A Nonlinear Viseme Model for Triphone-Based Speech Synthesis

Robert Bargmann
MPI Informatik
Saarbrücken, Germany

Volker Blanz
Fachgruppe Medieninformatik
Universität Siegen, Germany

Hans-Peter Seidel
MPI Informatik
Saarbrücken, Germany

Abstract

This paper presents a new learning-based approach to speech synthesis that achieves mouth movements with rich and expressive articulation for novel audio input. From a database of 3D triphone motions, our algorithm picks the optimal sequences based on a triphone similarity measure, and concatenates them to create new utterances that include coarticulation effects. By using a Locally Linear Embedding (LLE) representation of feature points on 3D scans, we propose a model that defines a measure of similarity among visemes, and a system of viseme categories, which are used to define triphone substitution rules and a cost function. Moreover, we compute deformation vectors for several facial expressions, allowing expression variation to be smoothly added to the speech animation.

In an entirely data-driven approach, our automated procedure for defining viseme categories closely reproduces the groups of related visemes that are defined in the phonetics literature. The structure of our selection method is intrinsic to the nature of speech and generates a substitution table that can be reused as-is in different speech animation systems.

1. Introduction

High realism of speech, as seen in animated movies, is still obtained by a direct mapping of the articulation taken from a real actor to a CG model. The great challenge for automatic speech generation is to reach comparable results synthetically, given only a text or an audio file. This aspect of speech animation has been addressed in many different ways, and in this work, we follow a data-driven approach.

Data driven techniques exploit real motion samples, which hold all the necessary information for natural behavior, and strive to apply appropriate data representations and generalization rules to carry as much of that information as possible from data acquisition to the final animation rendering process.

We investigate the statistical distribution of mouth configurations using a *Locally Linear Embedding* (LLE) data analysis for nonlinear dimension reduction. The representation of data in the LLE-space reflects an intuitive distribution of the data on which we build a *nonlinear viseme model*. *Visemes* are the basic elements of visual speech and form the visual counterparts of phonemes. Our viseme model empirically reproduces some standard viseme taxonomies from the *articulatory phonetics* literature.

In natural speech, however, each viseme is influenced significantly by the previous and following visemes, which is known as *coarticulation* effect. Due to coarticulation, a direct data-driven approach would need many viseme samples with different contexts, and choose the most appropriate one for the synthesis of a new articulation. To get better quality, the size of the database has to be increased and quickly becomes problematically large. In this paper, we directly address that problem by proposing a new substitution rule that allows the selection of appropriate visemes even when they were not originally found in exactly the same context as in the new sequence. The selection framework is based on a similarity measure that compares the natural behavior of visemes and allows us to build a hierarchy which will determine valid substitutions.

For generating of new sentences, we use a triphone-based approach [6]. *Triphones* are short pieces of motion sequence that span three phonemes, so each viseme is stored with its context and therefore captures all of the coarticulation effect caused by the direct neighbors. Our similarity measure is easily extended from visemes to triphones, and we can thus find the best overlapping triphone sequences in our database that match any new sentences that needs to be synthesized. Unlike earlier work on triphone methods [6], our work is based on dense 3D surface scans, which makes it more versatile than image-based techniques.

The new contributions of this paper are: (a) triphone-based 3D speech animation, (b) a nonlinear statistical analysis of viseme distributions using LLE, (c) a similarity measure for visemes, (d) a criterion for substitution of triphones for animation synthesis, (e) a rigorous empirical basis for the classification schemes found in the literature on *articulatory phonetics*, (f) a substitution table that can be reused as-is in different speech animation systems.

Related Work Facial animation is facing two different challenges: the first challenge is to produce realistic face

shapes in every single frame of the animation, and the second challenge is to create a dynamically realistic motion over time.

In contrast to physical models which generate deformations of the skin by simulating the facial tissue and muscles [1, 22, 24], data-driven methods observe and generalize the appearance of real faces, based on a statistical model. This model may be based on marker point positions [7, 9, 15, 18], 3D scans [3, 14, 25, 28, 30] or images [6, 12].

All of these approaches are facing the problem of defining how the parameters of the model vary over time. For speech synthesis, this involves the problem of coarticulation. *Cohen and Massaro*[10] define dominance functions of phonemes that control the interaction between subsequent phonemes as applied to muscle-based systems [1, 16]. *Pelachaud et al.*[20] assign to each phoneme a deformability and context sensitivity value, and derive rules for their mutual influence.

Unlike rule-based methods, statistical techniques derive general properties of motion trajectories from training data. *Voice Puppetry*[5] uses a Hidden Markov Model to learn the dynamics of speech from audio, and transfer this information to a face model. Another learning-based approach was proposed by *Ezzat et al.*[12] and reused by *Kim & Ko*[15], which uses regularization techniques to compute smooth curves for the model parameters over time. In this model, coarticulation is due to the smoothness of the curve and a statistical representation of the variance of each viseme.

Instead of synthesizing motion entirely, *Video Rewrite*[6] stores triphone motions in a database, and stitches them together to produce new utterances. The triphones capture coarticulation in a natural way. If a desired triphone sample is not available, substitutes are selected instead which belong to the same viseme class[19]. In our work, which is closely related to Video Rewrite, we do not make use of *a priori* assumptions of phoneme classes, but deduce a phoneme similarity measure. This quantitative similarity measure relaxes the selection rule of viseme grouping and offers further substitution options. Similarly to *Video Rewrite*, the optimal triphone sequence for the synthetic animation is found by minimizing an error function that takes both, viseme similarity and smoothness of the animation into account. These criteria, however, differ in our work from the ones proposed in *Video Rewrite*.

In data-driven methods, *visyllables*[17] have been proposed as an alternative to triphones with advantages in terms of storage requirements. Another concept of motion representation is based on *Animes*[7, 8, 15], which contain motion sequences of phonemes. These are stored in an *Anime Graph* that captures the context dependencies of individual instances of phonemes. By selecting Animes with an appropriate context from the graph, the algorithm synthesizes animation with coarticulation effects. Expression and speech are separated by an Independent Component Analysis [8]. Unlike *Anime Graph* our approach considers viseme substitutions with motion segments with which they were not associated to it in the first place and thus increases the number of valid candidates. *Wampler et al.*[27] also use an Anime-based graph algorithm, but rely on a bilinear model for separating expression and speech. Another graph-based method for coarticulation that uses Viterbi search in animation has been presented by *Ma et al.*[18].

Sifakis et al.[23] propose the concept of *physemes* to describe the time dependency of muscle activations over extended intervals, and use a physics-based simulation to generate speech animation with coarticulation.

In our work, we use Locally Linear Embedding (LLE) [21] for a statistical analysis of the distribution of mouth configurations in high-dimensional face space. This method estimates a low-dimensional, nonlinear manifold from a set of data points. In our system, LLE allows us to derive a highly specific criterion for viseme similarity that dictates appropriate triphone substitutions. LLE has been used previously by *Wang et al.*[28] as a representation that allows the separation of expression style from expression content with a bilinear model. Using a closely related Isomap method, *Deng and Neumann*[11] present a data-driven approach to speech animation where users can edit facial expressions in sequences.

System Overview Our system is divided into two main parts: the setup part which is performed only once, and the animation synthesis part. The **setup** starts with the *acquisition* and the *registration* of the data (Sec. 2). In the LLE space (Sec. 3), we study the behavior of the different visemes by deriving a *nonlinear model* (Sec. 4). This model allows the generation of a *substitution graph* among the viseme classes (Sec. 5), which is the central tool for the selection of motion sequences from our database to create animations. In the **synthesis** step (Sec. 6), a novel audio file is decomposed into a sequence of phonemes. For each phoneme triplet, candidate triphones are selected from the database each associated with a *substitution cost* based on the substitution graph. The optimal selection of samples from these lists minimizes the combination of a *concatenation cost* with the substitution costs and provides the most natural articulation.

Throughout the paper, we will use the following notation: phonemes (audio) are written with slashes, *e.g.* /AH/, and visemes (visual) are written with vertical bars, *e.g.* |AH|. Triphones, a sequence of three consecutive phonemes or visemes, are written with angle brackets, *e.g.* ⟨AH,R,V⟩.

2. Capturing 3D Motion Data

Our corpus of a 3D talking face was recorded with a structured-light dynamic 3D scanner at a frame rate of 40 3D-scans per second and contains a total of about 17'000

frames (about 7mins without long silences) with a video resolution of 480x640 pixels. The data is preprocessed and then brought into correspondence based on an optical flow algorithm similar to *Blanz & Vetter* [4]. As a result, each 3D frame is represented as a shape vector $\mathbf{S}$ in a morphable model that consists of a collection of all x, y and z coordinates of all n points of the model, combined into a single $3n$-dimensional vector. In a further step we perform a Principal Component Analysis (PCA) on the set of every 100th shape vector. Using all vectors is computationally expensive, and the data is likely to be highly redundant. The PCA is used for data compression of all frames, allowing us to store only a set of model coefficients for each frame. Like in [2], the PCA is computed only on the region around the mouth in order to cancel the influence of eye movements. In our system, we use the 50 coefficients of the first, most relevant principal components.

Expression Vectors The data registration stores deformation vectors relative to a selected reference face. We are thus able to determine typical deformations to several expressions like fear, happiness or sadness (see Fig. 1), which can then easily be added to the generated animation of our talking head. In the final animation process, we use only a single texture of the face for all the generated frames, the wrinkle effect is thus not emphasized by the texture and results only from geometric deformations. Nevertheless, expression reproduction is not our main focus in this paper.

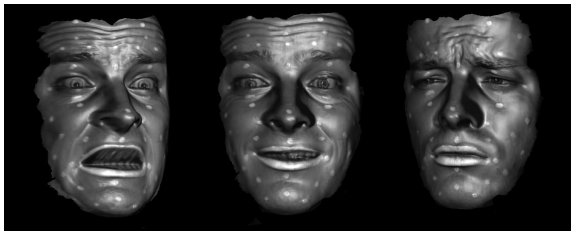

Figure 1. The recorded expression vectors for fear, happiness and sadness.

3. Viseme Representation

For the exploration and representation of motion data, we investigate two different statistical methods: PCA and LLE. In the reduced space of the PCA, the observation of the articulation curves for each viseme cluster does not reveal any specific pattern or structure (Fig. 2) and we therefore turn to an alternative representation using LLE.

Locally Linear Embedding We perform a Locally Linear Embedding (LLE) [21] to get a low dimensional representation of the data manifold. This reduction separates the most important deformations and gives a more intuitive representation of the actual behavior of the mouth, making the visemes easier to analyze (see Fig. 3). LLE maps

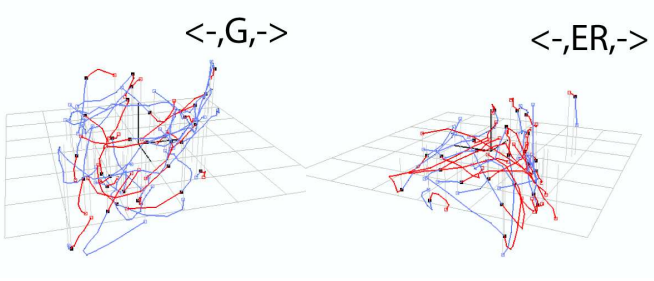

Figure 2. Triphone trajectories with common central phonemes /G/ and /ER/ respectively, in the space spanned by the 3 first PCA dimensions. The first frame of the occurrence of the central phoneme is marked as a black point. The blue and the red segments are the coarticulation path to the preceding and the succeeding phonemes.

a globally curved manifold into a linear space of a given, pre-selected dimension by finding sets of nearest neighbors locally, mapping these points into a low-dimensional subspace, and combining all local subspaces into a global space. Since mouth configurations form a continuous set and are defined by a small number of parameters (muscle activations), LLE is a promising method to estimate the manifold of mouth configurations. Compared to a linear analysis, we capture more of the structure of the data with less dimensions in a representation that is adapted to its underlying manifold.

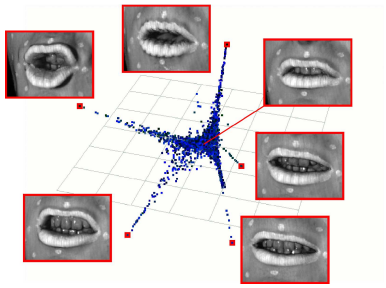

Figure 3. A Locally Linear Embedding (LLE) over the recorded data shows an intuitive and intrinsic low dimensional representation of the data. The central region of the star-shaped manifold holds neutral mouth configurations, the four most important directions correspond to mouth shapes that are typical for |EH|, |W|, |AW| and |I|.

In order to use the LLE in an efficient way, our data has to be adapted and reduced to a lower dimensionality. In order to keep the information of the mouth deformations, we select nine points around the lips (see Fig 4-left) that best reflect the movements of the mouth and combine them into point vectors $\mathbf{v} = (x_1, y_1, z_1, ..., z_9) \in \mathbb{R}^{27}$ on which we perform a PCA, reducing the dimensionality further down to 8. We now have for each recorded frame a representative vector $\mathbf{v}' \in \mathbb{R}^8$. This reduction not only simplifies the LLE computation, but also maps outliers in the measurements to plausible configurations. This smoothing process helps in making the data separation more distinctive. (refer to

Sec. 7-*Statistical Data Analysis*).

By the dynamic nature of our recorded data, its distribution is not homogeneous but samples are aligned along their motion curves and their closest neighbors are thus likely to be the same ones as the ones on the recording timeline. This correlation among the neighbors has a strong impact on the LLE results. LLE constructs a representation that is based on the spatial relation among the samples. To get a better representation of the underlying manifold inside the data, we must first select a sub-group of samples that distribute evenly in the original space. By measuring the distance between any pair of samples (Euclidean distance in the reduced space of the marker points), we consider the samples successively. If one lies closer than a given threshold k to another sample, we remove it and add its frame label to the remaining sample. With increasing k, neighbors lying on a common articulation curve are slowly removed, until the threshold reaches the typical distance that separates the different curves, and samples start to disappear much more quickly (see decay on curve from Fig. 4-right at $k = 0.7$). The LLE is finally computed on the reduced data-set. We

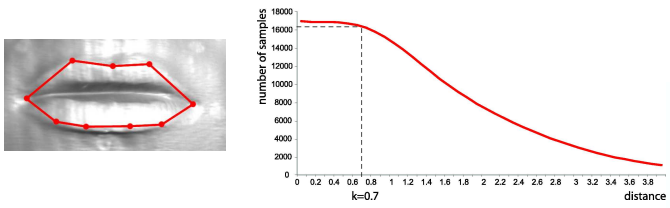

Figure 4. *Left*: for generating the LLE, we reduced the computations by considering the three-dimensional movement of only a few points around the lips. The selected nine points we selected in our scenario are sufficient to represent the most important mouth deformations. *Right*: in order to get a more homogeneous distribution of the data and reduce the amount of redundant samples, we successively remove those which lie closer than a threshold distance k from the others. This procedure ensures that only the most redundant samples get removed.

obtain the best separation of the data by selecting a neighborhood connectivity of six samples and a target dimensionality of six (LLE parameters). The reduction generates the star shaped representation shown in Fig. 3; the central region of the star holds neutral mouth configurations, the four directions (we ignore one as it represents too few samples) go respectively towards mouth shapes that correspond to typical |EH|, |W|, |AW| and |I|. Our algorithm defines viseme clusters in the LLE space along these four main directions.

4. A Measure for the Similarity for Visemes

The audio from the recorded corpus is decomposed into a sequence of phonemes using the CMU-SPHINX software[13] which gives us the time interval during which a phoneme is heard. We take the first frame of each sequences as a viseme sample, and investigate the distribution of these samples in our LLE representation.

The difficulty in a learning-based approach is that the natural number of possible phoneme combinations forming triphones is too large to be possibly recorded, without even considering that several samples of each would be necessary. However, different triphones have strongly correlated motions. By identifying similar triphones, we offer more valid substitution options to ensure good transitions between visemes during coarticulation. The substitution becomes even more essential when desired triphones are not directly available. We propose to do this with a LLE-based similarity measure of visemes, and a viseme categorization that defines substitution rules. For our algorithm we define a data-driven distance measure between visemes that we extend to distances between triphones.

In the original corpus, all first frames of phoneme sequences are mapped to the LLE representation where they form viseme clusters (see Fig. 5 for two examples). The different clusters vary not only in how they spread along the branches of the manifold, but also how they are distributed. We use these two criteria to distinguish between the clusters and to measure their similarities.

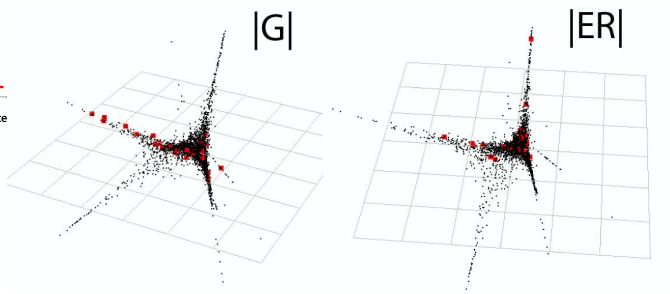

Figure 5. The representation in the LLE space of the recorded viseme candidates for the different phonemes. The black points show all the recorded frames from our corpus, the red squares illustrate the distribution of the different viseme clusters.

We model our LLE representation by a star-shaped manifold composed of four segments (see Fig. 6). The spread of the clusters along each segment is measured after a chosen quantization; in our case they range from 0 to 6. By taking the segments in a fixed order, we are able to describe the distribution of a cluster by a quartet that describes how far the cluster extends on each branch.

This model is implemented the following way: we define four segments (axes) $i \in \{1..4\}$ that connect each of the manifold's extremities to the origin. The length of these segments are then normalized to 1 and all samples are projected onto their clostest axis. We then measure the averaged squared distance d_i of the samples to the origin. Hence, for a given phoneme p, if X_i is a random process describing the position of the samples along the axis i we compute:

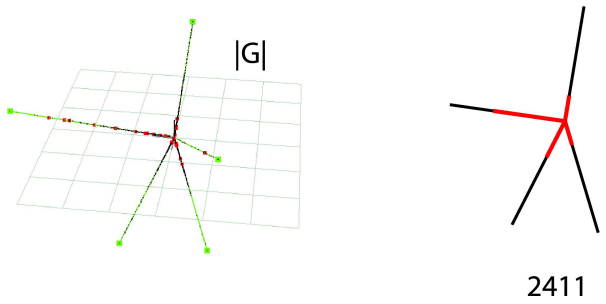

Figure 6. Each viseme cluster defines a distribution on the LLE representation. Projected along the main axes we computed a quantization for the spread along each axes, which taken in a fixed order describes the distribution by simple fourtets.

$$d_{X_i} = \frac{1}{n_i} \sum_{j=0}^{n_i} x_{j,i}^2 \tag{1}$$

where n_i is the number of visemes on axis i. With N being the total number of visemes for p, the quantization $q_p(i)$ of the spread along the axis i becomes:

$$q_p(i) = \frac{n_i}{N} d_{X_i} = \frac{1}{N} \sum_{j=0}^{n_i} x_{j,i}^2 \tag{2}$$

$q_p(i)$ is weighted according to the distributions on the different axes in order to provides a more accurate description of the behavior of the different viseme clusters: when a cluster has a large spreading along several axes, we lower their quantization to favor other clusters which spread on lesser axes and the final quantization will better characterize them. The maximum $q_p(i)$ over all phonemes in P is then normalized to $Q = 6$ (the desired quantization steps[1]) and the remaining $q_p(i)$ scaled accordingly. Finally, rounding to the closest integer gives us the quantization of the spreadings along the axes.

In Table 1 we give the conversion to our quartet notation for each of the visemes. The categorization in this table is an intrinsic statistical property of speech and can thus be directly reused in further animation systems.

PIT	0000	D	1123	P,B	2113	HH	3032
ZH	0002	IH	1221	TH, L, N, S, IY, DH		V	3112
SH	0212	ER	1213		2122	OW	3211
CH	1011	JH	1312	M	2123	R	3212
AY	1022	OY	2010	AX	2212	EY	3331
DX	1023	AW	2013	UW	2221	Y	3312
Z	1032	NG	2022	K, T	2222	W	6111
F	1111	EH	2023	SIL	2223		
AH	1112	AXR	2111	G, UH	2411		
AE	1122	AA, AO	2112	IX	3031		

Table 1. The phonemes and quartet notation with 6 quantization steps. PIT is the neutral head (slightly open mouth), a generalization to all phonemes and prevents our substitution rule from being degenerative. /SIL/ is the "silence phoneme".

[1] |W| is particularly concentrated on the first axis. We set $Q = 6$ to keep a sufficient resolution to be able to distinguish among the other visemes.

5. Substitution Rules for Visemes

We can now formulate a rule for replacing visemes by others when we select triphone candidates. Consider two clusters |A| and |B| with a smaller sample distribution for the latter. In the quartet notation, this implies that all digits of the quartet [B] are pairwise smaller or equal to the digits in the quartet [A], and we write: $|B| \subset |A|$. We argue that samples in |B| are potential instances for the viseme |A| by this *inclusion rule*, but not vice versa.

The *substitution cost* (or distance) is defined as the sum of the digit differences of the desired viseme to its possible substitute. By extension, the distance of a triphone to its substitute, is the sum of its three visemes' distances to their substitutes. We penalize the substitution of the central viseme of the triphone by taking the square of its distance in that sum.

With this rule, we built a substitution graph which is used when we look for best matching triphones. The nodes hold the quartet values and the edges indicate valid substitutions and are weighted with their respective substitution cost. Fig. 7 shows a sub-graph example. Following the edges of the graph, all nodes that we cross are valid candidates. At every jump, at least one of the digits decreases until we reach the quartet 0000 (|PIT|). This quartet is a non-viseme cluster that doesn't spread over any axis. If we reach PIT without having found a fitting quartet, the neutral face will be used in the substitution. However, experiments showed that our rule is generous enough for the substitution algorithm to never reach that case.

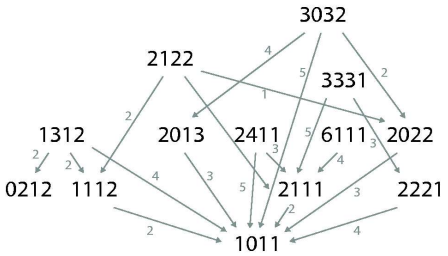

Figure 7. The acyclic directed substitution graph (here only a subset) defines the valid visemes candidates and their substitution.

6. Sentence Generation

The audio file of a new sentence is decomposed into a sequence of phonemes. In order to account for coarticulation and to generate the most natural transition between the visemes, the phonemes of the sequence are transformed into overlapping target triphones. The phonemes in each required triphone are mapped to quartets (Table 1) and the algorithm searches for the available triphone candidates in the database that match the given target triphone. For each target, the system generates a list of all possible candidates by verifying that every pair of visemes inside the triphones are connected in the graph. For each of the candidates in the

lists, a *substitution cost* is attributed. Because triphones are sampled from different portions of the sequence of training scans, they will not describe a continuous curve and may even be far apart in terms of face shape in the PCA space.

If two successive triphones lie far apart or move in opposite directions, the transition will most likely look abrupt. For this reason, it is necessary to collect a list of all possible triphone candidates in the database, and then compute the sequence of triphones that has a minimal *concatenation cost*. In the following of this section, we present how we computed this concatenation cost and the morphing between two triphones. To find the best suited triphone sequence, the sum of all substitution costs and concatenation costs is minimized.

Concatenation of Triphones The concatenation cost φ evaluates the compatibility of two triphones. This is, it gives a measure of the smoothness between these two. A triphone $\langle v_1, v_2, v_3 \rangle$, with v_i as PCA vectors, is composed of a central visemes $|v_2|$ and two neighbors $|v_1|$ and $|v_3|$. When we concatenate two successive triphones $\langle v_1, v_2, v_3 \rangle$ and $\langle w_1, w_2, w_3 \rangle$, we want to generate a curve that traverses v_2 and w_2 and keeps the tangentiality and the direction of the original curve at these points. We generate the "quadriphone" $\langle v_1, v_2, w_2, w_3 \rangle$ from which we only retain the $\langle v_2, w_2 \rangle$ segment (see Fig. 9-left). Let α be the angle between the curve tangents at $|v_2|$ and $|w_2|$ and we define our concatenation cost function as: $\varphi(\alpha) = -\cos\alpha$. Additional criteria, such as the distance between $|v_2|$ and $|w_2|$, did not improve the results further and thus can be omitted.

Given the set of desired target triphones for a new sentence, we find the sequence of triphones from our database that minimizes both the substitution cost θ and the concatenation cost φ. The new sentence is decomposed into a sequence of N triphones $i \in 1, .., N$. To compute the solution, the system provides a list of all M_i possible candidate substitutions $C_{i,j}$ in the database for each triphone i in the sentence. A graph is constructed whose edges connect every triphone from these lists, to all candidates in the preceding and the next list (see Fig. 8). Each of the candidates holds a *substitution cost* $\theta_{i,j}$ (the distance to the requested target triphone) and the *concatenation costs* $\varphi_{i,j,k}$ for the connection of $C_{i,j}$ to one of the preceding candidate nodes $C_{i-1,k}$.

We want to find the path that minimizes the total cost

$$\Theta_{min} = \sum_{i=0}^{N} \min_{j \in \{0,\ldots,M_i\}} \min_{k \in \{0,\ldots,M_{i-1}\}} (\theta_{i,j} + \varphi_{i,j,k}) \quad (3)$$

We use a shortest path algorithm over the graph to find the optimal solution.

Morphing the Optimal Triphones Consider the triphones from the previous paragraph. The transition $\langle v_2, w_2 \rangle$ has to be computed knowing that: (1) the number of frames available in $\langle v_2, v_3 \rangle$ is not the same as in $\langle w_1, w_2 \rangle$; (2) the number of frames to be generated for the new animation is

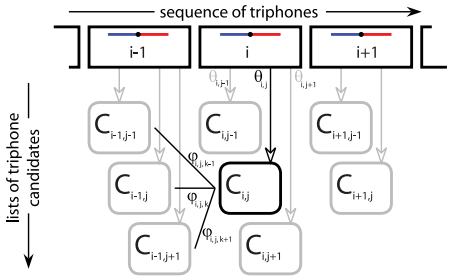

Figure 8. The optimal sequence of triphone substitutions is the one that minimizes the substitution cost θ and the concatenation cost φ. This optimal selection is found by computing the shortest path on a graph generated over the triphone candidates.

dictated by the phoneme sequence in the target audio file (3) and variations in the speed of articulation between the two transitions have to be taken into account.

For each frame t of a candidate triphone, we have initially transformed the original shape vector $\mathbf{S}$ into 50 PCA coefficients $\mathbf{a} = (a_1, a_2, ..., a_{50})^T$, so we have a curved trajectory $\mathbf{a}(t)$ in coefficient space that we can map back to shape vectors $\mathbf{S}$ and render on the screen for reproducing the original motion. We use a piecewise linear function to interpolate between the discrete time steps of the original frames. The argument $t \in [t_0, t_1]$ of this function can be substituted by a variable $u \in [0, 1]$, $u = \frac{t-t_0}{t_1-t_0}$ to obtain a time-normalized function $\mathbf{a}(u)$. If the lengths of the transition phases in two subsequent triphones differ, this will be corrected by the substitution of t by u (see Fig. 9-right).

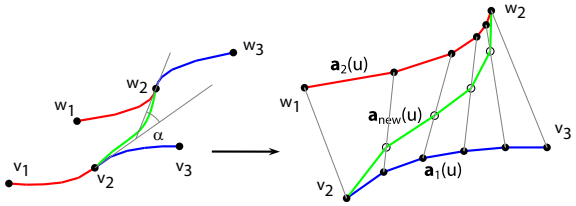

Figure 9. Left: the angle α between two triphones $\langle v1, v2, v3 \rangle$ and $\langle w1, w2, w3 \rangle$. Right: when morphing two triphones, we transform the time axis to a normalized variable $u \in [0, 1]$. The curves $\mathbf{a}_1(u)$ and $\mathbf{a}_2(u)$ preserve the original velocities (time spacing along the red and blue line).

The final animation curve is passed through a low-pass filter which automatically smoothes the transitions where $(\theta + \varphi)$ is most costly, and removes the noise present in the measured data.

7. Results

Besides generating synthetic visual speech from audio data, which we will discuss further below in this section, our approach gives also some insights into the intrinsic structure of natural articulation. Therefore, we discuss the statistical distribution of data, and compare our empirical taxonomy

of phonemes to the standard viseme grouping in the field of *articulator phonetics*.

Our data took a star-shaped distribution within the 6 dimensional manifold that was found by LLE. This means that towards extreme mouth configurations the data are, in fact, locally one-dimensional. This is in contrast to two- or more-dimensional manifolds usually found by LLE, but it is consistent with findings in the phonetics literature. The following reasoning makes this data distribution more intuitive: the representation constructed by the LLE bases only on mouth configurations used in normal speech context. This is, not all the physically plausible mouth shapes are considered. When the mouth goes towards an extreme configuration such as the viseme |O|, it looses degrees of freedom due to the physical constraints such as tissue strain. In the step where the data gets homogeneously distributed, the constrained regions are getting less populated and are therefore more separated in the LLE process. The observation of the representation shows that the mouth stretches to only 4 typical configurations; the many phonemes are then further modulated with the tongue, the nose or the epiglottis.

Phoneme Similarity In our data analysis, we start with no *a priori* assumptions of phoneme similarity. It is interesting to compare our viseme groups with those found in the literature on *articulatory phonetics*[19, 26].

In Table 2, we compare popular viseme associations to our nonlinear model. *Yau et al.*[29] use the MPEG-4 standard that actually comes close to the model proposed by *Binnie et al.* which is discussed in *Owens and Blazek*[19]. We highlight in bold face where our viseme encoding matches closely the standard grouping. Our model matches well both classifications from *Owens and Blazek* and from *Walden et al.* (refer again to [19]). A greater corpus size would make our model converge more, but the triphones would become more redundant. These results provide a rigorous empirical methodology for verifying the classification schemes for the *articulatory phonetics* literature. Our viseme associations are less restrictive for substitutions than the standard viseme grouping as visemes who have little correlations can still be considered for substitution according to the context they are taken for.

Animation The supplemental video to this paper shows examples of synthetic speech generated with our system. In the generation of the video, high frequency motions are smoothed out by a low-pass filter, while we made sure to keep the cutoff frequency above the spectral range of natural mouth movements.

According to Section 2, expressions were added to make the results look more lively. Furthermore, we added some eye-candies that are learned from the acquired data. Eyelid movements were recorded used for eye-blinking as well as following the movement of the iris. Eye-movements are produced by a simple, yet effective texture displacement.

A PCA-Based Similarity Measure In order to evaluate the LLE-based similarity measure as a substitution rule, we implemented a different rule based on the PCA representation. In the PCA space, we computed for every cluster its average shape. Taken two by two, the Euclidean distance between these average visemes gives the substitution cost. Unlike the LLE-based substitution rule, the PCA-based rule allows substitutions between any cluster with varying costs.

Syntheses with the full corpus show almost equivalent results using both methods with slight disturbances for the PCA scenario. In fact, with the full corpus, only few substitutions occur and the algorithm turns out to find triphones samples with the desired central viseme in the triphones. In order to force substitutions, we reduced the size of the corpus and generated new animations. In three stages, the size of the corpus is each time divided by 2. At each reduction, the quality of the animation decreases for both methods. However, we noticed that because triphones are time synchronized to the novel audio track, the rhythm of the mouth movement produces by its own a realist effect. In order to analyze the difference between the two methods, we had to focus on the actual movements associated to each phoneme. The LLE-based animations show on that aspect more credible movements than the ones based on the PCA.

8. Conclusion

In this work, we proposed a new data-driven system to automatic speech animation. Our novel selection method takes full advantage of the dual association between phonemes and visemes: not only can a phoneme take the visual appearance of several visemes, but visemes can be attributed to different phonemes as well. Our method determines visemes that can be used as a valid substitution for a specific phoneme even if there is no such association in the original corpus.

We developed a similarity measure among visemes that goes beyond using standard viseme groupings defined in *articulatory phonetics*: we propose a gradual measure, based on a *Locally Linear Embedding* (LLE) of the data, to distinguish visemes and build a hierarchical substitution rule. Furthermore, our measure could be directly extended to triphone similarities. The benefit of the our approach is in the tradeoff between database size and realism: while we keep a relatively small corpus of real data information, we generate realistic articulation motions by finding the optimal combination of triphones. Finally, the hierarchical structure of our selection method that we derived from the data is intrinsic to the nature of mouth articulation and can thus be reused as-is for different speech animation systems.

MPEG-4 (*Binnie et al.*)	*Owens and Blazek*	*Walden et al.*	Nonlinear Model
/P,B,M/	**/P,B,M/**	**/P,B,M/**	**2113 2113 2123**
/F,V/	**/F,V/**	**/F,V/**	**1111 3112** ($\|F\| \subset \|V\|$)
/TH,DH/	**/TH,DH/**	**/TH,DH/**	**2122 2122**
/T,D/	-	-	2222 1123
/S,Z/	-	-	2122 1032
-	/T,D,S,Z/*	-	2222 1123 2122 1032
/K,G/	-	-	2222 2411
/N,L/	-		**2122 2122**
-	**/K,G,N,L/***	-	**2222 2411 2122 2122**
-	-	/T,D,S,Z,K,G,N,L/*	2222 1123 2122 1032 2222 2411 2122 2122
/CH,JH,SH/	-	-	**1011 1312 0212** ($\|CH\|+\|SH\| \subset \|JH\|$)
-	**/CH,JH,SH,ZH/***	**/CH,JH,SH,ZH/***	**1011 1312 0212 0002** ($\|CH\|+\|SH\|+\|ZH\| \subset \|JH\|$)
/W/	-	-	**6111****
/R/	-	-	3212
-	/W,R/*	/W,R/*	6111 3212

($*$) depend on the vowel context and can thus be further divided.[19] ($**$) $|W|$ has a specific distribution that matched no other viseme.

Table 2. Our nonlinear model is compared to popular viseme groups. We highlighted in bold face where the viseme encoding matches the standard grouping closely.

References

[1] I. Albrecht, J. Haber, and H.-P. Seidel. Speech Synchronization for Physics-based Facial Animation. In V. Skala, editor, *Proc. 10th Int. Conf. on Computer Graphics, Visualization and Computer Vision (WSCG 2002)*, pages 9–16. UNION Agency, 2002.

[2] R. Bargmann, V. Blanz, and H.-P. Seidel. Learning-based facial rearticulation using streams of 3d scans. In *Proceedings of Pacific Graphics 2006*, pages 232–241, 2006.

[3] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In P. Brunet and D. Fellner, editors, *Computer Graphics Forum, Vol. 22, No. 3 EUROGRAPHICS 2003*, pages 641–650, Granada, Spain, 2003.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[5] M. Brand. Voice puppetry. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999.

[6] C. Bregler, M. Covell, and M. Slaney. Video rewrite:driving visual speech with audio. In *Computer Graphics Proc. SIGGRAPH'97*, pages 67–74, 1997.

[7] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin. Real-time speech motion synthesis from recorded motions. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 345–353. Eurographics Association, 2004.

[8] Y. Cao, W. C. Tien, P. Faloutsos, and F. Pighin. Expressive speech-driven facial animation. *ACM Trans. Graph.*, 24(4):1283–1302, 2005.

[9] E. Chuang and C. Bregler. Mood swings: expressive speech animation. *ACM Trans. Graph.*, 24(2):331–347, 2005.

[10] M. M. Cohen and D. W. Massaro. Modeling coarticulation in synthetic visual speech. In N. Magnenat Thalmann and D. Thalmann, editors, *Models and Techniques in Computer Animation*, pages 139–156. Springer, Tokyo, 1994.

[11] Z. Deng and U. Neumann. eFASE: expressive facial animation synthesis and editing with phoneme-isomap controls. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 251–260. Eurographics Association, 2006.

[12] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 388–398. ACM Press, 2002.

[13] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The SPHINX-II speech recognition system: an overview. *Computer Speech and Language*, 7, 2:137–148, 1993.

[14] G. A. Kalberer, P. Müller, and L. J. V. Gool. Speech animation using viseme space. In *VMV*, pages 463–470, 2002.

[15] I.-J. Kim and H.-S. Ko. 3d lip-synch generation with data-faithful machine learning. In *Computer Graphics Forum, Vol. 26, No. 3 EUROGRAPHICS 2007*, 2007.

[16] S. A. King and R. E. Parent. Creating speech-synchronized animation. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):341–352, 2005.

[17] S. Kshirsagar and N. Magnenat-Thalmann. Visyllable based speech animation. In P. Brunet and D. Fellner, editors, *Computer Graphics Forum, Vol. 22, No. 3 EUROGRAPHICS 2003*, pages 631–639, Granada, Spain, 2003.

[18] J. Ma, R. Cole, B. Pellom, W. Ward, and B. Wise. Accurate visible speech synthesis based on concatenating variable length motion capture data. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):266–276, 2006.

[19] E. Owens and B. Blazek. Visemes observed by hearing-impaired and normal-hearing adult viewers. *Journal of Speech and Hearing Research*, 28:381–393, 1985.

[20] C. Pelachaud, N. I. Badler, and M. Steedman. Generating facial expressions for speech. *Cognitive Science*, 20(1):1–46, 1996.

[21] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[22] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.*, 24(3):417–425, 2005.

[23] E. Sifakis, A. Selle, A. Robinson-Mosher, and R. Fedkiw. Simulating speech with a physics-based facial muscle model. In *SCA '06*, pages 261–270. Eurographics Association, 2006.

[24] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):569–579, 1993.

[25] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24(3):426–433, 2005.

[26] B. Walden, S. Erdman, A. Montgomery, D. Schwartz, and R. Prosek. Some effects of training on speech recognition by hearing-impaired adults. *Journal of speech and hearing research*, 24:207–16, 1981.

[27] K. Wampler, D. Sasaki, L. Zhang, and Z. Popović. Dynamic, expressive speech animation from a single mesh. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 53–62. Eurographics Association, 2007.

[28] A. Wang, M. Emmi, and P. Faloutsos. Assembling an expressive facial animation system. In *Sandbox '07: Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, pages 21–26. ACM Press, 2007.

[29] W. C. Yau, D. K. Kumar, and S. P. Arjunan. Voiceless speech recognition using dynamic visual speech features. In *VisHCI '06: Proceedings of the HCSNet workshop on Use of vision in human-computer interaction*, pages 93–101. Australian Computer Society, Inc., 2006.

[30] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 548–558. ACM Press, 2004.