



Adaptation of a tongue shape model by local feature transformations

Chao Qin, Miguel Á. Carreira-Perpiñán and Mohsen Farhadloo

EECS, School of Engineering, University of California, Merced, USA

{cqin, mcarreira-perpinan, mfarhadloo}@ucmerced.edu

Abstract

Reconstructing the full contour of the tongue from the position of 3 to 4 landmarks on it is useful in articulatory speech work. This can be done with submillimetric accuracy using nonlinear predictive mappings trained on hundreds or thousands of contours extracted from ultrasound images. Collecting and segmenting this amount of data from a speaker is difficult, so a more practical solution is to adapt a well-trained model from a reference speaker to a new speaker using a small amount of data from the latter. Previous work proposed an adaptation model with only 6 parameters and demonstrated fast, accurate results using data from one speaker only. However, the estimates of this model are biased, and we show that, when adapting to a different speaker, its performance stagnates quickly with the amount of adaptation data. We then propose an unbiased adaptation approach, based on local transformations at each contour point, that achieves a significantly lower reconstruction error with a moderate amount of adaptation data.

Index Terms: tongue model, speaker adaptation, ultrasound, radial basis functions.

1. Introduction

Reconstructing the full tongue shape from a few landmarks has applications in articulatory models, articulatory inversion, visualization, speech production studies, and in tracking the tongue in ultrasound images. Highly accurate (0.2–0.3 mm per point on the tongue) reconstruction for one speaker can be achieved by training a nonlinear predictive mapping from 3–4 landmarks to contours given a dataset with a large (hundreds to thousands) number of contours extracted from ultrasound images [1, 2, 3] (the measurement error is about 0.4 mm per tongue point). Recording and accurately segmenting the ultrasound images is currently a cumbersome, lengthy process involving much intervention by an expert user (since automatic methods are unreliable). Quick, automatic adaptation of an existing well-trained model for a reference speaker given a small number of segmented contours from the new speaker becomes attractive.

Previously, we proposed a simple but effective adaptation algorithm for reconstructing 2D tongue shapes [4], based on a feature normalization approach related to that used in acoustic HMM adaptation [5, 6]. The algorithm uses the adaptation data to learn an invertible linear transformation between the full tongue shape of both speakers. The adapted model maps the new speaker’s landmarks to the old speaker’s, then applies the old predictive mapping, and finally maps back (with the inverse transformation) the full contour to the new speaker space. The algorithm’s results were very good, achieving an accuracy close to the one from the reference model in just seconds of CPU, but it was only tested with data from a single speaker and synthetic transformations. The algorithm was later extended to adapt a

full-contour model given only partial contours [7] and was able to reconstruct the tongue shapes in articulatory databases (Wisconsin XRMB [8] and MOCHA [9]) even though the latter provide not a single full contour.

As we show later, the accuracy of this algorithm deteriorates somewhat when adapting to a completely different speaker; specifically, the prediction error stagnates quickly (with just 5 to 10 adaptation contours) and far from the optimal one that we would achieve if training with abundant data (0.3 to 0.7 mm more, which is over twice that error). There are two basic reasons for this. The algorithm uses a *global feature transformation* in that every 2D point in the contour (including the landmarks) undergoes exactly the same linear transformation, resulting in only 6 adaptation parameters. While this works perfectly under translation, rotation, scaling and global shearing, it does not allow for different transformations in different points of the tongue. This is very restrictive, because we should expect more complex variations arising from anatomical factors, sex and age (for example, the new speaker might have a longer tip but a shorter dorsum than the old one) as well as other factors such as speaking style, language, etc. A second problem with the global feature transformation in [4] is that, in order to simplify the optimization, a proxy objective function was used that introduces a (small) bias in the transformation.

We propose an extension of this algorithm that eliminates the bias and mitigates the stagnation problem. The idea is to use *local, linear feature transformations* at each landmark and at each contour point. This increases the number of parameters and thus the flexibility of the adaptation, which can take advantage of a larger number of adaptation contours and achieve an error much close to the optimal one (0.1 to 0.3 mm more); besides, we optimize the real reconstruction error without bias. In addition, we use a regularization term from [7] that reduces variance if using very few contours. We describe the predictive method, the new adaptation method, and the experiments in sections 2–4.

2. The predictive model of tongue shapes

We want to predict the full tongue contour $\mathbf{y} = (\mathbf{y}_1^T, \dots, \mathbf{y}_P^T)^T \in \mathbb{R}^{2P}$ consisting of P points $\mathbf{y}_i \in \mathbb{R}^2$ given only the positions $\mathbf{x} = (\mathbf{x}_1^T, \dots, \mathbf{x}_K^T)^T \in \mathbb{R}^{2K}$ of K landmarks $\mathbf{x}_i \in \mathbb{R}^2$ (fig. 1). The approach proposed in [1] for *linear mappings* and in [4] for *radial basis function (RBF) networks* fits a predictive mapping \mathbf{f} by minimizing the predictive square error $E(\mathbf{f}) = \sum_{n=1}^N \|\mathbf{y}_n - \mathbf{f}(\mathbf{x}_n)\|^2$ (plus a regularization term for RBFs) given a sufficiently large training set, and $\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{w}$ (linear) or $\mathbf{f}(\mathbf{x}) = \mathbf{W}\Phi(\mathbf{x}) + \mathbf{w}$ (RBF) with M basis functions $\phi_m(\mathbf{x}) = \exp(-\frac{1}{2}\|(\mathbf{x} - \boldsymbol{\mu}_m)/\sigma\|^2)$. The RBF is trained in an efficient but slightly suboptimal way (as commonly done) by fixing the centers $\boldsymbol{\mu}_m$ by k -means and cross-validating the width σ and the regularization parameter λ .

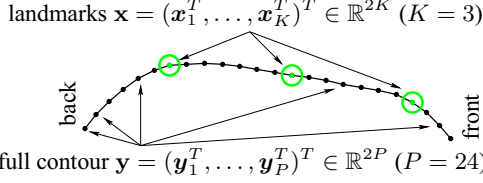


Fig. 1. The prediction problem: given the 2D locations of K landmarks located on the tongue midsagittal contour (\mathbf{x}), reconstruct the entire contour (\mathbf{y}), represented by P 2D points.

3. Adaptation with local transformations

Given a small N -contour adaptation dataset $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, we adapt an existing predictive mapping \mathbf{f} by estimating two invertible linear mappings \mathbf{g}_x and \mathbf{g}_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:

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

The adapted predictive mapping is given by $\mathbf{g}_y^{-1} \circ \mathbf{f} \circ \mathbf{g}_x$ and requires estimating $6(K + P)$ parameters that we write collectively as $(\mathbf{A}^x, \mathbf{b}^x, \mathbf{A}^y, \mathbf{b}^y)$. 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 a radial basis function network. In the global transformation method of [4], $\mathbf{A}_i^x = \mathbf{A}_j^y = \mathbf{A}$ and $\mathbf{b}_i^x = \mathbf{b}_j^y = \mathbf{b}$, so there were only 6 parameters.

Objective function. To estimate $(\mathbf{A}^x, \mathbf{b}^x, \mathbf{A}^y, \mathbf{b}^y)$, we minimize the predictive squared error $E(\mathbf{A}^x, \mathbf{b}^x, \mathbf{C}^y, \mathbf{d}^y)$:

$$\min E(\mathbf{A}^x, \mathbf{b}^x, \mathbf{C}^y, \mathbf{d}^y) = \sum_{n=1}^N \|\mathbf{y}_n - \mathbf{g}_y^{-1} \mathbf{f}(\mathbf{g}_x(\mathbf{x}_n))\|^2$$

where we introduce new parameters $\mathbf{C}_j^y, \mathbf{d}_j^y$, so we work with

$$\mathbf{y} = \mathbf{g}_y^{-1}(\tilde{\mathbf{y}}) = \begin{pmatrix} \mathbf{C}_1^y \tilde{\mathbf{y}}_1 + \mathbf{d}_1^y \\ \dots \\ \mathbf{C}_P^y \tilde{\mathbf{y}}_P + \mathbf{d}_P^y \end{pmatrix}, \quad \mathbf{C}_j^y = (\mathbf{A}_j^y)^{-1} \\ \mathbf{d}_j^y = -(\mathbf{A}_j^y)^{-1} \mathbf{b}_j^y$$

instead of \mathbf{g}_y , simplifying the optimization (no matrix appears as an inverse). In [4], we optimized a proxy function $F(\mathbf{A}, \mathbf{b})$:

$$\min_{\mathbf{A}, \mathbf{b}} F(\mathbf{A}, \mathbf{b}) = \sum_{n=1}^N \|\mathbf{g}_y(\mathbf{y}_n) - \mathbf{f}(\mathbf{g}_x(\mathbf{x}_n))\|^2$$

because $E(\mathbf{A}, \mathbf{b})$ must contain both \mathbf{A} and \mathbf{A}^{-1} and its gradient and optimization are more complicated. Our new approach has several advantages over this (apart from being more flexible): (1) As mentioned in [4], the (\mathbf{A}, \mathbf{b}) that minimize F differ somewhat from those optimizing E and are thus suboptimal. (2) Optimizing E in the new approach is quite simpler because the parameters of \mathbf{g}_x and \mathbf{g}_y are decoupled (see gradients below), in fact E separates over each $(\mathbf{C}_j^y, \mathbf{d}_j^y)$ for fixed $(\mathbf{A}^x, \mathbf{b}^x)$. Besides, the function F is not useful with the new parameters because it has a trivial solution: setting $\mathbf{A}^x, \mathbf{b}^x$ and \mathbf{C}^y to zero then each term in F is a constant that \mathbf{d}^y can pick up, so $F = 0$. However, the local transformation approach does not carry over to the case where the adaptation data contains only partial contours [7] because then we have no data to fit $(\mathbf{C}_j^y, \mathbf{d}_j^y)$.

Optimizing E . The gradients of E are $(\text{vec}(\cdot))$ concatenates the columns of its argument into a single column vector)

$$\begin{aligned} \frac{\partial E}{\partial \text{vec}(\mathbf{A}^x)} &= 2 \sum_{n=1}^N \mathbf{r}_n^T \mathbf{P}_n^x & \mathbf{P}_n^x &= \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{A}^x)} \\ \frac{\partial E}{\partial \text{vec}(\mathbf{b}^x)} &= 2 \sum_{n=1}^N \mathbf{r}_n^T \mathbf{Q}_n^x & \mathbf{Q}_n^x &= \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{b}^x)} \\ \frac{\partial E}{\partial \text{vec}(\mathbf{C}^y)} &= 2 \sum_{n=1}^N \mathbf{r}_n^T \mathbf{P}_n^y & \mathbf{P}_n^y &= \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{C}^y)} \\ \frac{\partial E}{\partial \text{vec}(\mathbf{d}^y)} &= 2 \sum_{n=1}^N \mathbf{r}_n^T \mathbf{Q}_n^y & \mathbf{Q}_n^y &= \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{d}^y)} \end{aligned}$$

where $\mathbf{r}_n(\mathbf{A}^x, \mathbf{b}^x, \mathbf{C}^y, \mathbf{d}^y) = \mathbf{y}_n - \text{diag}(\mathbf{C}_1^y, \dots, \mathbf{C}_P^y) \mathbf{z}_n - \text{vec}(\mathbf{d}^y)$ and $\mathbf{z}_n = (z_{n1}^T, \dots, z_{nP}^T)^T = \mathbf{f}(\mathbf{g}_x(\mathbf{x}_n))$. For the linear mapping function we obtain (\otimes is the Kronecker product)

$$\begin{aligned} \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{A}^x)} &= -\text{diag}(\mathbf{C}_j^y) \mathbf{W} \text{diag}(\mathbf{x}_{n1}^T \otimes \mathbf{I}_2, \dots, \mathbf{x}_{nK}^T \otimes \mathbf{I}_2) \\ \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{b}^x)} &= -\text{diag}(\mathbf{C}_j^y) \mathbf{W} \\ \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{C}^y)} &= -\text{diag}(z_{n1}^T \otimes \mathbf{I}_2, \dots, z_{nP}^T \otimes \mathbf{I}_2) \\ \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{d}^y)} &= -\mathbf{I}_{2P} \end{aligned}$$

and for the RBF mapping we obtain (notation as in [4])

$$\begin{aligned} \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{A}^x)} &= \frac{1}{\sigma^2} \text{diag}(\mathbf{C}_j^y) \mathbf{W} \text{diag}(\Phi'_n) (\tilde{\mathbf{x}}_n \mathbf{1}_M^T - \mathbf{M})^T \text{diag}(\mathbf{x}_{ni}^T \otimes \mathbf{I}_2) \\ \frac{\partial \mathbf{r}_n}{\partial \text{vec}(\mathbf{b}^x)} &= \frac{1}{\sigma^2} \text{diag}(\mathbf{C}_j^y) \mathbf{W} \text{diag}(\Phi'_n) (\tilde{\mathbf{x}}_n \mathbf{1}_M^T - \mathbf{M})^T \end{aligned}$$

and the same formulas for $(\mathbf{C}^y, \mathbf{d}^y)$. The solution for both linear and RBF cases requires nonlinear optimization of E using these gradient equations. As in [4], we found BFGS to be effective and reliable. E has local optima and we initialize BFGS from the solution obtained by the global adaptation method.

BFGS constructs approximate inverse Hessian matrices of order $6(P + K)$, so it will not work if P is large (consider a detailed 2D tongue shape representation of $P = 100 \times 100 = 10^4$ points in 3D). The fact that E decouples over each $(\mathbf{C}_j^y, \mathbf{d}_j^y)$ for fixed $(\mathbf{A}^x, \mathbf{b}^x)$ suggests alternating minimization of E :

1. Fix $(\mathbf{A}^x, \mathbf{b}^x)$ (thus fixing $\mathbf{f}(\mathbf{g}_x(\mathbf{x}_n))$) and minimize E over each $(\mathbf{C}_j^y, \mathbf{d}_j^y)$. Since E is linear over the latter, the unique solution is given by P linear systems of 6×6 .
2. Fix $(\mathbf{C}^y, \mathbf{d}^y)$ and minimize E over $(\mathbf{A}^x, \mathbf{b}^x)$ with BFGS. This requires matrices of order $6K$ only.

A disadvantage of the alternating optimization is that it converges very slowly. We use the full BFGS in our experiments.

Regularizing E . As in [7], we can penalize \mathbf{A}^x and \mathbf{C}^y with large condition numbers by adding the following term to E :

$$\begin{aligned} \lambda \mathcal{C}(\mathbf{A}^x, \mathbf{C}^y) &= \lambda \left(\sum_{i=1}^K C(\mathbf{A}_i^x) + \sum_{i=1}^P C(\mathbf{C}_i^y) \right) \\ \lambda > 0, \quad C(\mathbf{A}) &= \text{tr}(\mathbf{A}^T \mathbf{A}) - D(\det(\mathbf{A}^T \mathbf{A}))^{1/D}. \end{aligned}$$

With very little adaptation data ($N = 10$), this increases robustness to misspecification of landmarks and reduces overfitting.

4. Experiments

Using data from two speakers (male and female) with significantly different shapes, our experiments show the global method gives a reasonable adaptation but stagnates with as few as 5 to 10 contours. The local method keeps reducing the error with more contours and stagnation happens only with many more contours, producing reconstruction results close to retraining the predictive model for the new speaker on abundant data. With very few contours ($N < 10$), the local method needs regularization to reduce its variance, and performs worse than the global one. With more than 50 contours, retraining is the better option. Thus, the user has options to guide data collection and achieve the best result in each application. All these statements hold for various numbers of landmarks K . The local adaptation method takes 3 (linear) and 10 (RBF) minutes of CPU time in a workstation for $N = 50$ and $K = 3$.

Datasets. We use the ultrasound database [3] created at Queen Margaret University and the University of Edinburgh. It contains two speakers (one male, *maaw0*, and one female, *feal0*)

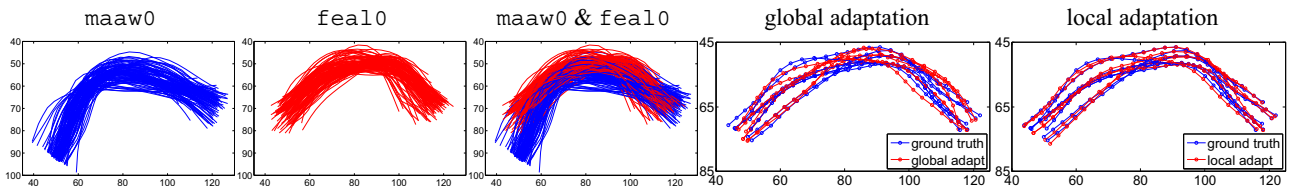


Fig. 2. Left 3 plots: speaker datasets. Right plots: maaw0 aligned to feal0 ($K = 3$). Only a subset of contours plotted to avoid clutter.

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 session, respectively. Each tongue contour contains $P = 24$ points for both speakers. Fig. 2 shows the contour datasets from both speakers, which display significant differences in shape. (Some differences in the tongue root of feal0 are due to its being poorly visible in the ultrasound; this poses an additional challenge for the adaptation algorithms.)

Adaptation task. We adapt a predictive model for maaw0 (learned on 2 236 tongue contours from its first session) to feal0. We use up to 500 contours from feal0 for adaptation/retraining and the remaining 2 409 contours for testing.

Predictive models. As in [4], we use (1) an RBF mapping with $M = 500$ basis functions, width $\sigma = 55$ mm and regularization parameter $\lambda = 10^{-4}$, trained by cross-validation; and (2) a linear mapping, given as a baseline (as it is consistently outperformed by the RBF mapping); we also use it to obtain initial (\mathbf{A}, \mathbf{b}) for the RBF. The K landmarks were chosen optimally from the P contour points as in [3].

Comparison methods. We compare adaptation with the global and the local transformations. We initialize the local method with the parameters of the global one. We also compare with retraining the predictive mapping from scratch on the adaptation data, and with PCA alignment using neither correspondences nor predictive mapping as in [4]; it finds a global (\mathbf{A}, \mathbf{b}) by matching the mean and covariance of the original and the adaptation datasets, each considered as a collection of 2D points (i.e., all the points from all the contours). The optimal baseline is achieved by retraining the predictive model with abundant data. All the error values we quote are RMSE predictive errors E per contour point in mm on the test data.

Results. Figs. 3–4 plot the errors after adaptation/retraining as a function of the number of adaptation contours N , and the number of landmarks K . Using the predictive model of maaw0 directly on feal0 without adaptation would incur an error > 2 mm. With global adaptation, the RBF beats the linear \mathbf{f} consistently by over 0.4 mm as in [4]. With just one contour, the RBF achieves an error of 0.9 mm robustly (note the tight errorbars). However, while the error decreases with N , it stagnates when $N = 10$ far from reaching the optimal value (retraining with abundant data). The performance gap is around 0.3 mm. With local adaptation, both the linear and RBF \mathbf{f} work very well with $N \geq 20$ contours, consistently and significantly outperforming the global adaptation. Surprisingly, the adaptation error of the linear and RBF cases are now comparable. With $N < 7$ to 20 contours, the local adaptation is less stable and has an average error larger than the global one. This is likely an overfitting effect, since the local method has now more parameters. The error decreases with N , stagnating around $N = 50$ but very close to the ground truth (less than 0.1 mm worse). PCA alignment (not shown in the figures) is consistently worse than both

global and local adaptation and with a larger variance even for larger N . Retraining catches up local adaptation for $N \approx 50$ to 90 contours and is essentially useless for $N < 20$.

From fig. 4, as K increases, the predictive error decreases (it is easier to reconstruct the contour given more landmarks), and the adaptation error closely tracks it. The local adaptation consistently beats the global one by 40% across K if using $N > 20$ contours. The articulatory databases use $K = 3$ (MOCHA) and $K = 4$ (Wisconsin XRMB), and the advantage of the local adaptation over the global one is strongest in this region.

Suitable amounts of regularization (linear: $\lambda = 10$, RBF: $\lambda = 10^4$) reduce the error for the local adaptation (RBF in particular) if using very few contours. Global adaptation benefits marginally from regularization.

5. Conclusions

We have introduced a new method for fast adaptation of a tongue model based on local transformations that align each contour point separately. This is more flexible than the global method of [4] and eliminates its estimation bias. The local method asymptotes close to retraining with abundant data, and distinctly outperforms retraining and the global method when the number of adaptation contours is not very small (10 to 50). Thus the user should use the global, local, or retraining methods with less than 10, 10 to 50, and more than 50 contours, respectively.

Acknowledgements. We thank Alan Wrench (Queen Margaret University) for providing us with the speaker data. Work funded by NSF award IIS-0711186.

6. References

- [1] T. Kaburagi and M. Honda, “Determination of sagittal tongue shape from the positions of points on the tongue surface,” *JASA*, 1994.
- [2] P. Badin, E. Baricchin, and A. Vilain, “Determining tongue articulation: From discrete fleshpoints to continuous shadow,” in *Proc. Eurospeech*, 1997, pp. 47–50.
- [3] 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.
- [4] C. Qin and M. Á. Carreira-Perpiñán, “Adaptation of a predictive model of tongue shapes,” in *Proc. Interspeech*, 2009, pp. 772–775.
- [5] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech and Language*, 1995.
- [6] P. C. Woodland, “Speaker adaptation for continuous density HMMs: A review,” in *Adaptation Methods for Speech Recognition, ISCA Tutorial and Research Workshop (ITRW)*, 2001, pp. 11–19.
- [7] C. Qin and M. Á. Carreira-Perpiñán, “Reconstructing the full tongue contour from EMA/X-Ray microbeam,” in *ICASSP*, 2010.
- [8] J. R. Westbury, *X-Ray Microbeam Speech Production Database User’s Handbook Version 1.0*, University of Wisconsin, Jun. 1994.
- [9] A. A. Wrench, “A multi-channel/multi-speaker articulatory database for continuous speech recognition research,” in *Phonus 5*, Institute of Phonetics, University of Saarland, 2000, pp. 1–13.

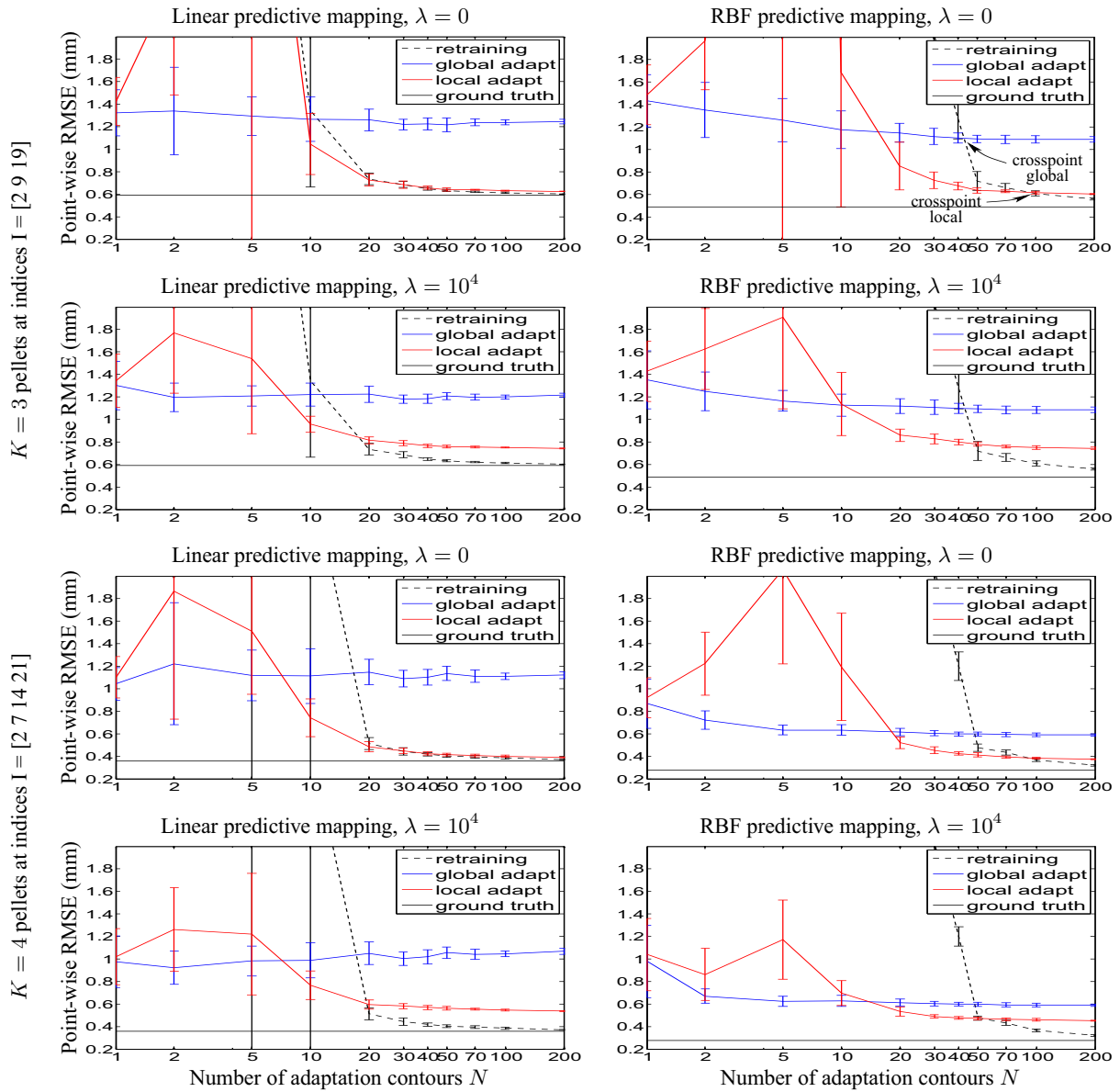


Fig. 3. Predictive error E (as RMSE per contour point in mm) after adaptation as a function of the number of adaptation contours N (for given (K, λ)). Errorbars over 10 random choices of the N adaptation contours. Note the crosspoints with the retraining curve.

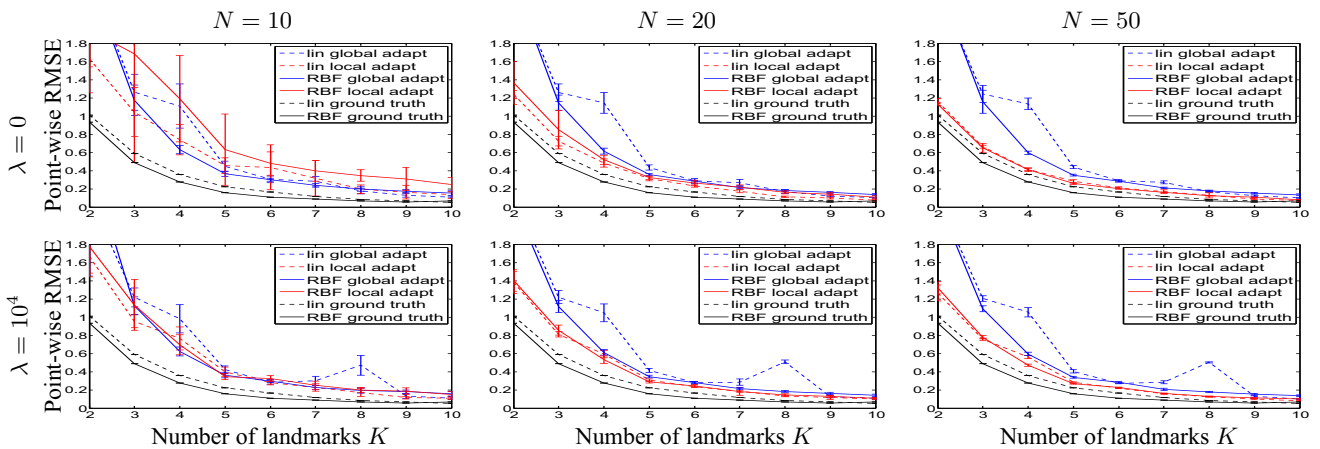


Fig. 4. As fig. 3 but as a function of the number of landmarks K (for given (N, λ)).