Automated detection of the tongue surface in sequences of ultrasound images

Michael Unser
Biomedical Engineering and Instrumentation Program, Building 13, Room 3W13, National Center for Research Resources, National Institutes of Health, Bethesda, Maryland 20892

Maureen Stone
Department of Rehabilitation Medicine, Building 10, Room 6S235, National Institutes of Health, Bethesda, Maryland 20892

(Received 7 June 1991; accepted for publication 31 December 1991)

An image processing system has been developed for a Macintosh II personal computer. It is designed to process sequences of sagittal tongue sections that are digitized in real time and stored in standard tagged image file format (TIFF). The successive processing steps are: (a) a low-pass filter for noise reduction, (b) a resampling of the sector of interest in polar coordinates, (c) a matched filter (vertical differentiator) for the enhancement of the tissue/air interface in the surface region of the tongue, and (d) an extraction of border points by searching for an optimal radial path along the angular dimension. This latter task is achieved by dynamic programming, which has the following advantages. First, due to the use of a global criterion to guide the detection, it is very robust. Second, as a result of certain restrictions of the allowable transitions, the extracted contours are smooth. Finally, the method permits the specification of particular predefined contour points. This system was implemented in a program that can handle image sequences in a fully automatic mode. Results obtained using ultrasound data are presented.

PACS numbers: 43.70.Jt, 43.85.Ta

INTRODUCTION

Imaging techniques such as magnetic resonance imaging (MRI), computed tomography (CT), or x-ray provide valuable tools for the study of the configurations of the vocal tract (Baer *et al.*, 1987, 1988; Johansson *et al.*, 1983; Keller and Ostry, 1983). Ultrasound, in particular, is an attractive way to monitor the movement of the tongue as it allows for measurements of time-varying features, and does not expose the subject to radiation (Stone *et al.*, 1987, 1988; Watkin and Rubin, 1989; Stone, 1990). However, quantification of these data using personal computers has been slow and tedious, and, for these reasons, usually limited to specific target images. The purpose of this paper is to address the problem by providing a method to automate and increase the speed of analysis of ultrasound images of the tongue using a PC based system. The ultimate goal is to analyze every single frame.

Ultrasound images of the tongue are usually quite noisy. Further, they exhibit a number of high-contrast edges unrelated to the structure of interest. In addition, the edge corresponding to the tongue surface may be interrupted in places. Accordingly, the automatic detection of these borders is by no means a trivial task to be solved by standard image processing techniques, e.g., gray-level thresholding, segmentation or edge detection (Pratt, 1978; Jain, 1989). Consequently, it was necessary to develop specific processing methods. Previous work designed to detect myocardial borders in *M*-mode echocardiograms provided a basis for approaching this problem (Unser *et al.*, 1989).

In our design, the image processing system is modular; it is represented schematically in Fig. 1. The first component is a preprocessing unit that reduces noise. The second step is a change of spatial coordinate system that also provides the delimitation of the area of interest. The next component is a matched filter that enhances the air/tissue interface. Finally, a contour tracking algorithm extracts the tongue's surface by making explicit use of connectivity and smoothness constraints.

I. IMAGE PROCESSING SYSTEM

The purpose of this section is to describe the different components of the image processing system in Fig. 1. The images at the various stages of processing are represented by the following two-dimensional arrays:

$f(k,l)$: initial digitized ultrasound image;

$g(k,l)$: preprocessed image (after noise reduction);

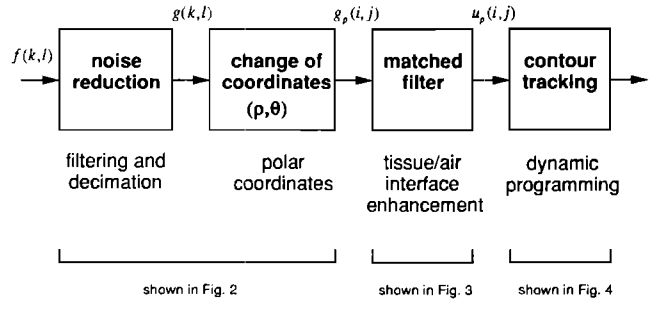

FIG. 1. General block diagram summarizing the contour detection algorithm. Illustrations of the effect of individual processing modules are shown in Figs. 2–4.

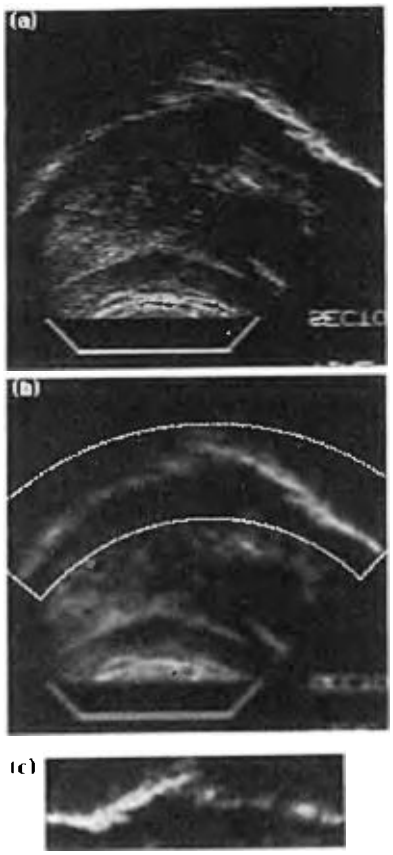

FIG. 2. Noise reduction and specification of a polar region of interest: (a) original ultrasound image, (b) smoothed image with overlaid region of interest, (c) extracted sector in polar coordinates.

$g_p(i,j)$: area of interest in the transformed coordinate system;

$u_p(i,j)$: result of matched filtering in the transformed coordinate system.

A. Preprocessing

Most ultrasound images have a relatively high noise level because of intrinsic limitations in the measurement device. A standard approach for noise reduction is to apply a low-pass digital filter such as a moving average. An example of a typical tongue profile before and after low-pass filtering is shown in Fig. 2. We found a useful extension to this technique by down sampling (i.e., reducing) the image by a factor of 2 after low-pass filtering. In order to minimize the loss of information, we use a least-squares cubic spline approximation technique that involves the design of a specific prefilter (Unser *et al.*, in press). This filter is implemented very efficiently using the recursive algorithm described in Unser *et al.* (1991). To the naked eye, the effect of this operation is essentially size reduction without noticeable distortion (e.g., smoothing, or blurring); more details can be found in Unser *et al.* (in press, Part II). This procedure has several advantages. First, it improves the signal-to-noise ratio. The distortion on the slowly varying signal component is minimal, while the noise variance is reduced by a factor of 4. Second, the reduction of the number of pixels in the original image will translate into a net increase in execution speed by a factor of 4 for all subsequent image processing operations. Finally, the fact that a reduced number of contour points is used to represent the tongue surface improves the overall robustness of the whole detection scheme. It also tends to improve the smoothness of the extracted contour segments. Accordingly, we chose to use this form of preprocessing for all the experimental examples in Sec. III.

B. Change of coordinate system

The purpose of a change of coordinate system is to present the data in a form that is better suited for the contour tracking algorithm described in Sec. I D. The main requirement is to be able to parametrize the contour of the tongue surface along one of the dimensions of the data. The ideal situation would be to define a transformation that maps the tongue surface into a horizontal line. Since the curved tongue surface is usually well represented by the arc of a circle, this situation can be approximated by using an appropriate polar coordinate system. Such a polar transformation is characterized by a center point (x_0, y_0) and two starting values ρ_0 and θ_0 for the radius and the angle, respectively. This geometrical transformation is implemented by appropriately resampling the preprocessed data

$$g_p(i,j) = g(x = x_{ij}, y = y_{ij}), \tag{1}$$

where $g(x,y)$ is a continuous function that interpolates $\{g(k,l)\}$ and where

$$\begin{aligned} x_{ij} &= x_0 + (\rho_0 + i\,\Delta\rho)\cos(\theta_0 + j\,\Delta\theta), \\ y_{ij} &= y_0 + (\rho_0 + i\,\Delta\rho)\sin(\theta_0 + j\,\Delta\theta). \end{aligned} \tag{2}$$

The quantities $\Delta\rho$ and $\Delta\theta$ are the radial and angular sampling steps, respectively. The interpolated values in (1) are obtained through a bilinear or cubic spline interpolation (Unser *et al.*, 1991). The parameters of the polar system are determined either by specifying three contour points (such as beginning, intermediate, and end points), or by searching for the circle that fits a previously detected contour most closely. A computational procedure that determines the parameters of a circle that provides the least-squares fit of a set of contour points $\{(x_j,y_j), j = 1,...,N\}$ with $N \geqslant 3$ is described in the Appendix. The specification of a transformed coordinate system also provides the definition of an area of interest for subsequent contour detection. This process is illustrated in Fig. 2. In this example, the coordinate system was specified interactively by the user. This task involved clicking on three points along the contour in order to define the geometry, and then making a radial selection with the computer's mouse, specifying the area of interest in terms of its minimum and maximum radii. In a typical sequence of ultrasound images, this initialization procedure needs to be performed only once, and the parameters of the coordinate system are kept fixed during the rest of the analysis. This procedure can be readily adapted for many other geometrical transformations.

C. Matched filter

On the ultrasound image, the bright light central line is the reflection of the air at the surface of the tongue. The tongue surface itself is the intersection of that line and the

dark area immediately beneath. In order to enhance the tissue/air interface, we use a standard matched filter technique (Jain, 1989). As the change of coordinate system is intended to flatten the tongue surface horizontally, the filtering is performed selectively in the perpendicular (vertical) dimension.

To a first approximation, the contour of the tongue surface can be approximated by a step function. Thus the location of the corresponding edge can be detected by using a convolution detector of the form

$$u_p(i,j) = g_p(i,j-1) - g_p(i,j+1), \tag{3}$$

where $u_p(i,j)$ denotes the output of the matched filter. In essence, this operation is a vertical differentiation; it will produce a maximum when there is a sharp vertical transition from black to white, as illustrated in Fig. 3. This detection scheme can be improved by using a larger filtering kernel that provides a better approximation of the characteristic signal response associated with an air/tissue medium transition. Such a reference template can, for example, be obtained by averaging a sequence of vertical signal profiles extracted along a reference contour traced by an experienced experimenter (Unser *et al.*, 1989). However, we found that such refinements were not necessary in the present application, and that a simple vertical differentiator was sufficient for our purpose when used in conjunction with the contour detection algorithm that is considered next. This latter algorithm is very robust by design; this also means that the exact shape and size of the matched filter have relatively little effect on overall performance.

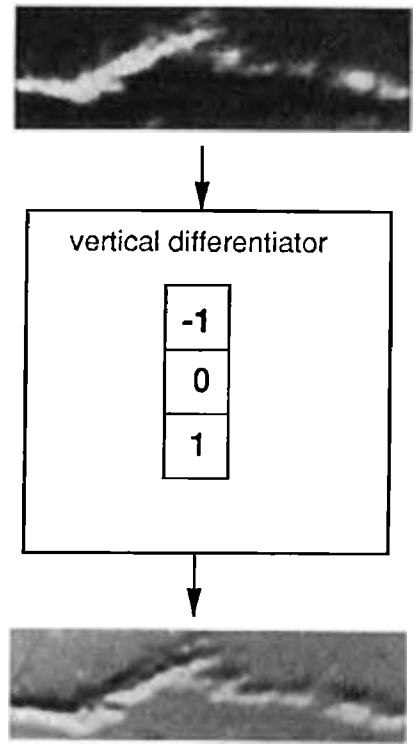

FIG. 3. Edge enhancement by matched filtering: The input image (top) is convolved with the template in the central box (finite difference approximation of a vertical differentiator). This operator acts as a correlation detector highlighting regions where there is a vertical transition from white to black.

D. Contour tracking

There are several special requirements for the design of the present contour tracking algorithm. The first constraint is that we are interested in extracting a single contour segment outlining the tongue surface; all other edges in the image need to be ignored. Second, the algorithm has to be able to accommodate to high noise levels. Third, the extracted contour points need to be connected—the algorithm should fill-in contour gaps. Fourth, the procedure should produce a smooth contour line.

A convenient way of dealing with all these issues simultaneously is to reformulate the present task as an optimization problem:

> Among all allowable contour segments T_k satisfying particular prescribed connectivity and smoothness constraints, find the one that maximizes some criterion $\zeta(T_k)$.

In this formulation, a contour segment of length N is represented by a sequence of points in the transformed coordinate system: $T_k = [(i_1,1),(i_2,2),\ldots,(i_j,j),\ldots,(i_N,N)]$. In effect, such a contour segment is parametrized by the angular variable j, which implicitly imposes the connectivity constraint. A smoothness constraint is introduced by limiting the maximal radial displacement from one contour point to the next: $|i_{j-1} - i_j| < \Delta i, j = 2,\ldots,N$, where Δi is some prescribed constant (typically, $\Delta i = 1$). Such a sequence of points defines a graph in the $i-j$ plane as illustrated in Fig. 4. In our system, we have chosen the simplest possible figure of merit; that is, the cumulative sum of the vertical derivative values along the contour segment:

$$\zeta(T_k) = \sum_{j=1}^{N} u_p(i_j, j). \tag{4}$$

Given these definitions, our contour extraction problem can, at least in principle, be solved by enumerating all allowable contour segments, evaluating their respective figures of merit, and finally selecting the one that maximizes (4). By taking advantage of the fact that $\zeta(T_k)$ can be decomposed into a sum of elementary contributions, each of which depends only on the previous transition, the problem is solved effectively through dynamic programming. This technique uses the Bellman principle of optimality (Bellman, 1957), which, in the present case, may be restated as follows: If the best path goes through a given point (i,j), then the best path includes, as a portion of it, the best partial path to the grid point (i,j). Accordingly, it is sufficient to restrict the search to subtrajectories that are partially optimal.

The dynamic programming algorithm scans iteratively through all values of i and j, successively. At each step, it determines $\zeta^*_{i,j}$, the optimal figure of merit for reaching the grid point (i,j) starting from $j = 1$. The initialization sequence for $j = 1$ is

$$\zeta^*_{i,1} = u_p(i,1). \tag{5}$$

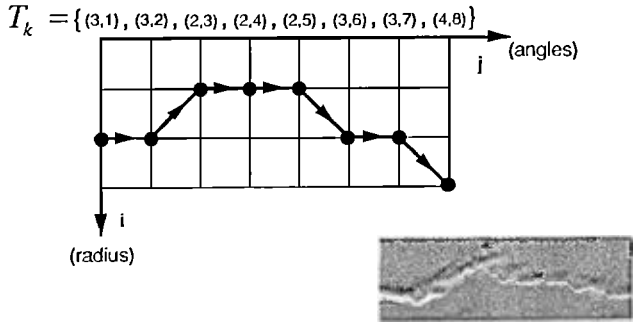

FIG. 4. Representation of contours: A potential contour candidate T_k is represented by a sequence of integer coordinates that define a directed graph in the $i-j$ plane. Among all allowable contours, there is one that maximizes the figure of merit $\zeta(T_k)$. The optimal contour for the example in Fig. 3 is displayed on the lower right.

For $j > 1$, the partially optimal figures of merit are updated as

$$\zeta^*_{i,j} = u_p(i,j) + \max\{\zeta^*_{k,j-1}, k = i - \Delta i, \ldots, i + \Delta i\} \quad (j = 2, \ldots, N). \tag{6}$$

This procedure amounts to considering all possible predecessors of a grid point (i,j) and selecting the most favorable transition. At the end of the cycle, the optimal figure of merit is found by searching for the maximum of $\zeta^*_{i,j=N}$.

The retrieval of the optimal trajectory is achieved by backtracking. This procedure is simplified by storing the most favorable predecessor of any grid point (i,j) in an auxiliary bidimensional array at each step of the algorithm. As only the values of the figure of merit at $j-1$ are required for a given j, it is sufficient to store $\zeta^*_{i,j-1}$ in a temporary one-dimensional array that is updated with each increment of j.

The dynamic programming procedure described above is readily adapted to those cases in which particular grid points are required to be on the path. Such constraints may be provided in particularly difficult images for which the unconstrained algorithm has failed. In such a case, the global trajectory is divided into subsections, each of which is specified by a start and end node prescribed by the operator, with no additional constraints. Optimization is then performed for each segment independently. Specification of a start point (i_s, j_s) is taken into account by appropriately restricting the allowable predecessors of all nodes scanned successively by the algorithm. The specification of an end point $(i_e j_e)$ is straightforward since the optimal figure of merit to reach this point is given by $\zeta^*_{i_e,j_e}$. However, for typical sequences of ultrasound images, the use of start and end points is rarely necessary and the algorithm is capable of fully automated border extraction.

II. APPLICATION

A. Speech materials

Two data sets were used to illustrate the behavior of the algorithm. They included single frames chosen for their poor image quality, and two syllables extracted from running speech, [ʃa] and [sa], demonstrating rotation and segmental displacement. The speech materials were spoken by a normal native English speaker.

B. Data acquisition

The midsagittal views of the tongue were obtained using a mechanical sectoring real-time ultrasonic scanner (Advanced Technology Laboratory, Inc., Bellevue, WA) with a 5-MHz transducer which produces a complete scan 30 times per second, as described by Stone *et al.* (1987, 1988). The images were recorded on a Sony U-Matic video tape recorder at a rate of 60 frames per second.

Successive image fields are then digitized from video tape using a Macintosh IIci equipped with a QuickCapture frame grabber board (Data Translation 2255). The images are saved on hard disk in standard TIFF (tagged image file format) with 8 bits per pixel. This process is controlled using Image 1.30 software, a public domain program, created by Wayne Rasband of the National Institutes of Health.

C. Special purpose image processing software

Our special purpose image processing software was developed in FORTRAN on a Macintosh II computer; it requires a hardware configuration with a minimum of 4Mbytes of RAM. The program accepts images that are stored in TIFF format. The processing results (series of $x-y$ contours coordinates) can be displayed graphically and saved in standard spreadsheet format. The program is menu driven and has a "user-friendly" Macintosh interface. The organization of the menus more or less replicates the block diagram in Fig. 1.

A typical image processing session involves the following manipulations. The experimenter first selects a representative image in the sequence that will be used to define the spatial coordinate system, as described in Sec. I C. This image is read using the *Open TIFF* command under the *File* menu. The preprocessed image is obtained by activating the *Region of interest* command, which prompts for a rectangular selection, and *Reduce* which shrinks the input image by a factor of two. Next, the user defines the local geometry and an area of interest by executing the *Polar system* and *Radial sector* operations. The next step is to extract the contour of the tongue surface, achieved through the *Matched filter* and *Tracking* commands. At this stage, the extracted contour segment is superimposed onto the input image. At the operator's option, the detected contour can be smoothed by selecting *Smooth contour*, which applies a Gaussian low-pass filter to the contour sequence in polar coordinates. Finally, the user can specify a full image sequence using the *Define sequence* command. This sequence is then analyzed automatically by executing the macro command which processes all images successively. When operating in macro mode, all user input is suppressed and the program uses default parameter values that replicate the previous manual input. At the end of processing, the extracted contour segments are saved by selecting the contour window and choosing the

Save as ... commands. Unacceptable contours, determined by visual inspection, can be remeasured individually.

Additional features of the program include the ability to generate figures and to extract contour features such as the center of gravity or the radius of the best approximating circle. Further, the user can combine several contour segments into an average profile.

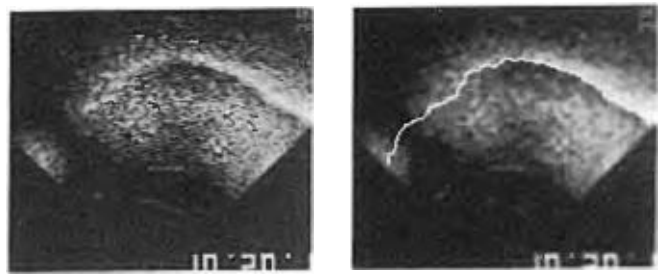

FIG. 6. Tongue surface profile no. 2 exhibiting a discontinuity on the left. The algorithm fills in the gap.

III. RESULTS AND DISCUSSION

A. Experimental results

Experimental results for four examples of sagittal sections of the tongue are shown in Figs. 5–8. The posterior tongue is on the left-hand side of the ultrasound image. These figures depict several tongue surface profiles, each presenting a different edge detection problem. The displays on the left represent the original ultrasound images reduced by a factor of 2. The (unsmoothed) extracted contours are superimposed on the low-pass filtered preprocessed images (3×3 moving average). The experimental procedure for these examples was essentially that outlined in the previous section. The area of interest was chosen so as to include the entire angular section of the ultrasound cone. Despite the fact that the surface of the tongue is not always well defined in these images, the extracted contour segments appear to be quite accurate. Figures 6 and 8, in particular, demonstrate the capacity of the algorithm to fill in contour gaps. The contour extraction procedure appears to be relatively immune to artifacts such as blind spots (left part of Fig. 6) or loss of echo intensity (upper part of the tongue in Fig. 8).

In order to validate these results, we performed a comparison between the automatically detected edges and the manual tracings of an experienced experimenter. Two independent manual tracings [(a) and (b)], separated in time by several months, were recorded for each image; these contours were drawn in Image 1.30 and superimposed onto the reduced images. The manually traced edges were entered into our program as reference contours. A quantitative performance evaluation was performed by computing the root-mean-square error:

$$\epsilon = \left(\frac{1}{N} \sum_{i=1}^{N} (x_i - x_i^*)^2 + (y_i - y_i^*)^2 \right)^{1/2}, \qquad (7)$$

where $\{(x_i,y_i), i = 1,...,N\}$ and $\{(x_i^*,y_i^*), i = 1,...,N\}$ represent the extracted and reference contour points, respectively; it is also assumed that both sequences of contour points were obtained using the same parametrization. The results of these calculations are given in Table I. The observed discrepancy was within the range of the manual intermeasurement variation, i.e., the error between references (a) and (b).

What may be the most valuable feature of our system is its ability to handle dynamic image sequences in a quasi-autonomous fashion. In most cases, the user need only specify the local geometry and area of interest on the first image of the sequence; all subsequent computations can be performed in batch mode. Figure 9 displays two examples of such an analysis.

In Fig. 9(a), a series of six successive contour configurations have been superimposed onto the first image of the sequence. In this example, the tongue appears to rotate around a pivot point: the intersection of these contour segments. The movement pattern indicates tongue-tip lowering and dorsal raising during the [ʃ]-to-[a] movement. The simultaneous timing of anterior raising and dorsal lowering, typical of this C-to-V movement, cannot be demonstrated from maximal and minimal frames alone.

Figure 9(b) displays successive contours from [s]-to-[a]. This movement pattern demonstrates inward displacement of the posterior tongue and simultaneous upward movement of middle tongue and anterior portions of the tongue. As in Fig. 9(a), the timing relationships among the various tongue segments cannot be demonstrated from the maximal and minimal frames.

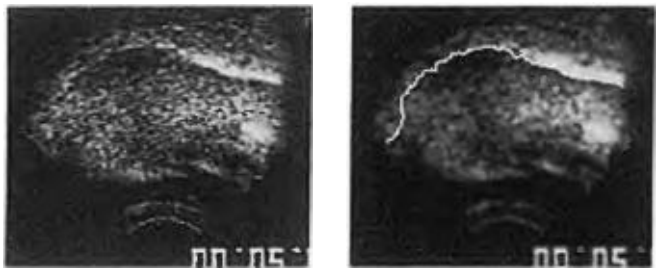

FIG. 5. Tongue surface profile no. 1 exhibiting a low contrast edge on the left side of the image. The algorithm finds the beginning of the curve and tracks the faint edge.

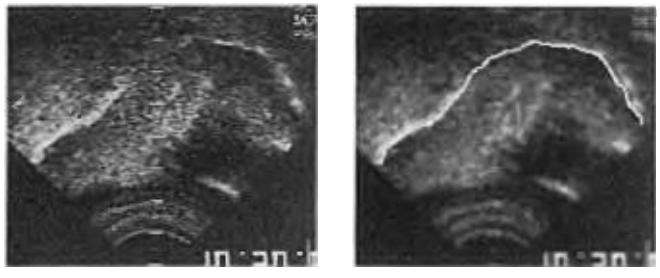

FIG. 7. Tongue surface profile no. 3 exhibiting a loss of edge definition in the midportion. The algorithm tracks across the faint portion.

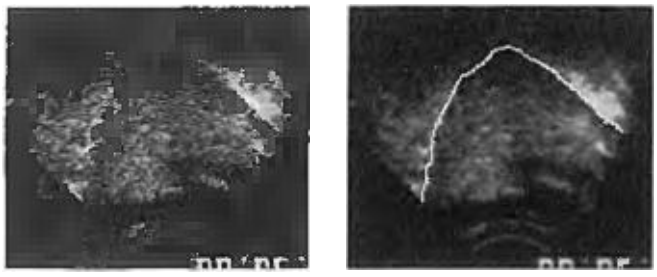

FIG. 8. Tongue surface profile no. 4 exhibiting a loss of edge intensity accompanied by a large change of slope. The algorithm successfully tracks the edge.

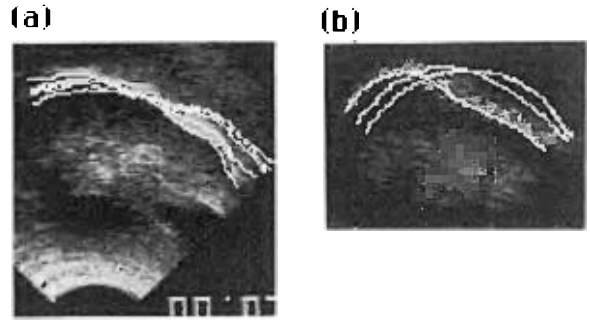

FIG. 9. Two examples of image sequence tracking. (a) Shows tongue rotation during the [ʃ]-to-[a] movement. (b) Shows segmental displacement during the [s]-to-[a] movement.

B. Discussion

The edge detection and quantification scheme presented here makes feasible the rapid analysis of two-dimensional time-varying ultrasound images of tongue movement. Earlier manual tracking of the tongue surface profile in a sequence of scans was prohibitively slow, as even one second of data provided 60 tongue surface contours (video fields). As a result, previous analyses of such data focused on maximal and minimal tongue positions rather than transitional movements. However, tongue movements are often circular rather than linear, so clear-cut maxima are not apparent. In addition, parts of the tongue often reach maximum while others are still moving. Continuous semiautomatic measurement should permit movement and positional subleties, tongue rotation, and segmental movement patterns to emerge. With this information, we can better define two-dimensional and three-dimensional tongue shapes for both the static and the time-varying case.

From a practical point of view, the most obvious advantage of the present approach is its time saving potential. Other beneficial features are its accuracy, which has been demonstrated experimentally, and the reproducibility of the measurements.

One way of improving the present scheme may be to include the temporal dimension in the formulation of the contour tracking problem to insure the continuity of the contour displacement from frame to frame. Such an extension of the work would enable "bridging" a short time period of very low image quality. The less fortunate aspect is that it would result in an increase of complexity by several orders of magnitudes, at least, if one uses dynamic programming. Fortunately, this type of improvement does not seem to be necessary for most practical applications: It is clear from our experiments that the present algorithm is adequate for ultrasound data of reasonable quality.

TABLE I. Comparison between manual and automated tracing of the tongue surface. The test images 1 to 4 are displayed in Figs. 5–8, respectively. The values in parentheses correspond to an additional level of smoothing by Gaussian filtering. Error values are in pixel units.

Experiment	Reference (a)	Reference (b)
Test 1:		
manual versus manual	···	$\epsilon = 1.40$
automatic versus manual	$\epsilon = 1.38$ (1.17)	$\epsilon = 1.37$ (1.19)
Test 2:		
manual versus manual	···	$\epsilon = 1.09$
automatic versus manual	$\epsilon = 1.42$ (1.37)	$\epsilon = 1.23$ (1.21)
Test 3:		
manual versus manual	···	$\epsilon = 1.49$
automatic versus manual	$\epsilon = 1.72$ (1.70)	$\epsilon = 1.48$ (1.38)
Test 4:		
manual versus manual	···	$\epsilon = 1.29$
automatic versus manual	$\epsilon = 1.56$ (1.55)	$\epsilon = 1.82$ (1.56)

IV. CONCLUSION

In this paper, we have described an image processing system that allows an automated detection of the tongue surface in sequences of ultrasound images. This system has been implemented by "user-friendly" special purpose image processing software that runs on any Macintosh II personal computer. One of the key components is the contour extraction algorithm that uses dynamic programming optimization.

The present analysis method exhibits the following advantageous features. The contour tracking algorithm has built-in smoothness constraints. The contour detection is robust and the system is capable of handling relatively high levels of noise. The approach is fast and offers reproducible analyses. It can handle large amounts of data and process whole sequences of images in a fully automatic mode, although the detection can be readily constrained by specifying a small number of contour points.

Finally, the present method should also be applicable to other tongue imaging modalities such as x-ray, computed tomography and magnetic resonance imaging.

APPENDIX: POLAR COORDINATE SYSTEM FITTING

We first consider the problem of finding the center (x_0, y_0) and radius R of a circle that goes through three refer-

ence points: (x_1,y_1), (x_2,y_2), and (x_3,y_3). These parameters must satisfy the following constraints:

$$(x_i - x_0)^2 + (y_i - y_0)^2 = R^2 \quad (i = 1,...,3). \tag{A1}$$

By combining these equations, it is not difficult to show that the coordinates of the center (x_0,y_0) are solutions of the linear system of equations:

$$\begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) \\ 2(x_1 - x_3) & 2(y_1 - y_3) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 \end{bmatrix}. \tag{A2}$$

This system is invertible provided that the three points are not collinear. The radius is obtained by taking the distance between any of these points and the center.

Next, we consider the problem of finding the parameters of the circle that provides the best least-squares fit of a set of points $\{(x_i,y_i), i = 1,...,N\}$. More precisely, we want to determine the values x_0, y_0, and R that minimize the error criterion:

$$\epsilon^2(x_0,y_0,R) = \sum_{i=1}^{N} (R - \rho_i)^2, \tag{A3}$$

where

$$\rho_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}. \tag{A4}$$

The optimum values are found by setting the partial derivatives of ϵ^2 to zero, which yields the system of nonlinear equations:

$$\frac{\partial \epsilon^2}{\partial R} = \sum_{i=1}^{N} 2(R - \rho_i) = 0, \tag{A5}$$

$$\frac{\partial \epsilon^2}{\partial x_0} = \sum_{i=1}^{N} 2\left(\frac{R - \rho_i}{\rho_i}\right)(x_i - x_0) = 0, \tag{A6}$$

$$\frac{\partial \epsilon^2}{\partial y_0} = \sum_{i=1}^{N} 2\left(\frac{R - \rho_i}{\rho_i}\right)(y_i - y_0) = 0. \tag{A7}$$

These equations can also be rewritten as

$$R = \frac{1}{N} \sum_{i=1}^{N} \rho_i, \tag{A8}$$

$$x_0 = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \frac{R}{\rho_i}(x_i - x_0)\right)$$

$$= x_0 - \frac{1}{N} \sum_{i=1}^{N} \left(\frac{R}{\rho_i} - 1\right)(x_i - x_0), \tag{A9}$$

$$y_0 = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \frac{R}{\rho_i}(y_i - y_0)\right)$$

$$= y_0 - \frac{1}{N} \sum_{i=1}^{N} \left(\frac{R}{\rho_i} - 1\right)(y_i - y_0). \tag{A10}$$

These last expressions suggest an iterative algorithm that finds the solutions by successive refinement. The initial parameter values $R^{(0)}$, $x_0^{(0)}$, and $y_0^{(0)}$ can be determined from the circle that passes through the three points (x_1,y_1), $(x_{N/2}, y_{N/2})$, and (x_N,y_N); these parameters are found by solving a system of equations equivalent to (A2). For each iteration cycle k, these values are then updated according to the following procedure:

$$R^{(k)} = \frac{1}{N} \sum_{i=1}^{N} \rho_i^{(k-1)}, \tag{A11}$$

$$x_0^{(k)} = x_0^{(k-1)} - \frac{1}{N} \sum_{i=1}^{N} \left(\frac{R^{(k-1)} - \rho_i^{(k-1)}}{\rho_i^{(k-1)}}\right)(x_i - x_0^{(k-1)}), \tag{A12}$$

$$y_0^{(k)} = y_0^{(k-1)} - \frac{1}{N} \sum_{i=1}^{N} \left(\frac{R^{(k-1)} - \rho_i^{(k-1)}}{\rho_i^{(k-1)}}\right)(y_i - y_0^{(k-1)}), \tag{A13}$$

where $\rho_i^{(k-1)}$ denotes the individual radius values associated to the center position $(x_0^{(k-1)}, y_0^{(k-1)})$. The procedure is iterated until no more change occurs. Interestingly enough, the update terms in (A12) and (A13) are proportional to the corresponding partial derivatives of the error criterion [Eqs. (A6) and (A7)]. This algorithm is therefore in essence a steepest descent procedure (Press *et al.*, 1986), although the step size in the present case is fixed. We have verified experimentally that this algorithm is well behaved and that it usually converges in less than 20 cycles. We have not encountered any convergence problems.

Baer, T., Gore, J., Boyce, S., and Nye, P. (**1987**). "Application of MRI to the analysis of speech," Magnet. Reson. Imag. **5**, 1–7.

Baer, T., Gore, J., Gracco, L., and Nye, P. (**1988**). "Vocal tract dimensions obtained from magnetic resonance imaging," J. Acoust. Soc. Am. Suppl. 1 **84**, S125.

Bellman, R. (**1957**). *Dynamic Programming* (Princeton U. P., Princeton, NJ).

Jain, A. K. (**1989**). *Fundamentals of Digital Image Processing* (Prentice-Hall, Englewood Cliffs, NJ).

Johansson, C., Sundberg, J., Wilbrand, H., and Ytterbergh, C. (**1983**). "From sagittal distance to area: A study of transverse cross sectional area in the Pharynx by means of computer tomography," R. Inst. Technol. Stockholm Trans. Lab. Progr. Stat. Rep. **4**, 39–49.

Keller, E., and Ostry, D. (**1983**). "Computer measurement of tongue dorsum movement with pulsed echo ultrasound," J. Acoust. Soc. Am. **73**, 1309–1315.

Pratt, W. K. (**1978**). *Digital Image Processing* (Wiley, New York).

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (**1986**). *Numerical Recipes* (Cambridge U. P., Cambridge, GB).

Stone, M., Morrish, K., Sonies, B., and Shawker, T. (**1987**). "Tongue curvature: A model of shape during vowel production," Folia Phoniatr. **39**, 302–315.

Stone, M., Shawker, T., Talbot, T., and Rich, A. (**1988**). "Cross-sectional tongue shapes during vowel production," J. Acoust. Soc. Am. **83**, 1586–1596.

Stone, M. (**1990**). "A three-dimensional model of tongue movement based on ultrasound and x-ray microbeam data," J. Acoust. Soc. Am. **87**, 2207–2217.

Unser, M., Pelle, G., Brun, P., Eden, M. (**1989**). "Automated extraction of serial myocardial borders from *M*-mode echocardiograms," IEEE Trans. Med. Imag. **MI-8**(1), 96–103.

Unser, M., Aldroubi, A., and Eden, M. (**1991**). "Fast *B*-spline transforms for continuous image representation and interpolation," IEEE Trans. Pattern Anal. Machine Intell. **13**(3), 277–285.

Unser, M., Aldroubi, A., and Eden, M. (**in press**). "*B*-spline signal processing. Part I: theory; Part II: efficient design and applications," IEEE Trans. Signal Processing.

Watkin, K., and Rubin, J. (**1989**). "Pseudo-three-dimensional reconstruction of ultrasonic images of the tongue," J. Acoust. Soc. Am. **85**, 496–499.