SPEECH SYNTHESIS USING STOCHASTIC MARKOV GRAPHS

Matthias Eichner, Matthias Wolff, Sebastian Ohnewald and Rüdiger Hoffmann
Dresden University of Technology
Laboratory of Acoustics and Speech Communication
D-01062 Dresden, Germany

ABSTRACT

Speech synthesis systems basing on concatenation of natural speech segments achieve a high quality in terms of naturalness and intelligibility. However, in many applications such systems are not easy to apply because of the huge demand for storage capacity. Speech synthesis systems based on HMMs could be an alternative to concatenative speech synthesis systems but do not yet achieve the quality needed for use in applications.

In one of our research projects we investigate the possibility of combining speech synthesis and speech recognition to a unified system using the same databases and similar algorithms for synthesis and recognition. In this context we are examining the suitability of Stochastic Markov Graphs instead of HMMs to improve the performance of such synthesis systems. This paper describes the training procedure we used to train the SMGs, explains the synthesis process and introduces an algorithm for state selection and state duration modeling. We focus particularly on issues which arise using SMGs instead of HMMs.

1. MOTIVATION

Various speech synthesis systems based on Hidden Markov Models have been developed in the past (see e.g. [1], [4], [6]). This observation emphasizes the tendency that the methods and databases of speech synthesizers obtain more and more similarity with those of speech recognizers. This seems to offer new possibilities in combining systems for speech synthesis and speech recognition. Considering this fact, we introduced a dialogue system with the synthesis and recognition components using unified databases in [7].

Combining speech synthesis and speech recognition in a single system allows "Analysis by Synthesis" to evaluate the performance of the recognition process in detail. It allows an inverse function of the recognizer which will enable us to improve the performance of the system. In this paper we investigated the suitability of Stochastic Markov Graphs (SMG) instead of HMMs as models in a parametric speech synthesizer. SMGs were first introduced in [5] for the recognition task. The advantage of using SMGs lies in their enhanced capability of modeling trajectories in the feature space. However, a powerful prediction of feature trajectories is even more important for synthesis. SMG based speech synthesizers require less resources than concatenative synthesis systems and promise to improve the limited quality of HMM based synthesizers.

In the following sections we will explain the procedure of training the SMGs and we will describe our speech synthesis algorithm. We focus particularly on issues which arise using SMGs instead of HMM and introduce an adapted method for state selection and state duration modeling.

2. STOCHASTIC MARKOV GRAPHS

In the following sections we describe a number of graph processing algorithms. We will use the following symbols and notations: $\gamma(U,\Psi_{UU})$ denotes a directed graph with the state set $U=\{u_1,u_2,...,u_N\}$ and the incidence relation $\Psi_{UU}:U\times U\rightarrow\{\varnothing,1\}$. We denote an edge between two states u_i and u_k as $(u_i\rightarrow u_k)$. We do not allow parallel edges, hence the edge set is implicitly defined by the incidence relation. A particular edge exists in the graph if, and only if, the incidence relation is not empty for the ordered tuple (u_i,u_k), i.e. $exist(u_i\rightarrow u_k)\Leftrightarrow\psi_{u_iu_k}\neq\varnothing$. The transition probability of an edge $(u_i\rightarrow u_k)$, estimated by the graph training process, is written as $P(\psi_{u_i,u_k})$. A consecutive sequence of edges in a graph is called a path *q*. We refer to the *i*th state of the path as $q(i)$. SMG graphs contain circles. We distinguish between trivial circles and non-trivial circles. A trivial circle consists of exactly one edge which starts and ends in the same state. A non-trivial circle is a path with more than one edge.

3. TRAINING

Our training procedure mainly resembles the procedure described in [5]. Figure 1 gives an overview of the training procedure.

We use 20 MFC coefficients, their deltas and delta-deltas and one energy value as acoustic features. After the feature extraction we apply an eigenvector transformation and sort the transformed features by their standard deviation. We split the feature space into 24 most significant features (MSF) and 37 least significant features (LSF). We use the MSF for building the acoustic models. The LSF are modeled by simpler statistics (mean vectors and standard deviation) and will be used for the acoustic synthesis. The SMG training procedure also gathers statistics on non-acoustic features:

- per SMG state:
  - Number of non-trivial circles per sample occurring at the state (*re-entry statistic*)
  - Phone duration of samples using the state
- per SMG phone model:
  - Path length used by samples, disregarding trivial circles

See section 4 for a detailed discussion of these non-acoustic features.

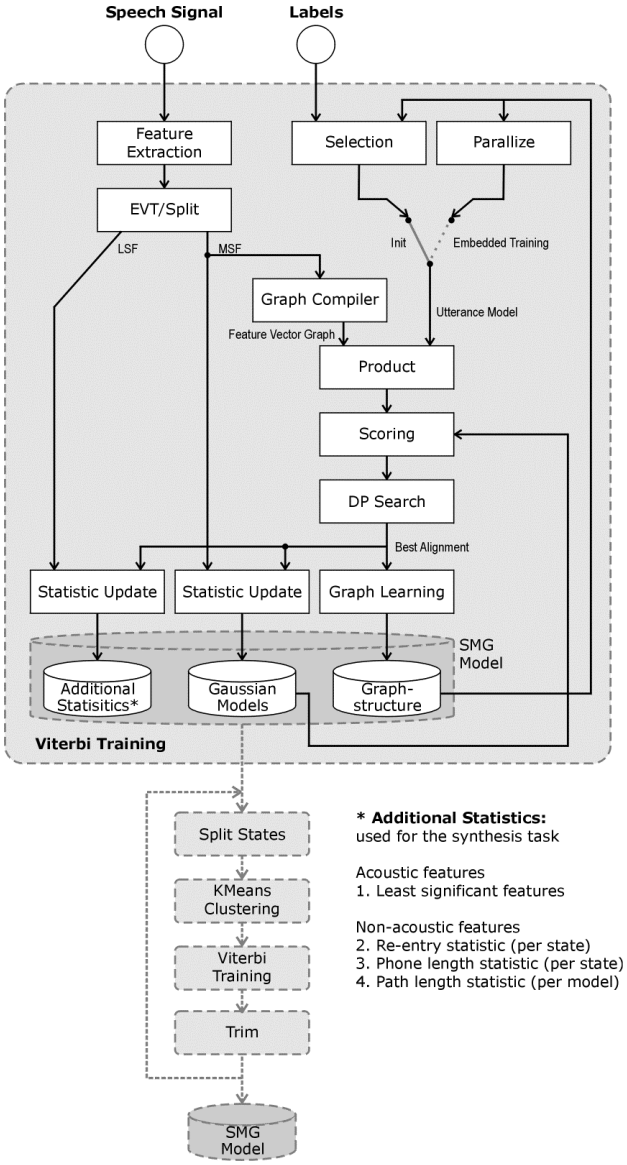

Figure 1: Overview of the SMG training procedure

We start the training with a conventional three-state forward connected HMM structure. In the initialization stage, each state is assigned to exactly one Gaussian distribution. We train this model with the standard Viterbi algorithm. After the training, each state of the model is split into two.

$$\forall_{u_i \in U} : U = U \cup \{u_i'\} \tag{1}$$

The resulting states inherit all transitions to the predecessors and successors of the original state.

$$\begin{aligned} &\forall_{u_i, u_k \in U} \left(\psi_{u_i u_k} \neq \varnothing\right) : \psi_{u_i' u_k} = 1 \\ &\forall_{u_i, u_k \in U} \left(\psi_{u_k u_i} \neq \varnothing\right) : \psi_{u_k u_i'} = 1 \end{aligned} \tag{2}$$

Further they will be connected with each other by two additional edges.

$$\psi_{u_i' u_i} = 1 \quad \text{and} \quad \psi_{u_i u_i'} = 1 \tag{3}$$

Together with the original states, we split the Gaussian distributions along the axis of their greatest standard deviation. After splitting, we cluster the Gaussian distributions with the k-means algorithm. Then the entire SMG model is retrained.

In a last stage the SMG models are trimmed. By the term *trim* we denote the process of removing improbable edges and paths from an SMG. This procedure consists of two stages. In the first stage, all edges whith transition probabilities less than a given threshold p^* are removed from the graph:

$$\forall_{u_i, u_k \in U} \left(\psi_{u_i u_k} \neq \varnothing \wedge p_{u_i u_k} < p^*\right) : \psi_{u_i u_k} = \varnothing \tag{4}$$

Removing edges from a graph can produce dead paths. We call a path dead if it is not part of a consecutive path from a start state to an end state of the graph. Such dead paths are removed from the graph in a second stage by iteratively applying the following algorithm on the graph:

$$\begin{aligned} &\forall_{u_k \in U} \left(\nexists_{u_i \in U} \psi_{u_i u_k} \neq \varnothing \vee \nexists_{u_j \in U} \psi_{u_k u_j} \neq \varnothing \right) : \\ &\cdots \begin{cases} U = U \setminus \{u_k\} \\ \forall_{u_m \in U} \psi_{u_k u_m} \neq \varnothing : \psi_{u_k u_m} = \varnothing \end{cases} \end{aligned} \tag{5}$$

The procedure is repeated until the condition on the left hand side of equation (5) is false for all u_k .

We repeat the process of splitting, clustering, retraining and trimming of the SMG states iteratively until

a) a maximum total number of states is reached *or*

b) the number of states which are trimmed in the last stage of the iteration is greater than $0.3 \cdot 2^I$ (where I is the number of state splits since the initialization).

4. SPEECH SYNTHESIS USING SMGs

The synthesis algorithm using SMGs introduces a different algorithm for state selection and state duration modeling in comparison to HMM speech synthesis (cf. [6]). It uses the phoneme symbol sequence and target duration as input information and includes the following processing steps:

- selecting a state sequence (path through the SMG) according to a demanded sequence length (target phoneme duration) and modeling the duration of each state in the path
- assembling the feature vector sequence for the chosen path by extracting the means of the corresponding Gaussians
- generating the speech signal using the MLSA filter[3]

In comparison to HMMs, SMGs introduce a more complex graph structure. In general an SMG contains parallel paths. Thus during synthesis the choice for one of these paths is necessary and crucial for the resulting speech quality. This state selection has to find the optimal state sequence for a given

target phone duration. At the first glance this looks like a search for the most probable path under the constraint of a given path length. Since the SMGs are Markov models of first order the probability of a trivial circle does not depend on how often this circle has already been passed. A Viterbi search maximizing the overall transition probability would not result in a typical state sequence but would select the shortest path containing the most probable trivial circle since the iteration in this circle collects the highest scores.

Using higher order Markov models (see e.g. [8]) would solve the problem but is not feasible because of the scarcity of training data needed to train such models and the increasing computational effort during state selection. To solve the problem we additionally derived non-acoustic statistics for each state of the SMG model (see below).

4.1 State Selection

As stated above, a simple Viterbi search on the SMG model is not suitable to find a representative state sequence. In the following two subsections we describe a two stage approach to find an optimal state sequence for the acoustic synthesis which bases on a transformed SMG structure and additional statistic data obtained during the SMG training.

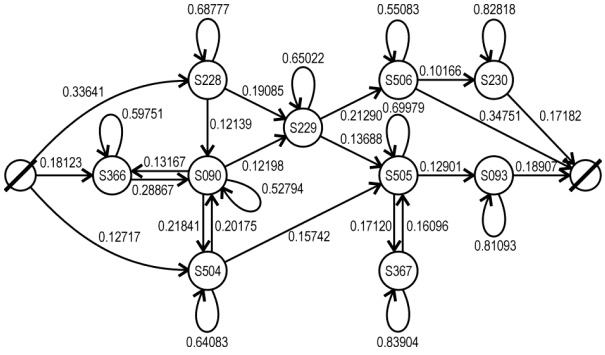

Figure 2: Example of an SMG model (phone /i:/, trained with approx. 1 hour read German speech of one male speaker)

In a first stage we transform the trained SMGs γ (Figure 2 shows an example) into an alternative representation γ' by the transformation *TE*:

$$\gamma'\left(U', \Psi'_{U'U'}\right) = TE\left(\gamma\left(U, \Psi_{UU}\right)\right) \tag{6}$$

TE is basically a tree expansion. However, as SMGs contain circles, we have to modify the tree expansion algorithm in the following two points:

- disregard trivial circles
- limit the number of passes through non-trivial circles.

Disregarding the trivial circles does not introduce any problems as we can easily re-insert these circles into γ'. The limitation of passes through non-trivial circles requires some information on how often a particular state can occur in a non-trivial circle. We gather this information by means of *re-entry statistics* during the SMG training. These statistics describe, how often a particular state occurs in a training sample disregarding immediate repetitions (or trivial circles). Column μ_{Nent} of Table 1 shows an example of the re-entry statistic for the phone /i:/:

u	$\mu_{T,u}$ [Frames]	$\sigma_{T,u}$ [Frames]	μ_{Nent}
S365	19,62	6,40	1.03
S228	16,59	6,33	1.02
S090	19,31	5,48	1.15
S504	18,25	6,87	1.01
S229	21,25	6,40	1.00
S506	19,24	7,18	1.01
S505	19,45	6,12	1.15
S367	23,16	11,23	1.00
S230	18,29	6,24	1.01
S093	20,80	7,43	1.01

Table 1: Example of non-acoustic per-state features (model /i:/)

$\mu_{pathlength}$[Frames]	$\sigma_{pathlength}$[Frames]
5.11	1.45

Table 2: Example of non-acoustic per-model features (model /i:/)

The figures of μ_{Nent} are representative for all states of all SMG models. With μ_{Nent} being the number of entries into a state, we see, that the re-entry into a state using a non-trivial circle is a very rare event. So for the transformation algorithm we simply did not allow non-trivial circles.

Thus we apply the following transformation algorithm *TE*:

$$① \quad \lambda\left(u'^0_0\right) = u_0$$

$$② \left[\begin{array}{l} \underset{u_i'^{t-1} \in U'}{\forall} \left(\underset{u_k \in U}{\forall} \left(\psi_{\lambda\left(u_i'^{t-1}\right) u_k} \neq \varnothing \wedge c\left(u_i'^{t-1}\right) \leq 1 \right) \right) : \\ \quad \left\{ \begin{array}{l} U' = U' \cup \left\{ u'^t_k \right\} \\ \lambda\left(u'^t_k\right) = u_k \\ \psi'_{u_i'^{t-1} u'^t_k} = 1 \end{array} \right. \end{array} \right]_{t=1}^{T} \tag{7}$$

where

- t denotes the current tree depth and T denotes the maximum tree depth. The value of T is obtained from a per-model statistic gathered during the training. Table 2 shows this statistic for our example model /i:/.

[1] For better legibility we simplified the formula in two points: firstly on the left hand side trivial circles should be excluded and secondly on the right hand side duplicate instances of the transformed state u'^t_k should be distinguished.

- $\lambda : U' \to U$ describes the relation between the transformed states $u_i'^t$ and the original state u_i
- $c\left(u_i'^t\right)$ denotes a counter function for the re-occurrence of the original state $\lambda\left(u_i'^t\right)$ in a path $q_i'^t$ from state $u_0'^0$ to state $u_i'^t$ through the transformed SMG:

$$c\left(u_i'^t\right) = \sum_{n=0}^{t} \begin{cases} \lambda\left(q_i'^t(n)\right) = \lambda\left(u_i'^t\right) & :1 \\ \text{else} & :0 \end{cases} \tag{8}$$

γ' lists all potential paths through the SMG. To find the best one, each path q' is traversed and a score is calculated given the demanded phone duration. This score consists of two parts. The first part is the product of the transition probabilities of all transitions in the path and the second part represents the probability that a certain state $q'(i)$ belonged to a training sample of length *T*. The parameters for this distribution have been extracted in a separate statistic during training.

$$\begin{aligned} s(q' \mid T, \gamma') &= P_{trans}(q') \cdot p_{state}(q', T) \\ &= \prod_{i=0}^{N-1} P\left(\psi'_{q'(i)q'(i+1)}\right) \cdot \prod_{i=1}^{N} p(T, \mu_{q'(i)}, \sigma_{q'(i)}) \end{aligned} \tag{9}$$

The selected path q'_{res} is the one that maximizes:

$$q'_{res} = \arg\max_{q'} s(q' \mid T, \gamma') . \tag{10}$$

4.2 State Duration Modeling

In a third step the duration (number of passes through the trivial circles $(u_i \to u_i)$) of each state of the selected path is determined. Since the optimal sequence of states q'_{res} is already chosen in γ', the duration modeling is limited to the distribution of repetitions of all the states in the path until the demanded phone length is reached. In our approach we use the ratio between the loop probabilities to assign the appropriate number of repetitions. In contrast to [2], we do not consider the variability of the state durations. We compute relative length of state $q'(i)$ by relative duration values:

$$d'_i = \frac{1}{1 - P\left(\psi'_{q'(i)q'(i)}\right)} \tag{11}$$

These duration values can be interpreted as the mean duration in a state as observed in the training. Let q'_{res} be the optimal path with *N* states and let $D'_{q'_{res}}$ be the sum of all duration values in that path, then under the constraint $N \le T$ the number of repetitions to distribute *L* can be calculated as the difference of the number of feature vectors to emit *T* and the number of states existing in the selected path *N*:

$$L'res = T - N \tag{12}$$

The factor

$$l'_i = \frac{d'_i}{D'_{q'_{res}}} \tag{13}$$

is used to obtain the resulting repetition number for each state:

$$L'_i = l'_i \cdot L'_{res} = \frac{d'_i}{D'_{q'_{res}}} \cdot (T - N) \tag{14}$$

Thus the optimal path can be written as:

$$q'_{opt} = \left\{ [q'_{res}(0)]^{1+L'_0}, [q'_{res}(1)]^{1+L'_1}, \ldots, [q'_{res}(N)]^{1+L'_N} \right\} \tag{15}$$

The final step of the synthesis process assembles the feature vector sequence by extracting the means from the corresponding Gaussian distributions according to q'_{opt}. This feature vector sequence is finally feed into to the synthesis filter to generate the speech signal.

5. CONCLUSION

Applying SMGs as acoustic models in a speech synthesis system raises the task of finding the optimal state sequence within a phoneme model in a given context. The proposed algorithm uses additional statistics derived during the training of the SMGs which model the probability of using and remaining in a certain state depending on a demanded phone duration. This approach reduces the computational complexity in comparison to solutions using higher order Markov chains. Future work will include extensive evaluation and optimization of the algorithm.

6. REFERENCES

[1] Falaschi, A., Giustiniani, M., and Verola, M., "A hidden Markov model approach to speech synthesis", *Proc. Eurospeech 1989, Paris*, 187 - 190.

[2] Tokuda, K., et al., "Speech parameter generation algorithms for HMM-based speech synthesis", *Proc. ICASSP 2000, Istanbul*, 1315 - 1318.

[3] Imai, S., Sumita, K., and Furuichi, C., "Mel log spectrum approximation (MLSA) filter for speech synthesis", *Trans. IECE, vol. J66-A*, 122-129, 1983.

[4] Tychtl, Z., and Psutka, J., "Speech production based on the mel-frequency cepstral coefficients", *Proc. Eurospeech 1999, Budapest*, vol. 5, 2335 – 2338.

[5] Wolfertstetter, F., and Ruske, G., "Structured Markov models for speech recognition", *Proc. ICASSP 1995, Detroit*, 544-547.

[6] Masuko, T., Tokuda, K., Kobayashi, T., and Imai, S., "Speech synthesis using HMMs with dynamic features", *Proc. ICASSP 1996, Atlanta*, 389-392.

[7] Eichner, M., Wolff, M., Hoffmann, R., "A unified approach for speech synthesis and speech recognition using Stochastic Markov Graphs", *Proc. ICSLP 2000, Beijing,* vol. 1, 701-704

[8] Bühlmann, P., Wyner, A.J., "Variable length Markov chains", *Annals of Statistics, vol. 27,* 480-513, 1999