = \sum_{i=1}^{n-1} d_{v_i v_{i+1}} \cos\angle(\overrightarrow{v_i v_{i+1}}, \vec{x})/W, \tag{5}$$

where:

$$d_{v_i v_{i+1}} = \begin{cases} ||v_i \vec{v}_{i+1}||, & \text{if } ||v_i \vec{v}_{i+1}|| \le 2\sqrt{2} \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

W is the image width and $\vec{x}$ represent the x-axis. In the second measure, consecutive skeletal points are considered to be connected if they have a distance less than $2\sqrt{2}$ pixels. The final score is computed based on the combination of these two scores:

$$\Gamma = \Gamma_1^{\eta_1} \Gamma_2^{\eta_2}, \tag{7}$$

where η_1 and η_2 are chosen empirically.

Within a window of X frames from the starting frame, we choose the frame with the highest segmentation reliability score Γ 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 3.1. In our experiments we set $X = 10$.

3.3. Re-initialization

Any tongue contour tracker may temporarily or permanently lose

the trajectory and fail due to a variety of reasons. Inspired by Xu et al. [17], we added a module to our framework that is able to automatically re-initialize tracking from time to time.

The criteria used to do these resets look for two types of situations. One is when the similarity between the current frame and last chosen initial frame is low. For this, we used Structural Similarity index measure (SSIM) which measures the similarity between two images [18]. The second criterion is to do the automatic reset when the number of particles from the semi-automatic tracker ([16]) gets bigger than a particular threshold. 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. In our experiments, we set the thresholds for these two criteria as the following: $t1 = 0.9$ and $t2 = 400$, where the first threshold is applied on SSIM and the second threshold is used on the number of particles. These numbers were empirically chosen to minimize segmentation errors (MSD) in two validation US video sequences. Every time that either of these criteria is met, the semi-automatic module is paused and the automatic tongue segmentation is performed anew according to the procedure described in Section 3.2.

4. Experiments

4.1. Data acquisition

The main set of data used in this work is the same US video sequences that were presented in Ref. [16]. In this setup, the machine used for recording is a Sonosite 180 plus US scanner with a micro-convex 8-5 MHz bandwidth transducer set at a 84° field of view. For this application, gain, TGC and depth were chosen for best tongue visibility in each data set. The resulting imaging depths ranged from 7.4 cm to 15 cm. After recording, the US video sequences were manually segmented by a trained operator using the interface provided by the Autotrace software [10]. In our experiments, we used the 16 free speech US video segments described by Laporte and Ménard [16] containing a total of 23776 frames, where 2 of these US video segments containing 2121 frames were used for fine-tuning of our system parameters leaving the other 14 video sequences for testing. Each segment was between 20 s and 84 s 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. These data were acquired with approval by the Research Ethics Boards of Université du Québec à Montréal and Sainte-Justine University Hospital Research Centre.

The manual segmentations were used as ground truth to validate our segmentation approach. To ensure the quality of this gold standard, the intra-rater and inter-rater variability of the manual segmentation process were evaluated using additional manual segmentations on a small subset of the original video data, specifically a 154 frame (5 s) continuous speech segment from subject C6. To measure intra-rater variability, the primary rater segmented this small subset over two months after completing the segmentation of the entire video data set. A second rater, also an experienced operator with expert knowledge of tongue US images, also segmented the 154 frame subset of the data, providing the means to measure inter-rater variability. Intra-rater and inter-rater variabilities were assessed using the mean-sum-of-distances measure (see Section 4.2) and respectively evaluated to 0.86 mm ± 0.23 mm and 1.05 mm ± 0.39 mm.

4.2. Segmentation error measures

To evaluate the accuracy of segmentation and tracking algorithms studied in this paper, we used three error measures. One is the mean sum of distances (MSD) proposed by Li et al. [4], and the other two are shape-based measures.

The 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, 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:

$$MSD(U, V) = \frac{\sum_{j=1}^{n} \min_i ||v_j - u_i|| + \sum_{i=1}^{n} \min_j ||u_i - v_j||}{2n}. \quad (8)$$

The other error measures we take into consideration are related to the curvature and asymmetry of the segmented tongue contour, and inspired by the definitions of Ménard et al. [28]. These capture linguistically relevant shape features. Let us consider 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. 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 point C is the point of the tongue contour that is farthest away from the line jointed A and B, and point D is its projection on the mentioned line (see Fig. 6).

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

$$\kappa = \frac{||CD||}{||AB||}, \ \gamma = \frac{||AD||}{||DB||}. \quad (9)$$

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

$$\text{acc}_\kappa = 1 - \frac{|\kappa_{\text{met}} - \kappa_{\text{gt}}|}{\kappa_{\text{gt}}}, \ \text{acc}_\gamma = 1 - \frac{|\gamma_{\text{met}} - \gamma_{\text{gt}}|}{\gamma_{\text{gt}}}, \quad (10)$$

where κ_{met} is the score of a contour computed by a specific method, and κ_{gt} is the curvature score computed for the ground truth data.

4.3. 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 3.2, and the other is the semi-automated

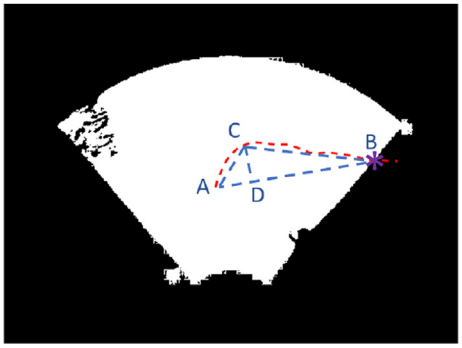

Fig. 6. Assuming the dashed red line is representing the computed tongue contour, this figure shows how our approach computes the three points A, B, C and D. The purple star shows the intersection of tongue contours with the mask on either side.

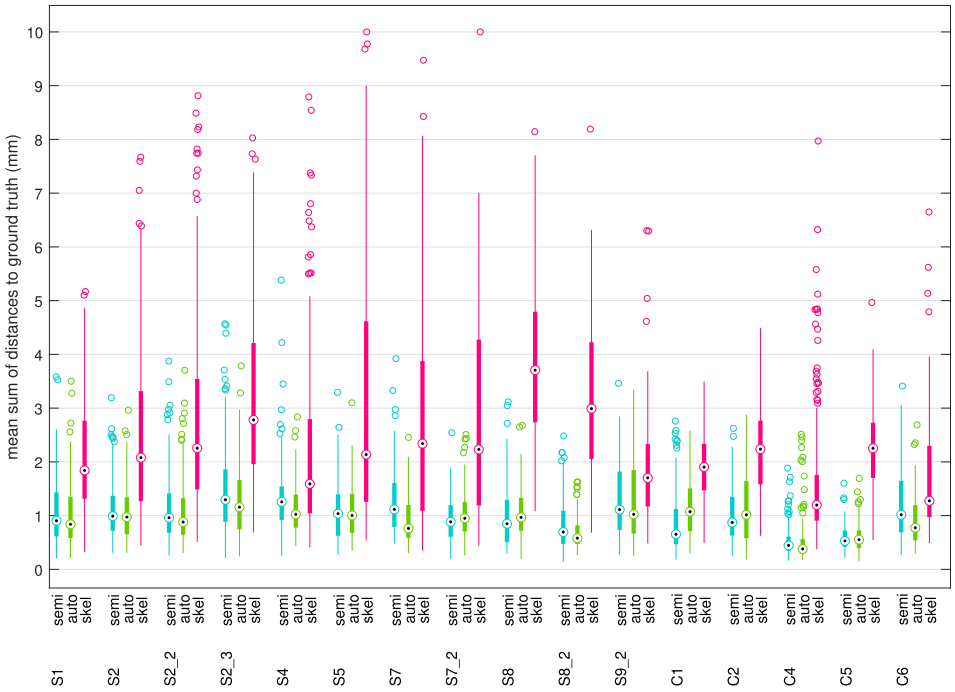

Fig. 7. This figure compares the MSD values of tongue contour points computed from three approaches: our automatic segmentation approach before snake fitting (**skel**), our fully automatic tracking approach (**auto**), and the semi-automatic approach of [16] (**semi**), where all three are compared 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 this figure, the central mark indicates the median, and the bottom and top edges of the box indicate the 25^{th} and 75^{th} percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the 'o' symbol.

method (labeled "semi") of [16], 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 section. Fig. 7 compares the MSD across these three methods for our 16 different video sequences. 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: 2.84 ± 1.58, semi: 1.05 ± 0.63, auto: 1.01 ± 0.57, averaged over all video sequences). 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 (temporal information) and therefore is not highly regularized. We examine some of the failure cases of frame-by-frame segmentation in comparison with the other two approaches in section 4.4. Note that the same segmentation method, when used to automatically initialize tracking from a carefully selected frame, yields MSD scores quite similar to the semi-automatic approach where the initial points are captured manually, and on the same order of magnitude as the inter-rater manual segmentation variability. This means that our approach can be used to automatically initialize the tracker without loss of accuracy.

We also note that the accuracy of the frame-by-frame segmentation method appears to be generally poorer and more variable in video sequences from speakers with Steinert's disease than in speakers from the control group. This is likely because image quality tends to be worse for the impaired speakers, for whom tuning the acquisition setup is more difficult due to their physical limitations, and whose motor control difficulties sometimes lead them to produce more lateral tongue motion (impairing its visibility in the image) than speakers in the control group. Interestingly, even though the frame-by-frame segmentation produced by our method is less accurate in these speakers, careful selection of a frame to use for initialization of the tracker, as proposed in Section 3.2, generally yields accurate results (see Section 4.5 for a detailed analysis), prior to the tracking algorithm taking over the segmentation task, leading to overall tracking-based segmentation accuracy and precision comparable to that measured in the control group.

Measuring the extent to which the proposed algorithm would preserve linguistically relevant shape features is important, as one of the ultimate goals of this research work is to detect tongue contours for shape analysis purposes. Considering this point, in addition to MSD, we measured tongue curvature and asymmetry similarity scores (see Fig. 8). This figure shows the accuracy measures that are based on shape similarities between each of these methods 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. Furthermore, the results show the same trend as the MSD results in terms of the differences in the performance of the frame-by-frame segmentation method between impaired and control speakers. Again, these differences essentially vanish when the method is used to initialize the tracker from an automatically selected frame.

4.4. Sample results and challenges

This section demonstrates some tracking results including different examples of successes and failures for the various methods tested in this paper, where they are all computed 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 segmentation and tracking tasks. We start with cases where all approaches are finding accurate tongue contours. Fig. 9 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 all three algorithms were able to find proper sets of points that were close to the ground truth data.

Fig. 10 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, which, as we will see in Section 4.5, was successfully accomplished using the proposed segmentation reliability measures. 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

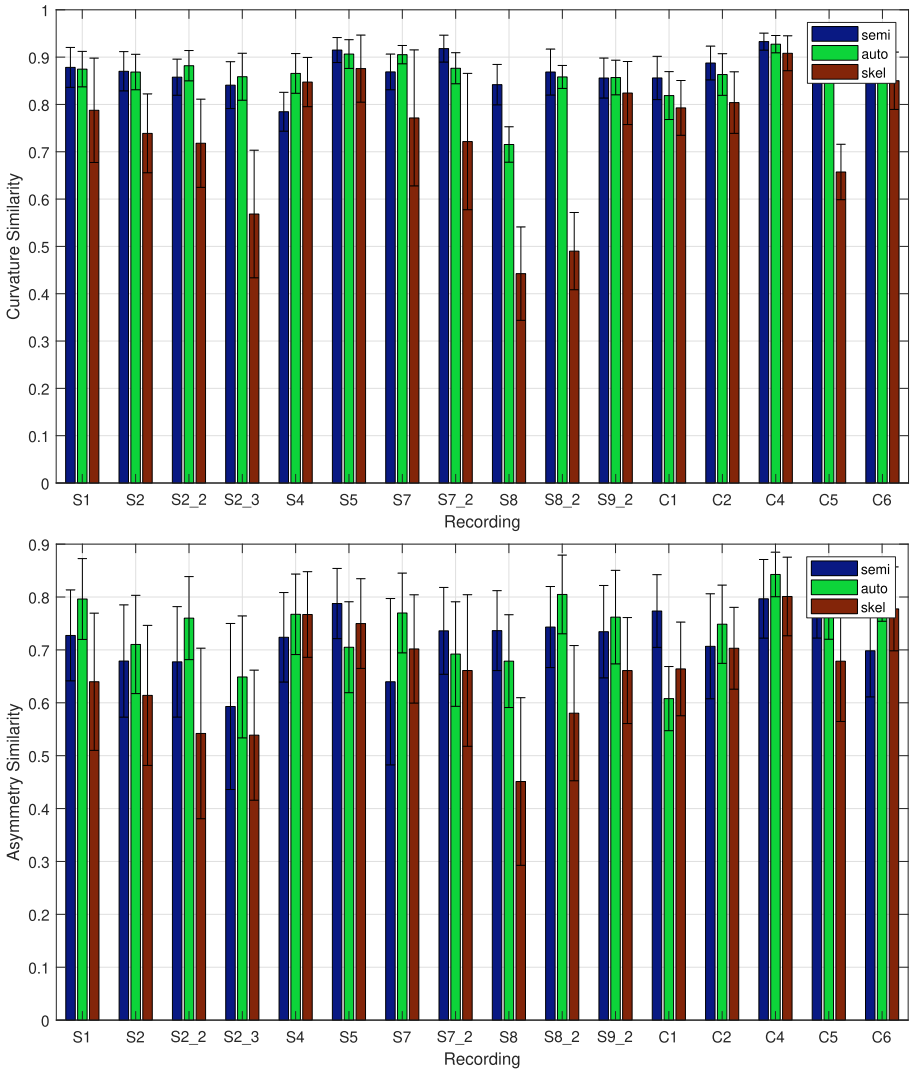

Fig. 8. This figure shows box plots of curvature (top) and asymmetry (bottom) similarity between contours extracted using each of the three methods of Fig. 7 and the ground truth data. Error bars represent one standard deviation above and below the average.

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. 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 Figs. 11 and 12).

Section 4.7 includes examples illustrating how our re-initialization strategy would resolve some of these cases.

4.5. Analyzing reliability scores

To validate the reliability measures introduced in Section 3.2, we examine their relationship to the MSD between the automatically segmented skeletal points and the manually segmented ground truth tongue contours. A contour is assumed to be segmented well if the

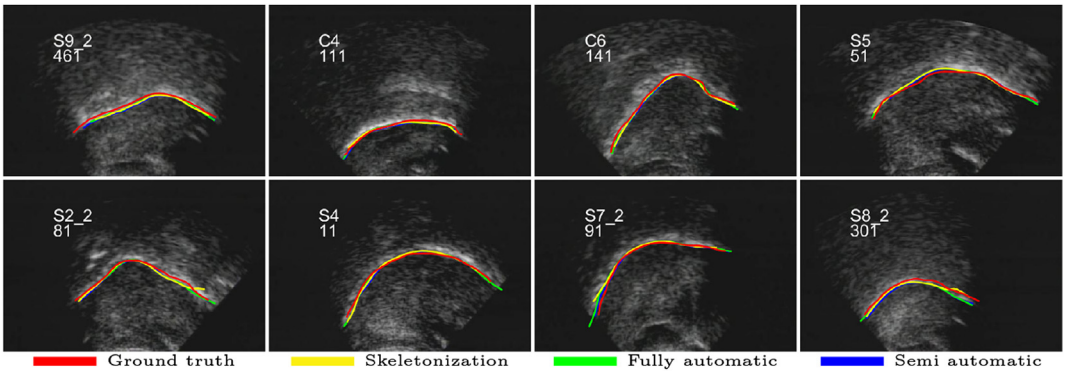

Fig. 9. Sample contour points obtained from different approaches where the computed points are similar to ground truth data.

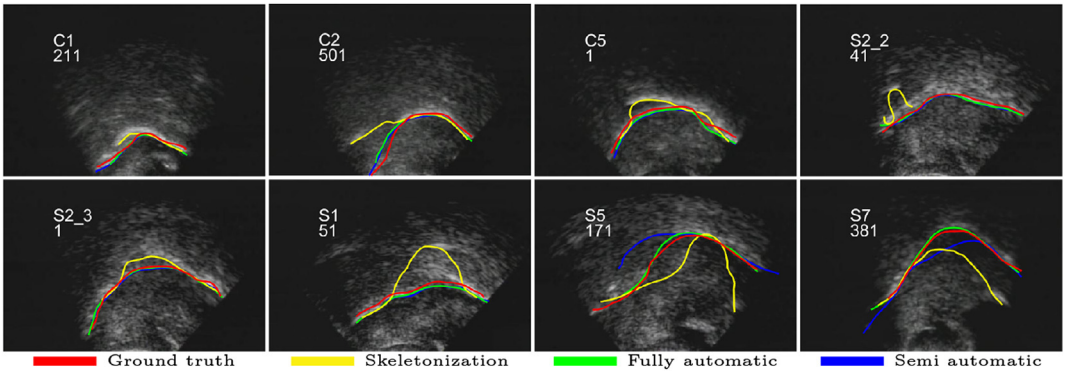

Fig. 10. Example cases where skeletal points fail in segmentation but the fully automatic approach generates points close to ground truth manually segmented points.

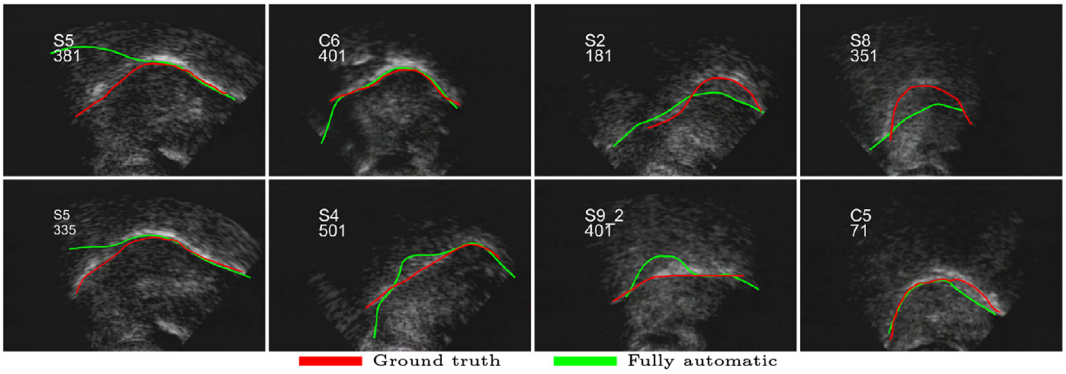

Fig. 11. Example cases where the fully-automatic approach fails in tracking.

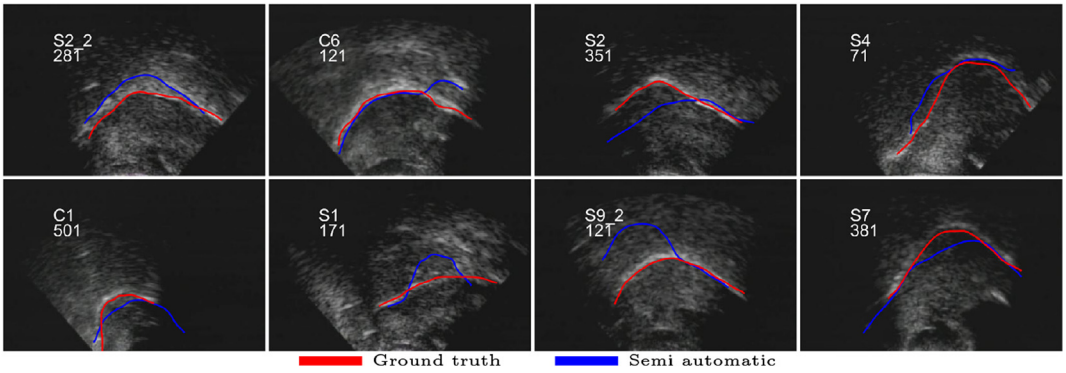

Fig. 12. Example cases where the semi-automatic approach fails in tracking.

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:

$$g(V_i^{sk}, V_i^{gt}) = \frac{1}{MSD(V_i^{sk}, V_i^{gt})}, \tag{11}$$

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 g by its maximum value over all contours of all US frames in our experiments, yielding scores between 0 and 1:

$$f(V_i^{sk}, V_i^{gt}) = \frac{g(V_i^{sk}, V_i^{gt})}{\max_i g(V_i^{sk}, V_i^{gt})}. \tag{12}$$

We examine the relationship of $f(V_i^{sk}, V_i^{gt})$ to the combination of the two reliability 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 (12)). Fig. 13a 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 Fig. 13a and 13b. Logarithmic scales are used for improved visualization. Though reliability scores are fairly broadly distributed in Fig. 13a and b, both of these plots show a similar trend to the inverse MSD. Fig. 13c shows the result of combining the Γ_1 and Γ_2 scores according to Equation (7) in relation to sorted f scores. Fig. 13d shows the boxplot of the ratio of the Γ score to the normalized inverse MSD (f score) along each video separately. This ratio is clearly quite close to 1 in all cases, indicating that there is a

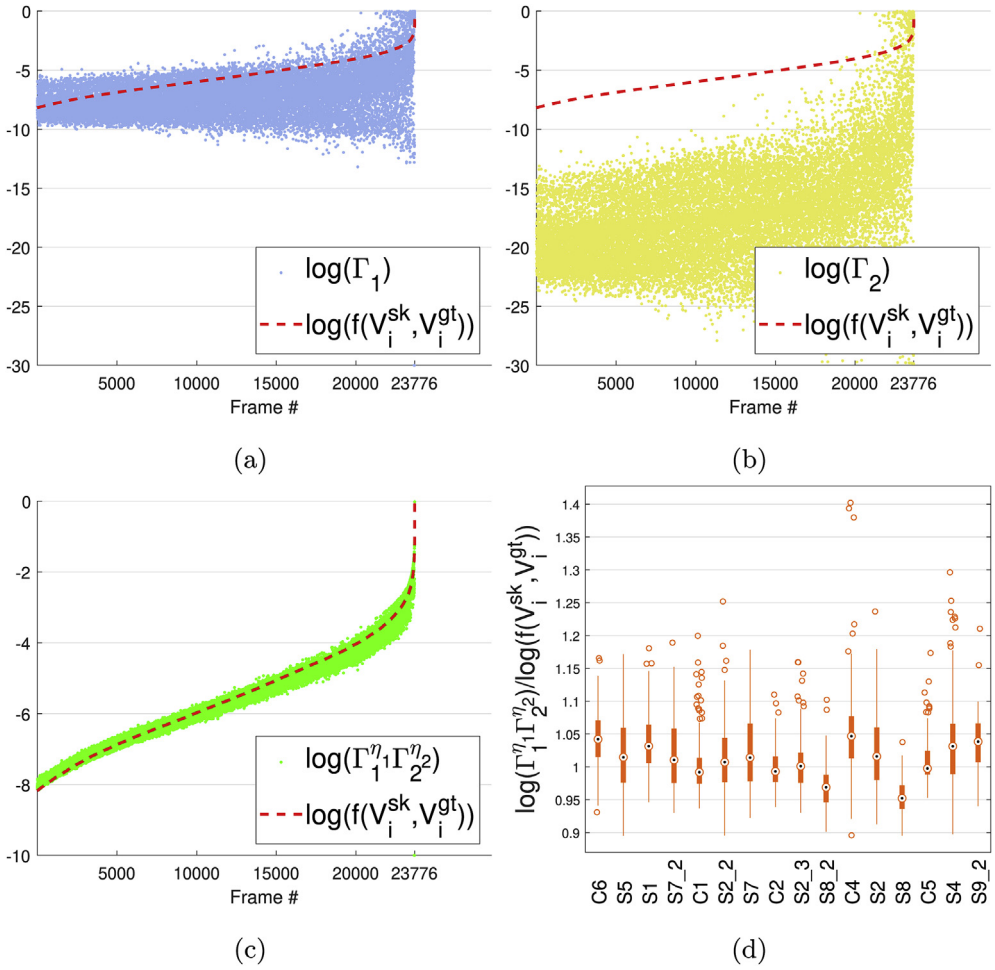

Fig. 13. Analysis of the Γ score introduced in section 3.2 (see the text for more information).

strong relationship between the Γ score and the inverse MSD. This suggests that the proposed Γ score which is the combination of both scores Γ_1 and Γ_2, 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 4.3, which showed little difference in error between the fully automated and semi-automated tracking methods.

4.6. Robustness to changes in acquisition setup

To validate the reliability of our proposed methodology when performing with different US acquisition conditions, we used an additional data set in our experiments, which is publicly available as a companion to the TongueTrack software [15,29]. The setup used for this data set is a General Electric Logiq Alpha 100 MP US scanner with a model E72 6.5 MHz transducer that uses a 114° microconvex array. There were two video sequences available with manual segmentations, one with 545 frames and one with 436 frames, where the video camera capture rate is 30 frames per second. The parameters of our proposed segmentation method were kept to the same values as in all our other experiments and were in no way fine-tuned for the new data set. MSD results are shown for semi-automatic tracking, fully automatic tracking and frame-by-frame segmentation are shown in Fig. 14. While the MSD achieved by the frame by frame method on this additional data set is slightly larger than for the main data sets presented in this paper, it remains on the same order of magnitude. More importantly, the fully automatic tracking method, which uses this type of frame-by-frame analysis only for one carefully and automatically selected image, yields results of very similar quality as in the previously demonstrated data sets, while also performing quite closely to the semi-automatic tracking method. This demonstrates the robustness of the proposed approach (including the choice of adjustable parameters) with respect to imaging conditions.

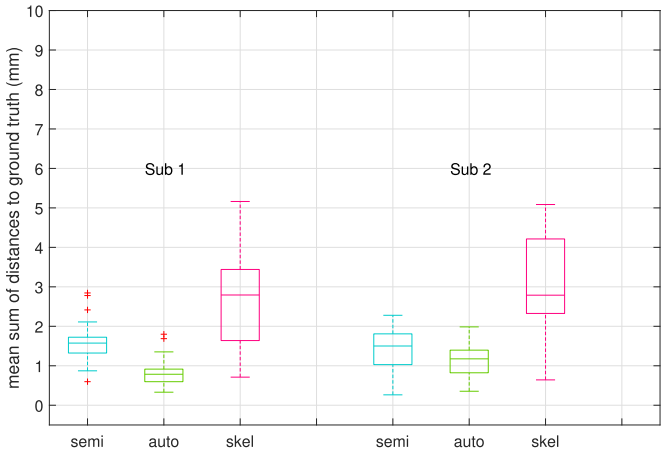

Fig. 14. This figure compares the MSD values of tongue contour points computed from three approaches: our automatic segmentation approach before snake fitting (skel), our fully automatic tracking approach (auto), and the semi-automatic approach of [16] (semi) for the TongueTrack demonstration data set.

Table 1
Comparison of the mean and standard deviation of all MSD values across 16 videos for the three methods of **semi**, **auto**, and **re-init**.

Measure	Fully-auto	Semi-auto	Re-initialization
MSD Mean (mm)	1.01	1.05	0.63
MSD Standard Deviation (mm)	0.57	0.63	0.35

4.7. Re-initialization

This section, discusses experiments done using the tracker re-initialization approach described in section 3.3. 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 and Ménard [16], 2) the fully automatic approach proposed in this paper and 3) the fully automatic approach proposed in this paper with an added re-initialization module. Putting the three approaches in the experimental setup as before, and comparing results we see that the re-initialization approach decreases the MSD compared to both the approaches without re-initialization. Table 1 shows the performance boost that re-initialization 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).

5. 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 paper 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 contributions of this paper can be summarized as follows:

1. The automatic tongue segmentation approach is original compared to existing ones because 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, 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. The proposed tongue segmentation reliability scores are novel and 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.
3. This proposed system offers a variety of benefits over the state-of-the-art. Its accuracy, when used to automatically initialize Ménard and Laporte's semi-automatic tracking method [16], is indistinguishable from that of the original semi-automatic approach, which was in turn shown to be more accurate than Edgetrak [27], TongueTrack [15] and Autotrace [10]. Moreover, using automatically generated contour points for initialization has never been reported in previous work. Foregoing manual initialization in this fashion represents an important step towards facilitating analysis of the large US data sets acquired for articulatory studies, as tongue contour extraction over large numbers of videos can be computed as part of a batch-processing script. Previously, this could only be achieved using learning-based methods such as Autotrace [10], at the expense of requiring copious amounts of correctly segmented training data. 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 3.3). While such a re-initialization approach has been proposed before by Xu et al. [17], its application was previously limited to searching for images that were similar to the one used for manual initialization as there was no other reliable tongue contour available. Automatic segmentation, as proposed here, has allowed us to include a new re-initialization criterion based on the uncertainty of the tracker. The resulting system yields even lower MSD error than Laporte and Ménard's method. While the strict significance of this result is limited by the intra-rater and inter-rater variability of the manual segmentations used as ground truth, the system on the whole represents a significant gain of time, repeatability and objectivity for speech scientists who must segment large amounts of tongue US images on a regular basis.

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 conceivably be extended to work in 3D US images. The masking, phase symmetry filtering, binarization and skeletonization steps are all transferable to 3D. While the slow volume rates and limited spatial resolution currently available for 3D US scanning may to some extent limit this prospect, the additional information afforded by 3D images (e.g. information regarding structure connectivity between adjacent para-sagittal slices) may also help constrain the solution space and reduce spurious structure detections and improve accuracy.

The final outcome of such a system would be a set of tongue surface points. Extend the system proposed in this paper 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.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada and the Fonds de Recherche Québécois - Nature et Technologies.

The authors also wish to thank Mr. Lambert Beaudry for his assistance in evaluating the inter-rater variability of manual segmentation.

References

[1] M. Stone, A guide to analysing tongue motion from ultrasound images, Clin. Linguist. Phon. 19 (6–7) (2005) 455–501.
[2] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vis. 1 (4) (1988) 321–331.
[3] Y.S. Akgul, C. Kambhamettu, M. Stone, Extraction and tracking of the tongue surface from ultrasound image sequences, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1998, pp. 298–303.
[4] M. Li, C. Kambhamettu, M. Stone, Automatic contour tracking in ultrasound images, Clin. Linguist. Phon. 19 (6–7) (2005) 545–554.
[5] A. Roussos, A. Katsamanis, P. Maragos, Tongue tracking in ultrasound images with active appearance models, IEEE International Conference on Image Processing, IEEE, 2009, pp. 1733–1736.
[6] G. Hamarneh, T. Gustavsson, Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images, Comput. Cardiol. 2000 (2000) 115–118.
[7] S. Ghrenassia, C. Laporte, L. Ménard, Statistical Shape Analysis in Ultrasound Video Sequences: Tongue Tracking and Population Analysis, Ultrafest VI, (2013), pp. 53–55.
[8] C.-Y. Chien, J.-W. Chen, C.-H. Chang, C.-C. Huang, Tracking dynamic tongue motion in ultrasound images for obstructive sleep apnea, Ultrasound Med. Biol. 43 (12)

(2017) 2791–2805.
[9] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, Proceedings of the DARPA Image Understanding Workshop, 1984, pp. 121–130.
[10] I. Fasel, J. Berry, Deep belief networks for real-time extraction of tongue contours from ultrasound during speech, 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 1493–1496.
[11] D. Fabre, T. Hueber, F. Bocquelet, P. Badin, Tongue tracking in ultrasound images using eigentongue decomposition and artificial neural networks, in: Sixteenth Annual Conference of the International Speech Communication Association, pp. 2410–2414.
[12] T. Hueber, G. Aversano, G. Cholle, B. Denby, G. Dreyfus, Y. Oussar, P. Roussel, M. Stone, Eigentongue feature extraction for an ultrasound-based silent speech interface, IEEE International Conference on Acoustics, Speech and Signal Processing, 1 IEEE, 2007, pp. I–1245.
[13] A. Jaumard-Hakoun, K. Xu, P. Roussel-Ragot, G. Dreyfus, B. Denby, Tongue Contour Extraction from Ultrasound Images Based on Deep Neural Network, arXiv preprint arXiv:1605.05912.
[14] T.G. Csapó, S.M. Lulich, Error analysis of extracted tongue contours from 2d ultrasound images, Sixteenth Annual Conference of the International Speech Communication Association, 2015.
[15] L. Tang, T. Bressmann, G. Hamarneh, Tongue contour tracking in dynamic ultrasound via higher-order mrfs and efficient fusion moves, Med. Image Anal. 16 (8) (2012) 1503–1520.
[16] C. Laporte, L. Ménard, Multi-hypothesis tracking of the tongue surface in ultrasound video recordings of normal and impaired speech, Med. Image Anal. 44 (2018) 98–114.
[17] K. Xu, T. Gábor Csapó, P. Roussel, B. Denby, A comparative study on the contour tracking algorithms in ultrasound tongue images with automatic re-initialization, J. Acoust. Soc. Am. 139 (5) (2016) EL154–EL160.
[18] Z. Wang, A.C. Bovik, A universal image quality index, IEEE Signal Process. Lett. 9 (3) (2002) 81–84.
[19] B. Bernhardt, B. Gick, P. Bacsfalvi, M. Adler-Bock, Ultrasound in speech therapy with adolescents and adults, Clin. Linguist. Phon. 19 (6–7) (2005) 605–617.
[20] P. Kovesi, et al., Symmetry and asymmetry from local phase, Tenth Australian Joint Conference on Artificial Intelligence, 190 1997, pp. 2–4.
[21] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[22] H. Blum, A transformation for extracting new descriptors of shape, Models for the Perception of Speech and Visual Form, 5 1967, pp. 362–380.
[23] P. Dimitrov, J.N. Damon, K. Siddiqi, Flux invariants for shape, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1 IEEE, 2003.
[24] M. Rezanejad, K. Siddiqi, Flux graphs for 2d shape analysis, Shape Perception in Human and Computer Vision, Springer, 2013, pp. 41–54.
[25] M. Rezanejad, B. Samari, I. Rekleitis, K. Siddiqi, G. Dudek, Robust environment mapping using flux skeletons, IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2015, pp. 5700–5705.
[26] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, KDD (1996) 226–231.
[27] M. Li, C. Kambhamettu, M. Stone, Tongue motion averaging from contour sequences, Clin. Linguist. Phon. 19 (6–7) (2005) 515–528.
[28] L. Ménard, J. Aubin, M. Thibeault, G. Richard, Measuring tongue shapes and positions with ultrasound imaging: a validation experiment using an articulatory model, Folia Phoniatrica Logop. 64 (2) (2012) 64–72.
[29] L. Tang, G. Hamarneh, T. Bressmann, A machine learning approach to tongue motion analysis in 2d ultrasound image sequences, International Workshop on Machine Learning in Medical Imaging, Springer, 2011, pp. 151–158.