Admissible Stopping in Viterbi Beam Search for Unit Selection
Speech Synthesis

Shinsuke SAKAI∗ and Tatsuya KAWAHARA†, Members

SUMMARY  Corpus-based concatenative speech synthesis has been widely investigated and deployed in recent years since it provides a highly natural synthesized speech quality. The amount of computation required in the run time, however, can often be quite large. In this paper, we propose early stopping schemes for Viterbi beam search in the unit selection, with which we can stop early in the local Viterbi minimization for each unit as well as in the exploration of candidate units for a given target. It takes advantage of the fact that the space of the acoustic parameters of the database units is fixed and certain lower bounds of the concatenation costs can be precomputed. The proposed method for early stopping is admissible in that it does not change the result of the Viterbi beam search. Experiments using probability-based concatenation costs as well as distance-based costs show that the proposed methods of admissible stopping effectively reduce the amount of computation required in the Viterbi beam search while keeping its result unchanged. Furthermore, the reduction effect of computation turned out to be much larger if the available lower bound for concatenation costs is tighter.

key words:  speech synthesis, unit selection, concatenation cost, Viterbi search

1. Introduction

The corpus-based concatenative approach to speech synthesis by unit selection has been widely explored in the research community in recent years [1]–[7]. In this approach, an optimal sequence of synthesis units with various granularities (e.g. Hidden Markov Model (HMM) state, half-phone, phone, or non-uniform contiguous sequence of them extracted from the corpus) are chosen from a large inventory of units to synthesize speech for the input text through the minimization of the overall cost on the unit sequence. The overall cost is typically modeled as the weighted sum of target costs and concatenation (or join) costs defined on various features of synthesis units such as spectral shape, intonation contour, and segmental duration. The sequence of units to be concatenated to form the output is usually chosen by some sort of Viterbi algorithm with beam pruning where an optimal unit sequence is obtained by accumulated cost minimization based on the dynamic programming principle. The amount of computation, however, is often quite large due to the large size of the unit database that sometimes amounts to more than ten hours of recorded speech. Various techniques have so far been proposed to reduce the amount of run-time computation, such as caching of concatenation costs [8] and segment preselection based on usage statistics [9]. These techniques have been shown to be effective and can be applied together with our methods presented in this paper, which are independent of these techniques.

In this paper, we propose two novel schemes for reducing the amount of computation in the Viterbi beam search for unit selection, by taking advantage of the prior knowledge about the fixed acoustic space of the unit database [10]. Specifically, we use the knowledge of lower bounds of the concatenation costs. One of the two schemes is “admissible stopping” in the local minimization*, with which we can stop early in the local Viterbi minimization over the partial sequences of units up to the previous target position for a new candidate unit retrieved from the database in the current target position. The other scheme is “admissible stopping” for the beam”, in which we can stop early in the exploration of candidate units from the database for the current target.

These “admissible stopping” schemes are named after “admissible heuristic functions” h(n) used in the A* algorithm for graph search [11]. A heuristic function h(n) in graph search is called admissible if it always gives a lower bound of (i.e. the value smaller than or equal to) the true cost of reaching the goal from the current node n. It is also known that the search is usually faster if the heuristic function h(n) is closer to (i.e. the better estimate of) the true cost h*(n). By metaphor with admissible heuristic functions, the proposed early stopping schemes are named “admissible stoppings” since they utilize lower bounds of the concatenation costs and are guaranteed to yield exactly the same results as the ordinary Viterbi beam search as long as the lower bounds of concatenation costs are correctly computed. It is also naturally expected that the search requires less computation if these bounds get tighter i.e. closer to the true minimums of the costs.

In the next section, we describe the basic Viterbi beam search algorithm utilized in the unit selection. In the succeeding two sections, we present the two early stopping schemes, namely, the “admissible stopping in the local minimization” and the “admissible stopping for the beam” with mathematical rationale that allows us to stop in the middle of the procedure without any approximation errors. In Sect. 5, we present experimental results to show that the proposed scheme of admissible stoppings are effective with concatenation costs based on a probabilistic method as well as a distance-based method. We also demonstrate that we have larger reduction of computation if we have a tighter lower

PAPER

Copyright © 2013 The Institute of Electronics, Information and Communication Engineers
bound for the concatenation costs available. The final section presents our conclusions.

2. Unit Selection with Viterbi Beam Search

In this section, we review the basic Viterbi beam search framework for unit selection in concatenative speech synthesis. At the beginning of the unit selection, we are provided with a sequence of $I$ target feature vectors, $t_1, \cdots, t_I$, generated from the text processing module. Each of these feature vectors $t_i$ usually comprises phonetic and prosodic properties, such as phone context, duration, and $F_0$, that we wish the resultant units to have. Given the sequence of target feature vectors $t_1, \cdots, t_I$, we are to find a sequence of waveform fragments, or units, $U = u_1, \cdots, u_I$, that minimizes the total cost $C(U)$. This total cost $C(U)$ is defined as the sum of all target costs over the unit sequence $u_1, \cdots, u_I$ and all concatenation costs over the sequence,

$$
C(U) = \sum_{i=1}^{I} L_t(u_i) + \sum_{i=2}^{I} L_c(u_{i-1}, u_i),
$$

where $L_t(u_i)$ is the local target cost for the unit $u_i$ and $L_c(u_{i-1}, u_i)$ is the local concatenation cost for having the unit $u_i$ after $u_{i-1}$. Minimization of the total cost $C(U)$ is done efficiently by the Viterbi algorithm. Figure 1 is a schematic diagram that depicts a local minimization step in the algorithm. The Viterbi algorithm performs global optimization efficiently by repeating local optimizations. However, when the number of candidate units is very large, the amount of computation can get too large to be practical. Therefore, beam pruning is usually employed and only a limited number of partial sequences of units are kept after local optimizations at each target position to overcome this problem. The basic algorithm of this Viterbi beam search is depicted in Fig. 2.

---

**Basic Viterbi beam search**

(Notation)

- $u_i(k)$: $k$-th database unit for the $i$-th target.
- $K_i$: the number of database units for the $i$-th target.
- $K_0$: the beam width or the number of hypotheses (partial unit sequences) retained at each stage of the iteration.
- $L_t(u)$: the local target cost for the unit $u$.
- $L_c(u_1, u_2)$: the local concatenation cost for having the unit $u_2$ after the unit $u_1$.
- $C^*(u)$: the accumulated cost for the hypothesis ending with the unit $u$.
- $bt(u)$: backtrace information, i.e. the predecessor of the unit $u$ determined by the local minimization.
- $\{u_1, u_2, \ldots\}$: a set of hypotheses each of which is identified by its last (i.e. right-most) unit, $u_1, u_2, \ldots$.

1. **Initialization**

$$
C^*(u_{2}(k)) = L_t(u_{2}(k)) \quad \text{for} \quad k = 1, \cdots, K_1.
$$

Prune the initial set of hypotheses, $\{u_1(1), \cdots, u_1(K_1)\}$, preferring hypotheses with lower costs to keep at most $K_0$ units.

2. **Iteration**

Repeat the following for the target indices $i = 2, \cdots, I$:

For all the unit indices $k = 1, \cdots, K_i$ for the $i$-th target:

$$
C^*(u_i(k)) = \min_j \{C^*(u_{i-1}(j)) + L_t(u_{i-1}(j), u_i(k)) + L_c(u_i(k))
$$

$$
bt(u_i(k)) = \arg\min_j [C^*(u_{i-1}(j)) + L_t(u_{i-1}(j), u_i(k))]
$$

Prune the new set of hypotheses up to the $i$-th target position, $\{u_{i}(1), \cdots, u_{i}(K_i)\}$, to keep at most $K_0$ hypotheses preferring hypotheses with lower values of $C^*(u_i(k))$.

3. **Termination**

$$
u^*_i = \arg\min_k C^*(u_i(k))
$$

Starting from $u^*_I$, backtrace $bt(u^*_I)$ recursively, and retrieve the $u_i(k)$’s for $i = 1, \cdots, I - 1$ that lead to $u^*_I$.

---

**Fig. 1** A schematic diagram that depicts local minimization in the Viterbi algorithm. A gray rectangle labeled $t_i$ stands for the $i$-th target. Dark rounded-corner rectangles labeled $k = 1, \cdots, k = K_i$ are candidate units for the $i$-th target shown above them.

**Fig. 2** Basic Viterbi beam search for the unit selection.
error, if the cumulative cost up to the last target position, $C^*(\tilde{u}_{i-1}(j_0))$, is large enough such that

$$C^*(\tilde{u}_{i-1}(j_0)) + \text{lbound} \left[ L_c(\tilde{u}_{i-1}(j), u_i(k)) \right]$$

$$> \min_{j' < j_0} \left[ C^*(\tilde{u}_{i-1}(j')) + L_c(\tilde{u}_{i-1}(j'), u_i(k)) \right],$$

where “lbound” stands for a lower bound of $L_c(\tilde{u}_{i-1}(j), u_i(k))$ for all possible values of $j$. This lower bound can be given by a table of the minimums of the concatenation costs of the database units for all phone bigram contexts, for example. To justify this stopping condition, we first note that

$$L_c(\tilde{u}_{i-1}(j), u_i(k)) \geq \text{lbound} \left[ L_c(\tilde{u}_{i-1}(j), u_i(k)) \right]$$

holds for any $j$, as the property of a lower bound. Since the list of hypotheses up to the $(i-1)$-th target position is sorted in the ascending order of cumulative costs, it holds that

$$C^*(\tilde{u}_{i-1}(j)) \geq C^*(\tilde{u}_{i-1}(j_0))$$

for all $j$ such that $j > j_0$. Therefore, summing up (4) and (5), we note its relationship with the stopping condition (3),

$$C^*(\tilde{u}_{i-1}(j)) + L_c(\tilde{u}_{i-1}(j), u_i(k))$$

$$\geq C^*(\tilde{u}_{i-1}(j_0)) + \text{lbound} \left[ L_c(\tilde{u}_{i-1}(j), u_i(k)) \right]$$

$$> \min_{j' < j_0} \left[ C^*(\tilde{u}_{i-1}(j')) + L_c(\tilde{u}_{i-1}(j'), u_i(k)) \right],$$

for all $j$ such that $j > j_0$. This means that once the condition (3) holds, the sum of the previous cumulative cost and the concatenation cost will never get smaller than the current minimum and therefore the minimization is over at this point. The run-time concatenation cost computation can thus be avoided for $j > j_0$.

4. Admissible Stopping for the Beam

In the previous section, we presented an early stopping scheme in the local minimization loop. Now we look at (possibly an enormous number of) candidate units coming from the unit database at the stage for the $i$-th target. Suppose we have $K_i$ candidate units, $u_i(1), \cdots, u_i(K_i)$, retrieved from the unit database. Before we perform the beam pruning to retain just $K_o$ new hypotheses, we need to perform the local minimization (described in the previous section) for each of these units. This may be inefficient if $K_i$ is considerably larger than $K_o$, for example, $K_i = 2,000$ and $K_o = 200$. We can speed up the search if we can stop in the middle of examining all candidate units $u_i(1), \cdots, u_i(K_i)$ for local Viterbi minimization at the $i$-th target position.

Toward this objective, we make use of the prior knowledge of the lower bounds of concatenation costs once again. We sort the set of candidate units retrieved from the unit database for the $i$-th target in an ascending order of the local target cost $L_c(\cdot)$ and keep them in the ordered list $[u_i(1), \cdots, u_i(K_i)]$. We also keep newly created hypotheses, i.e., candidate units so far concatenated with past partial unit
sequences, in the ordered list \((\tilde{u}_i(1), \cdots, \tilde{u}_i(k))\) in an ascending order of new cumulative costs, \(C^*(\tilde{u}_i(1)), \cdots, C^*(\tilde{u}_i(k))\), after local minimizations are done up to the \(k\)-th candidate unit. Note that the square brackets “[” and “]” are used to denote an ordered list of units and the angle brackets “(” and “)“ are used for an ordered list of hypotheses. We denote units in the sorted hypothesis list as \(\tilde{u}_i(k)\) to distinguish them from those in the sorted unit list denoted \(u_i(k)\), since \(k\)-th elements of these two lists do not necessarily refer to the same unit. As we can see in Fig. 4, after we have explored \(K_0\) units in the \(i\)-th stage, we can stop if the target cost for some \(K_{0,i}\)-th unit \(L_{C^*}(\tilde{u}_i(K_0))\) is large enough such that

\[
\min_{j} C^*(\tilde{u}_{i-1}(j)) + \text{lbound}[L_{C^*}(\tilde{u}_{i-1}(j), u_i(k))] + L_{C^*}(u_i(k)) > C^*(\tilde{u}_i(K_0)).
\]

This lower bound can also be given by a table of the minimums of the concatenation costs, as in the last section. To see the validity of this condition, we first note that \(L_{C^*}(u_i(k))\) is larger than or equal to \(L_{C^*}(u_i(K_0))\) for all \(k\) such that \(k > K_0\). Then, if the inequality (7) holds for some \(K_0\), we have

\[
C^*(u_i(k)) = \min_{j} \{ C^*(\tilde{u}_{i-1}(j)) + \text{lbound}[L_{C^*}(\tilde{u}_{i-1}(j), u_i(k))] + L_{C^*}(u_i(k)) \}.
\]

1. **Initialization**

Hypotheses (partial unit sequences) up to the \((i-1)\)-th stage are listed in an ascending order of cumulative cost \(C^*(\tilde{u}_{i-1}(j))\).

Set \(j_{\text{min}} = \text{none} \) and \(\text{cost}_{\text{min}} = \infty\).

2. **Iteration**

Starting from \(j = 1\), repeat the following for \(j = 1, \cdots, K_{i-1}\) until \(C^*(\tilde{u}_{i-1}(j))\) is large enough such that

\[
C^*(\tilde{u}_{i-1}(j)) + \text{lbound}[L_{C^*}(\tilde{u}_{i-1}(j'), u_i(k))] > \text{cost}_{\text{min}}:
\]

if \(C^*(u_{i-1}(j)) + L_{C^*}(u_{i-1}(j), u_i(k)) < \text{cost}_{\text{min}}\), then

\[
\text{cost}_{\text{min}} = C^*(u_{i-1}(j)) + L_{C^*}(u_{i-1}(j), u_i(k)),
\]

and \(j_{\text{min}} = j\).

3. **Termination**

\[
C^*(u_i(k)) = \text{cost}_{\text{min}} + L_{C^*}(u_i(k))
\]

\[
\text{bt}(u_i(k)) = u_{i-1}(j_{\text{min}})
\]

with no approximation error.

The modified Viterbi beam search algorithm that incorporates the two admissible stopping schemes described in this section and the previous section is depicted in Figs. 5 and 6.

5. **Experiments and Results**

We implemented the two admissible stopping methods presented in the previous sections in a concatenative speech synthesis system [12], [13]. Synthesis units are uniformly phone-sized. The unit database was developed using the speaker SLT of the CMU Arctic speech databases [14]. It is spoken by a female speaker of American English and consists of 1,132 utterances. The total duration is roughly 50 minutes. The target and concatenation models were all trained using this database.

The total target cost for each unit is a sum of spectral, duration, and \(F_0\) target costs which are negatives of the log probabilities coming from the probabilistic target models [13], [15]. As the costs of concatenating synthesis units, we have developed a probabilistic concatenation model based on conditional Gaussian densities [12]. In order to demonstrate the wide applicability of the proposed method, we also implemented a distance-based concatenation model using the Euclidean distance, which is more widely adopted in the speech synthesis community [16]. These two schemes both compute the concatenation cost based on the near-boundary spectral features of the two units to be concatenated. The spectral feature parameters used in the target and concatenation models were both \(8-\)
Viterbi beam search with admissible stoppings

(Notation)

(The definitions of \( u_t(k), K_I, K_\theta, L_\nu(u), L_c(u_1, u_2), C^*(u) \), and \( \hat{b}(u) \) are the same as Fig. 2.)

\[
[u_1, u_2, \ldots] : \text{an ordered list of units.}
\]

\[
\langle u_1, u_2, \ldots \rangle : \text{an ordered list of hypotheses each of which is identified by its last unit, } u_1, u_2, \ldots.
\]

1. Initialization

Retrieve the set of units for the first target from the unit database. Sort them in an ascending order of the target cost, yielding a sorted list of units \([u_1(1), \cdots, u_1(K_1)]\).

Set \( C^*(u_t(k)) = L_c(u_t(k)) \) for \( k = 1, \cdots, K_1 \).

Prune the initial hypothesis list \( \langle u_1(1), \cdots, u_1(K_1) \rangle \), preferring hypotheses with smaller costs and keep at most \( K_\theta \) units.

2. Iteration

Repeat the following for the target indices \( i = 2, \cdots, I \):

Retrieve the set of units for the \( i \)-th target from the unit database and sort them in an ascending order of the target cost, yielding a sorted list of units \([u_i(1), \cdots, u_i(K_i)]\).

Starting from \( k = 1 \), repeat the local minimization procedure shown in Fig. 5, keeping the new hypotheses in the list \( \langle \tilde{u}_i(1), \cdots, \tilde{u}_i(k) \rangle \) sorted in ascending order of the accumulated costs just calculated, for unit indices \( k = 1, \cdots, K_i \).

Stop, however, if \( k > K_\theta \) and the inequality

\[
\min_j C^*(\tilde{u}_{i-1}(j)) + \text{lbound}[L_c(\tilde{u}_{i-1}(j), u_i(k'))] + L_c(u_i(k)) > C^*(\tilde{u}_i(K_i))
\]

holds.

Prune the list of new hypotheses up to the \( i \)-th target, \( \langle \tilde{u}_i(1), \tilde{u}_i(2), \cdots \rangle \) to keep at most \( K_\theta \) units.

3. Termination

(The same as “3. Termination” in Fig. 2.)

dimensional feature vectors obtained by principal component analysis on 14 MFCC coefficients. For modeling of \( F_0 \) and duration targets, fundamental frequencies and durations in seconds were directly used without any transformations.

In the experiments using the Euclidean distance, we adopted two kinds of lower bounds for the concatenation costs that are different in their tightness. We also employed the popular heuristics of assigning zero cost when the units concatenated were adjacent in the original corpus in order to see whether the proposed method is effective as well with this heuristics applied. The lower bounds of the concatenation costs were precomputed for all the phone pair contexts. In the current implementation using 50 phones, these lower bounds are stored in a table with 50 \( \times \) 50 entries. Ten conversational sentences extracted from the Blizzard Challenge 2005 test set were used as input text in the speech synthesis tests reported in the following subsections.

A. Results with conditional Gaussian models

The concatenation cost of having unit \( v \) just after \( u \) based on the conditional Gaussian models is defined as

\[
L_c(u, v) = - \log \mathcal{N}(h(v) | B \ t(u) + b, \ \Sigma), \tag{9}
\]

where \( t(u) \) and \( h(v) \) indicate near-boundary feature vectors of the units \( u \) and \( v \), respectively. The conditional Gaussian model parameters \( B, b, \) and \( \Sigma \) are determined by the current phonetic context for the units [17].

We first look at how admissible stopping for the beam presented in Sect. 4 is effective on its own. Table 1 shows the average number of units retrieved from the unit database per target while synthesizing the test utterances (column 2) and the number of units actually examined for concatenation before the early termination by admissible stopping for the beam (column 3). The actual number is also plotted in Fig. 7. From the table and the figure, we see that the number

| beam width | # units examined (%) |
|------------|----------------------|
| 2000       | 1268 (98.59)         |
| 600        | 1028 (79.94)         |
| 200        | 741 (57.64)          |
| 60         | 501 (39.02)          |

Fig. 7 The average number (left axis) of the database units examined for concatenation per target during beam search for four beam widths with conditional Gaussian concatenation models. The right axis represent their proportions to the number of all the units retrieved from the database. The graph visualizes Table 1.
of units examined for concatenation was effectively reduced by admissible stopping for the beam. Naturally, its effect gets larger when the beam width gets smaller, which is expected from Fig. 4. (In Fig. 4, we see that the vertical broken line showing the cumulative cost at $K_0$ should move toward left when the beam width $K_0$ gets smaller, thus making the admissible stopping occur earlier.)

The reduction of the number of concatenation cost computations achieved by two admissible stopping schemes applied independently and together is summarized in Fig. 8. In the figure, we see that the number of concatenation cost computations is effectively reduced by the two admissible stopping schemes. The right-most bars (in orange color) of Fig. 8 shows that the number of concatenation cost computations is closer to the true minimum. When the Euclidean distance is employed, the concatenation cost is defined to be

$$L_c(u, v) = \| h(v) - t(u) \|,$$

where $t(u)$ and $h(v)$ indicate near-boundary spectral feature vectors of the units $u$ and $v$, respectively.

Figure 9 shows the average number of units actually examined out of all units retrieved from the database per target before the early termination by the admissible stopping for the beam. In Fig. 9 and succeeding figures, $\text{euc (corpus)}$ represents the results with corpus-based lower bounds and $\text{euc (zero)}$ represents the results with lower bounds set to all zero. The results using lower bounds all zero as well as zero cost heuristics for corpus adjacency is represented as $\text{euc (zero+a)}$. From the figure, we see that the admissible stopping for the beam is also and further effective with the Euclidean distance. As seen with the conditional Gaussian models, we see that the effect gets larger when the beam width gets narrower. For example, only around 10% of the units retrieved from the database were examined for concatenation when the beam width is 60. By comparing the results for $\text{euc (corpus)}$ and $\text{euc (zero)}$ in Fig. 9, we also note that the number of units examined is smaller with $\text{euc (corpus)}$ than $\text{euc (zero)}$, which indicates that a greater reduction effect is achieved when the lower bounds for concatenation cost is closer to the true minimum.

Figure 10 summarizes the reduction effects on the number of concatenation cost computations when the both of the two admissible stopping schemes are applied with the Euclidean distance. From Fig. 10, we see that the use of admissible stoppings effectively reduces the number of concatenation cost computations as well when the Euclidean distance.
distance is employed for concatenation cost. In fact, we notice that the reduction rate is much larger with the Euclidean distance than the conditional Gaussian when we compare Fig. 8 and Fig. 10. To understand this result, we looked at the numerical values of concatenation costs appearing in the search experiments and we found out that the dynamic range of the concatenation costs by the Euclidean distance is much smaller than the costs given by conditional Gaussian-based concatenation models. This much smaller dynamic range leads to much tighter lower bounds and stops the local Viterbi minimization much earlier.

Comparing the results for \texttt{euc (corpus)} and \texttt{euc (zero)} in Fig. 10, we also see that the reduction effect on the number of concatenation cost computations is greater with \texttt{euc (corpus)} than \texttt{euc (zero)}. Therefore, we again confirm that the reduction effect gets greater due to earlier occurrences of admissible stoppings when the lower bounds are closer to the true minimum, which is expected when we note the widths of \textit{lbound} $L_c$ in Figs. 3 and 4.

From the entries for \texttt{euc (zero+a)} in Fig. 9 and Fig. 10, we see that the admissible stopping is still effective when the popular heuristics of assigning the concatenation cost zero to units that happen to be adjacent in the corpus. Comparing \texttt{euc (zero)} and \texttt{euc (zero+a)} in Fig. 10, we see that the number of concatenation cost computations is a little further reduced with \texttt{euc (zero+a)}, i.e. when the zero cost heuristics is employed. This is because zero cost has the effect of making the right-hand side of the inequality (3) smaller, which, in turn, leads to an earlier termination of the local Viterbi minimization.

C. Time measurements

In order to demonstrate the contribution of admissible stopping to the actual processing speed improvement, we measured the elapsed time spent for unit selection. The machine is equipped with Intel Core2 Extreme (Q6850) with 3.0 GHz clock and 8 GB of memory. The operating system is Red Hat Enterprise Linux release 5. The average time required for unit selection for an utterance is depicted in Fig. 11 for conditional Gaussian-based concatenation costs with four possible combinations of admissible stoppings and in Fig. 12 for Euclidean distance-based concatenation costs with both of two admissible stoppings applied. The average length of a synthesized utterance varied between 2.46 seconds and 2.53 seconds depending on the model and beam conditions. Observing the similarity of Fig. 11 and Fig. 8, we see that the reduction in the number of concatenation cost computations leads to the almost proportional reduction of unit selection time with conditional Gaussian concatenation models. Therefore, the additional overhead time
for sorting the hypotheses and units required in admissible stoppings were negligible compared to the reduction effect of processing time for concatenation cost computations.

When we compare Fig. 10 and Fig. 12, we also see the similar trends between the number of concatenation cost computations and the elapsed time for unit selection. Therefore, the proposed method works effectively with the Euclidean distance as well. Since the computation time for concatenation cost with the Euclidean distance is roughly the half of conditional Gaussian, the overhead time for sorting is relatively larger than when we adopt the conditional Gaussian model. Thus, we see a little slowing effect by the overhead of sorting with the Euclidean distance.

By the way, the absolute value of the elapsed time is roughly 20 times larger with conditional Gaussian (Fig. 11, b:Y 1:Y) than with Euclidean distance (Fig. 12). This is because reduction rate of the number of concatenation cost computation is one order of magnitude larger with Euclidean distance as discussed in the last subsection and the time for concatenation cost computation itself is roughly the half with Euclidean distance.

Overall, we see that the actual elapsed time for unit selection is indeed reduced, for example, to below 40% for the conditional Gaussian models and to below 12% for the Euclidean distance, respectively.

6. Conclusion

In this paper, we proposed two methods of admissible stoppings for the Viterbi beam search in unit selection for concatenative speech synthesis systems that reduce computation without changing the search result. One is the admissible stopping in the local minimization, which can terminate the computation over the list of hypotheses in the middle. The other is the admissible stopping for the beam, which makes it possible to avoid examining the database units with large target costs for concatenation without introducing any approximation error. The experimental results have shown that both of the admissible stopping methods effectively speed up unit selection by reducing the number of concatenation cost computations with concatenation modeling based on conditional Gaussian models as well as the Euclidean distance. The whole unit selection time was reduced to 30–40% with the conditional Gaussian models and to 3–12% with the Euclidean distance.

References

[1] N. Iwahashi, N. Kaiki, and Y. Sagisaka, “Speech segment selection for concatenative synthesis based on spectral distortion minimization,” IEICE Trans. Fundamentals, vol.E76-A, no.11, pp.1942–1948, Nov. 1993.
[2] A. Hunt and A. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” Proc. ICASSP ’96, pp.373–376, 1996.
[3] E. Eide, A. Aaron, R. Bukis, P. Cohen, R. Donovan, W. Hamza, T. Mathes, M. Picheny, M. Polkosky, M. Smith, and M. Viswanathan, “Recent improvements to the IBM trainable speech synthesis system,” Proc. ICASSP 2003, pp.1–708–I–711, 2003.
[4] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, “Microsoft Mulan – A bilingual TTS system,” Proc. ICASSP 2003, pp.I–264–I–267, 2003.
[5] H. Kawai, T. Toda, J. Ni, M. Tsuzaki, and K. Tokuda, “Ximera: A new TTS from ATR based on corpus-based technologies,” Proc. ISCA 5th Speech Synthesis Workshop, pp.179–184, 2004.
[6] Z.H. Ling and R.H. Wang, “HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion,” Proc. ICASSP-07, pp.1245–1248, Honolulu, April 2007.
[7] J.Z. Yan, Y. Qian, and F.K. Soong, “Rich-context unit selection (RUS) approach to high quality TTS,” Proc. ICASSP 2010, 2010.
[8] M. Beutragel, M. Mohri, and M. Riley, “Rapid unit selection from a large speech corpus for concatenative speech synthesis,” Proc. EUROSPEECH ’99, pp.607–610, 1999.
[9] W. Hamza and R. Donovan, “Data-driven segment preselection in the IBM trainable speech synthesis system,” Proc. ICISLP 2002, pp.2609–2612, Denver, 2002.
[10] S. Sakai, T. Kawahara, and S. Nakamura, “Admissible stopping in viterbi beam search for unit selection in concatenative speech synthesis,” ICASSP-2008, pp.4613–4616, 2008.
[11] N.J. Nilsson, Principles of Artificial Intelligence, Springer, 1982.
[12] S. Sakai and T. Kawahara, “Decision tree-based training of probabilistic concatenation models for corpus-based speech synthesis,” Proc. Interspeech 2006, Pittsburgh, PA, Sept. 2006.
[13] S. Sakai and H. Shiu, “A probabilistic approach to unit selection for corpus-based speech synthesis,” Proc. Interspeech 2005, pp.81–84, Lisbon, Portugal, Sept. 2005.
[14] J. Kominnek and A. Black, “The CMU ARCTIC speech databases for speech synthesis research,” Tech. Rep. CMU/LTI-03-177, Language Technologies Institute, CMU, 2003.
[15] S. Sakai, “Fundamental frequency modeling for speech synthesis based on a statistical learning technique,” IEICE Trans. Inf. & Syst., vol.E88-D, no.3, pp.489–495, March 2005.
[16] Y. Stylianou and A.K. Syrdal, “Perceptual and objective detection of discontinuities in concatenative speech synthesis,” Proc. ICASSP 2001, Salt Lake City, USA, 2001.
[17] S. Sakai, T. Kawahara, and H. Kawai, “Probabilistic concatenation modeling for corpus-based speech synthesis,” IEICE Trans. Inf. & Syst., vol.E94-D, no.10, pp.2006–2014, Oct. 2011.

Shinsuke Sakai received his B.E. in 1982, and M.E. in 1984, and Ph.D. in 2012 from Kyoto University, Kyoto, Japan. He joined NEC Corp. in 1984, where he worked on various aspects of speech and natural language processing for machine translation and speech recognition. Between 1991–1993, he was a visiting scientist at Massachusetts Institute of Technology, Cambridge, MA, U.S.A. Between 2002–2004, he was with the Laboratory for Computer Science and Computer Science and Artificial Intelligence Laboratory at MIT, where he worked on speech synthesis. He has been affiliated with the Graduate School of Informatics at Kyoto University from 2005 to 2008. He worked as a senior researcher at ATR Spoken Language Communication Laboratories in Kyoto, Japan between 2006–2009 and as an expert researcher at Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology between 2009–2011. Currently, he is affiliated with Academic Center for Computing and Media Studies, Kyoto University. His research interests include exploring speech technology that can be used in various daily life situations, such as in the office, living room and on the street. He is a member of the Information Processing Society of Japan, the Acoustical Society of Japan, and the IEEE.
Tatsuya Kawahara received B.E. in 1987, M.E. in 1989, and Ph.D. in 1995, all in information science, from Kyoto University, Kyoto, Japan. In 1990, he became a Research Associate in the Department of Information Science, Kyoto University. From 1995 to 1996, he was a Visiting Researcher at Bell Laboratories, Murray Hill, NJ, USA. Currently, he is a Professor in the Academic Center for Computing and Media Studies and an Affiliated Professor in the School of Informatics, Kyoto University. He has also been an Invited Researcher at ATR and NICT. He has published more than 250 technical papers on speech recognition, spoken language processing, and spoken dialogue systems. He has been conducting several speech-related projects in Japan including free large vocabulary continuous speech recognition software (http://julius.sourceforge.jp/) and the automatic transcription system for the Japanese Parliament (Diet). Dr. Kawahara received the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology (MEXT) in 2012. From 2003 to 2006, he was a member of IEEE SPS Speech Technical Committee. From 2011, he is a secretary of IEEE SPS Japan Chapter. He was a general chair of IEEE Automatic Speech Recognition & Understanding workshop (ASRU 2007). He also served as a Tutorial Chair of INTERSPEECH 2010 and a Local Arrangement Chair of IEEE ICASSP 2012. He is an editorial board member of Elsevier Journal of Computer Speech and Language, ACM Transactions on Speech and Language Processing, and APSIPA Transactions on Signal and Information. He is a senior member of IEEE.