LEARNING AND ADAPTATION OF A TONGUE SHAPE MODEL WITH MISSING DATA

Mohsen Farhadloo *Miguel Á. Carreira-Perpiñán*

EECS, School of Engineering, University of California, Merced. Merced, CA, USA
Email: {mfarhadloo,mcarreira-perpinan}@ucmerced.edu

ABSTRACT

Using data-driven techniques and ultrasound data, it is possible to learn models that reconstruct the tongue shape of a speaker with submillimetric accuracy given the location of 3–4 fleshpoints, and to adapt these models to a new speaker for which little data is available. In practice, tongue contours extracted from ultrasound imaging are often incomplete because of shadowing, noise and other factors. We extend these models to deal with missing data during learning and adaptation, and show that submillimetric accuracy can still be achieved even with relatively large amounts of missing data.

***Index Terms*—** tongue model, model adaptation, missing data, articulatory databases, ultrasound.

1. INTRODUCTION

Realistic models of the shape of the vocal tract, in particular the tongue, are useful for talking heads [1], articulatory synthesis and inversion, tracking in ultrasound and MRI, and reconstructing the tongue contour in articulatory databases such as MOCHA [2], among other applications. Landmark-based models [3, 4, 5] use as control parameters the location on the tongue contour of a fixed number of fleshpoints (landmarks), given which the entire tongue shape is reconstructed (fig. 1). The predictive mapping from landmarks to contours is learned given a dataset of full contours obtained from ultrasound. Nonlinear mappings [5] achieve submillimetric errors (0.2–0.3 mm per point on the tongue). Since collecting contours is costly, it is convenient to adapt automatically a predictive mapping trained on lots of data from one speaker to a new speaker given only a few full contours from the latter. This can be achieved using a feature-transformation approach [6, 7], resulting in errors just slightly larger than training with a large dataset (0.1–0.3 mm more).

In practice, limitations of the recording technique typically cause missing values in the contour dataset. With ultrasound, contours can appear incomplete for several reasons (fig. 2): disturbances such as noise and shadows (e.g. by the hyoid bone on the back of the tongue) occlude portions of the contour; the back or the tip of the tongue may exit the window of visibility of the probe if moving excessively forward or backward; tongue surfaces disappear if they become approximately parallel to the probe (e.g. in sounds where the tongue tip curves upwards). In addition, the (manual or automatic) segmentation of the tongue contour can also be incomplete. Missing data can also be created artificially: one can increase the temporal rate of ultrasound by having the probe skip scan lines, so that each image has lower resolution (thus trading off missing data in the temporal and spatial domains) [8]. This can be useful with sounds such as clicks, for which the tongue moves extremely fast.

All these situations result in partially complete contours for learning or adaptation. The simplest option, discarding incomplete contours, is wasteful, because recording and segmentation are costly

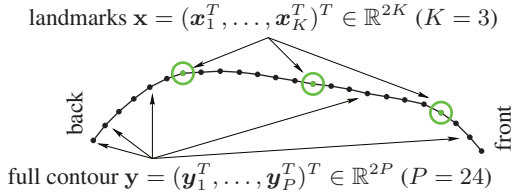

Fig. 1. Prediction problem: given the 2D locations of K landmarks on the tongue midsagittal contour $\mathbf{x}$, reconstruct the entire contour $\mathbf{y}$, represented by P 2D points, by a predictive mapping $\mathbf{y} = \mathbf{f}(\mathbf{x})$.

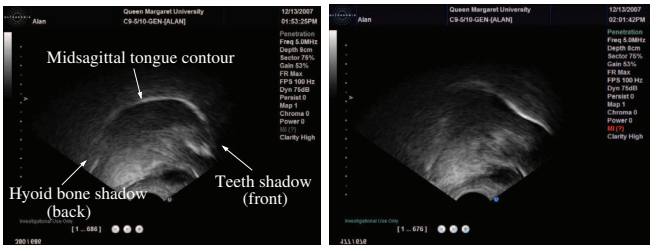

Fig. 2. Missing tongue portions in typical ultrasound images.

and cumbersome (requiring significant expert intervention); and it can severely reduce the number of complete contours available, particularly in the adaptation setting, where very few contours are collected in the first place. This makes it imperative to use all contours, complete or not. Another approach to the problem is to reconstruct first the contours and then learn or adapt the model. This might be achieved with a matrix completion algorithm, or directly during the segmentation itself, or combining ultrasound with another recording technology (such as MRI) to complete the contours. In this paper we follow a third approach: we assume we are given a dataset of full contours with missing values and learn or adapt a model *without explicitly reconstructing the contours*, by exploiting the implicit temporal and spatial redundancy of the tongue contours. We make no specific assumptions about the mechanism that caused the values to be missing. We use a generic approach called *missing data deleted* [9] that is applicable to regression models in which the variables having missing values appear as a sum-of-squares form. In this case the objective function has a single term for each data item, and we drop the ones that are missing. As long as the remaining terms sufficiently constrain the model—in our case thanks to the temporal and spatial smoothness of the tongue contours—the resulting mapping will be able to achieve accurate predictions. We review the learning and adaptation of tongue shape models and describe our extension to missing data in section 2, and demonstrate their empirical performance in section 3. Although our experiments focus on the midsagittal tongue contour, our algorithms carry over to 3D shapes of the tongue or other vocal tract articulators.

2. LEARNING AND ADAPTING THE TONGUE MODEL

2.1. Predictive model with missing data

We define the tongue reconstruction problem (fig. 1) as in [5]. Of the P points along the contour, we choose K (say, 3) to represent the landmarks, mimicking electromagnetic articulography (EMA) pellets affixed to the tongue; call this vector $\mathbf{x} \in \mathbb{R}^{2K}$. The goal then is to predict all P points (or rather, the remaining $P-K$) using a mapping $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ that we estimate from a training set. Consider a training set $\{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$, collected in matrices $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ of $2K \times N$ and $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$ of $2P \times N$. Each $\mathbf{x}_n$ is a contour subset $\mathbf{x} = (\boldsymbol{x}_1^T, \dots, \boldsymbol{x}_K^T)^T \in \mathbb{R}^{2K}$ consisting of K landmarks $\boldsymbol{x}_i \in \mathbb{R}^2$, and each $\mathbf{y}_n$ is a full contour $\mathbf{y} = (\boldsymbol{y}_1^T, \dots, \boldsymbol{y}_P^T)^T \in \mathbb{R}^{2P}$. The goal is to fit a predictive mapping $\mathbf{f}\colon \mathbf{y} = \mathbf{f}(\mathbf{x})$ by least squares from $\mathbf{x}$ to $\mathbf{y}$. When the training set is complete (without any missing values) the objective function of [5] consists of $2PN$ additive terms:

$$\min_{\mathbf{f}} E(\mathbf{f}) = \sum_{n=1}^{N} \sum_{j=1}^{2P} (y_{jn} - (\mathbf{f}(\mathbf{x}_n))_j)^2 \tag{1}$$

where y_{jn} and $(\mathbf{f}(\mathbf{x}_n))_j$ are the j^{th} component of the n^{th} training and predicted output vectors, respectively. In the missing data deleted approach, and assuming that no landmarks $\mathbf{x}$ are missing, we drop terms corresponding to missing values y_{jn}:

$$\min_{\mathbf{f}} E(\mathbf{f}) = \sum_{\text{present } n,j} (y_{jn} - (\mathbf{f}(\mathbf{x}_n))_j)^2. \tag{2}$$

As long as we have enough values present, $\mathbf{f}$ is still determined by the data. We have developed this approach for linear mappings and radial basis function (RBF) networks, but report here only the latter. Consider an RBF network with M Gaussian basis functions $\phi_m(\mathbf{x}) = \exp(-\frac{1}{2}\|(\mathbf{x}-\boldsymbol{\mu}_m)/\sigma\|^2)$ of width σ and centre $\boldsymbol{\mu}_m$:

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}) + \mathbf{w} = \sum_{m=1}^{M} \mathbf{w}_m \phi_m(\mathbf{x}) + \mathbf{w}.$$

As is common with RBFs, we apply a suboptimal but efficient training strategy. We first obtain the basis function centres with k-means on the input data and fix them. Then we find σ by cross-validation. To avoid overfitting we add a quadratic regularisation term on $\mathbf{W}$ with weight $\lambda \geq 0$ to the objective function. Finally, the minimum over the weights $\mathbf{W}$ is the unique solution of the linear system

$$\begin{gathered}\Big(\boldsymbol{\Phi}_{j,\mathcal{P}}\boldsymbol{\Phi}_{j,\mathcal{P}}^T - \tfrac{1}{N_{j,\mathcal{P}}}(\boldsymbol{\Phi}_{j,\mathcal{P}}\mathbf{1}_{j,\mathcal{P}})(\boldsymbol{\Phi}_{j,\mathcal{P}}\mathbf{1}_{j,\mathcal{P}})^T + \lambda\boldsymbol{\Phi}(\boldsymbol{\mu})\Big)\mathbf{w}_j = \\ \boldsymbol{\Phi}_{j,\mathcal{P}}\big(\mathbf{I} - \tfrac{1}{N_{j,\mathcal{P}}}\mathbf{1}_{j,\mathcal{P}}\mathbf{1}_{j,\mathcal{P}}^T\big)\mathbf{y}_{j,\mathcal{P}}^T \\ w_j = \tfrac{1}{N_{j,\mathcal{P}}}\mathbf{1}_{j,\mathcal{P}}^T\left(\mathbf{y}_{j,\mathcal{P}} - \boldsymbol{\Phi}_{j,\mathcal{P}}^T\mathbf{w}_j\right)\end{gathered}$$

where the $N_{j,\mathcal{P}} \times 1$ vector $\mathbf{y}_{j,\mathcal{P}}$ contains the present components of the j^{th} row of matrix $\mathbf{Y}$, $\mathbf{1}_{j,\mathcal{P}}$ is a $N_{j,\mathcal{P}} \times 1$ vector of ones, $\boldsymbol{\Phi}_{j,\mathcal{P}}$ contains the columns of $\boldsymbol{\Phi}$ corresponding to $\mathbf{y}_{j,\mathcal{P}}$, and $\mathbf{w}_j^T$ is the j^{th} row of $\mathbf{W}$.

2.2. Adaptation with missing data

We are now given a small number of full contours from a new speaker, insufficient to train reliably a predictive mapping. Instead, we adapt an existing mapping $\mathbf{f}$ from another speaker that was trained with lots of data. We follow the local feature transformation approach of [7], where we estimate two invertible linear transformations $\mathbf{g}_\mathbf{x}$ and $\mathbf{g}_\mathbf{y}$ (with few parameters) that map new data to old data in the landmark ($\mathbf{x}$) and contour ($\mathbf{y}$) spaces, respectively. Each mapping $\mathbf{g}$ is defined as a concatenation of separate, local linear mappings that map a 2D point to another 2D point:

$$\tilde{\mathbf{x}} = \mathbf{g}_\mathbf{x}(\mathbf{x}) = \begin{pmatrix} \mathbf{A}_1^\mathbf{x}\boldsymbol{x}_1 + \mathbf{b}_1^\mathbf{x} \\ \cdots \\ \mathbf{A}_K^\mathbf{x}\boldsymbol{x}_K + \mathbf{b}_K^\mathbf{x} \end{pmatrix}, \quad \tilde{\mathbf{y}} = \mathbf{g}_\mathbf{y}(\mathbf{y}) = \begin{pmatrix} \mathbf{A}_1^\mathbf{y}\boldsymbol{y}_1 + \mathbf{b}_1^\mathbf{y} \\ \cdots \\ \mathbf{A}_P^\mathbf{y}\boldsymbol{y}_P + \mathbf{b}_P^\mathbf{y} \end{pmatrix}.$$

The adapted predictive mapping is given by $\mathbf{g}_\mathbf{y}^{-1} \circ \mathbf{f} \circ \mathbf{g}_\mathbf{x}$ and requires estimating $6(K+P)$ parameters that we write collectively as $(\mathbf{A}^\mathbf{x}, \mathbf{b}^\mathbf{x}, \mathbf{A}^\mathbf{y}, \mathbf{b}^\mathbf{y})$; this is far fewer parameters than training $\mathbf{f}$ directly in (1). The adapted model is linear if $\mathbf{f}$ was linear, and a basis function network where the basis functions are non-radial if $\mathbf{f}$ was an RBF network. When the adaptation dataset is complete (no missing data) the objective function of [7] is

$$E(\mathbf{A}^\mathbf{x}, \mathbf{b}^\mathbf{x}, \mathbf{C}^\mathbf{y}, \mathbf{d}^\mathbf{y}) = \sum_{n=1}^{N} \left\|\mathbf{y}_n - \mathbf{g}_\mathbf{y}^{-1}(\mathbf{f}(\mathbf{g}_\mathbf{x}(\mathbf{x})))\right\|^2 \tag{3}$$

where we introduce $\mathbf{C}_j^\mathbf{y} = (\mathbf{A}_j^\mathbf{y})^{-1}$, $\mathbf{d}_j^\mathbf{y} = -(\mathbf{A}_j^\mathbf{y})^{-1}\mathbf{b}_j^\mathbf{y}$, simplifying the optimisation (no matrix appears as an inverse). We can add a regularisation term to E as in [7, 10], particularly with small datasets, but in this paper we do not do so to keep the experiments simple. With missing values in the adaptation contours and again assuming no landmarks are missing, we drop the terms corresponding to missing values y_{jn} and write the following objective function:

$$E(\mathbf{A}^\mathbf{x}, \mathbf{b}^\mathbf{x}, \mathbf{C}^\mathbf{y}, \mathbf{d}^\mathbf{y}) = \sum_{n=1}^{N} \left\|\mathbf{m}_n \circ [\mathbf{y}_n - \mathbf{g}_\mathbf{y}^{-1}(\mathbf{f}(\mathbf{g}_\mathbf{x}(\mathbf{x})))]\right\|^2 \tag{4}$$

where we define the $2P \times N$ binary matrix $\mathbf{M} = (\mathbf{m}_1, \dots, \mathbf{m}_N)$ so $m_{in} = 0$ or 1 indicates y_{in} is missing or present, respectively, and $\circ$ denotes elementwise product. (In the actual code we do not use $\mathbf{M}$ and simply skip zero terms.) To minimise E we need its gradients:

$$\begin{aligned}
\tfrac{\partial E}{\partial \operatorname{vec}(\mathbf{A}^\mathbf{x})} &= 2\textstyle\sum_{n=1}^{N} \mathbf{r}_{n,\mathcal{M}}^T \mathbf{P}_{n,\mathcal{M}}^\mathbf{x} & \mathbf{P}_{n,\mathcal{M}}^\mathbf{x} &= \tfrac{\partial \mathbf{r}_{n,\mathcal{M}}}{\partial \operatorname{vec}(\mathbf{A}^\mathbf{x})} \\
\tfrac{\partial E}{\partial \operatorname{vec}(\mathbf{b}^\mathbf{x})} &= 2\textstyle\sum_{n=1}^{N} \mathbf{r}_{n,\mathcal{M}}^T \mathbf{Q}_{n,\mathcal{M}}^\mathbf{x} & \mathbf{Q}_{n,\mathcal{M}}^\mathbf{x} &= \tfrac{\partial \mathbf{r}_{n,\mathcal{M}}}{\partial \operatorname{vec}(\mathbf{b}^\mathbf{x})} \\
\tfrac{\partial E}{\partial \operatorname{vec}(\mathbf{C}^\mathbf{y})} &= 2\textstyle\sum_{n=1}^{N} \mathbf{r}_{n,\mathcal{M}}^T \mathbf{P}_{n,\mathcal{M}}^\mathbf{y} & \mathbf{P}_{n,\mathcal{M}}^\mathbf{y} &= \tfrac{\partial \mathbf{r}_{n,\mathcal{M}}}{\partial \operatorname{vec}(\mathbf{C}^\mathbf{y})} \\
\tfrac{\partial E}{\partial \operatorname{vec}(\mathbf{d}^\mathbf{y})} &= 2\textstyle\sum_{n=1}^{N} \mathbf{r}_{n,\mathcal{M}}^T \mathbf{Q}_{n,\mathcal{M}}^\mathbf{y} & \mathbf{Q}_{n,\mathcal{M}}^\mathbf{y} &= \tfrac{\partial \mathbf{r}_{n,\mathcal{M}}}{\partial \operatorname{vec}(\mathbf{d}^\mathbf{y})}
\end{aligned}$$

where $\mathbf{r}_{n,\mathcal{M}} = \mathbf{m}_n \circ [\mathbf{y}_n - \operatorname{diag}\left(\boldsymbol{z}_{in}^T \otimes \mathbf{I}_2\right) \operatorname{vec}(\mathbf{C}^\mathbf{y}) - \operatorname{vec}(\mathbf{d}^\mathbf{y})] = \mathbf{m}_n \circ [\mathbf{y}_n - \operatorname{diag}(\mathbf{C}_i^\mathbf{y})\,\mathbf{z}_n - \operatorname{vec}(\mathbf{d}^\mathbf{y})]$ and $\mathbf{z}_n = (\boldsymbol{z}_{n1}^T, \dots, \boldsymbol{z}_{nP}^T)^T = \mathbf{f}(\mathbf{g}_\mathbf{x}(\mathbf{x}_n))$. For a linear mapping $\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{w}$ we obtain ($\otimes$ is the Kronecker product):

$$\begin{aligned}
\mathbf{P}_{n,\mathcal{M}}^\mathbf{x} &= (\mathbf{m}_n\mathbf{1}^T) \circ (-\operatorname{diag}(\mathbf{C}_i^\mathbf{y})\,\mathbf{W}\operatorname{diag}\left(\boldsymbol{x}_i^T \otimes \mathbf{I}_2\right)) \\
\mathbf{Q}_{n,\mathcal{M}}^\mathbf{x} &= (\mathbf{m}_n\mathbf{1}^T) \circ (-\operatorname{diag}(\mathbf{C}_i^\mathbf{y})\,\mathbf{W}) \\
\mathbf{P}_{n,\mathcal{M}}^\mathbf{y} &= (\mathbf{m}_n\mathbf{1}^T) \circ (-\operatorname{diag}\left(\boldsymbol{z}_{in}^T \otimes \mathbf{I}_2\right)) \\
\mathbf{Q}_{n,\mathcal{M}}^\mathbf{y} &= (\mathbf{m}_n\mathbf{1}^T) \circ (-\mathbf{I}_{2P})
\end{aligned}$$

and for an RBF network (where $\mathbf{K} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M)$):

$$\begin{aligned}
\mathbf{P}_{n,\mathcal{M}}^\mathbf{x} &= (\mathbf{m}_n\mathbf{1}^T) \circ \left(\tfrac{1}{\sigma^2}\operatorname{diag}(\mathbf{C}_i^\mathbf{y})\mathbf{W}\operatorname{diag}(\Phi'_n)(\tilde{\mathbf{x}}_n\mathbf{1}_M^T - \mathbf{K})^T\operatorname{diag}(\boldsymbol{x}_i^T \otimes \mathbf{I}_2)\right) \\
\mathbf{Q}_{n,\mathcal{M}}^\mathbf{x} &= (\mathbf{m}_n\mathbf{1}^T) \circ \left(\tfrac{1}{\sigma^2}\operatorname{diag}(\mathbf{C}_i^\mathbf{y})\mathbf{W}\operatorname{diag}(\Phi'_n)(\tilde{\mathbf{x}}_n\mathbf{1}_M^T - \mathbf{K})^T\right)
\end{aligned}$$

and the same equations for $\mathbf{P}_{n,\mathcal{M}}^\mathbf{y}$ and $\mathbf{Q}_{n,\mathcal{M}}^\mathbf{y}$ as for the linear mapping. Both the linear and RBF cases require nonlinear optimisation of E using these gradient equations. We found the BFGS algorithm [11] to be effective and reliable. Since E has local optima, we initialise it from the solution obtained by a PCA adaptation method [7].

2.3. Computational complexity

Both missing-data objective functions, (2) for training and (4) for adaptation, have $(1-\rho)PN$ terms where $\rho \in [0,1]$ is the proportion of missing data. Since training/adaptation have a cost linear in NP, the missing-data optimisation is actually faster (assuming the same number of iterations), though the accuracy degrades if ρ is too large. Imputation methods (e.g. with splines, sec. 3), are slower because they first reconstruct all contours (so $\rho = 0$), and then optimise.

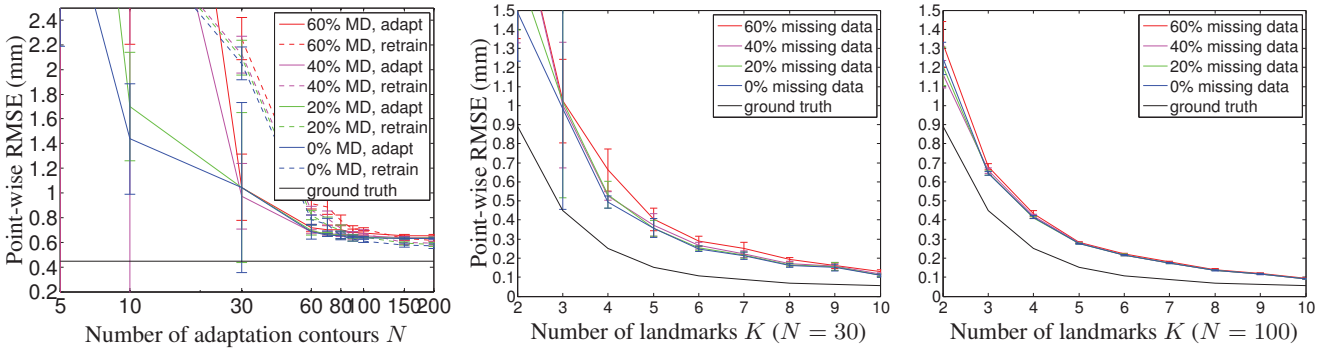

Fig. 3. Missing at random pattern. *Left plot*: predictive error E for adaptation and retraining with different amounts of missing data, as a function of the number of adaptation contours N (for $K = 3$ landmarks). *Right two plots*: predictive error E for adaptation as a function of the number of landmarks K (for $N = 30$ and $N = 100$ adaptation contours, respectively). The ground truth line is the optimal baseline (training with lots of contours without missing data). Errorbars over 5 random choices of the N adaptation contours.

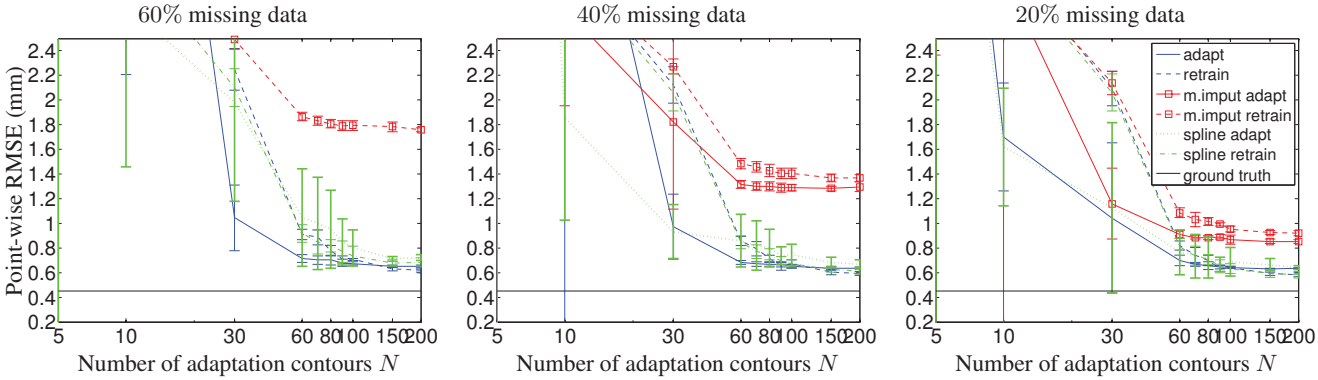

Fig. 4. Missing at random pattern. Like fig. 3(left) but comparing our algorithms (blue lines) with retraining/adaptation after mean imputation and spline imputation.

3. EXPERIMENTS WITH ULTRASOUND CONTOURS

Dataset We used the ultrasound database created at Queen Margaret University and the University of Edinburgh [5]. It contains two speakers (one male, `maaw0`, and one female, `feal0`) with different Scottish accents. Each speaker recorded a set of 20 British TIMIT sentences designed to be phonetically balanced. Recordings for `maaw0` and `feal0` were done in two and one sessions respectively; we used only the first session of `maaw0`. We used `maaw0` to obtain a reference model, which we adapted to data from `feal0` as target speaker having missing values. We partitioned the data at random so `maaw0` contained 2 236 training frames and 1 491 testing, and `feal0` contained up to 4 363 training and 2 909 testing. Each tongue contour contains $P = 24$ points for both speakers. For most experiments, we select $K = 3$ pellets at indices (2,9,19), roughly corresponding to the MOCHA pellet positioning [2].

Missing data patterns We considered two different patterns: (1) *missing at random*, where any point in any contour has a uniform probability ρ of being missing (for $\rho = 0\%, 20\%, 40\%, 60\%$). This is representative of random ultrasound noise. (2) *Missing runs*, where each contour has a single sequence of 8 consecutive points that are missing, at the front, middle or back of the tongue (a proportion of 33% missing data). This is representative of shadowing and other effects.

Comparison methods In all cases we use RBF networks for the predictive mapping $\mathbf{f}$. We compared our missing-data adaptation algorithm of sec. 2.2 with: (1) an *optimal baseline* where we train the model on a large amount of complete full contours from the new speaker. (2) *Retraining* a new model from scratch (i.e., disregarding the reference model $\mathbf{f}$) using only the adaptation, incomplete contours of the new speaker, with our missing-data learning algorithm of section 2.1. (3) *Mean imputation*: reconstructing each missing value y_{jn} with the mean of all present values y_{jm} and then running the complete-data adaptation algorithm of [7]. (4) *Spline imputation*: reconstructing the missing points by fitting a cubic spline separately to each contour using the points present in it and then running the complete-data adaptation algorithm of [7]. With the spline, we assume missing points are equidistant within a run, and for missing runs at either end of the contour, we estimate the run length proportionally to the runs that are present in that same contour.

Results Fig. 3(left) plots the error after adaptation/retraining as a function of number of contours N for different proportions of missing data. Both our learning and adaptation algorithms can tolerate

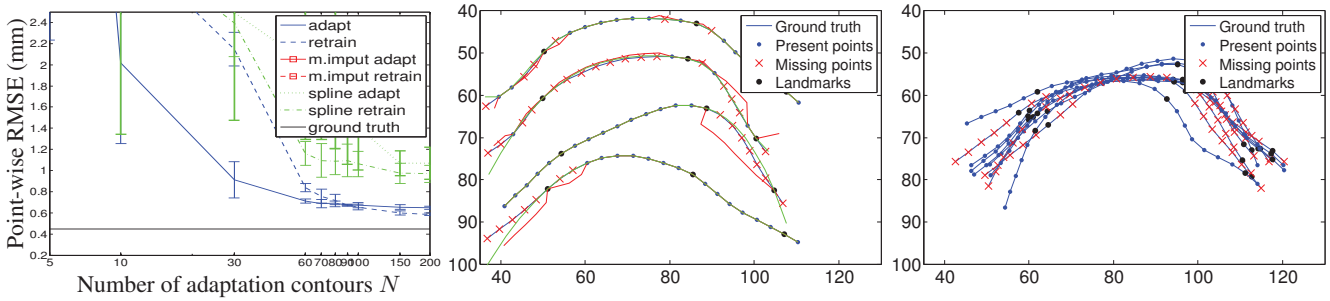

Fig. 5. Missing runs pattern. *Left*: like fig. 4. *Middle*: illustration for some contours of the mean and spline imputation with the random and runs missing data patterns. *Right*: sample contours with missing runs.

up to 60% random missing data with little performance decrease. In adaptation, with as few as $N = 30$ contours and almost independently of the proportion of missing data (up to 60%), we achieve an error which is less than 0.5 mm from the optimal baseline (note the ultrasound measurement error itself is about 0.4 mm). This is also seen on fig. 3(right) over the range of K. With very few contours ($N < 30$), up to 20% missing data is still tolerated. Retraining is also largely insensitive to missing data up to 60%, but it is useless for $N < 30$ and it only catches up with adaptation for $N = 80$ to 150 contours (the larger N the more values are missing), validating the effectiveness of adaptation with small datasets.

Figs. 4–5 compare our adaptation/retraining algorithm (blue lines) with mean and spline imputation. From fig. 5(middle) we see that mean imputation often produces significantly distorted tongue reconstructions, while spline imputation can be very good depending on which points are missing; it tends to produce errors when long, curved runs are missing, particularly at the contour ends. The predictive errors for retraining and adaptation are consistent with this: mean imputation does very poorly (off the plot in missing runs); spline imputation does slightly worse than our algorithm for random missing data, but significantly worse for missing runs (additional error of 0.4 mm or more). Thus, our algorithm is preferable in terms of accuracy, computation time and preprocessing needed.

Adapting with our algorithm with $N = 60$ contours takes 7 minutes with no missing data and 5.6 min. with 60% missing data.

4. DISCUSSION

Our algorithms are practically convenient because of their simplicity (no special preprocessing required of the missing data), fast runtime and good accuracy. A limitation is that none of the landmark locations can be missing. This is not very problematic in the practically common case of missing runs if landmarks correspond to the location of pellets in EMA as in MOCHA [2]: missing runs typically occur at the back or tip of the tongue, where pellets are not located (pellets attached to the tip easily drop, while the gag reflex prevents attaching pellets deep inside the throat). However, more sophisticated algorithms that can deal with missing inputs are possible and should be studied in future work.

The algorithm in [12] considers a special type of adaptation with missing data, where in each adaptation contour all points are missing except the landmarks. That algorithm is specifically intended to reconstruct the tongue contour in EMA databases such as MOCHA [2], and is not applicable to our local feature transformation model.

5. CONCLUSION

We have extended a landmark-based model of the tongue shape to deal with missing data at both learning and adaptation times. Compared to the case where there is no missing data, the new algorithms achieve a comparable accuracy with less computation time even when much of the data is in fact missing, thanks to the temporal and spatial redundancy of the data. They require no special user preprocessing beyond indicating what values are missing. A limitation of our approach, which future work may address, is that the landmarks themselves cannot be missing.

Acknowledgments. Work funded by NSF award IIS–0711186.

6. REFERENCES

[1] M. M. Cohen, J. Beskow, and D. W. Massaro, "Recent developments in facial animation: An inside view," in *AVSP*, 1998.

[2] A. A. Wrench, "A multi-channel/multi-speaker articulatory database for continuous speech recognition research," in *Phonus*, vol. 5, University of Saarland, 2000.

[3] T. Kaburagi and M. Honda, "Determination of sagittal tongue shape from the positions of points on the tongue surface," *J. Acoustic Soc. Amer.*, vol. 96, no. 3, pp. 1356–1366, Sept. 1994.

[4] P. Badin, E. Baricchin, and A. Vilain, "Determining tongue articulation: From discrete fleshpoints to continuous shadow," in *Proc. Eurospeech*, 1997, pp. 47–50.

[5] C. Qin, M. Á. Carreira-Perpiñán, K. Richmond, A. Wrench, and S. Renals, "Predicting tongue shapes from a few landmark locations," in *Proc. Interspeech*, 2008, pp. 2306–2309.

[6] C. Qin and M. Á. Carreira-Perpiñán, "Adaptation of a predictive model of tongue shapes," in *Proc. Interspeech*, 2009.

[7] C. Qin, M. Á. Carreira-Perpiñán, and M. Farhadloo, "Adaptation of a tongue shape model by local feature transformations," in *Proc. Interspeech*, 2010, pp. 1596–1599.

[8] A. A. Wrench and J. M. Scobbie, "Very high frame rate ultrasound tongue imaging," in *Proc. 9th Int. Seminar on Speech Production (ISSP)*, 2011, pp. 155–162.

[9] A. Gifi, *Nonlinear Multivariate Analysis*, Wiley, 1990.

[10] M. Farhadloo and M. Á. Carreira-Perpiñán, "Regularising an adaptation algorithm for tongue shape models," *ICASSP*, 2012.

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, second edition, 2006.

[12] C. Qin and M. Á. Carreira-Perpiñán, "Reconstructing the full tongue contour from EMA/X–ray microbeam," *ICASSP*, 2010.