Tongue motion averaging from contour sequences

Min Li†, Chandra Kambhamettu†, Maureen Stone‡
†University of Delaware, ‡University of Maryland Dental School

(Received March 11, 2004; accepted September 1, 2004)

Contact Address
Min Li
Video/Image Modelling and Synthesis Lab
Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716 USA
Fax: 302-831-8458 /Phone: 302-831-0556
E-mail: mli@cis.udel.edu

Abstract

In this paper, a method to get the best representation of a speech motion from several repetitions is presented. Each repetition is a representation of the same speech captured at different times by sequence of ultrasound images and is composed of a set of 2D spatio-temporal contours. These 2D contours in different repetitions are time aligned first by a shape based Dynamic Programming (DP) method. The best representation of the speech motion is then obtained by averaging the time aligned contours from different repetitions. Procrustes analysis is used to measure the contour similarity in the time alignment process and to get the averaged best representation. To get the point correspondence for Procrustes analysis, a nonrigid point correspondence recovery method based on a local stretching model and a global constraint is developed. Synthetic validations and experiments on real tongue motion are also presented in this paper.

Keywords: Motion analysis; Nonrigid motion; Dynamic programming; Time alignment

1 Introduction

The tongue motion in a fixed plane can be recorded with a sequence of images. The imaging techniques include ultrasound, X-ray , MRI and many others. Tongue surfaces can be extracted from images (Li, Kambhamettu and Stone, 2003; Akgul, Kambhamettu and Stone, 1999) thus 2D tongue motion in a fixed plane is represented by a sequence of spatio-temporal 2D contours. Each contour describes the tongue shape at a certain time instance. To best capture the tongue motion, the same speech is repeated by the same subject several times. Because biological systems are imprecise, humans do not produce identical repetitions of the same item. To provide the best representation of an utterance, averaging of different repetitions is a useful technique. However in different repetitions, the subject varies in speaking rates, and small tongue shape differences may result in arbitrary spatial shifts. These shifts may be different for different repetitions. Therefore a time-warping algorithm is needed to align temporal variations in multiple repetitions before averaging. Dynamic Programming (DP) algorithm has been used successfully to eliminate the effect of inter and intra-speaker variation in speech recognition (Sakoe and Chiba, 1978). It has also been used by Yang and Stone (Yang and Stone, 2002) to align repetitions of the same utterance in different planes, for 3D tongue reconstruction purpose. These DP algorithms find the optimal time registration between two repetitions based on the minimum total distance measure of the acoustic feature. However, an acoustic signal is not always available, for example, when studying swallowing. Shape based time alignment is a useful alternate for tongue analysis.

In this paper, we use shape based Dynamic Programming to align different repetitions of the same utterance. The best representation of the speech motion is then obtained by averaging the time aligned contours from different repetitions. Procrustes analysis is used to measure the contour similarity in the time alignment process and to get the averaged best representation. The utterance is recorded with ultrasound images. With the Head and Transducer Support System (HATS) (Stone and Davis, 1995), the head of the subject is fixed and the transducer is placed below the chin at a known position. Accurate and reliable ultrasound images can be obtained.

2D tongue contours are extracted from the ultrasound images with an automatic contour tracking system (Li, Kambhamettu and Stone, 2003).

The problem we are trying to solve is: given a set of 2D contour sequences, where each sequence is a representation of the same object motion recorded at different times and the time-varying gesture differs across sequences, construct a mean sequence of all available sequences that best models the given object motion.

The outline of our tongue motion averaging method is below:

1. Input m contour sequences $S_1, S_2, ..., S_k, ..., S_m$. The k^{th} sequence S_k consists of n_k contours $C_{k1}, C_{k2}, ..., C_{kn_k}$ and each contour is represented by a set of object boundary points.

2. Time alignment: perform time alignment algorithm between contour sequences $S_1, S_2, ..., S_k, ..., S_m$, and get the time aligned sequences $S'_1, S'_2, ..., S'_3, ..., S'_m$. Each time aligned sequence has the same number n of contours.

3. Mean shape computing: calculate the mean shape $\hat{S}$ of $S'_1, S'_2, ..., S'_3, ..., S'_m$. The i^{th} contour of $\hat{S}$ is the mean of all the i^{th} contours of $S'_1, S'_2, ..., S'_3, ..., S'_m$ where $1 \leq i \leq n$.

2 Time alignment

Since different sequences S_k of the same motion are captured in different repetitions, time variations exist among these sequences due to the small tongue shape differences and the change of motion velocity in different repetitions. There are two types of time variations. First, the sequence length is different for different repetitions. Second, a given contour (say, the i^{th} contour) in different sequences is not the representation of the same time instance of the motion. A time registration method is necessary to align different sequences in time and to get the best representation of the motion.

The method we used for time alignment is based on DP and it is performed between each sequence and the selected reference sequence. After each sequence is time aligned with the refer-

ence, the time alignment is established between any two sequences by transitivity. All sequences have the same number of contours as the reference by linear interpolation between contours after time registration. The reference sequence we selected is the longest sequence among all available sequences because more information of the raw data can be kept after time registration.

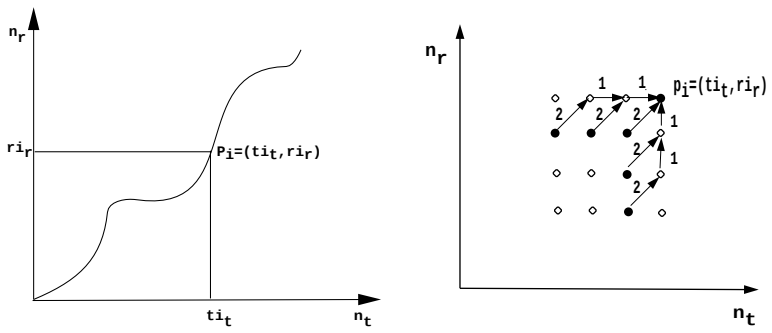
Consider the time alignment between one test sequence S_t and the reference sequence S_r. The contours in these two sequence are $C_{t1}, C_{t2}, ..., C_{tn_t}$ and $C_{r1}, C_{r2}, ..., C_{rn_r}$ and the sequence lengths are n_t and n_r for these two sequences respectively. Contours in these two sequences define a grid of n_t by n_r. As shown in figure 1(a), a possible path $P = p_1 p_2 ... p_L$ of this grid has L points and the point $p_i = (ti_t, ri_r)$ along this path is defined by the indices of contour C_{ti_t} and C_{ri_r} where $1 \leq i_t \leq n_t$ and $1 \leq i_r \leq n_r$.

The optimal path among all possible paths is the path which has the minimum cost. It represents a best matching between the two sequences (S_t and S_r).

If $P = p_1 p_2 ... p_L$ is the optimal path, let $d(ti_t, ri_r) = d(C_{ti_t}, C_{ri_r})$ denote the contour matching cost between contours C_{ti_t} and C_{ri_r} where $p_i = (ti_t, ri_r)$ is a grid point along this path; let $D(ti_t, ri_r) = D(C_{ti_t}, C_{ri_r})$ denote the accumulated path cost from the first grid point to point $p_i = (ti_t, ri_r)$ along this path. The sub-path from point p_{i-1} should also be optimal. If we constrict the choice of p_{i-1} and define the sub-path cost by using a template shown in figure 1(b), then:

$$D(ti_t, ri_r) = \min \begin{cases} D(ti_t - 3, ri_r - 1) + 2d(ti_t - 2, ri_r) + d(ti_t - 1, ri_r) + d(ti_t, ri_r) \\ D(ti_t - 2, ri_r - 1) + 2d(ti_t - 1, ri_r) + d(ti_t, ri_r) \\ D(ti_t - 1, ri_r - 1) + 2d(ti_t, ri_r) \\ D(ti_t - 1, ri_r - 2) + 2d(ti_t, ri_r - 1) + d(ti_t, ri_r) \\ D(ti_t - 1, ri_r - 3) + 2d(ti_t, ri_r - 2) + d(ti_t, ri_r - 1) + d(ti_t, ri_r) \end{cases} \quad (1)$$

This symmetrical template is similar to what was used by Yang et al. (Yang and Stone, 2002) and it was reported to have good performance by Sakoe and Chiba (Sakoe and Chiba, 1978). The slope weighting coefficients in the template control the distribution of the local distance for each



(a) (b)

Figure 1: (a) A possible path P and one grid point $p_i = (ti_t, ri_r)$ along this path. (b) The sub path template.

path.

The optimal path cost is defined as the accumulated cost of the last point p_L along this path.

The whole process of the DP algorithm is as follows:

1. Initialization: For each point $p_i = (ti_t, ri_r)$, calculate the contour matching cost $d(ti_t, ri_r) = d(C_{ti_t}, C_{ri_r})$. Let $D(1,1) = d(1,1)$.

2. Forward calculation : For each point $p_i = (ti_t, ri_r)$, calculate $D(ti_t, ri_r)$ according to Equation 1 and record its sub path.

3. Termination: In the last column of the grid, find the point p_L which has the minimum accumulated cost. p_L is the last point of the optimal path.

4. Backward tracing: from point p_L, back trace the track.

The contour matching cost we selected is the Procrustes distance[1] between two contours. Procrustes distance has been frequently used in medical image analysis (Duta, Jain and Dubuisson-Jolly, 2001) and it is suitable for our application due to the following reasons: First, it is designed to measure shape similarity and the time alignment algorithm aims to find pairs of the most similar contours from two different sequences. Second, it provides a convenient way to calculate the

[1]Procrustes was the nickname of a robber who lived on the road from Eleusis to Athens. He offered travelers a room with a bed and he would fit them into the bed by stretching them if they were too short or cutting off their legs if they were too tall (Dryden and Mardia, 1998).

mean shape from a set of time aligned shapes. But the Procrustes distance should be calculated between sets of corresponding points which are not available from the input data, therefore we developed an algorithm to set up point correspondences between two contours that is presented in Section 4.

3 Mean shape and the Procrustes analysis

After the time alignment, the mean shape of the object at any aligned time instance is defined as the Procrustes average of contours from all sequences at the same time instance. To calculate the Procrustes average of a set of contours, the Procrustes distance between any two contours and the mean contour should be defined first. The Procrustes distance is also the contour matching error we used in the Dynamic Programming for time alignment.

Consider two centered contours $C_1 = (y_1, y_2, ...y_k)^T$ and $C_2 = (w_1, w_2, ..., w_k)^T$ where y_i and w_i are the i^{th} point vectors of C_1 and C_2 respectively and these two points correspond to each other. In order to compare these two shapes, the Procrustes analysis fits C_2 to C_1 with a similarity transformation and the difference between the fitted C_2 and observed C_1 indicates the magnitude of the shape difference between C_1 and C_2.

The similarity transformation in Procrustes analysis is defined as a sequence of transformations T of rotation, scale and translation. The Procrustes distance is defined as:

$$d_P(C_1, C_2) = \inf_T |(\frac{C_1}{|C_1|} - \frac{T(C_2)}{|T(C_2)|})|. \tag{2}$$

The Procrustes distance between C_1 and C_2 is the Euclidean distance from C_1 to the best fitted C_2 with the similarity transformation T. It is also normalised to get rid of the shape scale effects. The Procrustes average of a set of contours $C_1, C_2, ..., C_n$ is the mean shape of these contours and the summation of the Procrustes distance of all contours to the mean should be minimised. So

the mean $\hat{C}$ of $C_1, C_2, ..., C_n$ is:

$$\hat{C} = \arg\inf_{C} \sum_{i}^{n} d_P^2(C_i, C). \tag{3}$$

The details of Procrustes analysis can be found in (Dryden and Mardia, 1998).

4 Point correspondence recovery

Both the time alignment and averaging steps need point correspondence between contours. However, the problem of recovering nonrigid point correspondence is not trivial. Many researchers directly compare shapes to get the point correspondence, e.g. Wang et al. (Wang, Peterson and Staib, 2000) minimise the distance, curvature, and normal differences between corresponding point sets, while Belongie et al. (Belongie, Malik and Puzicha, 2001) match two shapes by comparing the shape context, which is a set of points around the point of interest. Another approach of nonrigid point recovery is modelling the nonrigid transformation between shapes. Amini et al. (Amini and Duncan, 1992) minimise the bending and stretching energies between shapes to recover point correspondence while Kambhamettu et al. (Kambhamettu and Goldgof, 1994) find the point correspondence by recovering a pre-defined nonrigid motion model.

The method we used for nonrigid point correspondence recovery in this paper is based on (Kambhamettu and Goldgof, 1994) but with some extensions: first, the motion in (Kambhamettu and Goldgof, 1994) can only be conformal motion while it is not restricted in our method in condition that the shape difference is small. This is important for the speech analysis since the tongue motion is not conformal. Second, only the change of curvature is modeled in (Kambhamettu and Goldgof, 1994) but the relationship between the changes of arc length and the curvature is modeled in our method. Third, a global constraint is added in addition to a local stretching model in our method while in (Kambhamettu and Goldgof, 1994) each point finds its correspondence independently without a global constraint. With this global constraint, it is guaranteed that several points on one contour will not be mapped onto the same point on another contour.

In our method, to compare a contour C to another contour C', the point correspondence is not

recovered for all original points (original points refer to all points on a contour) on C. Instead, contour C is represented by some sample points from the original contour points. These sample points are equally distributed along C and the distance between consecutive sample points is about 2-3 pixels. To recover the corresponding points on C' for these sample points, the sample points are compared with all possible original points on C'. In this way, we can avoid the local noise and point contention problem. This approach is motivated by and similar to (Duta, Jain and Dubuisson-Jolly, 2001).

The Procrustes distance between two contours used in previous sections is calculated between corresponding sample points. The Procrustes average of a set of contours of the same time instance is therefore represented by the averaged sample points. The whole contour of the Procrustes average then can be obtained by the cubic spline approximation of these averaged sample points.

4.1 Local stretch modelling

To recover the point correspondence between two contours, we first model the local stretching of a contour. Consider two contours C and C'. For a sample point P on C, the possible hypotheses about its corresponding point is formed in some small neighbourhood around the closest point to P among all original points on contour C'. The closest point to P is decided by aligning C to C' using the rigid Iterative Closest Point (ICP) (Besl and Mckay, 1992) method and C' is extended (Parthasarathy, Stone and Prince, 2003) to make sure there is a corresponding point on C' for any point at the ends of C. Figure 2(a) shows the correspondence hypotheses of P on contour C'. Here, P can correspond to any original point within some region, W. W is the region we check for point correspondences. It is defined by assuming that there is only small difference between C and C'. In this figure, P can correspond to any of the 5 points in the window, W. Then the point correspondence reliabilities of the correspondence hypotheses will be checked. In the estimation of point correspondences, original points in the neighbourhood are also considered. i.e, a local curve α around the sample point P. The mapping of a set of neighbouring points p_i of P onto another set of neighbouring points p'_i of the point correspondence hypothesis P' satisfies our local

stretch model which is presented below. The local curve α of point P is shown in figure 2(b). It is compared to the local curve α' of each point correspondence hypothesis P'. The matching error between α and α' indicates the correspondence reliability between points P and P'.

Let p_i denote the i^{th} point on the local curve α of point P; let p'_i denote the i^{th} point on the local curve α' of the point correspondence hypothesis P'; let $s(p_i)$ and $s'(p'_i)$ denote the curve arc length at p_i and p'_i on α and α' respectively. To compare two curve segments, we model the shape difference of these two curve segments by way of a nonrigid motion model. A simple nonrigid motion example is homothetic motion (Kambhamettu and Goldgof, 1994). For 2D homothetic motion, the expansion or contraction is uniform and the following relationship is true for all i:

$$\frac{s'(p'_i)}{s(p_i)} = \frac{k(p_i)}{k(p'_i)} \tag{4}$$

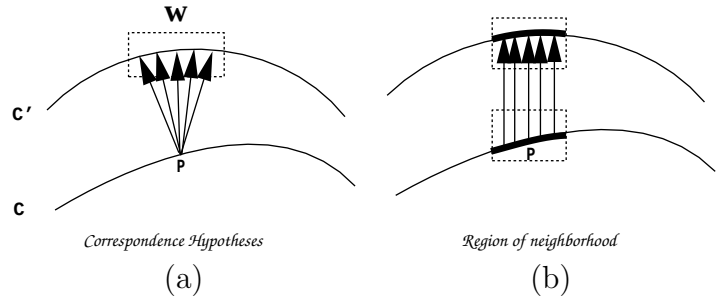
where $k(p_i)$ and $k(p'_i)$ are the curvatures of points p_i and p'_i respectively.

The homothetic motion models uniform expansion or contraction in the radial direction for all points along a curve. For most interesting medical object motions such as human tongue and heart motion, if the motion is small, one can assume that the motion is along the radial direction, however the motion is not uniform. The actual motion has little variance compared with the homothetic motion. The relationship between the arc length stretching and the curvature changes is not same as Equation 4 but the actual relationship is not far away from Equation 4. We model the arc length stretching as a linear function of the curvature change for the non-homothetic situation:

$$\frac{s'(p'_i)}{s(p_i)} = a\frac{k(p_i)}{k(p'_i)} + b \tag{5}$$

where a and b are the linear function parameters.

With the above linear model, the reliability of correspondence between P and P' can be esti-



(a) (b)

Figure 2: The correspondence hypotheses and the region of neighbourhood.

mated by the residual error:

$$er_L(P, P') = \sum_{p_i \in \alpha, p_i^{'} \in \alpha'} (s'(p_i^{'}) - s(p_i) \cdot (a \cdot \frac{k(p_i)}{k(p_i^{'})} + b))^2. \tag{6}$$

If P and P' are the best matching, then:

$$\frac{\partial er_L}{\partial a} = -2 \sum_{p_i \in \alpha, p_i^{'} \in \alpha'} (s_2(p_i^{'}) - s_1(p_i) \cdot (a \cdot \frac{k(p_i)}{k(p_i^{'})} + b)) \cdot \frac{k(p_i)}{k(p_i^{'})} \cdot s(p_i) = 0 \tag{7}$$

$$\frac{\partial er_L}{\partial b} = -2 \sum_{p_i \in \alpha, p_i^{'} \in \alpha'} (s_2(p_i^{'}) - s_1(p_i) \cdot (a \cdot \frac{k(p_i)}{k(p_i^{'})} + b)) \cdot s(p_i) = 0. \tag{8}$$

The values of the linear model parameters a and b can be obtained from Equation 7 and 8. If their values are substituted in Equation 6, the residual error $er_L(P, P')$ by fitting α and α' to the linear motion model of Equation 5 can be calculated. For point P, the residual error is calculated for each point correspondence hypothesis P' in the region of interest, W. The point P' with smaller residual error is more reliable as the correspondence of P.

4.2 Global stretching constraint

With the above local stretch modelling constraint, every sample point P on contour C is looking for its best matched point on C' independently. The correspondence information from the consecutive

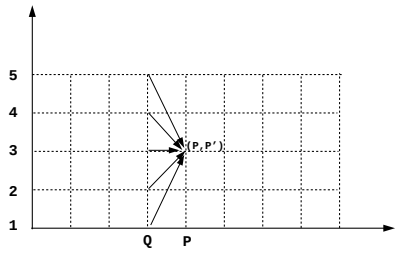


Figure 3: The grid and sub path template for point correspondence recovery.

sample points of P is not taken into account. Several points on C may be mapped onto the same point on C' (many-to-one mapping problem). We add a global constraint to the correspondence recovery by minimizing the total stretching between C and C'. This constraint is similar to what is used in (Amini and Duncan, 1991). The many-to-one mapping problem will be avoided with this global constraint.

Let Q and P denote two consecutive sample points in contour C; let Q' denote the correspondence point of Q; let P' denote the under-consideration correspondence hypothesis of P. The global stretching constraint defines an error at point P as:

$$er_E(P, P') = S(Q, P) - S(Q', P') \tag{9}$$

where $S(Q, P)$ is the arc length between points Q and P along contour C, $S(Q', P')$ is the arc length between points Q' and P' along contour C'. This error minimises the stretching energy between two contours. With this constraint, the arc length between Q' and P' tends to be similar as the arc length between Q and P, thus avoiding Q and P to map onto the same point.

Combining two constraints together, the error that needs to be minimised for all sample points is:

$$er = \sum_P \alpha er_L(P, P') + \beta er_E(P, P') \tag{10}$$

where α and β are the weighting parameters.

er is minimised again with a Dynamic Programming algorithm. A m by k grid is defined by

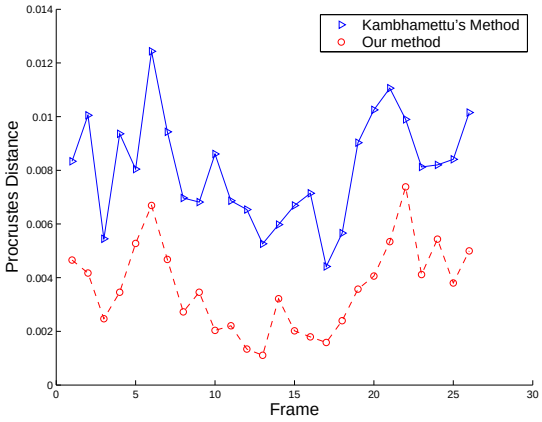


Figure 4: For each pair of time aligned contours, the Procrustes distance is calculated according to our correspondence recovery method (circle) and Kambhamettu's method (triangular).

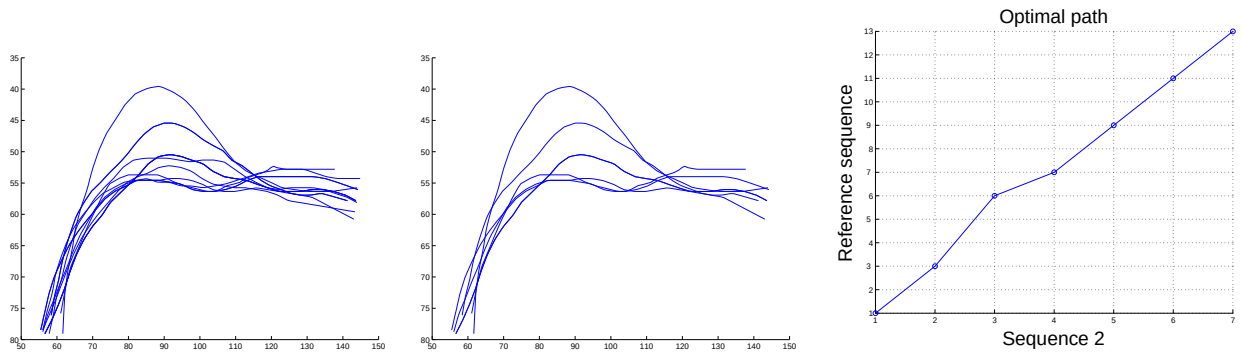
all sample points P of contour C and the searching window W, where m is the number of sample points of C and k is the size of W. This grid and the searching template is shown in figure 3. In this figure, the size of searching window is 5. A grid node, e.g (P, P') in column P tests all nodes in the previous column Q to decide the optimal sub path to it. Each node on the final optimal path represents a pair of corresponding points.

5 Validation

Several experiments are conducted to validate our point correspondence recovery and time alignment methods.

5.1 Performance of point correspondence recovery

In the time alignment step, point correspondence between contours is required to calculate the Procrustes distance. If contour C_1 of one sequence is best aligned in time with contour C_2 of the second sequence, the Procrustes distance between C_1 and C_2 should be smaller (if the Procrustes distance is calculated using accurate point correspondences). This fact supplies a way to evaluate our point correspondence recovery method. Two sequences S_1 and S_2 of the same speech



(a) (b) (c)

Figure 5: (a) 13 contours of sequence 1. (b) 7 contours of sequence 2. (c) Optimal path. See text for details.

Sequence 1	1	2	3	4	5	6	7	8	9	10	11	12	13
Original	1'	2'	2'	3'	4'	4'	4'	5'	6'	7'	7'	8'	9'

Table 1: Frames of the created sequence 1 and the corresponding frames of the original sequence.

'he taught' are time aligned manually (there are 26 contours in both S_1 and S_2). Procrustes distances between time aligned contours are calculated with our correspondence recovery method and Kambhamettu's method (Kambhamettu and Goldgof, 1994), respectively. The comparison between these two methods is shown in figure 4. In this figure, the Procrustes distance (circle) calculated according to our point recovery method is smaller than the Procrustes distance (triangular) from Kambhamettu's method for each pair of time aligned contours. One can see that our point correspondence recovery method works better for the time alignment problem.

5.2 Validation of time alignment

We have performed several experiments to validate our time alignment algorithm. We present one such experiment below. First, one repetition of the word 'golly' was selected as the original sequence. Then two sequences were created from the original sequence for the experiment. For clarity, let n' denote the n^{th} frame in the original sequence, let n denote the n^{th} frame in the created sequence.

In this experiment, sequence 1 was created with 13 frames, which are frames $1', 2', 2', 3', 4', 4', 4', 5',$

Sequence 2	1	2	3	4	5	6	7
Original	1'	2'	4'	4'	6'	7'	9'

Table 2: Frames of the created sequence 2 and the corresponding frames of the original sequence.

$6', 7', 7', 8'$ and $9'$ of the original sequence. Sequence 2 was created as a sparse data set which has 7 frames which are frames $1', 2', 4', 4', 6', 7'$ and $9'$ of the original sequence. The frame numbers of these two created sequences and corresponding frame numbers of the original sequence are listed in table 1 and 2 for sequence 1 and 2, respectively. Frames in these two sequences were selected with random duplication, and some frames in sequence 1 are omitted in sequence 2. These two sequences are shown in figure 5(a) and figure 5(b). The obtained optimal path between them is shown in figure 5(c). The longest sequence is always selected as the reference sequence. The obtained optimal path is $(1,1),(2,3),(3,6),(4,7),(5,9),(6,11),(7,13)$, i.e. (1',1'),(2',2'),(4',4'),(4',4'),(6',6'),(7',7'),(9',9') of the original sequence. All contours are correctly aligned. After alignment the shorter sequence is interpolated to the length of the longer one.

6 Experiments

The algorithm was applied to three datasets to demonstrate its use. All utterances were repeated at one second intervals. The first utterance is the word 'golly', which has four repetitions. This word requires considerable high-to-low motion and deformation of the tongue within the vocal tract space. The second utterance is five repetitions of the phrase 'he taught', which has front-to-back tongue motions. The third utterance consists of four repetitions of 'ee'-'aa'. The alternating 'ee'-'aa' motions involve the two most extreme positions of the tongue in the vocal tract: high-front (ee) and low-back (aa). These three utterances are from three different subjects. The standard deviation was calculated at each of the individual sample points for all the repetitions. These individual standard deviations were then averaged for each contour to produce a global measure of variation: the average standard deviation.

Figures 6-8 display the three datasets in several ways. In each figure, (a) shows a time-motion

display (x, y, t) of the averaged midsagittal tongue contour deforming over time. The right axis shows frame numbers and the left axis shows the tongue height. The x-axis shows the xy-coordinate points for each contour. The tongue tip is on the right and the back on the left. Figures 6-8 (b) show the same material projected on XT-plane. Local tongue displacement goes from very high (black) to very low (white). Figures 6-8 (c) display the average standard deviation for each frame after time aligning all repetitions. Figures 6-8 (d) depict the standard deviation for each point in each frame.

Figure 6 shows the tongue contour sequence for 'golly'. Figure 6(a) and (b) show that during the transition from /ɑ/ to /l/, the tongue's maximum displacement shifts backward and the tip elevates (contour 9-10). The maximum displacement shifts forward again for /i/. Figure 6(c) shows a small average SD for most contours. The exception, number 15 has a 1.7 mm SD. This value is explained in figure 6(d) by the large variance in the tongue root, due probably to extrapolation error when extending the contour. The average SD for the entire contour sequence was 0.97 mm.

The alignment of 'he taught' presents a variant on this error pattern. Figure 7 (a) shows less high-to-low variation than 'golly', as expected. Figure 7(b) provides a clearer display of the exact transition frames. For example frames 5-6 and 15-16 depict the very anterior displacement of the two /t/'s, respectively. The last 12 frames are a pause. Figure 7(c) shows large variability at frames 14-15. Figure 7(b) shows this is the transition from /ɔ/ to /t /. This spike in variability (SD=3.3mm) was larger than that expected from normal variability, suggesting a misalignment due to a poor interpolation caused by the inadequate sampling rate of ultrasound. The back-vowel to front-stop motion uses rapid tongue motions that cover a long distance, and the ultrasound frame rate of 30 Hz under samples this motion. Multiple repetitions are likely to capture different moments in time and increase variability. The phonemes in this word all use a relatively closed vocal tract, therefore, its tongue movements are more likely due to local deformation (tongue only) than global deformation (contributed by jaw opening), and would be less well aligned using rigid body registration. The pause (frames 18-26) had a low variance (SD<1.5mm). In addition

spikes in variability are seen at frames 5, 6, 8, and 11. Figure 7(d) suggests that the large average SD's seen in these frames are due to large local SD's at the tongue tip. These are probably due to contour extension errors, which can be eliminated during later data analysis and, therefore, of little consequence.

The third data set, /i/-/ɑ/, contained only one motion, which had large local and global deformations, and a rapid motion from high-front to low-back tongue position. The within-vowel variability was between that of 'golly' and 'he taught' (max SD=1.4 mm) with the larger SD's occurring near the transition. Figure 8(a) and (b) show this utterance is simpler than the other two; the movement involves a single transition and long vowel steady states. Figure 8(c) shows that the largest errors occurred at the vowel-to-vowel transition (Frame 6) as expected. A large error is also observed in the middle of the /ɑ/ (Frame 13). For this particular data set, frame 13 of the reference sequence (the longest repetition) was not time aligned with any frame of the other sequences. Thus before averaging, frame 13 of each non-reference sequence was obtained by interpolation to make sure that all sequences had the same number of contours. Errors introduced during the interpolation in turn caused the large variation. The interpolation errors could be reduced if the ultrasound frame rate were increased.

The three data sets contain large changes in temporal and spatial patterns for the tongue. These are consistent with those found in speech. Variance up to 1.5 mm was typical in speech events and pause. Variance above that amount appeared due to contour extension, and interpolation error. The latter was usually caused by rapid tongue motion leading to ultrasound undersampling.

7 Conclusion

In this paper, a method to get the best representation of a speech motion from several repetitions is presented. A set of contour sequences is first time aligned by a shape based Dynamic programming method. The best representation of the speech is then obtained by averaging the time aligned contours from different repetitions. This method has been tested on both synthetic and real data

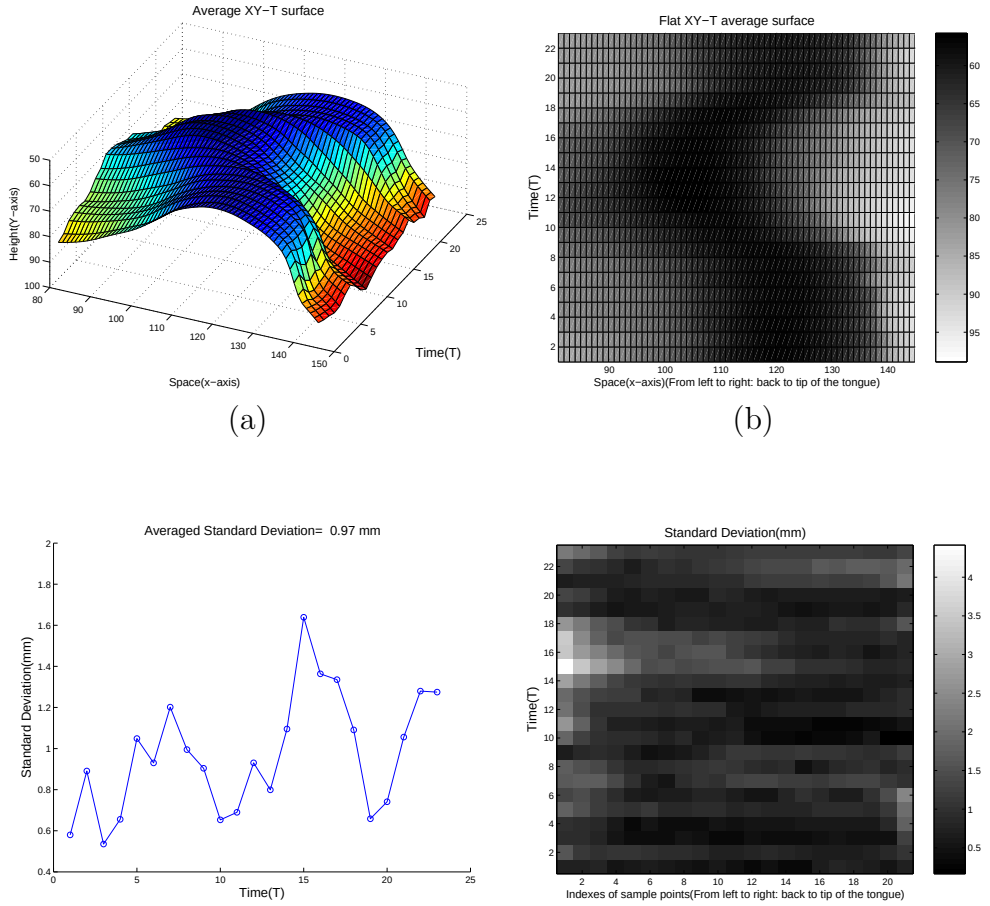


Figure 6: Results for 'golly' in the upright position. Average standard deviation of real distance=0.97 mm. (a) is the average sequence shown in the XY-T space. The average XY-T surface is also shown in the X-T plane in (b). Standard deviation of each frame is shown in (c). The standard deviation of each sample point is shown in the X-T plane in (d).

sets. Variance above 1.5 mm is observed due to errors resulting from contour extension which we used to get corresponding points of endpoints, and due to interpolation errors resulting primarily from ultrasound undersampling.

Acknowledgment

This research was funded in part by NIDCD/NIH grant number R01 DC01758.

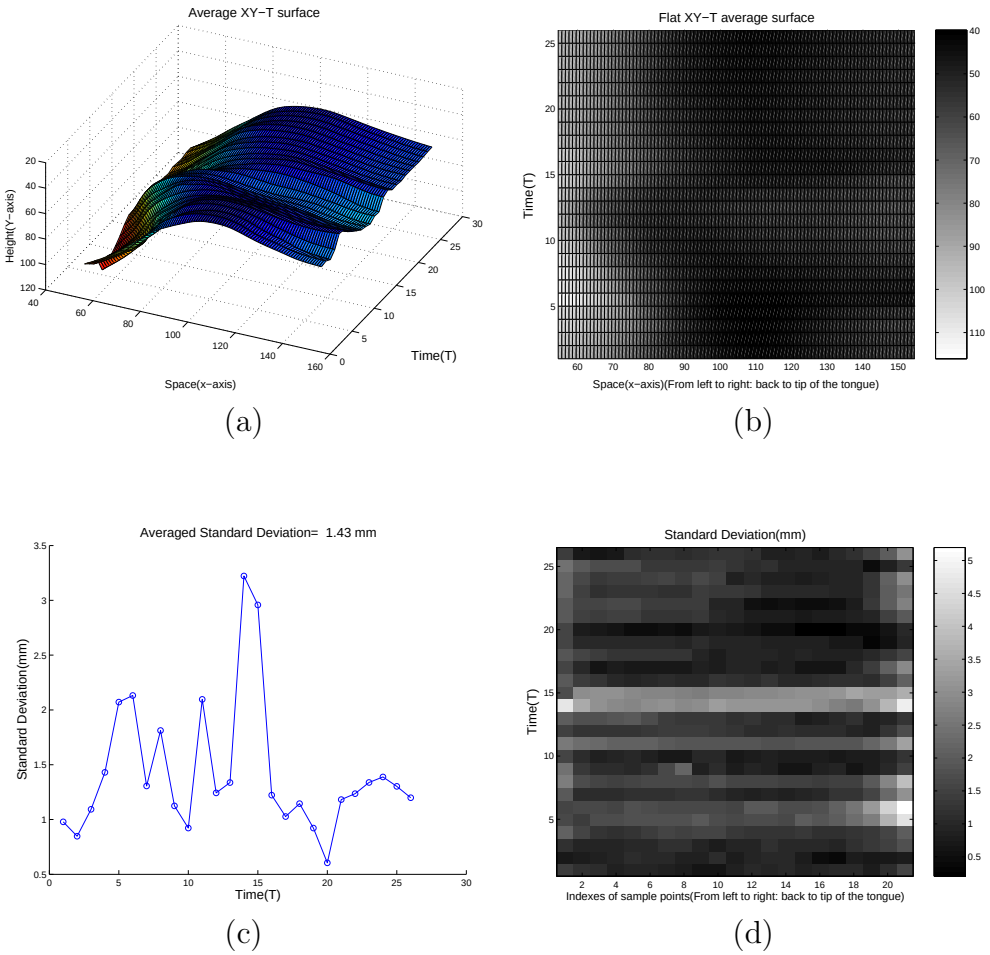


Figure 7: Results for 'he taught' in the upright position. Average standard deviation of real distance=1.43 mm. (a) is the average sequence shown in the XY-T space. The average XY-T surface is also shown in the X-T plane in (b). Standard deviation of each frame is shown in (c). The standard deviation of each sample point is shown in the X-T plane in (d).

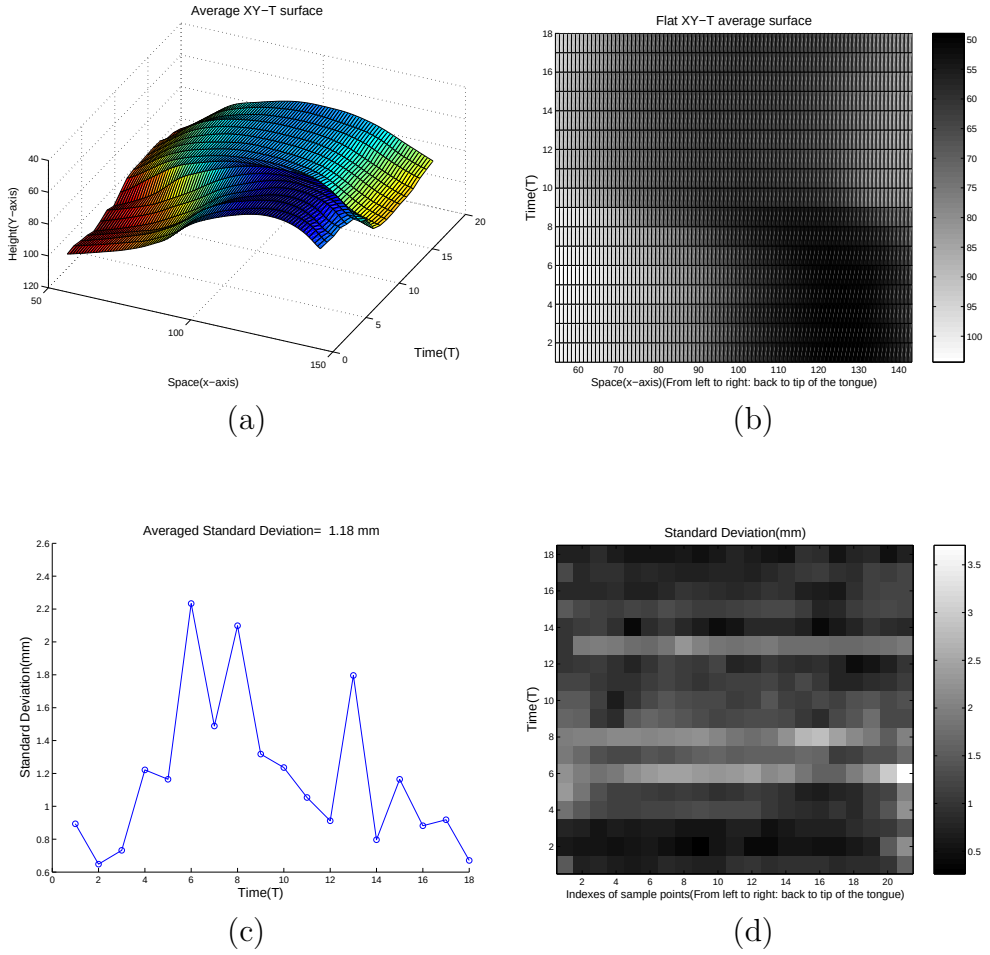


Figure 8: Results for 'ee'-'aa'. Average standard deviation of real distance=1.18 mm. (a) is the average sequence shown in the XY-T space. The average XY-T surface is also shown in the X-T plane in (b). Standard deviation of each frame is shown in (c). The standard deviation of each sample point is shown in the X-T plane in (d).

References

Akgul, Y.S., Kambhamettu, C., & Stone, M. (1999). Automatic extraction and tracking of the tongue contours. *IEEE Transactions on Medical Imaging*, 18(10), 1035-1045.

Amini, A.A., & Duncan, J.S. (1991). Pointwise tracking of left-ventricular motion in 3D. *MOTION91* (pp. 294-299).

Amini, A.A., & Duncan, J.S. (1992). Bending and stretching models for LV wall motion analysis from curves and surfaces. *IVC*, 10(6), 418-430.

Belongie, S., Malik, J., & Puzicha, J. (2001). Matching shapes. *ICCV01* (pp. I: 454-461).

Besl, P.J., & McKay, N.D. (1992). A method for registration of 3-D shapes. *PAMI*, 14(2), 239-256.

Dryden I.L., & Mardia K.V. (1998). Statistical shape analysis. New York: Wiley.

Duta, N., Jain, A.K., & Dubuisson-Jolly, M.P. (2001). Automatic construction of 2D shape models. *PAMI*, 23(5), 433-446.

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

Li, M., Kambhamettu, C., & Stone, M. (2003). Snake for band edge extraction and its applications. *6th IASTED International Conference on Computers, Graphics, and Imaging.*, Honolulu, Hawaii,USA.

Parthasarathy, V., Stone, M., & Prince, J.L. (2003). Spatiotemporal visualization of the tongue using Ultrasound and Kriging. *Proc. Of SPIE-Medical Imaging.*

Sakoe, M., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process*, 26, 588-595.

Stone, M., & Davis, E.P. (1995). A head and transducer support system for making ultrasound images of tongue/jaw movement. *The Journal of The Acoustical Society of America*, 6, 3107-3112.

Wang, Y., Peterson, B.S., & Staib, L.H. (2000). Shape-based 3D surface correspondence using geodesics and local geometry. *CVPR00* (pp. II:644-651).

Yang, C., & Stone, M. (2002). Dynamic programming method for temporal registration of three-dimensional tongue surface motion from multiple utterances. *Speech Communication*, 38(1-2), 201-209.