Speech-Driven Facial Animation Using A Shared Gaussian Process Latent Variable Model

Salil Deena and Aphrodite Galata

School of Computer Science, University of Manchester
Manchester, M13 9PL, United Kingdom
{deenas,agalata}@cs.man.ac.uk

Abstract. In this work, synthesis of facial animation is done by modelling the mapping between facial motion and speech using the shared Gaussian process latent variable model. Both data are processed separately and subsequently coupled together to yield a shared latent space. This method allows coarticulation to be modelled by having a dynamical model on the latent space. Synthesis of novel animation is done by first obtaining intermediate latent points from the audio data and then using a Gaussian Process mapping to predict the corresponding visual data. Statistical evaluation of generated visual features against ground truth data compares favourably with known methods of speech animation. The generated videos are found to show proper synchronisation with audio and exhibit correct facial dynamics.

1 Introduction

Synthesis of a talking face driven driven by speech audio has many applications from cinema, games, virtual enviroments, online tutoring and in devising better Human Computer Interaction (HCI) systems. Humans perceive speech by interpreting both the sounds produced by speech movements and the visual cues that accompany it. Suppression of one channel at the expense of the other results in ambiguities in speech perception as shown by McGurk and McDonald [1]. Moreover, given the high fine tunement in the way humans perceive speech, slight glitches in an animated character are very conspicuous. Thus, an animated character needs to exhibit plausible speech movements without jerks and with proper synchronisation with the audio.

The pioneering work on facial animation was done by Parke [2] where a 3D model of the face was built using a polygon mesh which was texture-mapped and animation was achieved by interpolating between prototypes or keyframes. Facial animation can also be done using anatomical models of the face constrained by the laws of physics [3], [4]. Whilst 3D models of the face offer a high level of flexibility to the animator, they are very labour intensive and fail to achieve very high levels of realism. Data-driven approaches to facial animation seek to use text or audio data to directly synthesise animation with minimal manual intervention. They can be grouped into text-to-visual synthesis [5] and audio-to-visual synthesis [6], [7], [8], [9], [10], [11]. Rendering for data-driven facial animation can be using 3D graphics-based models of the face [12], [13]; 2D image-based models [6], [7] or through hybrid appearance-based models [8], [9], [10], [11].

The basic unit of spoken language is the phoneme and the corresponding visual unit pertaining to different lip configuations is the viseme. The english language has a total of 41 phonemes [14] and according to the MPEG-4 standard, these are grouped into 14 visemes [15]. Thus, the mapping from phonemes to visemes is many-to-one. Moreover, the visual counterpart of speech is dependent on the context of the speech signal, which means that the same phoneme may produce a different visual output, depending on the phonemes preceeding and following it. This phenomenon is known as coarticualtion.

Our focus is on a data-driven approach to speech animation using machine learning techniques. Because the audio-visual mapping is many-to-one and modelling coarticulation involves taking context into account, regression techniques like artificial neural networks or support vector machines fail to produce appropriate results. Successful techniques that effectively model coarticulation include hidden Markov models [7], Gaussian phonetic models [8] and switching linear dynamical systems [11]. In this work, we make use of the Gaussian Process Latent Variable Model [16] (GPLVM) framework to learn a shared latent space between audio and visual data. The GPLVM is a non-linear dimensionality reduction technique and has recently been applied to multimodal data by learning a shared latent space between human silhouette features and 3D poses [17], [18]. This allows the inferrence of pose from silhouettes. We apply this framework to learn an audio-visual mapping and compare the results with Brand's Voice Puppetry [7].

2 Background and Related Work

We begin by providing some background on the Shared GPLVM (SGPLVM) and refer readers to [16] and [18] for more information.

2.1 The GPLVM

The GPLVM is a probabilistic dimensionality reduction technique that uses Gaussian Processes (GPs) to find a non-linear manifold of some data that seeks to preserve the variance of the data in latent space. The latent space $\mathbf{X} = [x_1, \ldots, x_N]$ is assumed to be related to the mean centered data set, $\mathbf{Y} = [y_1, \ldots, y_N]^T$ through a mapping f that is corrupted by noise:

$$y_n = f(x_n) + \epsilon \, . \tag{1}$$

By placing a GP prior of the mapping f and marginalising it, the likelihood function (2) is obtained, which is a product of D GPs and Φ are the hyperparameters of the covariance function, which is also referred to as the kernel.

$$p(\mathbf{Y}|\mathbf{X},\Phi) = \prod_{i=1}^{D} \frac{1}{(2\pi)^{\frac{N}{2}}|\mathbf{K}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} y_{:,i}^{T} \mathbf{K}^{-1} y_{:,i} \right) . \tag{2}$$

For non-linear mappings, a closed-form solution is not available and the likelihood function is optimised with respect to the latent values $\mathbf{X}$ using conjugate-gradient optimisation. Maximising the marginal likelihood (2) with respect to the latent points and the hyperparameters Φ results in the latent space representation of the GPLVM.

$$\{\hat{\mathbf{X}}, \hat{\Phi}\} = \arg\max_{\mathbf{X},\Phi} P(\mathbf{Y}|\mathbf{X},\Phi) . \tag{3}$$

Back Constraints The GPLVM, being a mapping from the latent space to the data space, ensures that points that are close on the latent space are found close on the data space. However, it does not ensure the opposite, i.e. points that are close in the data space to be mapped close on the latent space. The aim of the back constraints [19] is to enforce this distance preservation. It is done by using an inverse parametric mapping that maps points from the data space to the latent space. The mapping takes the following form:

$$x_i = g(y_i, \mathbf{W}) . \tag{4}$$

Where $\mathbf{W}$ are the parameters of the back-constraint kernel function, which can be any non-linear kernel such as the Radial Basis Function (RBF) or the Multilayer Perceptron (MLP). The optimisation in (3) is then done with respect to the back constraint parameters $\mathbf{W}$:

$$\{\hat{\mathbf{W}}, \hat{\Phi}\} = \arg\max_{\mathbf{W},\Phi} P(\mathbf{Y}|\mathbf{W},\Phi) . \tag{5}$$

Dynamics Wang et al. [20] proposed an extension of the GPLVM which produces a latent space that preserves sequential relationships between points on the data space, on the latent space. This is done by specifying a dynamical function over the sequence in latent space:

$$x_t = h(x_{t-1}) + \epsilon_{dyn} . \tag{6}$$

Where $\epsilon_{dyn} \sim N(\mathbf{0}, \beta_{dyn}^{-1}\mathbf{I})$. This is a first-order dynamics kernel that assumes that each latent point x_t is only conditioned on the preceeding frame, x_{t-1}. By placing a Gaussian Process prior over the function $h(x)$ and marginalising this mapping, a new objective function is obtained. Optimising this objective function results in latent points that preserve temporal relationships in the data. The new objective function is given by (7) with $\hat{\Phi}_{dyn}$ being the hyperparameters of the dynamics kernel.

$$\{\hat{\mathbf{X}}, \hat{\Phi}_Y, \hat{\Phi}_{dyn}\} = \underset{X, \Phi_Y, \Phi_{dyn}}{\arg\max} \, P(\mathbf{Y}|\mathbf{X}, \Phi_Y) P(\mathbf{X}|\Phi_{dyn}) \ . \tag{7}$$

2.2 The SGPLVM

To construct a shared latent space between two sets of variables, $\mathbf{Y}$ and $\mathbf{Z}$ and with a shared latent space $\mathbf{X}$, the likelihood function is taken to the the product of each individual likelihood function, conditioned on a common latent space. This leads to the optimisation of two different sets of hyperparameters for the two kernel functions. The joint likelihood of the two observation spaces is given by:

$$P(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \Phi_s) = P(\mathbf{Y}|\mathbf{X}, \Phi_Y) P(\mathbf{Z}|\mathbf{X}, \Phi_Z) \ . \tag{8}$$

Where $\Phi_S = \{\Phi_Y, \Phi_Z\}$ is a concatenation of the two different sets of hyperparameters.

Back-constraints can similarly be integrated, but with respect to only one data space, because in practice, two separate mappings from two different data spaces, that produce a common latent space cannot be defined. Moreover, a dynamics prior can also be placed on the latent space, just like for the GPLVM.

The Shared GPLVM (SGPLVM) used by [17] and [18] has been used to learn a mapping between pose and silhouette data. However, the mapping from silhouette to pose is one-to-many because silhouettes are ambiguous, especially when the figure is turning around. Ek et al. have addressed ambiguity in [18] by putting a back constraint with respect to poses, which forces a one-to-one relationship between the data and latent space. In [21], ambiguity has been catered for by using a Non-Consolidating Components Analysis (NCCA) whereby a private latent space for each of the observation spaces is learnt in addition to the shared latent space. This allows for the disambiguation of human pose estimation given silhouettes because the variance in both data spaces is retained. Thus, the variance from the space pertaining to the test data can used in the inferrence as a discriminant to resolve ambiguities. The same NCCA model has been used in [22] for mapping human facial expression data, represented by facial landmarks to a robotic face. The ambiguity in this case is with respect to robot poses, with multiple robot poses corresponding to a given facial expression vector.

The SGPLVM can be viewed as a non-linear extension of Canonical Correlation Analysis (CCA). CCA learns a correspondence between two datasets by maximising their joint correlation. Theobald and Wilkinson [10] use CCA to learn an audio-visual mapping. Modelling coarticulation is achieved by appending speech features to the right and to the left of each frame. This, however, leads to a combinatorial explosion and requires large amounts of data to provide adequate generalisation ability. Our approach, based on the SGPLVM framework allows for coarticulation to be modelled in two ways. Placing a back constraint with respect to audio features ensures distance preservation of speech features in the latent space, thus ensuring a smooth transition of latent points for test audio data. Moreover, placing a dynamical model on the latent space constraints the optimisation of latent points to respect the data's dynamics both in the training and synthesis phases.

3 Building an Audio-Visual Corpus

The *Democracy Now!* dataset [11] has been used for our experiments. It features an anchor giving news presentations under roughly the same camera and lighting conditions. We use the dissected video sequences mentioned in [11], featuring the anchor speaking sentences delimited by pauses for breath. However, we perform our own parameterisation of the visual and speech data. The video sequences are converted into frames sampled at the rate of 25 frames per second and cropped around the face region. High quality uncompressed audio has also been made available separately by the authors of [11], that match the dissected video sequences. We now detail how a compact parameterisation is obtained for both visual and audio data. A total of 236 video sequences, corresponding to about 20 minutes of video have been used, together with the corresponding uncompressed audio.

3.1 Visual Data Pre-Processing

Active Appearance Models (AAMs) [23] have been chosen for facial parameterisation because they capture the statistical variation in shape and texture and provide a generative model to extrapolate novel faces as a linear combination of basis shape and texture vectors. They require a training set of annotated prototype face images where the annotations provide for the shape data and the texture data is sampled from the convex hull spanned by these shape vectors (Figure 1a). AAM training first normalises the shape vectors by removing rotations and translations and aligns the the shape with respect to the mean shape by a piecewise affine warp. This requires a triangulation of the landmarks to be performed (Figure 1b). In our case, 31 landmarks have been used. PCA is then applied to the shape and texture data separately and then further on the concatenation of the PCA parameters for shape and texture. Ater training, AAM parameters can be extracted from novel images by projecting the shape and texture data to the corresponding retained eigenvectors of the PCA and then again on the combined eigenvectors. In addition, given a set of AAM parameters, novel frames can be generated by first reconstructing the shape and texture separately and then warping the texture to the shape (Figure 1d). AAMs can also be used for tracking landmarks on novel facial images (Figure 1c). By retaining 95% of the variance in the shape, texture and combined PCA, a 28-dimensional AAM feature vector is obtained.

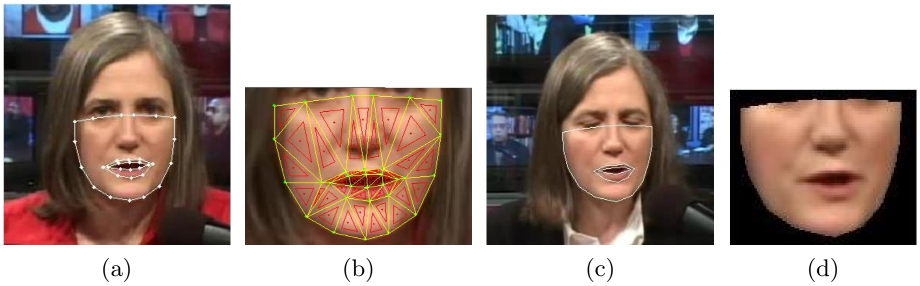

Fig. 1: (a) Annotations marked on a sample face image. (b) Triangulation of the landmarks for warping. (c) Results of AAM search on a new image. (d) Reconstruction of face from a set of AAM parameters

3.2 Speech Parameterisation

Speech needs to be parameterised so as to represent the acoustic variability within and between the different phonemes. This is done by extracting features from the speech signal that help distinguish between the phonemes. The most common speech feature extraction techniques are: Linear Predictive Coding (LPC), Mel-Frequency Cepstral Coefficients (MFCC), Line Spectral Frequencies (LSF) and Formants [14]. MFCCs have been chosen for speech feature extraction of our data because of its robustness to noise and also because we do not need accurate reconstructions provided by linear prediction methods such as LPC and LSF. The MFCC features have been computed at 25 Hz in order to match the sampling rate of the image frames.

4 Audio-Visual Mapping

Taking $\mathbf{Y}$ to be the MFCC feature vector and $\mathbf{Z}$ to be the AAM feature vector, an SGPLVM is learnt between $\mathbf{Y}$ and $\mathbf{Z}$. Whilst in [21] and [22], the data to be synthesised is ambiguous, in our case, we have a many-to-one mapping between audio and visual data. This leads to more flexibility in building the model and the NCCA model of [21] and [22] brings no benefit to our system. However, placing a back-constraint with respect to the audio favours the modelling of coarticulation by constraining similar audio features to be mapped close on the latent space. In addition, it allows the initialisation of latent points from novel audio using the back-constrained mapping. We place an MLP back-constraint with respect to the audio data. An autoregressive dynamics GP is also placed on the latent space. The graphical model of our system is shown in Figure 2. All the parameters of the model are optimised during training. The resulting latent space can be viewed as a non-linear embedding of both audio and visual data that can generate both spaces.

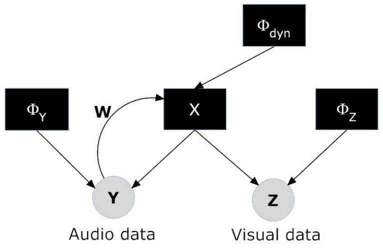

Fig. 2: Graphical model of the shared GPLVM with a back-constraint with respect to the audio and an autoregressive dynamics model on the latent space

4.1 Synthesis

Once the SGPLVM model is trained, audio-visual synthesis proceeds by first extracting MFCC features from test audio. AAM parameters $\hat{\mathbf{Z}}$ can then be synthesised from the test MFCC features $\hat{\mathbf{Y}}$ by first obtaining the corresponding latent points, $\hat{\mathbf{X}}$. The optimisation of latent points is done both with respect to the GP mapping from $\mathbf{X}$ to $\mathbf{Y}$ as well as with respect to the dynamical model, by formulating a joint likelihood given in (9). The likelihood is then optimised using conjugate gradient optimisation to find the most likely latent coordinates for a sequence of audio features.

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}_*} P(\hat{\mathbf{Y}}, \mathbf{X}_* | \mathbf{Y}, \mathbf{X}, \Phi_Y, \Phi_{dyn}) \ . \tag{9}$$

Where $\mathbf{X}_*$ is an initialisation of the latent points. Once $\hat{\mathbf{X}}$ is obtained, $\hat{\mathbf{Z}}$ is obtained from the mean prediction of the GP from $\mathbf{X}$ to $\mathbf{Z}$.

$$\hat{\mathbf{Z}} = k(\hat{\mathbf{X}}, \mathbf{X})^T \mathbf{K}^{-1} \mathbf{Z} \ . \tag{10}$$

4.2 Initialisation of Latent Points

Optimisation of (9) is likely to be highly multimodal with multiple local optima. Thus, a proper initialisation of the latent space, $\mathbf{X}_*$ is required to get good results. We use two initialisation techniques for $\mathbf{X}_*$. In the first method, the latent points obtained from the SGPLVM training are taken to be the states of a hidden Markov Model (HMM) and the training audio features are taken to be the observations. The transition log likelihood is computed as the GP point likelihood between each latent point and every other latent point and the observation log likelihood is obtained by computing the GP point likelihood between the test audio vector and each of the training latent points (states of the HMM). The optimal sequence of latent points $\mathbf{X}_*$ is obtained from the Viterbi algorithm in log space. This is analogous to choosing a set of latent points from the training set that best match the test audio. To speed computation when the number of training data points for the SGPLVM is very high, a subset of the points can be randomly chosen instead, for initialisation.

The second method of initialisation is from the back-constrained mapping from the audio space $\mathbf{Y}$ to the latent space $\mathbf{X}$, which can be obtained as follows:

$$\mathbf{X}_* = g(\hat{\mathbf{Y}}, \mathbf{W}) \ . \tag{11}$$

We shall call the method based on the back-constraint initialisation SGPLVM A and the method based on the HMM initialisation SGPLVM B.

4.3 Experiments

GPLVM training is quite expensive and has a complexity $O(N^3)$, where N is the number of data points. Various sparsification methods have been proposed [24] by making use of a subset of data at a time, called the active set. However, even with sparsification, optimisation of a GPLVM likelihood becomes intractable when the number of data points exceeds a few thousands. This is in contrast to other methods to audio-visual mapping such as HMMs, which can cope with tens of thousands of data points. In our experiments, we have used a repeated random subsampling method for choosing 50 sequences from the 236 audio-visual sequence pairs for training SGPLVM A and SGPLVM B, giving an average of 6000 frames. We fix the dimensionality of the latent space to be 8 as further increasing the dimensionality does not improve the reconstructions of AAM parameters. We then randomly choose 20 sequences for testing, such that the training and testing sets do not overlap. Only the audio features from this test set are used for inferring novel AAM parameters using SGPLVM A and SGPLVM B.

We have used Brand's Voice Puppetry [7] as a benchmark. We train the cross-modal HMMs using the same subsets of audio and visual features as used for the SGPLVM and use the same data for testing. The repeated random subsampling experiment is done ten times for both the SGPLVM and Brand's method.

4.4 Results

We present both quantitative and qualitative results from our experiments. Quantitative results are obtained by finding the Root Mean Square (RMS) error between test AAM feature vectors and ground truth. Figure 3 shows the results obtained accross the ten runs of the experiment. The results show no statistically significant difference between the errors obtained from Brand's method and the SGPLVM. In general, the errors for SGPLVM B are slightly higher than those for SGPLVM A, mostly due to a smoother latent space obtained from the back-constraint initialisation.

We also compare the trajectories of the first mouth landmark parameter reconstructed from the AAM parameters, of the three approaches against ground truth. Figures 5, 6 and 7 shows the results for Brand's Voice Puppetry, SGPLVM A and SGPLVM B respectively. The results for Brand's method show that the trajectories are smoothed out as compared to the SGPLVM approaches. This is

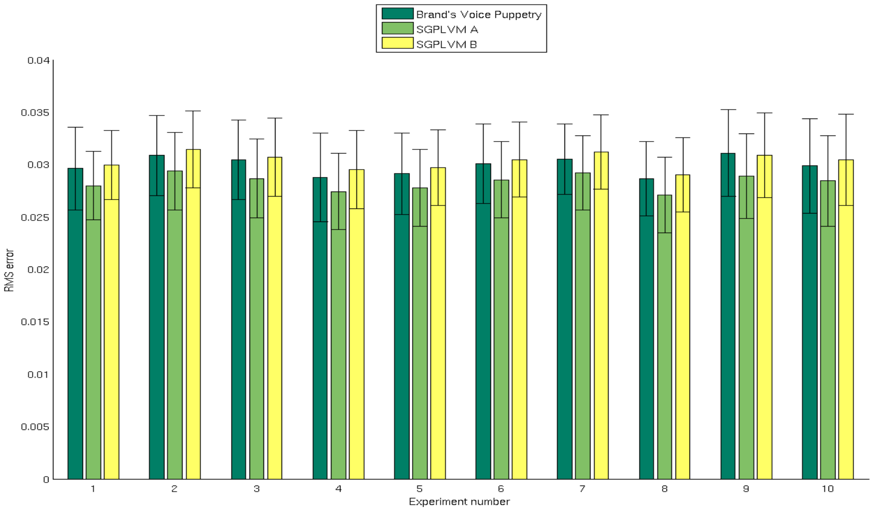

Fig. 3: RMS errors obtained between ground truth AAM feature vectors and 1) SGPLVM A 2) SGPLVM B and 3) Brand's Voice Puppetry. The plots also include the standard deviation of the errors

because Brand's approach involves synthesising AAM parameters from a state sequence, which represents Gaussian clusters, and is thus very approximative. The SGPLVM approaches, on the other hand, bypass this approximation and make use of the full variance of the visual data in synthesis.

Qualitative results are obtained by rendering frames from the AAM parameters in order to visualise the output. The videos show proper lip synchronisation with the audio with smooth lip movements. The results from SGPLVM A appears to be the best whilst SGPLVM B gives proper lip synchronisation but with a few jerks in the animation. The results from Brand's Voice Puppetry are overly smoothed with under articulation. Figure 4 shows ground truth frames as well as frames generated from AAM features obtained from Voice Puppetry and SGPLVM A. The audio contains a sentence which has 12 of the 14 visemes from the MPEG-4 standard [15].

5 Conclusions and Future Work

We have shown how the shared GPLVM can be applied to multimodal data comprising of audio and visual features, in order to synthesise speech animation. The results show that our methods are comparable to Brand's Voice Puppetry in terms of RMS errors of AAM features generated, but with more articulated lip movements.

In future work, a perceptual evaluation of the animation will be carried out where viewers would be asked to asses the realism of the generated videos as well

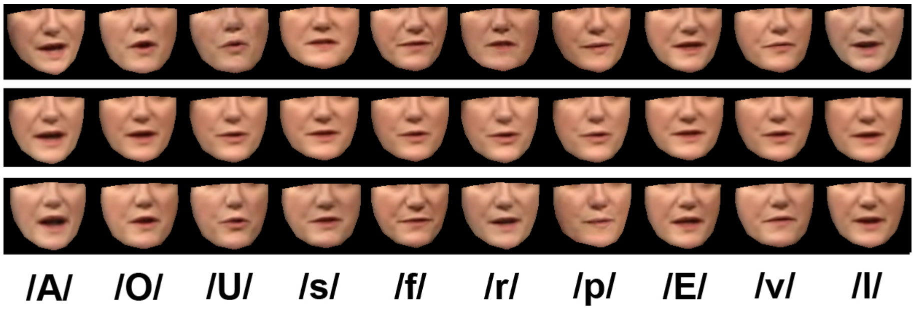

Fig. 4: Reconstructions from AAM features obtained from: ground truth (top row), Voice Puppetry (middle row) and SGPLVM A (bottom row). The audio used for synthesis contains the sentence: "*House of representatives has approved legislation*". The frames correspond to ten different visemes from the test audio sentence

as the intelligibility of the lip movements. Experiments will also be performed with different parameterisations of speech, which favour speaker independence. Moreover we would also investigate delta features in speech to more effectively capture context in speech animation.

Acknowledgements

The authors would like to thank Gwenn Englebienne and Magnus Rattray for their publicly available *Democracy Now!* dataset and to Neil Lawrence for his publicly available Gaussian Process software.

References

1. McGurk, H., MacDonald: Hearing lips and seeing voices. Nature **264** (1976) 746–748
2. Parke, F.I.: A parametric model of human faces. PhD thesis, University of Utah (1974)
3. Terzopoulos, D., Waters, K.: Analysis and synthesis of facial image sequences using physical and anatomical models. IEEE Trans. on Patt. Anal. and Mach. Intel. **15** (1993) 569–579
4. Kähler, K., Haber, J., Yamauchi, H., Seidel, H.P.: Head shop: generating animated head models with anatomical structure. In: SCA '02: Proc. of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation. (2002) 55–63
5. Ezzat, T., Poggio, T.: Miketalk: A talking facial display based on morphing visemes. In: Proc. of the Computer Animation Conference. (1998)
6. Bregler, C., Covell, M., Slaney, M.: Video rewrite: driving visual speech with audio. In: SIGGRAPH '97: Proc. of the 24th ACM annual conference on Computer graphics and interactive techniques. (1997) 353–360

7. Brand, M.: Voice puppetry. In: SIGGRAPH '99: Proc. of the ACM 26th annual conference on Computer graphics and interactive techniques. (1999) 21–28
8. Ezzat, T., Geiger, G., Poggio, T.: Trainable videorealistic speech animation. In: SIGGRAPH '02: Proceedings of the ACM 29th annual conference on Computer graphics and interactive techniques. (2002) 388 – 398
9. Cosker, D., Marshall, D., Rosin, P.L., Hicks, Y.: Speech driven facial animation using a hidden Markov coarticulation model. In: ICPR '04: Proc. of the IEEE 17th International Conference on Pattern Recognition. Volume 1. (2004) 128–131
10. Theobald, B.J., Wilkinson, N.: A real-time speech-driven talking head using active appearance models. In: AVSP'07: Proc. of the International Conference on Auditory-Visual Speech Processing. (2007)
11. Englebienne, G., Cootes, T.F., Rattray, M.: A probabilistic model for generating realistic lip movements from speech. In: NIPS'08: Avances in Neural Information Processing Systems 21. (2008) 401–408
12. Chai, J.X., Xiao, J., Hodgins, J.: Vision-based control of 3D facial animation. In: SCA '03: Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation. (2003) 193–206
13. Cao, Y., Faloutsos, P., Kohler, E., Pighin, F.: Real-time speech motion synthesis from recorded motions. In: SCA '04: Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation. (2004)
14. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall PTR (2001)
15. Tekalp, M., Ostermann, J.: Face and 2-D mesh animation in MPEG-4. Image Communication Journal (1999)
16. Lawrence, N.D.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of Machine Learning Research **6** (2005) 1783–1816
17. Shon, A., Grochow, K., Hertzmann, A., Rao, R.: Learning shared latent structure for image synthesis and robotic imitation. In: NIPS'05: Advances in Neural Information Processing Systems 18. (2005) 1233–1240
18. Ek, C.H., Torr, P.H.S., Lawrence, N.D.: Gaussian process latent variable models for human pose estimation. In: MLMI'07: Proc. of the 4th International Workshop on Machine Learning for Multimodal Interaction. Volume 4892 of Lecture Notes in Computer Science. (2007) 132–143
19. Lawrence, N.D., Quinonero-Candela, J.: Local distance preservation in the GP-LVM through back constraints. In: ICML '06: Proc. of the ACM 23rd International Conference on Machine learning. (2006) 513–520
20. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. In: NIPS'05: Advances in Neural Information Processing Systems 18. (2005)
21. Ek, C.H., Rihan, J., Torr, P.H., Rogez, G., Lawrence, N.D.: Ambiguity modeling in latent spaces. In: MLMI'08: Proc. of the conference on Machine Learning for Multimodal Interaction. (2008)
22. Ek, C.H., Jaeckel, P., Campbell, N., Lawrence, N.D., Melhuish, C.: Shared Gaussian process latent variable models for handling ambiguous facial expressions. In: American Institute of Physics Conference Series. (2009)
23. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: ECCV '98: Proc. of the 5th European Conference on Computer Vision-Volume II. (1998) 484–498
24. Lawrence, N.D.: Learning for larger datasets with the Gaussian process latent variable model. In: AISTATS'07: Proc. of of the Eleventh International Workshop on Artificial Intelligence and Statistics. (2007)

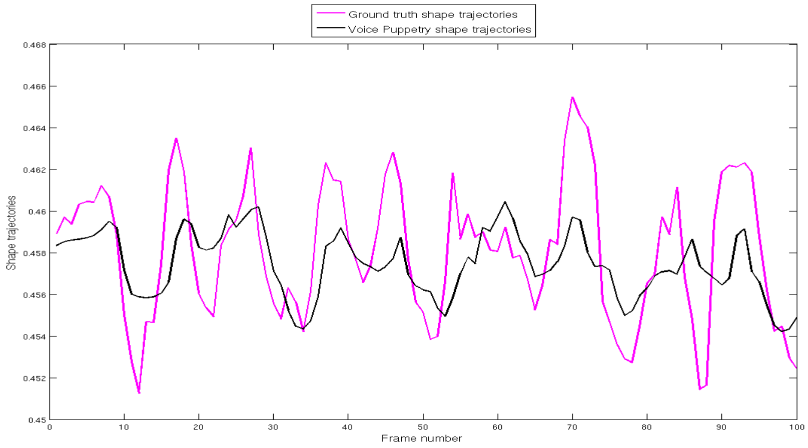

Fig. 5: Shape trajectories obtained from Brand's Voice Puppetry and the corresponding ground truth trajectories

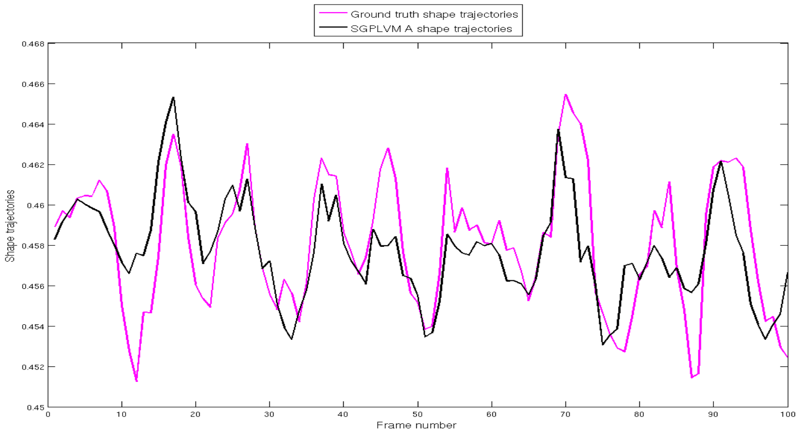

Fig. 6: Shape trajectories obtained from SGPLVM A and the corresponding ground truth trajectories

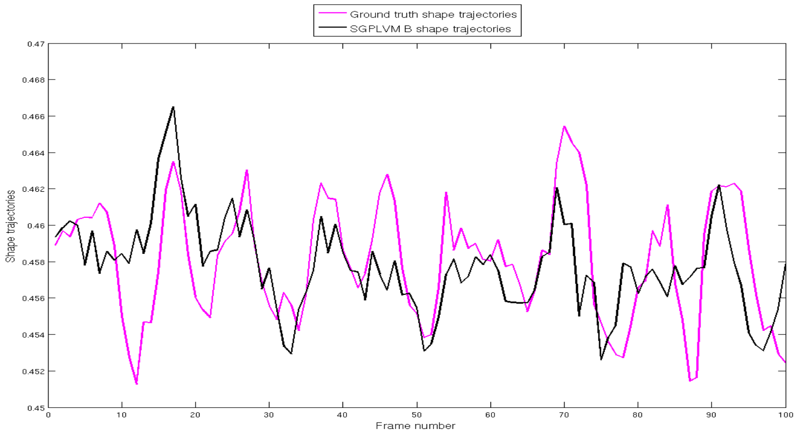

Fig. 7: Shape trajectories obtained from SGPLVM B and the corresponding ground truth trajectories