Using Robust Viterbi Algorithm and HMM-Modeling in Unit Selection TTS to Replace Units of Poor Quality

Hanna Silén[1], *Elina Helander*[1], *Jani Nurminen*[2], *Konsta Koppinen*[1], *Moncef Gabbouj*[1]

[1]Department of Signal Processing, Tampere University of Technology, Tampere, Finland
[2]Nokia Devices R&D, Tampere, Finland

hanna.silen@tut.fi, elina.helander@tut.fi

Abstract

In hidden Markov model-based unit selection synthesis, the benefits of both unit selection and statistical parametric speech synthesis are combined. However, conventional Viterbi algorithm is forced to do a selection also when no suitable units are available. This can drift the search and decrease the overall quality. Consequently, we propose to use *robust Viterbi algorithm* that can simultaneously detect bad units and select the best sequence. The unsuitable units are replaced using hidden Markov model-based synthesis. Evaluations indicate that the use of robust Viterbi algorithm combined with unit replacement increases the quality compared to the traditional algorithm.

Index Terms: robust Viterbi algorithm, unit selection, hidden Markov models

1. Introduction

Quality in text-to-speech (TTS) synthesis is typically a tradeoff between the voice quality and consistency. While concatenative unit selection TTS methods [1] preserve the naturalness of real speech at segmental level, they often suffer from larger single errors caused by the limited coverage of the database. On the other hand, statistical parametric synthesis methods [2] can produce stable quality, but the speech is typically less real-sounding due to the parameterization and model training.

In unit selection TTS, speech is generated by concatenating speech units taken from a recorded database. Since real speech and a limited amount of modification are employed, voice quality in synthesis is rather similar to the original recordings. However, the database coverage is essential in order to avoid audible mismatches in synthesis. Covering all possible context-dependent units of a language is practically impossible.

Statistical parametric hidden Markov model-based TTS (HMM-TTS) is able to avoid the problem of inconsistent quality and database coverage requirements common in unit selection. The speech data is used for training models that are used in generating parameter tracks for synthetic speech and unseen units can be predicted. Nevertheless, the modeling is not able to capture all fine speech variations resulting in averaged speech quality.

To combine the consistency of HMM-TTS with the real-sounding voice quality of unit selection TTS, hybrid approaches have been introduced e.g. in [3, 4, 5]. In HMM-based unit selection TTS [4], the selection of a unit sequence is guided by HMM-based prediction, but the units used in concatenation are real speech units. The underlying HMM-based prediction provides a stable overall quality without abrupt artifacts whereas the use of real speech in concatenation preserves natural variation that is difficult to model. In multiform speech synthesis, unit selection and HMM-TTS are mixed by synthesizing some parts of an utterance using unit selection and some parts using HMM-TTS [6].

In unit selection, the search of the most suitable database unit sequence is a dynamic programming problem that can be solved by the Viterbi algorithm. The selection is based on global minimization of a cost function: for each candidate unit taken from the database, a target and join cost representing its suitability and the smoothness of concatenation are determined. The traditional Viterbi algorithm is forced to make a decision also when all the candidates are unsuitable for the given context. There are two types of problems related to the traditional Viterbi search. First, the search can average the total cost over longer sequences in which case all the units are of average suitability or quality. Secondly, the procedure can end up selecting otherwise good units but some poor-quality units with larger errors. In [7], the problem is solved by rerunning Viterbi search iteratively and removing candidates classified as unnatural between iterations. This kind of repeated Viterbi search is however timeconsuming.

In this paper, we propose to use the robust Viterbi search [8] in the framework of HMM-based unit selection TTS and to replace the detected bad units using HMM-TTS employing the same parameterization as the unit selection TTS. The robust Viterbi algorithm is allowed to skip a pre-defined number of units during the search. Since the search is not forced to make a decision if there are only unsuitable candidates available, it is not drifted due to single outliers and errors can be compressed into shorter sections. It also enables the selection of longer continuous speech segments for synthesis, since the units between these segments can be ignored in the search.

The evaluations carried out for two different languages and parameterizations indicate that the robust Viterbi algorithm combined with the replacement of unsuitable units is able to improve the synthesis quality compared to the conventional selection procedure. Using the same parameterization in both unit selection and HMM-based TTS enables the switching between the methods without disturbing change in the voice quality. Because the prosody is guided by the same HMM models in both cases, the change in the synthesis method does not introduce any major prosodic differences.

This paper is organized as follows. Section 2 gives an overview of the cost function formulation in the conventional and HMM-based unit selection synthesis. Section 3 describes the use of robust Viterbi algorithm in unit selection speech synthesis. Experiments and results are presented in Section 4. Discussion and conclusions are given in Sections 5 and 6, respectively.

2. Cost function formulation for HMM-based unit selection

2.1. Unit selection procedure

The unit selection procedure aims at selecting a sequence of database units with a good match with the predicted target unit sequence and smooth transitions between the units. A cost value is assigned to each candidate unit and candidate unit combination. The sequence $\mathbf{u}^*$ with the lowest total cost $C^{tot}(\mathbf{u})$ over all possible sequences is the one to be selected

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathbf{U}} C^{tot}(\mathbf{u}) \tag{1}$$

where $\mathbf{U}$ is the set of all sequences of possible units for the sentence to be synthesized. The cost function is usually a sum of a target cost and a join cost. Total cost $C^{tot}(\mathbf{u})$ of a sequence $\mathbf{u}$ is

$$C^{tot}(\mathbf{u}) = \sum_{n=1}^{N} C^t(u_n) + \sum_{n=2}^{N} C^j(u_{n-1}, u_n) \tag{2}$$

where $C^t(u_n)$ and $C^j(u_{n-1}, u_n)$ are the target cost of candidate unit u_n and join cost of concatenating candidates u_{n-1} and u_n. The total length of the unit sequence is N.

In traditional unit selection TTS, the target cost typically consists of a small set of linguistic and phonetic features. They are assumed to correspond to acoustic properties of speech. Typical features are unit's stress, position, and context. The join cost measures the concatenation smoothness and it typically involves computation of the the frame spectra and fundamental frequency (F_0) distances around the concatenation boundary.

2.2. HMM-based unit selection

Conventional unit selection provides only a loose control on the speech features of a unit. HMM-based unit selection tackles the problem by using HMM-based prediction as a basis of the selection. However, despite the use of HMMs, the natural voice quality of the standard unit selection is preserved consequent on the use of real database units in concatenation.

In HMM-based unit selection, costs are formulated using the distance between a candidate unit parameterization and the corresponding HMM model. In [4], the target cost is computed as a weighted distance of a candidate unit and the corresponding statewise prediction

$$\begin{aligned} C^t(u_n) =& \alpha_1\lambda_n \sum_{i=2}^{T_n-1} d\left(\mathbf{o}_{n,i}, \boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i}\right) \\ &+ \alpha_2 d\left(T_n, \mu_n^{dur}, {\sigma_n^{dur}}^2\right) \end{aligned} \tag{3}$$

where $d(\mathbf{o}_{n,i}, \boldsymbol{\mu}_{n,i}, \boldsymbol{\Sigma}_{n,i})$ denotes the Mahalanobis distance between the parameterization of the ith frame of candidate unit u_n and the corresponding HMM model with mean $\boldsymbol{\mu}_{n,i}$ and covariance $\boldsymbol{\Sigma}_{n,i}$, while $d\left(T_n, \mu_n^{dur}, {\sigma_n^{dur}}^2\right)$ denotes the Mahalanobis distance between the observed candidate unit duration T_n and the corresponding duration model with mean μ_n^{dur} and variance ${\sigma_n^{dur}}^2$. Scaling factor $\lambda_n = T_n^P/T_n$ is the ratio of the predicted unit duration T_n^P and T_n.

The join cost is formulated in [4] as a weighted sum of the Mahalanobis distances between the parameterized boundary frames $\mathbf{o}_{n,1}$ and $\mathbf{o}_{n-1,T_{n-1}}$ and the corresponding HMM models, and the Mahalanobis distance of the boundary frame difference $\mathbf{o}_{n,1} - \mathbf{o}_{n-1,T_{n-1}}$ and the corresponding concatenation model with mean $\boldsymbol{\mu}_n^{con}$ and covariance $\boldsymbol{\Sigma}_n^{con}$

$$\begin{aligned} C^j(u_{n-1}, u_n) =& \alpha_1\lambda_n d(\mathbf{o}_{n,1}, \boldsymbol{\mu}_{n,1}, \boldsymbol{\Sigma}_{n,1}) \\ &+ \alpha_1\lambda_{n-1} d(\mathbf{o}_{n-1,T_{n-1}}, \boldsymbol{\mu}_{n-1,T_{n-1}}, \boldsymbol{\Sigma}_{n-1,T_{n-1}}) \\ &+ \alpha_3 d(\mathbf{o}_{n,1} - \mathbf{o}_{n-1,T_{n-1}}, \boldsymbol{\mu}_n^{con}, \boldsymbol{\Sigma}_n^{con}) \end{aligned} \tag{4}$$

In (3)–(4), α_1–α_3 refer to weights given to each subcost.

3. Robust Viterbi algorithm for unit selection TTS

3.1. Comparison of standard and robust Viterbi algorithm

Minimization of the total cost (1) can be solved by dynamic programming. Viterbi algorithm (VA) provides an efficient tool for searching the minimum cost path through the lattice of candidate units. VA proceeds in stages where each stage comprises of the search of the predecessor candidate resulting in the lowest possible cost for the the current candidate. Only the index of the predecessor and the current minimum cumulative cost for each candidate are stored. The best sequence is the one that has the lowest total cumulative cost.

The problem of VA is that it searches for a sequence with a globally minimum total cost. It is not allowed to ignore the outlier units for which no good candidates are found in the database. Different units e.g. phones occur, however, very unevenly in the database. There is typically a large amount of some phones whereas some of the phones can be very few in number or highly context-dependent. Standard VA can end up giving too much emphasis on rare units.

The effect of outliers can be alleviated using robust Viterbi algorithm (RVA) introduced in [8] for automatic recognition of noise-corrupted speech. Unlike VA, RVA is allowed to skip some of the units during the search and thus prevents single unsuitable candidates from drifting the search. In RVA, all possible subsequencies with up to a predefined number of excluded units are taken into account. Units with a high cost value are likely to get ignored and hence they do not corrupt the rest of the search. The detected poor-quality units can be replaced by better ones. In this paper, HMM-based synthesis is used for replacing.

3.2. Cost minimization using robust Viterbi algorithm

The decision of whether to include a unit or not is based on the costs resulting from excluding a unit and from retaining it. The maximum number of skipped units during the search is K. For each unit candidate, all possible amounts of earlier skips up to K are taken into account. The cost of selecting $u_n^{(j)}$, i.e. the jth candidate of the nth unit, when k units are excluded is the minimum of the costs of excluding the unit and the cost of retaining it. The cost of excluding the unit when $k-1$ units have already been skipped is

$$C^e(u_n^{(j)}, k) = \min_i [C^{tot}(u_{n-1}^{(i)}, k-1)] \tag{5}$$

The cost of excluding the unit is thus the cost of the best preceding candidate unit. The cost of retaining the unit when k units have already been excluded is

$$\begin{aligned} C^r(u_n^{(j)}, k) =& \min_i [C^{tot}(u_{n-1}^{(i)}, k) \\ &+ C^j(u_{n-1}^{(i)}, u_n^{(j)}) + C^t(u_n^{(j)})] \end{aligned} \tag{6}$$

The cost of retaining a unit is the minimum of the sum of the total cost of the best preceding unit, the join cost, and the current target cost. The total cost of a candidate unit is the minimum of the costs of excluding the unit and retaining it

$$C^{tot}(u_n^{(j)}, k) = \min[C^e(u_n^{(j)}, k), C^r(u_n^{(j)}, k)] \quad (7)$$

The minimization is done iteratively for $n = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M_n$, and $k = 0, 1, \ldots, K$, where N is the total length of the unit sequence and M_n is the number of candidate units at time n. For each candidate $u_n^{(j)}$ and number of skipped units k, we also store the information of the best previous unit and the corresponding number of skipped units.

We denote by $\gamma(u_n^{(j)}, k)$ the number of skipped units in the best preceding unit leading to $u_n^{(j)}$ with k skipped units. That is,

$$\gamma(u_n^{(j)}, k) = \begin{cases} k-1, & C^r(u_{n-1}^{(j)}, k) \geq C^e(u_{n-1}^{(j)}, k) \\ k, & \text{otherwise} \end{cases} \quad (8)$$

The index i of the best previous unit leading to $u_n^{(j)}$ is denoted by $\psi(u_n^{(j)}, k)$:

$$\psi(u_n^{(j)}, k) = \arg\min_i C^{tot}(u_n^{(j)}, k) \quad (9)$$

After recursive total cost computation, path backtracking is carried out starting from the minimum cost candidate at $n = N$. By setting $K = 0$, the search is reduced into traditional Viterbi search where no units are excluded. In our approach, excluded units are replaced using the underlying HMM-TTS system. Setting $K = 0$ leads to conventional unit selection procedure with no units excluded and replaced. When $K = N$, all the units are replaced and the synthesis is done using only HMM-TTS.

The use of RVA increases the computational load of the unit sequence selection. In theory, the load of RVA is $K+1$ times the load of VA [8]. However, since the target and join cost values are independent from the number of excluded units, there is no need to compute them again for every possible number of skips. Furthermore, all the sequences with less than K units excluded can be found based on the stored partial path metrics.

4. Evaluations and results

4.1. System implementation

For evaluations, a hybrid system consisting of an HMM-based target prediction and unit selection based-concatenation of parameterized phone-sized units and replacement of poor-quality units was used. For the selection of the best candidate unit sequence, RVA was used. The HMM-TTS system providing the synthesis target and the basis of the cost computation was trained using HMM-based speech synthesis system (HTS) [9] version 2.1. In synthesis, units with poor quality were taken from HMM-based synthesis and concatenated with the selected database unit parameterizations. As a reference, conventional unit selection with VA was used with all units taken from a database. The concatenation boundaries were not smoothed in either case.

For the unit selection procedure, cost functions based on HMM-modeling were used. The target cost was computed as in (3). The join cost (4) was modified to use directly the frame parameterization difference around a unit boundary instead of its distance from a pre-trained concatenation model. The speech features used in the cost computation contained delta-augmented mel-cepstral coefficients (MCCs) and F_0 values. The total cost was defined as in (7), where the robust Viterbi search was used to find the best unit candidate sequence with a certain amount of skips allowed. In the reference system, no skips were allowed in the search.

Evaluations were carried out for two databases, an English and a Finnish one. For English, a publicly available database of a female speaker *slt* in CMU ARCTIC (*http://festvox.org/cmu_arctic/*) collection was used. For Finnish, a prosodically rich male voice database of roughly 650 sentences was used. Labels for the data were aligned automatically with the Viterbi alignment using the HMM models trained for synthesis.

4.2. Speech parameterization and model training

The evaluations involved synthesis using two languages and two parameterizations. For the English voice, STRAIGHT [10] parameterization was used. STRAIGHT encodes the speech waveform into a spectrum and excitation parts. In the extraction of the spectrum part, all the periodic interferences caused by F_0 are removed. The excitation is parameterized into F_0 and aperiodicity of each spectrum component.

To overcome the problem of non-modal voice quality with STRAIGHT [11], an alternative representation was used for the Finnish voice. In this representation, conventional linear prediction (LP) analysis was used for approximating the vocal tract contribution, and the residual signal was represented using sinusoidal modeling. The parameters used in the sinusoidal model include F_0, voicing, energy and the residual amplitude spectrum. The benefit of the representation is two-fold: First, the parameter estimation process can properly treat the problematic low-frequency and vocal fry segments. Second, this parametric representation lends itself to very efficient compression of the unit database.

The speech parameters were further encoded into a form that allowed their use in HMM training and synthesis. The same parameterization was used for both the unit selection and HMM-based prediction and synthesis. The encoded parameterizations for the databases were:

- **English:** STRAIGHT spectral envelope represented as MCCs of order 24, logarithmic STRAIGHT F_0, and mean STRAIGHT aperiodicity of 5 frequency bands.
- **Finnish:** LP-spectrum represented as MCCs of order 24, logarithmic F_0, voicing cut-off frequency, logarithmic energy, and the shape of the residual amplitude spectrum as a discrete cosine transform of length 15.

4.3. Comparison test

Synthesis quality of the described hybrid synthesis system was evaluated by a pairwise comparison test. In the test, 10 sentences for both Finnish and English were synthesized using HMM-based unit selection (a) in a conventional form using VA and (b) as proposed here by using RVA combined with HMM-based synthesis to replace the detected unsuitable units. The evaluation sentences were selected randomly and the amount of poor-quality units varied. For Finnish, 7 native listeners and for English, 6 non-native listeners with good skills in English attended the test.

The listeners were asked to evaluate the quality of the sentence pairs and for each pair, to decide the one of better overall quality including the voice quality, prosodic naturalness, and brightness of the speech. The evaluation sentences were synthe-

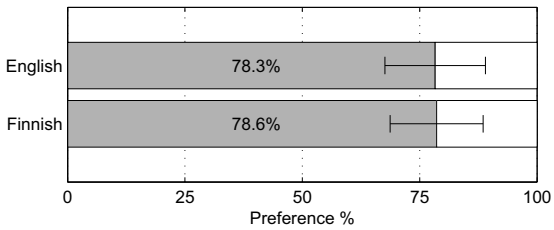

Figure 1: *Preference scores with 95% confidence intervals for HMM-based unit selection using robust Viterbi algorithm and replacing of the detected poor-quality units (gray) compared to the traditional Viterbi algorithm (white).*

sized using the two methods (a) and (b). The number of units replaced in (b) was set to be 20% of the sentence length. Since the join cost was ignored at the boundaries of an excluded unit, also the nearby frames around the unit were replaced.

The results of the comparison test are presented in Figure 1. For both the languages, the proposed approach (b) was preferred over the conventional selection and synthesis (a). For the Finnish voice, the preference percentage for (b) with 95% confidence interval was 78.6% $\pm$ 9.9%. For the English voice, the preference percentage was 78.3% $\pm$ 10.7%. The results indicate that by allowing the unit selection procedure to exclude poor-quality units, better overall quality can be achieved in synthesis. The use of RVA prevents single outliers from drifting the search and from deteriorating the quality of the rest of the sentence. The search of the best unit sequence and the detection of the poor quality units is done simultaneously and unsuitable units can therefore be effectively replaced. In a parameterized form, mixing of different synthesis methodologies can also be done without noticeable differences at switching boundaries. Synthesis samples are available at *http://www.cs.tut.fi/sgn/arg/silen/is2010/robust_viterbi.html*.

5. Discussion

After the robust Viterbi search, all the minimum cost sequences up to the maximum number of skips are available. Automatic selection of the best one out of these is however rather difficult. The total cost of a candidate unit sequence usually gets lower when the number of skips is increased. The total cost as such cannot thus be used as a decision criterion. Modified from [8], the change in the total cost caused by an increase in the number of excluded units could be used as a criterion instead of the cost itself. If ignoring a unit decreases the total cost less than a predefined threshold, no more units are excluded.

In addition to the optimal skip number problem, some of the phones suffer from poor quality in HMM-TTS and thus the trained models for them may not be proper for target cost calculation. Furthermore, in these cases also the speech segment used for replacing the unit is of poor quality. To avoid the problem, for instance detection of difficult segments could be added to prevent the selection algorithm from paying attention to them.

The results in [12] suggest that the mixing of synthetic and real units do not improve voice quality but rather makes it less natural due to the change in the voice quality when switching from unit selection to HMM-based synthesis. However, the usage of real speech without parameterization combined with speech built from parameterized features may cause too much variation in the quality. In the approach proposed in this paper, replacing and concatenating units in a parameterized form enabled the mixing of different synthesis methods without significant changes in voice quality at the boundaries. In addition, even when using speech at a parametric level for unit selection without mixing, some of the concatenation errors are forgiven but naturally at the cost of the quality.

6. Conclusions

In this paper, we have studied the use of robust Viterbi algorithm in the framework of HMM-based unit selection TTS. The robust Viterbi algorithm aims at excluding outliers in the search thus preventing them from ruining the rest of the search. Since the search also points out the ignored units, they can be easily replaced with new ones. In our approach, HMM-TTS was used in synthesizing detected unsuitable segments. The evaluations carried out for English and Finnish data using two different parameterizations indicate that excluding the poor-quality units and replacing them by HMM-based synthesis improves the synthesis quality compared to the conventional selection.

7. Acknowledgements

This work was supported by the Academy of Finland, (application number 129657, Finnish Programme for Centres of Excellence in Research 2006-2011).

8. References

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP*, 1996.

[2] A. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *ICASSP*, 2007.

[3] P. Taylor, "Unifying unit selection and hidden Markov model speech synthesis," in *Interspeech*, 2006.

[4] Z.-H. Ling, L. Qin, H. Lu, Y. Gao, L.-R. Dai, R.-H. Wang, Y. Jiang, Z.-W. Zhao, J.-H. Yang, J. Chen, and G.-P. Hu, "The USTC and iFlytek speech synthesis systems for Blizzard Challenge 2007," in *Blizzard Challenge Workshop*, 2007.

[5] A. Black, C. Bennett, B. Blanchard, K. Kominek, B. Langner, K. Prahallad, and A. Toth, "CMU Blizzard 2007: A hybrid acoustic unit selection system from statistically predicted parameters," in *Blizzard Challenge Workshop*, 2007.

[6] V. Pollet and A. Breen, "Synthesis by generation and concatenation of multiform segments," in *Interspeech*, 2008.

[7] D. Lin, Y. Zhao, F. Soong, M. Chu, and J. Zhao, "Iterative unit selection with unnatural prosody detection," in *Interspeech*, 2007.

[8] M. Siu and A. Chan, "A robust Viterbi algorithm against impulsive noise with application to speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, 2006.

[9] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *6th ISCA Workshop on Speech Synthesis*, 2007.

[10] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[11] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, "Parameterization of vocal fry in HMM-based speech synthesis," in *Interspeech*, 2009.

[12] M. Aylett and C. Pidcock, "The CereProc Blizzard Entry 2009: Some dumb algorithms that don't work," in *Blizzard Challenge Workshop*, 2009.