Visual Speech Synthesis by Modelling Coarticulation Dynamics using a Non-Parametric Switching State-Space Model

Salil Deena, Shaobo Hou and Aphrodite Galata
School of Computer Science
University of Manchester, UK
{sdeena,shou,agalata}@cs.man.ac.uk

ABSTRACT

We present a novel approach to speech-driven facial animation using a non-parametric switching state space model based on Gaussian processes. The model is an extension of the shared Gaussian process dynamical model, augmented with switching states. Audio and visual data from a talking head corpus are jointly modelled using the proposed method. The switching states are found using variable length Markov models trained on labelled phonetic data. We also propose a synthesis technique that takes into account both previous and future phonetic context, thus accounting for coarticulatory effects in speech.

Categories and Subject Descriptors

I.5.4 [**Image Processing and Computer Vision**]: Applications—*Computer vision, Signal processing*

Keywords

speech-driven facial animation, visual speech synthesis, artificial talking head

General Terms

algorithms, theory, experimentation

1. INTRODUCTION

Speech-driven facial animation is a challenging area of research and its aim is to synthesise a talking face uttering a novel speech sequence in a way such that the animation looks natural, life-like and respects the dynamics of the face. Potential applications include animating characters in 3D animated films, animating avatars in games and virtual environments, speech therapy for people with disabilities, as well as devising novel human-computer interaction (HCI) systems. Our focus is on creating speech-driven facial animation using automated methods as opposed to manual techniques. This requires building a generative model of the face that captures the shape and texture variation in a way that facial configurations can be represented by a compact set of parameters. Statistical models of shape and texture variation, known as appearance models [5, 15], are powerful generative models for capturing the distribution of facial expressions. These models can parameterise any face as a linear combination of basis vectors and they can also generate novel faces by reconstructing from the parameter space. It is possible to use speech signal to automatically drive the synthesis of facial animation because the visual and auditory components of speech are highly correlated. However, audio-visual mapping involves several factors. The basic unit of speech is the phoneme and the corresponding visual unit is the viseme. The British English Example Pronunciation Dictionary (BEEP) phone set [21], which represents the phonemes in the English language consists of 45 phonemes including silence, which can be grouped into 13 visemes. Thus, the mapping from phonemes to visemes is many-to-one. This mapping can also be considered many-to-many when noise is introduced, because some phonemes are visually ambiguous and vice-versa. This phenomenon has been referred to as the McGurk effect [19]. The visual appearance of a phoneme also depends on the phonemes that come before and after it, a phenomenon is known as coarticulation. A successful speech-driven animation technique needs to address both these issues. Moreover, the generated speech animation has to be smooth and respect the dynamics of the face, thus making synthesis a challenging task. In this work, a method is presented that jointly models audio and visual parameters using a shared latent space. The different dynamics involved during speech are catered for by augmenting the model with switching states, which are found automatically by training a variable length Markov model (VLMM) [20] on phonetic labels. In the next section, a survey of the state-of-the-art in speech driven facial animation is presented, followed by a description of the specific techniques used in our proposed model.

ICMI-MLMI'2010 November 8-12, 2010, Beijing, China.

2. BACKGROUND

Bregler *et al*[2] were one of the first to adopt an image-based approach, whereby 2D facial images are preprocessed and subsequently reordered and morphed to generate smooth animation. In his proposed method, facial animation was driven by audio, as opposed to being manually created by animators.

Others have also used audio to drive facial animation. Brand [1] trained a hidden Markov model (HMM) on vi-

sual data and remapped the underlying Gaussian clusters to audio data, thus allowing the talking head to be driven by audio. A geodesic interpolation method was used to generate a smooth trajectory of visual parameters from a state sequence of Gaussian clusters. Ezzat *et al*[11] dealt with phonetically aligned speech data and modelled visual parameters as Gaussian clusters indexed by phonemes. A trajectory synthesis method with regularisation was then presented for synthesis. Coarticulation was not explicitly but rather implicitly modelled in the trajectory synthesis technique. A linear dynamical system (LDS) was used in the work of Shiøiler *et al*[17] to jointly model audio and visual parameters. During synthesis, a Kalman filter was used to infer the underlying states from audio data, and a linear mapping was then used to generate the visual parameters from the inferred states.

Cao *et al*[3] explored motion graphs to synthesise novel speech animation using a greedy graph search algorithm, which implicitly models the coarticulatory effects. Englebienne *et al*[10] used a variant of switching linear dynamical systems (SLDS) to model visual data while audio data was modelled using a HMM. Both models were coupled by the phonemes, which represent the states of the HMM as well as the switching states of the SLDS. Their approach only modelled previous phoneme coarticulation, without taking into account future phonemes. In addition, certain information in the speech signal such as intonation and prosody was ignored because the synthesis process was driven by phonemes inferred from the speech signal, and not driven by the speech signal directly. More recently, Deena and Galata [6] tried to address both of these limitations by modeling speech and visual parameters jointly using a shared latent space based on Gaussian process observation and dynamical models. The shared latent space is found by optimising the Gaussian process latent variable model (GPLVM) [16] objective function with respect to two observation spaces instead of one.

A limitation of Deena and Galata's approach is that a single dynamical and observation model is used to account for the whole parameter space, which is not a valid assumption because speech involves multiple types of dynamics [10]. In this paper, we propose a way to circumvent this by augmenting the model with switching states that represent the different types of dynamics. In order to achieve it, we had two challenges to address: how to segment a corpus of data into switching states so as to explicitly model coarticulation and how to synthesise novel animation such there is no discontinuity across switching states. In our proposed method, we address the first problem by using variable length Markov models trained on phonetic data and propose two algorithms to synthesise visual features from audio.

2.1 Switching Shared Gaussian Process Dynamical Model

The switching shared Gaussian process dynamical model (SSGPDM) is a non-parametric switching state-space model proposed by Chen *et al*[4] to account for multiple types of dynamics when jointly modelling silhouettes and 3D pose data. We begin by describing the shared Gaussian process dynamical model (SGPDM), which has been previously used to synthesise speech animation in [6].

2.1.1 SGPDM

In the SGPDM, two observation spaces, $\mathbf{Y} = [\mathbf{y}_1, \dots \mathbf{y}_T]$ and $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_T]$, corresponding to audio and visual parameters respectively, are assumed to be generated from a common latent space $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$ using two non-linear mappings f and g with Gaussian distributed observation noise. Each latent point $\mathbf{x}_t$ is also generated by a dynamical mapping h from the previous latent point $\mathbf{x}_{t-1}$, again corrupted by Gaussian noise.

$$\mathbf{y}_t = f(\mathbf{x}_t) + \epsilon_y \quad \epsilon_y \sim \mathcal{N}(0, \beta_Y^{-1}\mathbf{I}) \tag{1}$$

$$\mathbf{z}_t = g(\mathbf{z}_t) + \epsilon_z \quad \epsilon_z \sim \mathcal{N}(0, \beta_Z^{-1}\mathbf{I}) \tag{2}$$

$$\mathbf{x}_t = h(\mathbf{x}_{t-1}) + \epsilon_{dyn} \quad \epsilon_{dyn} \sim \mathcal{N}(0, \beta_{dyn}^{-1}\mathbf{I}) \tag{3}$$

The model is trained in an unsupervised manner by placing Gaussian process priors over f, g and h and optimising the resulting likelihood function with respect to the Gaussian process (GP) hyperparameters and the latent points $\mathbf{X}$. The likelihood function is given by:

$$P(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \Phi) = P(\mathbf{Y}|\mathbf{X}, \Phi_Y)P(\mathbf{Z}|\mathbf{X}, \Phi_Z)P(\mathbf{X}|\Phi_{dyn}) \tag{4}$$

where $\Phi = [\Phi_Z, \Phi_Y, \Phi_{dyn}]$ is a concatenation of the hyperparameters of the GPs for $\mathbf{Y}$, $\mathbf{Z}$ and the dynamics. The likelihood function for $\mathbf{Y}$ is given by:

$$P(\mathbf{Y}|\mathbf{X}, \Phi_Y) = \frac{|\mathbf{W}_Y|^N}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}} \exp\left(-\frac{1}{2} tr(\mathbf{K}_Y^{-1}\mathbf{Y}\mathbf{W}^2\mathbf{Y}^T) \right) \tag{5}$$

where N is the number of data points and D is the dimensionality of the observation space for $\mathbf{Y}$. The matrix $\mathbf{W}_Y$ is a diagonal matrix of scaling parameters for each output dimension and is used to account for different variances in each output dimension.

The elements of the kernel matrix, $(\mathbf{K}_Y)_{i,j}$ are computed using the kernel function, $K_Y(\mathbf{x}_i, \mathbf{x}_j)$, which in our case is taken to be the radial basis function (RBF):

$$k_Y(\mathbf{x}, \mathbf{x}') = \alpha_Y \exp(-\frac{\gamma_Y}{2}||\mathbf{x}_i - \mathbf{x}_j||^2) \tag{6}$$

The hyperparameters $\Phi_Y = [\alpha_Y, \gamma_Y, \beta_Y]$ are: the variance of the RBF kernel, its inverse width and the variance of the noise term, respectively. The likelihood function, $P(\mathbf{Z}|\mathbf{X}, \Phi_Z)$ is very similar to that of $P(\mathbf{Y}|\mathbf{X}, \Phi_Y)$. The likelihood function for the autoregressive dynamics is given by:

$$p(\mathbf{X}|\Phi_{dyn}) = \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}} \exp\left(-\frac{1}{2} tr(\mathbf{K}_X^{-1}\mathbf{X}_{out}\mathbf{X}_{out}^T) \right) \tag{7}$$

where d denotes the dimensionality of the latent space of $\mathbf{X}$. $\mathbf{X}_{in} = [\mathbf{x}_1, \dots, \mathbf{x}_{N-1}]^T$, $\mathbf{X}_{\mathbf{out}} = [\mathbf{x}_2, \dots, \mathbf{x}_N]$, $\mathbf{K}_X$ is an RBF kernel matrix constructed from $\mathbf{X}_{in}$, and $p(\mathbf{x}_1)$ is an isotropic Gaussian prior.

Optimisation of Eq(4) is done through scaled conjugate gradient optimisation and this gives a regularised latent space that respects the dynamics of the data.

In order to provide a inverse mapping from $\mathbf{Y}$ or $\mathbf{Z}$ so as to ensure distance preservation from a observation space to the latent space, a back-constraint mapping can be used. Eq(8) illustrates a back constraint placed with respect to $\mathbf{Y}$:

$$\mathbf{x}_i = b(\mathbf{y}_i, \theta) \tag{8}$$

b is a parametric mapping such as multilayer perceptron

(MLP), radial basis function (RBF) or kernel-based regression (KBR) with parameters θ. A back-constraint with respect to $\mathbf{Y}$ can also be used to ensure a many-to-one mapping from $\mathbf{Y}$ to $\mathbf{Z}$ by ensuring a one-to-one correspondence between $\mathbf{Y}$ and the latent points $\mathbf{X}$. The graphical model for the SGPDM is shown in Figure 1(a).

2.1.2 Switching SGPDM

The Switching SGPDM is an extension of the SGPDM where multiple SGPDMs are indexed by switching states $\pi = [\pi_1, \ldots, \pi_N]$. The state-space equations then become:

$$\mathbf{y}_t = f_{\pi_t}(\mathbf{x}_t) + \epsilon_{y_{\pi_t}} \quad \epsilon_{y_{\pi_t}} \sim \mathcal{N}(0, \beta^{-1}_{Y_{\pi_t}}\mathbf{I}) \tag{9}$$

$$\mathbf{z}_t = g_{\pi_t}(\mathbf{x}_t) + \epsilon_{z_{\pi_t}} \quad \epsilon_{z_{\pi_t}} \sim \mathcal{N}(0, \beta^{-1}_{Z_{\pi_t}}\mathbf{I}) \tag{10}$$

$if\ \pi_t = \pi_{t-1}$

$$\mathbf{x}_t = h_{\pi_t}(\mathbf{x}_{t-1}) + \epsilon_{dyn_{\pi_t}} \quad \epsilon_{dyn_{\pi_t}} \sim \mathcal{N}(0, \beta^{-1}_{dyn_{\pi_t}}\mathbf{I}) \tag{11}$$

$if\ \pi_t \neq \pi_{t-1}$

$$\mathbf{x}_t \sim \mathcal{N}(\mu_{x_{\pi_t}}, \Sigma_{x_{\pi_t}}) \tag{12}$$

It is to be noted that the state-space equations of the SSGPDM are more similar to the stochastic segment model [7] than the switching linear dynamical system because continuous states are not propagated across switching states. If the switching states are known, then the training of SSGPDM reduces to training a separate SGPDM for each switching state. In this work, switching states are found automatically using variable length Markov models [13, 20, 12]. The graphical model for the SSGPDM is shown in Figure 1(b).

2.2 Variable Length Markov Model

Training higher-order Markov models is often infeasible as the number of higher-order states is exponential in the order of the Markov model, and thus requires a large amount of data to robustly estimate their parameters. It is noted that most higher-order states are only observed sparsely in the training data, if at all; and that higher-order states sometimes do not provide significantly better predictions than their lower-order counterparts. Variable length Markov models (VLMMs) [20] are a powerful extension of nth-order Markov models which take into account these observations and the memory length to vary locally based on the specific realisation of preceding states (i.e., the context).

A VLMM generally contains fewer states than an equivalent nth-order Markov model as higher-order states not supported by the training data are automatically pruned from the model during training, and is therefore much more efficient space-wise. VLMMs have been applied to behaviour modelling [12] to capture higher-order temporal dependencies in some part of the behaviours and lower-order temporal dependencies elsewhere. VLMMs have also been applied to language modelling [13].

In their work, Ron et al. [20] formulated a VLMM as a Probabilistic Finite State Automaton (PFSA). The PFSA is specified by $\mathcal{M} = (Q, \Sigma, \tau, \gamma, s)$, where Σ is a set of tokens representing the finite alphabet of the VLMM and Q is a finite set of model states. Each VLMM state corresponds to a string of tokens of at most length N, representing the memory in the conditional transition distribution of the VLMM. The transition function τ, the output probability function γ and the probability distribution over the initial states, s are given as follows:

$$\tau : Q \times \Sigma \rightarrow Q \quad \gamma : Q \times \Sigma \rightarrow [0, 1] \quad s : Q \rightarrow [0, 1]$$

Training a VLMM involves scanning through the training sequences and building a *prediction suffix tree* such that every contiguous subsequence (i.e., context) w of at most length $N-1$ is represented by a node in the suffix tree. The parent node of a context σw is its suffix w, consisting all but the earliest word σ. The predictive distribution $\hat{P}(\sigma'|w)$ for tokens appearing after the context w is computed from the training data using simple frequency counting. The prediction suffix tree is then pruned by removing any node whose context does not provide significant amount of new information compared to its parent. Given a context σw and its parent w, the amount of information gained by using $\hat{P}(\sigma'|\sigma w)$ for prediction instead of $\hat{P}(\sigma'|w)$ is measured using weighted Kullback-Leibler divergence (KL):

$$\Delta H(\sigma w, w) = \hat{P}(\sigma w) \sum_{\sigma'} \hat{P}(\sigma'|\sigma w) \log \frac{\hat{P}(\sigma'|\sigma w)}{\hat{P}(\sigma'|w)} \tag{13}$$

If $\Delta H(\sigma w, w)$ exceeds a given threshold ϵ, then the longer memory σw is retained, otherwise σw is pruned and the suffix w is used instead. The final stage of training involves converting the suffix tree to a PFSA representing the trained VLMM. A more detailed description of the VLMM training algorithm is given by Ron *et al*[20].

3. PROCESSING AUDIO AND VISUAL DATA

In this section, we describe the data used in our experiments, as well as techniques for parameterising audio and visual data.

3.1 Data

The proposed model is evaluated on the LIPS corpus [22], which consists of 278 high quality video sequences featuring a female subject speaking sentences from the Messiah corpus [21]. The original corpus consists of image frames of size 576×720 sampled at a rate of 50 fps. A frame from the LIPS corpus is shown in Figure 2(a). Audio data in the form of WAV files has also been made available, as well as the phonetic annotation for each frame. In the following section, we describe how to parameterise the audio and visual data for use in our experiments.

3.2 Visual processing

We downsampled the data to 25 fps by skipping every other frame in order to obtain a manageable corpus size. In our approach, we use the Active Appearance Model (AAM) [5] for visual parameterisation.

For training the AAM, we select 184 prototype images by randomly choosing 4 frames from each of the 45 phonemes and 1 silence frame. This has been done automatically using a Matlab script. 56 markup points are then placed around the face, lips and nose in each of the prototype images (Figure 2(b)). An AAM is built on the shapes and images, by first aligning the shapes using Procrustes analysis and then computing a mean shape. The texture sampled from the convex hull of the shape for each prototype is then warped to the mean shape using a piecewise affine warp algorithm. The piecewise affine warp requires that a Delaunay triangulation of the shape vertices to be performed (Figure 2(c)). PCA is then applied to the shape and texture separately and then again to the concatenated shape and texture PCA

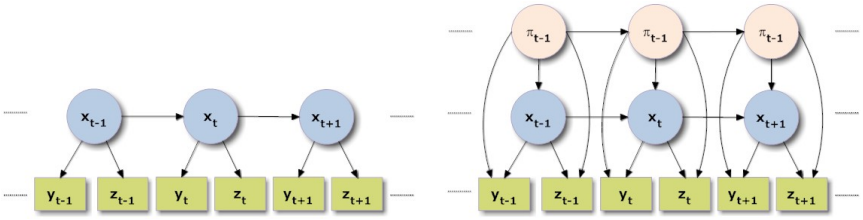


Figure 1: (a) Graphical Model for SGPDM. (b) Graphical Model for Switching SGPDM.

parameters. By retaining 99% of the variance of both the shape, texture and combined parameters, a 33-dimensional vector of AAM parameters is obtained.

The AAM search algorithm [5] is then applied to the whole corpus, after which the shape, texture and combined parameters are projected to the corresponding retained eigenvectors, in order to obtain the AAM parameters for the whole corpus. Reconstruction of an image is done by first reconstructing the combined shape and texture PCA coefficients from its AAM parameters. This is followed by projecting the shape and texture PCA parameters to the data space. Finally, the texture is warped from the mean shape to the reconstructed shape. A frame reconstructed from AAM parameters is shown in Figure 2(d).

It should also be noted that our proposed method requires visual data with minimal pose variation in order to obtain good results. Such variations in each sequence of a corpus can be minimised by normalising its AAM parameters and centering their modes of variation around zero. This is done by first computing the mean of the parameters for each mode of variation in a sequence and then subtracting that mean from the corresponding parameters. Reconstructions of sequences show that all the resulting sequences have the same uniform pose. In the LIPS corpus, approximately 75 sequences have major pose variation while other sequences exhibit minor variations and these are removed using the above method.

3.3 Audio processing

We extracted speech features using Mel-Frequency Cepstral Coefficients (MFCC) [14]. An auditory window of 60ms and a hop window of 40ms is used to obtain overlapping windows and capture dynamical properties in the speech signal. This results in a 25Hz sampling rate, so as to provide alignment with the visual parameters. Theobald *et al*[23] have shown that processing speech at 25Hz produces similar results in speech animation as compared to processing speech at higher sampling rates, whilst increasing the correlation between audio and visual features. 20 MFCC coefficients are used, which is slightly more than the 13 coefficients used in speech recognition. This is done in order to make the audio ($\mathbf{Y}$) and visual ($\mathbf{Z}$) spaces of comparable dimension, so that the likelihoods $P(\mathbf{Y}|\mathbf{X}, \Phi_Y)$ and $P(\mathbf{Z}|\mathbf{X}, \Phi_Z)$ are balanced.

4. PROPOSED METHOD

Our proposed method assumes audio signals can be segmented into commonly occurring fragments of speech, such that each fragments belongs to a primitive unit of speech, and each unit captures a different type of coarticulation dynamics above the level of phonemes. Synthesising speech driven animation therefore involves identifying the speech units in an input sequence and then estimating a sequence of visual parameters that respects the dynamics of the corresponding speech unit at each point in time.

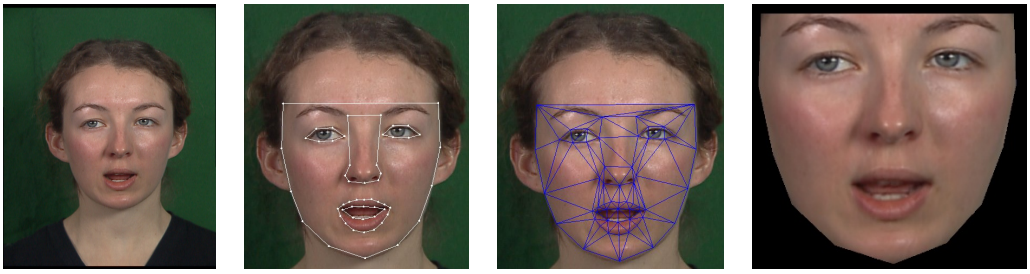
Instead of learning such speech units directly from audio signals, our method relies on the fact that each input sequence has been annotated with the underlying phonemes. These are in turn temporally aligned with the corresponding audio parameters. Such annotations can either be obtained manually or it can be estimated automatically using speech recognition systems such as HTK [27], as is the case for the LIPS corpus [22]. By treating the phonetic labels as tokens in an alphabet, a VLMM can be trained on the phonetic annotations to discover commonly occurring patterns of phonemes and estimate the transition probabilities between these patterns. The dynamics of each pattern or switching state are modelled using a SGPDM model.

Out of of 278 sequences, a random set of 250 sequences was used for training and the remaining 28 sequences were used for testing.

4.1 Training

Given a VLMM trained on phonetic labels, let π denote a VLMM state that correspond to a string of M_π phonemes. For each VLMM state π_t in the training corpus, the corresponding M_{π_t} frame-long sub-sequences of audio and visual features are extracted from the training data. The extracted sub-sequences for a particular VLMM state are modelled together within the same SGPDM as multiple sequences, rather than as a single sequence. Learning Gaussian process dynamical models using multiple sequences has been demonstrated by several researchers [24, 25]. Due to the nature of VLMM, the extracted frames for a particular VLMM state often overlap with other VLMM states, thus creating some redundancy in the model. These redundancies, however, help to model the coarticulation dynamics within each unit represented by a particular VLMM state.

A shared Gaussian process dynamical model (SGPDM) is learnt for each VLMM state. Each SGPDM is intialised by computing a linear subspace with respect to each data space using Principal Components Analysis (PCA), which are then averaged to form a shared latent space. We found in our experiments that a PCA initialisation produces a better model than a latent space initialised with respect to Canonical Correlation Analysis (CCA), primarily because CCA only finds directions of maximal correlation whilst neglecting the variance pertinent to each data space. No back constraints are used since the audio data corresponding to a particular VLMM state are less likely to be visually ambiguous, as opposed to when all the phonemes are modelled using a single SGPDM model [6]. The dynamics models used are an autoregressive Gaussian Process dynamics with the parameters



(a) (b) (c) (d)

Figure 2: (a) A frame from the LIPS corpus. (b) AAM Markup points. (c) Delaunay Triangulation. (d) AAM Reconstruction.

of the RBF and noise terms set such that the signal to noise ratio is: 100 : 1. In contrast to the method proposed by Englebienne *et al*[10], which uses a linear dynamical system to model the dynamics within each phoneme, our choice of non-linear dynamical models is motivated by the fact that phonetic contexts can exhibit highly non-linear behaviour. We use a 6-dimensional latent space because we found it to give decent reconstructions for both audio and visual data, without overfitting.

Figure 3 shows the first 3 dimensions of the latent spaces for SGPDM models corresponding to different VLMM states.

4.2 Synthesis

New speech animation is synthesised by determining the VLMM state π_t for each audio frame in the testing sequence and then inferring the corresponding latent point $\hat{\mathbf{x}}_t$ from the SGPDM associated with π_t. Visual data can then be generated from the latent points. In this work, we assume that the phonetic labels are available for the test sequence, so that the VLMM state for each frame can be found by simply traversing through the PFSA corresponding to the learnt VLMM, using its transition function (see section 2.2). Two methods are proposed for inferring the latent points $\hat{\mathbf{x}}_t$.

4.2.1 *Inferring VLMM States*

For the purpose of synthesis, we wish to associate each audio frame in the test sequence to a VLMM state, such it is not overlapping with frames belonging to any other VLMM states, so that we can synthesise from the corresponding SGPDM model. This is done by traversing the PFSA starting from the start state, and selecting the nodes corresponding to the incoming phoneme and moving to the corresponding VLMM state. This is repeated until all the phonemes in the testing sequence have been processed. Occasionally when traversing the PFSA, a VLMM state can be reached such that is no outgoing node for the next phoneme, in which case, the algorithm moves back to the start state and forgets all previous memory.

Once the last phoneme is reached, a backtracking step is carried out, starting from the last VLMM state π_T, which has a memory length of M_{π_T}. All previous M_{π_T} frames are marked as state π_T. We then move to frame $L - M_{\pi_T} - 1$, where L is the length of the sequence and find the VLMM state, π_t for that frame. Taking M_{π_t} to be the length of VLMM state π_t at frame t, the previous M_{π_t} frames are marked as state π_t. This is repeated until the beginning of the sentence is reached. This gives us a non-overlapping VLMM state sequence for the sentence.

4.2.2 *Optimising Latent Points*

In this section, we describe two different ways for estimating latent points from the inferred VLMM states and the test audio data. For both methods, an initial estimate of the latent points, $\mathbf{x}_*$, is found using a nearest-neighbour comparison of the test audio features, against the training audio features in the current SGPDM model.

Algorithm 1 describes a sequential optimisation algorithm which assumes the entire test sequence is available from the start, so that synthesis can be performed in an offline manner. Here, the latent point for each frame is locally optimised based on the current SGPDM model, which depends on the current VLMM state. If the current state is occupied by only the current frame, a GPLVM point optimisation is carried out [9, 8], whilst on the other hand, a GPLVM sequence optimisation is carried out [9, 8, 6].

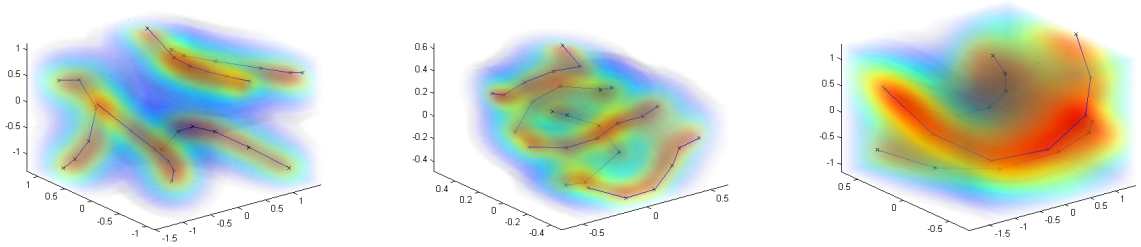
Algorithm 1 Sequential optimisation of latent points.

Let π_t be the VLMM state of the t^{th} frame, $\hat{\mathbf{y}}_t$ be the t^{th} audio frame and T be the length of the sequence
$t \leftarrow 1$
while t <= T **do**
  if $\pi_{t+1} \neq \pi_t$ **then**
    Eq(1):
    $\hat{\mathbf{x}}_t \leftarrow \arg\max_{\mathbf{x}_*} p(\hat{\mathbf{y}}_t|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}_{\pi_t}, \Phi_{Y_{\pi_t}})$
  else
    $t_i \leftarrow t$
    $\pi_s \leftarrow \pi_t$
    while $\pi_{t+1} \neq \pi_t$ **do**
      $t \leftarrow t + 1$
    end while
    $t_j \leftarrow t$
    Eq(2):
    $\hat{\mathbf{x}}_{t_i:t_j} \leftarrow \arg\max_{\mathbf{x}_*} p(\hat{\mathbf{y}}_{t_i:t_j}|\mathbf{x}_*, \mathbf{Y}_{\pi_s}, \mathbf{X}_{\pi_s}, \Phi_{Y_{\pi_s}}, \Phi_{dyn_{\pi_s}})$
  end if
end while

If the assumption is that the data is coming in an online fashion, then Algorithm 2 allows the prediction of the next frame from the previous. This is done using a GPLVM point optimisation for only the first frame in a given VLMM state and then, using the dynamical GP to predict the next frames for that state. This is repeated for all audio frames.

4.2.3 *Smoothness constraint*

Once the latent points are obtained, the visual features, $\hat{\mathbf{Z}} = [\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_T]$ can be obtained from the mean prediction of the visual observation GP, corresponding to the VLMM



(a) (b) (c)

Figure 3: (a) SGPDM for VLMM state y y uw uw uw. (b) SGPDM for VLMM state m aa aa aa aa. (c) SGPDM for VLMM state r ey ey ey ey ey.

Algorithm 2 Sequential prediction of latent points.

Let π_t be the state of the t^{th} frame, $\hat{\mathbf{y}}_t$ be the t^{th} audio frame and T be the length of the sequence
$t \leftarrow 1$
$\hat{\mathbf{x}}_t \leftarrow \arg\max_{\mathbf{x}_*} p(\hat{\mathbf{y}}_t|\mathbf{x}_*, \mathbf{Y}_{\pi_t}, \mathbf{X}_{\pi_t}, \Phi_{Y_{\pi_t}})$
while t <= T - 1 **do**
  if $\pi_{t+1} \neq \pi_t$ **then**
    Eq(3):
    $\hat{\mathbf{x}}_{t+1} \leftarrow \arg\max_{\mathbf{x}_*} p(\hat{\mathbf{y}}_{t+1}|\mathbf{x}_*, \mathbf{Y}_{\pi_{t+1}}, \mathbf{X}_{\pi_{t+1}}, \Phi_{Y_{\pi_{t+1}}})$
  else
    $\hat{\mathbf{x}}_{t+1} \leftarrow h_{\pi_t}(\hat{\mathbf{x}}_t)$
  end if
  $t \leftarrow t + 1$
end while

state at frame t, as depicted by Eq(14).

$$\hat{\mathbf{z}}_t = k(\hat{\mathbf{x}}_t, \mathbf{X}_{\pi_t})^T (\mathbf{K}_{\mathbf{z}})^{-1}_{\pi_t} \mathbf{Z}_{\pi_t} \tag{14}$$

The algorithms above consider the switching states to be independent. We found that by taking these states to be independent, discontinuities arise from when there is a transition from one state to the next. To deal with this, we introduce an additional term in the likelihood that is to be optimised to find latent points. First, the visual feature of the previous frame, $\hat{\mathbf{z}}_{t-1}$ is synthesised according to Eq(14). Then, the term $p(\hat{\mathbf{z}}_{t-1}|\hat{\mathbf{x}}_{t-1}, \mathbf{Z}_{\pi_{t-1}}, \mathbf{X}_{\pi_{t-1}}, \Phi_{Z_{\pi_{t-1}}})$ is multiplied to the term in the likelihood function of **Eq(1)** and **Eq(2)** in Algorithm 1 as well as in **Eq(3)** in Algorithm 2. This formulates a joint probability distribution between the test audio data and the previous visual features, which when optimised constrains the visual features synthesised from a given SGPDM model of VLMM state π_t, to be similar to the visual features belonging to the SGPDM model of the previous VLMM state, π_{t-1}, thus ensuring continuity across states. To further smooth the synthesised visual features and minimise jumps, a low-pass filter is applied to the data, using the Matlab function `interp`.

4.2.4 *Modelling coarticulation*

The method presented in this paper explicitly models coarticulation. The dynamics of phonetic contexts are captured by modelling each VLMM state using a SGPDM model. When synthesising the visual parameters, previous phonetic context is taken into account by the smoothness constraint mentioned in section 4.2.3. This accounts for *carryover coarticulation* [18]. In addition, each VLMM state encapsulates the context of phonemes. The sequential optimisation algorithm takes into account future phonemes that occur within a particular VLMM state, thus accounting for *anticipatory coarticulation* [18]. Hence, by using phonetic VLMM states as switches for the SSGPDM model, together with the sequential optimisation algorithm, we explicitly model both forward and backward coarticulation in synthesis of visual speech.

5. EVALUATION

The AAM features for the 28 test sequences are synthesised based on our proposed method. We compute Average Mean Squared Error (AMSE) as well as Average Correlation Coefficient (ACC) [26] between ground truth and synthesised AAM features for each of the test sequences. We compute the results using both the sequential optimisation and prediction algorithms. The results are also compared against the Voice Puppetry method [1] as well as a SSGPDM with phonemes as switching states, which does not explicitly model phonetic context. The SGPDM model with no switching states [6] is not directly comparable because its training is intractable on the 250 sequences, due to the $O(N^3)$ complexity of SGPDM training, where N is the number of data points in the model.

Table 1 shows the AMSE and the ACC between the ground truth and synthesised AAM features, obtained from the different techniques. The results show that the SSGPDM model with sequential optimisation gives the best results, thus hinting that a framework that explicitly models both *carryover* and *anticipatory* coarticulation is the most effective. Moreover, a joint model of audio and visual data is a more intuitive way of modelling speech animation, as opposed to learning Gaussian clusters on visual data and remapping them to audio data as in Brand's Voice Puppetry [1]. The quantitative results also support this.

Figure 4 shows frames synthesised by our proposed method for a test sequence, using sequential optimisation. The corresponding phonetic labels are shown underneath each frame.

6. CONCLUSIONS AND FUTURE WORK

We have presented the use of a non-parametric switching state-space model based on Gaussian process prediction and observation functions, to visual speech synthesis. The switching states are found using the variable length Markov model on phonetic data. In addition, we have devised two synthesis algorithms for generating visual features from au-

Method	Switching State	Synthesis method	AMSE	ACC
SSGPDM	Phoneme VLMM	Sequential Optimisation	0.04383 ± 0.04104	0.4438 ± 0.3838
SSGPDM	Phoneme VLMM	Sequential Prediction	0.05238 ± 0.04394	0.3410 ± 0.3278
SSGPDM	Phoneme	Sequential Optimisation	0.07252 ± 0.06167	0.1451 ± 0.2688
SSGPDM	Phoneme	Sequential Prediction	0.06594 ± 0.05601	0.1927 ± 0.3042
Brand's Voice Puppetry	N/A	Geodesic Interpolation [1]	0.05978 ± 0.04871	0.2631 ± 0.2757

Table 1: Quantitative evaluation results.

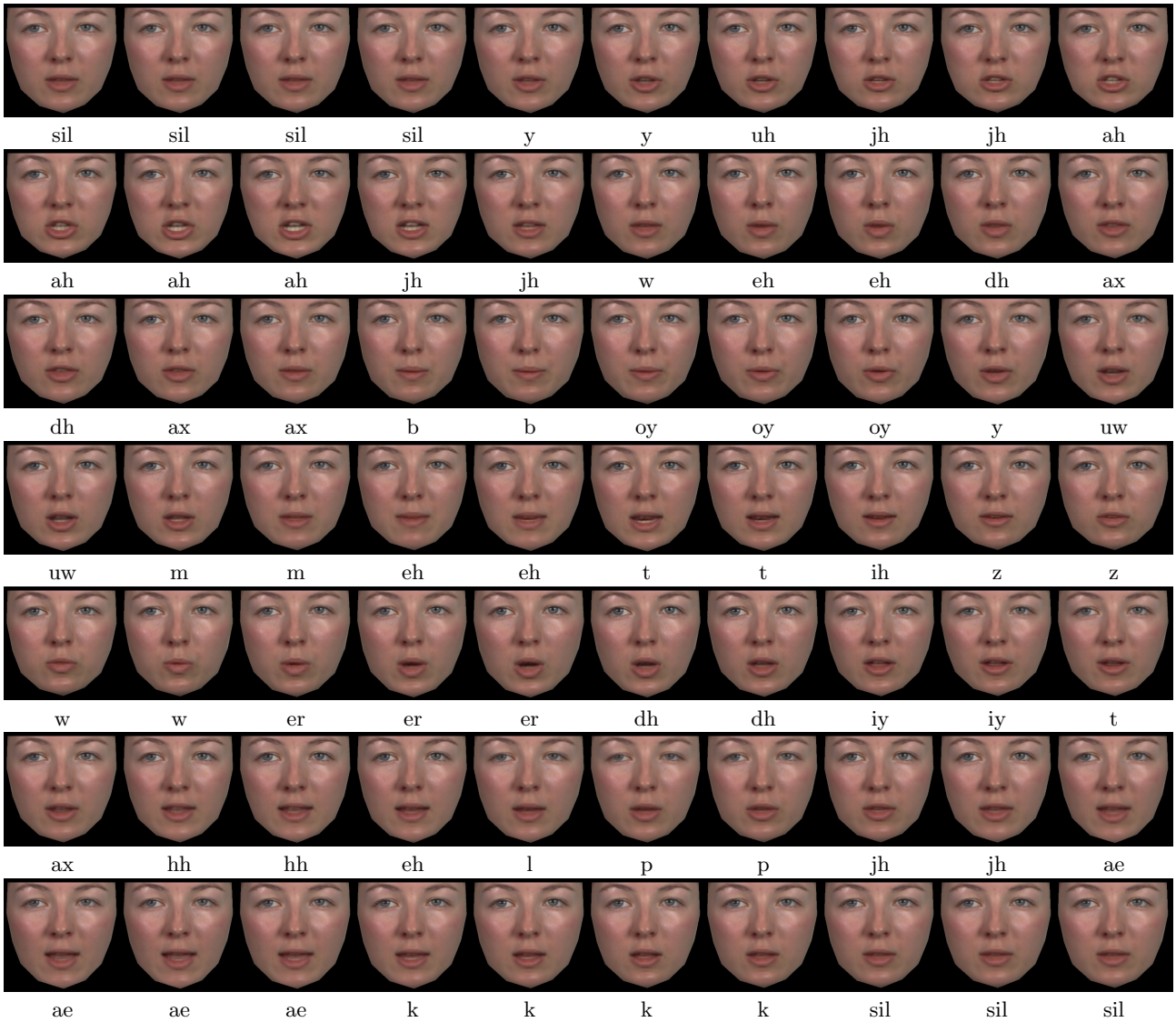


Figure 4: Synthesis results using the SSGPDM method with phoneme VLMM switching states and sequential optimisation method. The phonemes correspond to the sentence: "You judge whether the boy you met is worthy to help Jack".

dio signals that ensure smooth transitions across switching states. Quantitative experiments show that the switching shared Gaussian process dynamical model, which explicitly models forward and backward coarticulation gives the best results.

One limitation of our proposed method is that we need to have phonetic labels for both the training and test data. A direction for future work will be to investigate techniques for training and synthesis using unlabelled phonetic data. We also plan to conduct qualitative evaluations with human subjects to assess realism and intelligibility of animations generated by the proposed technique.

Acknowledgements

The authors would like to thank Barry-John Theobald and others for making the LIPS dataset available. The SSGPDM codes are built on Neil Lawrence's Matlab Gaussian Process toolboxes.

7. REFERENCES

[1] Brand, M. Voice puppetry. In *SIGGRAPH '99: Proc. of the ACM Conference on Computer Graphics and Interactive Techniques* (1999).

[2] Bregler, C., Covell, M., and Slaney, M. Video rewrite: driving visual speech with audio. In *SIGGRAPH '97: Proc. of the ACM Conference on Computer Graphics and Interactive Techniques* (1997).

[3] Cao, Y., Faloutsos, P., Kohler, E., and Pighin, F. Real-time speech motion synthesis from recorded motions. In *SCA '04: Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004).

[4] Chen, J., Kim, M., Wang, Y., and Ji, Q. Switching Gaussian process dynamic models for simultaneous composite motion tracking and recognition. In *CVPR'09: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 2655–2662.

[5] Cootes, T. F., Edwards, G. J., and Taylor, C. J. Active Appearance Models. *IEEE PAMI 23*, 6 (2001), 681–685.

[6] Deena, S., and Galata, A. Speech-driven facial animation using a shared Gaussian process latent variable model. In *ISVC'09: Proc. of the 5th International Symposium on Visual Computing* (2009).

[7] Digalakis, O., Ostendorf, M., and Digalakis, V. The stochastic segment model for continuous speech recognition. In *Proc. of the Asilomar Conference on Signals, Systems and Computers* (1991), pp. 964–968.

[8] Ek, C. H., Rihan, J., Torr, P. H. S., Rogez, G., and Lawrence, N. D. Ambiguity modeling in latent spaces. In *MLMI'08: Proc. 5th International Workshop on Machine Learning for Multimodal Interaction* (2008).

[9] Ek, C. H., Torr, P. H. S., and 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* (2007).

[10] Englebienne, G., Cootes, T. F., and Rattray, M. A probabilistic model for generating realistic lip movements from speech. In *NIPS'07: Advances in Neural Information Processing Systems* (2007).

[11] Ezzat, T., Geiger, G., and Poggio, T. Trainable videorealistic speech animation. In *SIGGRAPH '02: Proc. of the ACM conference on Computer graphics and interactive techniques* (2002).

[12] Galata, A., Johnson, N., and Hogg, D. Learning variable length Markov models of behaviour. *Computer Vision and Image Understanding 81*, 3 (2001), 398–413.

[13] Guyon, I., and Pereira, F. Design of a linguistic postprocessor using variable memory length Markov models. In *ICDAR'95: Proc. of IEEE International Conference on Document Analysis and Recognition* (1995), pp. 454–457.

[14] Huang, X., Acero, A., and Hon, H.-W. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, 2001.

[15] Jones, M. J., and Poggio, T. Multidimensional morphable models. In *ICCV '98: Proc. of IEEE the International Conference on Computer Vision* (1998).

[16] Lawrence, N. D. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *JMLR 6* (2005), 1783–1816.

[17] Lehn-Schiøler, T., Hansen, L. K., and Larsen, J. Mapping from speech to images using continuous state space models. In *MLMI'04: Proc. of 1st International Workshop on Machine Learning for Multimodal Interaction* (2005).

[18] Lindblom, B. *Speech Production and Speech Modeling*. Kluwer Academic, Dordrecht, 1990, ch. Explaining phonetic variation: A sketch of the H&H theory, pp. 403–439.

[19] McGurk, H., and MacDonald. Hearing lips and seeing voices. *Nature 264* (1976), 746–748.

[20] Ron, D., Singer, Y., and Tishby, N. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning 25* (1996), 117–149.

[21] Theobald, B.-J. *Visual speech synthesis using shape and appearance models*. PhD thesis, School of Information Systems, University of East Anglia, 2003.

[22] Theobald, B.-J., Fagel, S., Bailly, G., and Elisei, F. LIPS2008: Visual speech synthesis challenge. In *Proc. of Interspeech* (2008).

[23] Theobald, B.-J., and 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).

[24] Urtasun, R., Fleet, D. J., and Fua, P. 3D people tracking with Gaussian process dynamical models. In *CVPR'06: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (2006), pp. 238–245.

[25] Wang, J. M., Fleet, D. J., and Hertzmann, A. Gaussian process dynamical models for human motion. *IEEE PAMI 30*, 2 (2008), 283–298.

[26] Xie, L., and Liu, Z.-Q. A coupled HMM approach to video-realistic speech animation. *Pattern Recognition 40*, 8 (2007), 2325–2340.

[27] Young, S. The HTK Hidden Markov Model Toolkit: Design and philosophy. Tech. rep., University of Cambridge, Department of Engineering, 1993.