

# Low-Latency Incremental Speech Transcription in the Synface Project

Alexander Seward

Centre for Speech Technology  
KTH – Royal Institute of Technology, Stockholm, Sweden  
alec@speech.kth.se

## Abstract

In this paper, a real-time decoder for low-latency online speech transcription is presented. The system was developed within the Synface project, which aims to improve the possibilities for hard of hearing people to use conventional telephony by providing speech-synchronized multimodal feedback. This paper addresses the specific issues related to HMM-based incremental phone classification with real-time constraints. The decoding algorithm described in this work enables a trade-off to be made between improved recognition accuracy and reduced latency. By accepting a longer latency per output increment, more time can be ascribed to hypothesis look-ahead and by that improve classification accuracy. Experiments performed on the Swedish SpeechDat database show that it is possible to generate the same classification as is produced by non-incremental decoding using HTK, by adopting a latency of approx. 150 ms or more.

## 1. Introduction

The lack of visual information and ability to utilize lip-reading make it especially hard for hearing-impaired persons to communicate using ordinary telephony. Synface [1] is a European project under the IST programme that aims to improve the possibilities for hard of hearing people to use conventional telephony by providing multimodal feedback. The participating partners are IvD from the Netherlands, UCL and RNID from the UK, and KTH and Babel-Infovox from Sweden. The Synface is based on earlier experiences gained in the Swedish Teleface project [2]. In this project a prototype of a telephone communication aid was developed. This offline system controlled a parametrical synthetic 3-dimensional face [3] that articulated in synchrony with the speech signal received over the telephone. Experiments have shown that this visual feedback and lip-reading support enable substantial improvements in intelligibility [2]. This paper presents the real-time version of the Synface decoder. In the transition to a real-time online system, several specific issues had to be considered. A primary goal was to minimize the amount of latency induced by the decoder, and to make the remaining latency explicit, in order to obtain a better synchronization between the synthetic face and the speech signal. Since decoder output is incrementally transformed to visemes and instantly manifested visually, emitted output is immutable. This fact implies an additional difficulty to the problem, which also will be addressed in this paper.

### 1.1. Latency factors

Latency is defined here as the period that elapses from the point of capturing the speech signal until output is emitted by

the decoder and visually manifested. There are several factors that are potential sources of latency in a recognition system, all of which have to be considered in order to minimize the overall latency of the system. (1)*Frame latency*. Since the front-end, which handles audio handling and feature extraction, represents the audio samples as frames, there is an associated latency equal to the frame length. In general however, this is short (typically 10ms). (2)*Regression Analysis*; Augmentations of feature vectors with derivatives, such as delta and acceleration coefficients are potential sources of latency as these could involve succeeding frames in order to be computed. However, this can be approximated by applying derivative functions that only access preceding frames. (3)*Computational overhead*. High performance decoding algorithms are of course crucial in building a low-latency online decoder. One purpose of this work has been to minimize the computational overhead. (4)*Classification latency*. A HMM-based decoder usually maintains a large number of concurrent hypotheses in the search for the final solution. If the decoder must generate immutable output incrementally, there is a potential risk of promising hypotheses producing output that later prove to be incorrect. A potential solution, which is discussed later, is to use a short time-window for *hypothesis look-ahead*, during which the decoder can assess additional observations.

## 2. Decoder

### 2.1. Overview

The acoustic models consist of triphone models modeled by continuous-density Hidden Markov Models (HMMs) with multivariate Gaussian mixtures using Mel Frequency Cepstral Coefficients (MFCC) with derivatives. The acoustic models are used in conjunction with an optional context-dependent phone grammar (n-gram). The decoder employs a composite probabilistic model that integrates the acoustic model in the form of n-phone HMMs with an optional context-dependent phone grammar. This is performed by composition of these models to a cyclic weighted automaton for which each transition is assigned a tuple consisting of a weight and an output symbol, which can be a phone symbol or null. The states are associated with probability density functions (Gaussian mixtures) from the acoustic-phonetic HMMs.

In order to solve this decoding problem, a time-synchronous decoding algorithm is required, capable of emitting phone output in frame increments with minimal latency.

### 2.2. Classification latency

Generation of incremental output adds an additional dimension to the problem, if the output is immutable, as mentioned in the previous section. In conventional HMM-

based decoding, where output is generated on an utterance basis, the problem definition is relatively simple. By adoption of the Viterbi assumption as an optimality criterion, there exists a best-path solution. This solution is the sequence that corresponds to the hypothesis with MAP (maximum a posteriori) probability for the input signal given a probabilistic model of acoustic HMMs and a priori weighted models, e.g. lexical and grammatical models. However, in the case of incremental output it is not sufficient to periodically write the most promising partial hypothesis to an output tape. The reason is that a hypothesis, which at a time point during the decoding process has the highest accumulated likelihood, very well can prove to be incorrect, according to the probabilistic model, when additional input frames have been observed. This occurs when another hypothesis, which corresponds to a different path in the search space, as well as a different symbol sequence, takes the role of the currently best hypothesis. In order to solve this, the incorrectly emitted sequence must be corrected. In such speech-to-text systems e.g. [4, 5], updates of output are made continuously and words are substituted by one or several new, more probable words. Such a decoding strategy enables convergence of the output sequence obtained by an offline, non-incremental decoder. In this case however, correction of output is not an option. Since output results are not emitted as a growing text sequence, but instead directly translates to parametrical facial movements, the output is immutable. Consequently, special handling must be applied in order to determine when and how new more likely hypotheses should be allowed to override the present hypothesis and alter the movements of the face.



Figure 1: The synthetic parametrically controlled face.

Let us clarify this with a simple example. Consider Figure 2 of a (very small) HMM net. This net consists of three HMMs, labeled A, B and C, with three states each. The optimal HMM sequence for any input (i.e. the one with MAP probability) can be computed using Dynamic Programming. Under the Viterbi assumption this implies a simple shortest-path search for the single state sequence with the highest accumulated probability. However, if output must be emitted incrementally and immutable the situation suddenly becomes more intricate. Assume that the hypothesis with the highest accumulated probability, at a certain time  $t$ , is the state sequence  $A_1, A_1, A_2, A_3, A_3$ . Consequently it is reasonable to claim that A has been observed. At time  $t+1$ , where yet another input vector has been observed, the path  $A_1, A_2, A_3, B_1, B_2, B_3$  is instead the currently best state sequence. Yet another frame later, at  $t+2$ , the best path is  $A_1, A_1, A_2, A_3, C_1, C_2, C_2$ . How should this be interpreted and what output should be emitted? If the decoding algorithm blindly regards the best hypothesis at each time frame, the result would be an output of A at time  $t$ . At the next frame ( $t+1$ ) the total sequence corresponds to AB. If we assume the emitted A to be correct (although the hypothesized duration of A has shrunk from 5 to 3 frames), we can output the remaining B. At time  $t+2$ , the hypothesis

AC is the best hypothesis. Should we assume that the previously emitted B was incorrect and instead should have been C? In that case, should the talking face be corrected towards a C? This would mean that we would have emitted the sequence ABC, i.e. recognized 3 phones with a minimal duration of 3 frames each, using an observation sequence of only 7 frames. Clearly, these insertion errors would not have been introduced by a non-incremental decoder without real-time requirements for output increments.

### 2.2.1. Solutions

A possible strategy to stabilize the output sequence and minimize insertion errors is to apply a phone insertion penalty. Such a procedure would induce inertia into the system, and reduce volatility among currently best-ranked hypotheses, with a more stable output as a result. However, there are two problems with this approach. Firstly, it implies an alteration of the probabilistic model, only to cope with the incremental case. Results would no longer be directly comparable with the results of non-incremental decoding (for which model alteration is not needed). Secondly, modeling of inertia creates a latency that is very hard to quantify, making it unsuitable for this specific task.

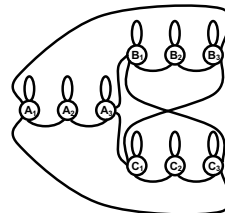


Figure 2: A weighted cyclic HMM transition net.

### 2.3. Hypothesis look-ahead

The strategy adopted in this work, is to retain the initial probabilistic model and yet try to obtain the same results despite the output being emitted incrementally. This is achieved by accepting a certain degree of classification latency, but by making this latency explicit in the decoder. In addition, the proposed decoding strategy allows the classification latency to be set dynamically. The algorithm utilizes this period of time for hypothesis look-ahead during decoding. Generation of output is continuously based on the symbol sequence that the currently best hypothesis corresponded to  $H$  frames earlier, where  $H$  is the hypothesis look-ahead in number of frames. The effect is that a best path that proves to be incorrect within  $H$  frames will not be manifested as errors in the output sequence. The more time that is assigned to hypothesis look-ahead  $H$ , the better the accuracy, but at the cost of a classification latency equal to the hypothesis look-ahead. If the hypothesis look-ahead is sufficiently long, the incremental decoder will give exactly the same output as an offline decoder. By adjusting the hypothesis look-ahead, an optimal balance between accuracy and latency can be obtained.

### 2.4. Decoder structure

As previously mentioned, the primary structure of the decoder consists of a weighted cyclic finite-state automaton that integrates all provided probabilistic models. Clearly, the number of states in this finite-state machine is equal to the total number of endpoints of all possible hypotheses, i.e. all possible state sequences. Each state also has the Markov

property, i.e. future paths depend only on the current state and not on the states in the past. In such a cyclic weighted automaton with  $n$  states there can only exist  $n$  concurrent partial paths that are interesting for breadth-first expansion. This can be handled by several decoding algorithms that can be implemented in a single pass e.g. Token Passing [6]. Due to the analogy with the Token Passing model, in terms of information being propagated over a weighted transition net, the forthcoming discussion will use the notion of tokens to denote placeholders of scored hypotheses. However, in addition to the incremental aspects of hypothesis look-ahead, there are some additional conceptual differences. The time-synchronous decoding algorithm used here must be indifferent to utterance length, since utterances are to be regarded as virtually infinite. Consequently, the decoder should occupy a fixed amount of resources, regardless of whether the input sequence is one second or one hour long. More precisely, creation and extension of hypotheses should be performed without increasing the memory footprint. This is also important with respect to the time-synchronization aspect, especially since the rate of output on phone basis is higher than on word basis. In order for the decoder to implement hypothesis look-ahead, the history of hypotheses must be kept. More specifically, it must be possible to access the specific symbol associated with each hypothesis up to  $H$  frames backward in time, where  $H$  is the hypothesis look-ahead. The fact that information on the actual state sequence is not needed, and that symbolic output only is emitted between HMMs, which cannot occur at a higher rate than what is specified by the topology of the HMMs, a very compact representation of hypothesis history can be formulated.

#### 2.4.1. History representation

Every state in the cyclic weighted automaton is a priori augmented with a token consisting of a placeholder of an accumulated score and a circular list implemented as a bounded array of equal length for all tokens. Each element of this list is a tuple that contains an HMM-label (a pointer) and a corresponding duration (an integer). Such a token represents a compact description of the currently best path to the specific state and its associated accumulated log-probability. The last updated row element in the circular list contains the label of the current HMM of the path, and the time it has been in this HMM. Previous elements, in a circular fashion, contain the labels of previous HMMs and the corresponding durations. By a priori deciding the maximum length of the hypothesis look-ahead, the length of the circular lists in all tokens/states can also be determined on the basis of the HMM topology. Let  $H_{\max}$  be the maximum hypothesis look-ahead we want to be able to set during decoding, and let  $d_{\min}$  be the minimum duration of each HMM. In order to ensure ability of the decoder to backtrack  $H_{\max}$  time units back in time, the lists should be  $N$  elements long, such that:

$$N \geq \frac{H_{\max}}{d_{\min}} \quad (1)$$

Usually, the acoustic models contain instances of special topologies, with shorter minimum duration, e.g. for silence modeling. Since several consecutive observations of these usually can be consolidated and accumulated on a single row in the circular list, the formula (1) can still be utilized without risk of losing information.

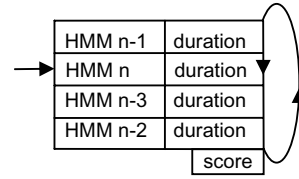


Figure 3: Token representation. A token consists of a circular list and an accumulated score, and is assigned to each state in the probabilistic model and describes the history of the most promising path to the state.

#### 2.4.2. Search algorithm

A decoding procedure for the proposed model becomes very straightforward. The decoder expands a set of hypotheses, optionally pruned by a beam criterion, including the most promising hypothesis. At each frame a token is passed from each state to states to which a transition can be made. The incoming token with the highest accumulated score is kept at each state. Except for an update of the accumulated score, a very simple update of the circular list is made. For tokens that remain in the same HMM, it is sufficient to increment the duration at the current row of the list by one frame unit (e.g. 10ms). For tokens that make a transition to another HMM, the circular list pointer is advanced to the next row where the existing element is replaced by the label of the new HMM and the corresponding duration is initialized to one frame unit (e.g. 10ms).

#### 2.4.3. Generation of output

As previously mentioned, there is value  $H$  denoting the hypothesis look-ahead. The decoder keeps track of the previously emitted symbol as well as the token with the highest accumulated score. The symbol/HMM label that exists  $H$  time units back in the circular list of this token, is selected. If this symbol differs from the previously emitted symbol, the newly selected symbol is emitted and translated to facial parameters.

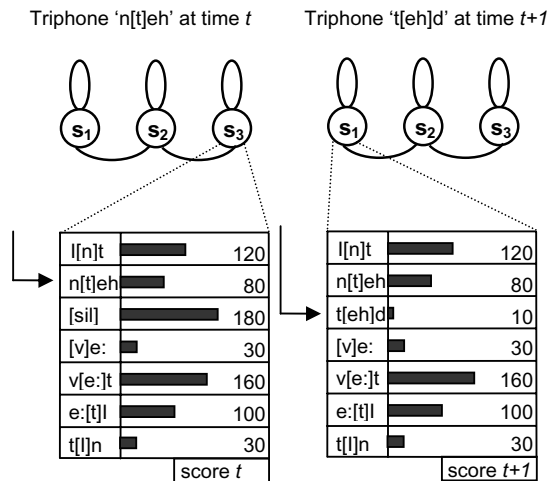


Figure 4: A decoding example of a partial user utterance in Swedish; “vet inte” [v e: t I n t eh] (Eng: don’t know). The figure illustrates a token being passed in the transition between the last two triphone-HMMs (10ms frames) in the recognized sequence, and the corresponding update of the circular list. The graphs in the lists denoting durations are added for illustrative

purposes only. The symbols surrounding the square brackets mark left and right phone context, respectively.

In order to exemplify this Figure 4 is used. Assume that the token in the first state of the right HMM (the right circular list) has the highest score of all tokens. The decoder will then, depending on the current hypothesis look-ahead (H), select a symbol as output. If  $H=0$  the present HMM 't[eh]d' is selected, despite the fact that the associated path only reached the first HMM state. If  $H=100\text{ms}$ , 'I[n]t' is instead selected, since  $10+80 < 100 < 10+80+120$ . Accordingly, for  $H=300\text{ms}$  the selected output is 'e:[t]l', since  $10+80+120+30 < 300 < 10+80+120+30+100$ .

### 3. Experimental results

The proposed decoding strategy was implemented in the prototype Synface real-time decoder. Initial experiments [7] were carried out on the Swedish SpeechDat telephone database [8, 9] with the purpose of evaluating phone recognition accuracy with different levels of hypothesis look-ahead. Offline, tests were conducted, using HTK [10] on the same task, as a baseline. For the tests an acoustic model consisting of 49 context-independent HMMs were used [11]. These were combined into a cyclic ergodic net without any phone-grammar weights. All HMMs were trained using HTK. Each HMM consisted of 3 states, with 32 Gaussian mixtures per state, modeling 39 Mel-Frequency Cepstral Coefficients (MFCC), including delta and acceleration coefficients. Derivates were computed using a symmetric regression window of 2 frames. The frame length was 10ms. The percentage of correctly classified frames for different levels of decoder latency is given in Table 1. The table also includes baseline, non-incremental results, produced by HTK/HVite.

Decoder latency (ms)	10	20	50	100	150	HTK
Frame accuracy (%)	34.71	36.57	39.9	40.92	41.46	41.47

Table 1: Frame recognition accuracy

These preliminary results indicate that without utilization of hypothesis look-ahead, recognition accuracy is substantially lower compared to non-incremental recognition. However, by applying a hypothesis look-ahead of 100 ms and hence delaying decoder output, the relative error was less than 1.3% compared with results obtained by non-incremental decoding. For output latencies above 150 ms, there was no virtually difference in accuracy between recognition with incremental irrevocable output and conventional recognition.

### 4. Conclusion

This paper presents a low-latency decoding algorithm, targeted for time-critical speech transcription tasks, in which results must be emitted incrementally and irrevocably. A major benefit of this method is the ability to explicitly assign a fixed period of time for hypothesis look-ahead. The more time that is assigned for this, the better the accuracy, but at the cost of higher latency. Experiments show that a good balance between accuracy and latency can be achieved through adjustment of the degree of hypothesis look-ahead.

### 5. Acknowledgements

This research was carried out at the Centre for Speech Technology supported by Vinnova (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organizations, in the context of the Synface project. I would like to thank Erik Pihl and Giampiero Salvi for their work in testing the system.

### 6. References

- [1] Granström, B., Karlsson, I. and Spens, K-E., "Synface – a project presentation", *TMH-QPSR – Fonetik 2002*, 44: 93-96, 2002.
- [2] Agelfors, E., Beskow, J., Granström, B., Dahlquist, M., Lundeberg, M., Spens, K-E., Öhman, T., "Synthetic faces as a lip-reading support". *Proc. ICSLP*, pp. 362-365, 1998.
- [3] Beskow, J., "Animation of Talking Agents", In *Proceedings of AVSP'97, ESCA Workshop on Audio-Visual Speech Processing*, pp. 149-152, 1997.
- [4] Ljolje, M., Riley, M., Hindle, D and Sproat, R., "The AT&T LVCSR-2000 system", *proc. NIST Large Vocabulary Conversational Speech Recognition Workshop*, 2000.
- [5] Gauvain, J.-L., Lamel, L. F., Adda, G. and Adda-Decker, M. "Speaker-independent continuous speech dictation". *Speech Communication*, 15:21-37, 1994.
- [6] Young, S.J., Russel, N.H, Thornton, JHS, "Token Passing: A simple conceptual model for connected speech recognition systems", *Technical report CUED/F-INFENG/TR38, Cambridge University Engineering Dept*, 1989.
- [7] Pihl, E., "Bottlenecks in the Synface telephone", Bachelor of Science thesis, Department of Speech, Music and Hearing, KTH, Royal Institute of Technology, Stockholm Sweden, 2003.
- [8] Elenius, K., "Experiences from Collecting Two Swedish Telephone Speech Databases", *International Journal of Speech Technology*, 3:119-127, 2000.
- [9] Höge, H., Troph, H., Winski, R., van den Heuvel, H, Haeb-Umbach, R. and Choukri, K., "European speech databases for telephone applications", *IEEE International Conference in Acoustics, Speech and Signal Processing*, 3:1771-1774.
- [10] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J.J., Ollason, D., Valtchev, V. and Woodland, P.C, "The HTK Book, Version 3.2", *Cambridge University Engineering Department*, 2002.
- [11] Lindberg, B., Johansen, F.T, Warakagoda, N., Lehtinen, G., Kacic, Z., Zgank, A., Elenius K., Salvi, G: "A noise robust multilingual reference recogniser based on SpeechDat(II)", *Proc. ICSLP*, 3:370-373, 2000