EFFICIENT COMBINATION OF LIKELIHOOD RECYCLING AND BATCH CALCULATION BASED ON CONDITIONAL FAST PROCESSING AND ACOUSTIC BACK-OFF

Atsunori Ogawa[1,2]*, Satoshi Takahashi*[2]*, and Atsushi Nakamura*[1]

[1]NTT Communication Science Laboratories, [2]NTT Cyber Space Laboratories, NTT Corporation

ABSTRACT

This paper proposes an efficient combination of state likelihood recycling and batch state likelihood calculation for accelerating acoustic likelihood calculation in an HMM-based speech recognizer. Recycling and batch calculation are each based on different technical approaches, i.e. the former is a purely algorithmic technique while the latter fully exploits PC architecture, and their good acceleration performances are reported in the literatures, respectively. To accelerate the recognition process further by combining them efficiently, we introduce *conditional fast processing* and *acoustic back-off* strategies. Our combination algorithm employs the conditional fast processing strategy that is conditioned by two criteria. The first *potential activity* criterion is used to control not only the recycling of state likelihoods at the current frame but also the precalculation of state likelihoods for several succeeding frames. The second *reliability* criterion and acoustic back-off are used to control the choice of recycled or batch calculated state likelihoods when they are contradictory in the combination and to prevent word accuracies from degrading. Large vocabulary spontaneous speech recognition experiments using four PCs with different specifications showed that, despite the PC specification dependence, the combined acceleration technique further reduced the total recognition time on all of the PCs.

Index Terms— fast acoustic likelihood calculation, state likelihood recycling, batch state likelihood calculation, combined acceleration technique, acoustic back-off

1. INTRODUCTION

It is well known that acoustic likelihood calculation is the most computationally expensive process in a hidden Markov model (HMM)-based speech recognizer. Generally speaking, in the total speech recognition process, more than 50% of the computational time is spent on acoustic likelihood calculation. Thus, to accelerate the speech recognition process, acoustic likelihood computation should be reduced. Many studies have attempted to solve this problem [1, 2, 3, 4, 5, 6]. And they can be roughly classified into the following two technical categories:

The first category consists of purely algorithmic techniques, such as Gaussian reduction [1], Gaussian selection [2], and state selection (or state likelihood recycling) [3]. All these techniques are based on approximations, i.e. simplifications of detailed model structures and/or detailed likelihood calculations. Thus, these techniques trade a slight degradation in recognition accuracy for process acceleration. However, their acceleration performances are essentially independent of the PC specifications. On the other hand, the second category consists of techniques based on PC architectures, such as MMX [4], SSE [5], and batch state likelihood calculation [6]. There is concern that their acceleration performances will depend heavily on the PC specifications. However, since none of these techniques use approximations, process acceleration can be obtained without degrading recognition accuracy.

In this paper, we propose an efficient technique for accelerating acoustic likelihood calculation. The proposed technique is based on a combination of state likelihood recycling [3] and batch state likelihood calculation [6]. As mentioned above, they have different technical characteristics, and their good acceleration performances are reported in [3] and [6], respectively. If we could combine them efficiently, further process acceleration could be expected. However, to the best of our knowledge, there have been no studies investigating their combination, and it is not known how much process acceleration could be obtained by using the combined technique. In this paper, we introduce *conditional fast processing* and *acoustic back-off* [7] strategies to the combined technique, and show its good acceleration performance through large vocabulary spontaneous speech recognition experiments using four PCs with different specifications.

2. EXISTING ACCELERATION TECHNIQUES

We will combine the following two existing fast HMM-state likelihood calculation techniques in the next section.

2.1. State Likelihood Recycling

The first existing acceleration technique is state likelihood recycling [3]. Henceforth, it is referred to as *recycling*. Figure 1 (A) is a state-frame likelihood table that shows the recycling procedure. Recycling assumes that monophones are approximated models of context-dependent (CD) phoneme-HMMs and, before decoding, all CD HMM states are linked to the monophone states on the condition that they are in the same phoneme cluster and in the same state position.

During the frame by frame decoding, we calculate the likelihoods of all the monophone states before calculating the likelihoods of the active CD HMM states. The computational costs of these precalculations are not so high because the number of monophone states is very small compared with the number of the CD HMM states. Then, the likelihood of the corresponding monophone state is referred in the likelihood calculation of each active CD HMM state. If it is Higher than the *recycling* threshold 1 ("circle+H"), the CD HMM state likelihood is calculated Normally ("circle+N"). Conversely, if the monophone state likelihood is Lower than the recycling threshold 1 ("circle+L"), it is Recycled as the approximated likelihood of the CD HMM state ("circle+R").

The recycling threshold 1 is given by multiplying the maximum monophone state likelihood by *recycling* coefficient α ($-\infty < \alpha < 1.0$) at each frame. As α becomes larger, the frequency of the likelihood recycling increases, thus the acoustic likelihood calculation could be accelerated but with a risk of degraded recognition accuracy. As α becomes smaller, the opposite effects could be obtained. Since recycling is a purely algorithmic technique, its acceleration performance is essentially independent of PC specifications.

2.2. Batch State Likelihood Calculation

The second existing acceleration technique is batch state likelihood calculation [6]. Henceforth, it is referred to as *batch calculation*. It is based on the following two experimental analyses: (i) Profiling shows that, in state likelihood calculation, much of the time is spent not on floating-point operations, but in fetching the state parameters (i.e. the mean vectors, covariance matrices and weighting factors of each Gaussian pdf in the state) from the main memory to the cache. (ii) If a state is activated at a frame, it tends to be activated for several succeeding frames.

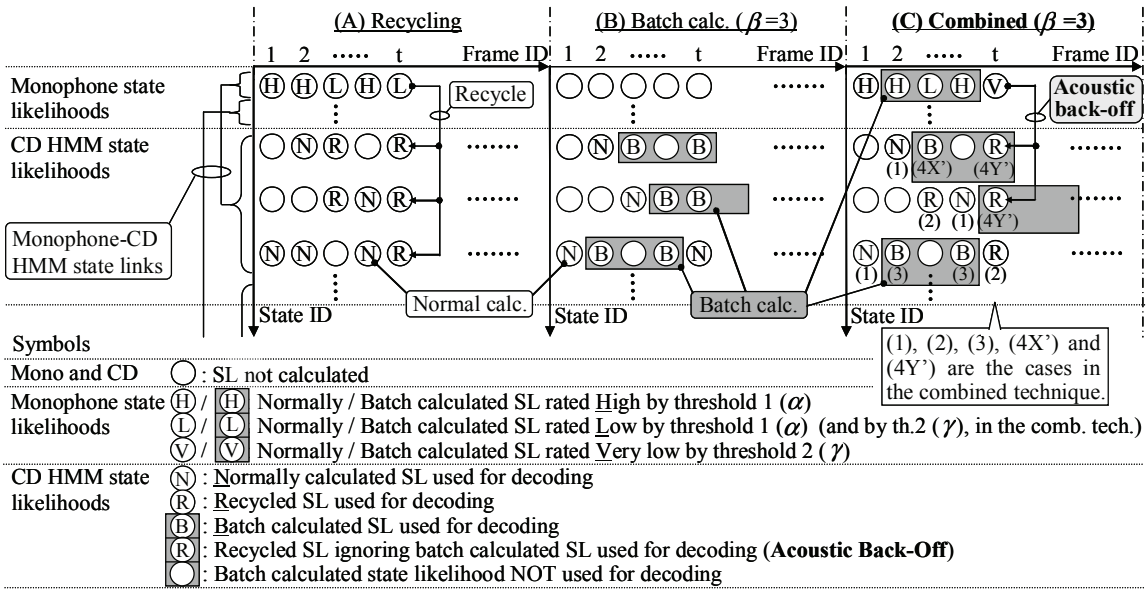

Fig. 1. Procedures of the three acceleration techniques in state-frame likelihood tables (SL: state likelihood).

The batch calculation procedure, which exploits the above two characteristics, is shown in Fig. 1 (B). If a CD HMM state is activated at a frame t, the state likelihoods are calculated and stored in the state-frame likelihood table not only for the current frame t ("circle+N") but also for succeeding β frames ("circle+rectangle". In Fig. 1 (B), β is set at 3). Then, for these look-ahead frames, $t+1, \cdots, t+\beta$, if the state likelihoods are required, they are looked up in the table (convert "circle+rectangle" to "circle+rectangle+B").

In batch calculation, the number of time-consuming state parameter fetching processes described in (i) is reduced, thus, we can expect the acoustic likelihood calculation to be accelerated. If the batch calculated state likelihoods ("circle+rectangle") are not used, they become redundant calculations. However, (ii) indicates that there are not so many of these redundant calculations. There is concern that the acceleration performance will depend heavily on the PC specifications. But, there is no degradation in recognition accuracy because there is no approximation in batch calculation.

3. COMBINED ACCELERATION TECHNIQUE

As described above, recycling and batch calculation have different technical characteristics, and their good acceleration performances are reported in [3] and [6], respectively. To further accelerate the recognition process by combining them efficiently, we introduce *conditional fast processing* and *acoustic back-off* [7] strategies.

Figure 2 shows our combination algorithm. We refer to it as the *conditional fast processing* strategy which is based on two criteria, namely *potential activity* and *reliability* criteria (thresholds or coefficients in the implementation). As with recycling, monophone state likelihoods are rated high or low by using a recycling threshold 1 based on a recycling coefficient α. And there are two possibilities for the corresponding CD HMM state likelihoods, one is *not calculated* and the other is *batch calculated*. Therefore, the combination basically consists of four cases, (1)–(4).

In case (1), as with recycling, we calculate the CD HMM state likelihoods normally. At the same time, as with batch calculation, we calculate the CD HMM state likelihoods for succeeding β frames. In case (2), as with recycling, we recycle the monophone state likelihoods as the approximated likelihoods of the corresponding CD HMM states. In case (3), as with batch calculation, we look up the batch calculated CD HMM state likelihoods in the state-frame likelihood table.

The function of recycling threshold 1 based on recycling coefficient α is strengthened with this combination algorithm. In recycling, as described in Section 2.1, it controls whether we calculate the CD HMM state likelihoods normally or approximate them by corresponding monophone state likelihoods only at the current frame. However, in the combined technique, as described in case (1), it also controls whether we calculate the CD HMM state likelihoods for succeeding β frames in advance with the estimation that these states would be activated in the future frames. Thus, in the combined technique, threshold 1 based on coefficient α is not just a recycling threshold, and we refer to it as a *potential activity* threshold.

In case (4), we must choose either of the two techniques' state likelihood calculation results. In this case, monophone state likelihoods are rated low. Thus, recycling estimates that the corresponding CD HMM state likelihoods are not worth calculating and could be approximated. On the other hand, several frame ago, based on the continuity of the state activation, batch calculation estimated that the CD HMM state likelihoods would be worth calculating and calculated them in advance. That is, in this case, the state likelihood calculation results of recycling and batch calculation are contradictory.

The straightforward choice in case (4) would be to look up the batch calculated CD HMM state likelihoods in the state-frame likelihood table as with case (3). This is because, in general, CD HMM state likelihoods are more precise than those of the corresponding monophone states. However, in this work, we adopt a more efficient method for preventing word accuracy degradation. It is based on the *reliabilities* of the state likelihoods and includes the straightforward choice as its special case. This reliability is a sort of frame level confidence measure and is estimated frame by frame. Our method divides case (4) into two cases with a *reliability* threshold. If the CD HMM state likelihoods are regarded as reliable (case (4X)), we look up them in the state-frame likelihood table as with case (3). On the other hand, if the CD HMM state likelihoods are regarded as unreliable (case (4Y)), we use some other reliable value in place of the unreliable CD HMM state likelihoods.

As more training data are assigned to the states (i.e. the larger the occupancy counts of the states are), the parameter estimations that can be performed for the Gaussian pdfs in the states become more robust, and the likelihoods obtained from these states become more reliable. If an acoustic model stores the occupancy count of each state, the counts can be used to estimate the reliabilities of the

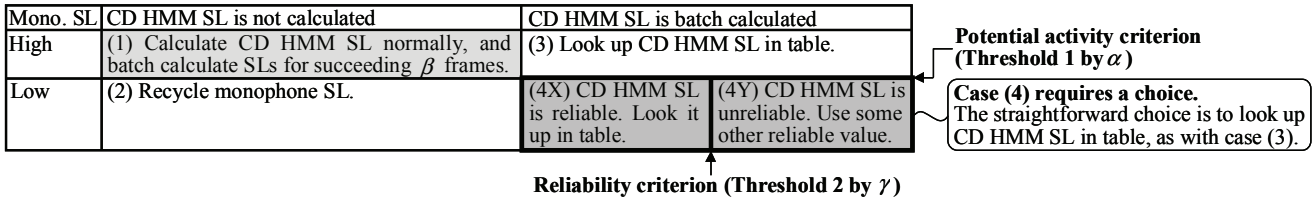

Fig. 2. Our combination algorithm, i.e. the conditional fast processing strategy (SL: state likelihood).

state likelihoods. Unfortunately, they are usually removed from the acoustic model. Therefore, instead of estimating the reliabilities of the CD HMM state likelihoods directly, we adopt another estimation method that uses corresponding monophone state likelihoods. This approach is based on the following consideration:

A monophone state is a representative version of the corresponding CD HMM states. Conversely, a CD HMM state is a detailed version of the corresponding monophone state. There should be a certain degree of correlation between monophone state likelihoods and the corresponding CD HMM state likelihoods. A monophone state is trained to cover all the data of a part (i.e. beginning, middle or ending part) of a phoneme segment. Since the occupancy counts of the monophone states are large, the parameters of the Gaussian pdfs in the monophone states are robustly estimated, and the reliabilities of the state likelihoods obtained from the monophone states are expected to be high. The training data of a monophone state are divided into parts according to the preceding and succeeding phoneme dependencies determined by the tree-based state clustering result. And each of the corresponding CD HMM states is individually trained using a part of the divided data. It is difficult to obtain a robust estimate of the Gaussian pdf parameters of a CD HMM state that covers the *low likelihood region* of the corresponding monophone state. In this *small occupancy count region*, Gaussian pdfs in the CD HMM state are over-tuned to the training data as shown in Fig. 3. Consequently, their covariances tend to be small. That is, if the monophone state likelihood is *very low* for an input feature vector, the state likelihoods of the corresponding CD HMM states for the input feature vector are unreliable. In some cases, even if the monophone state likelihood is very low for an input feature vector, the state likelihoods of the corresponding CD HMM states for the feature vector might be extremely high because of small covariances.

Based on the above consideration, we divide case (4) into two cases, (4X') and (4Y'), by introducing a new coefficient, i.e. the *reliability* coefficient γ, in addition to the *potential activity* coefficient α ($-\infty < \gamma \leq \alpha < 1.0$). γ gives threshold 2 that divides monophone state likelihoods into low or *very low* ranks (threshold 1 $\geq$ threshold 2). With (4X'), the monophone state likelihoods are rated low. Here, as with the straightforward choice, we look up the CD HMM state likelihoods in the state-frame likelihood table. With (4Y'), the monophone state likelihoods are rated very low. Thus, in this case, we estimate that the corresponding batch calculated CD HMM state likelihoods are unreliable and, as with recycling, we recycle the more reliable monophone state likelihoods as the approximated likelihoods of the CD HMM states. Recycling in case (4Y') is a sort of *acoustic back-off* [7], i.e. we postpone the detailed local frame scoring of the active hypotheses by giving them certain low but reliable state likelihoods. If γ is set at $-\infty$, case (4Y') (i.e. acoustic back-off) disappears, and our method becomes equivalent to the straightforward method. If γ is set equal at α, case (4X') disappears, and in case (4) (i.e. (4Y')), CD HMM state likelihoods are approximated by the corresponding monophone state likelihoods according to the acoustic back-off strategy.

Figure 1 (C) shows the procedure of our combined acceleration technique. It should be noted that, in the combined technique, the monophone state likelihood calculations are also accelerated by batch calculation.

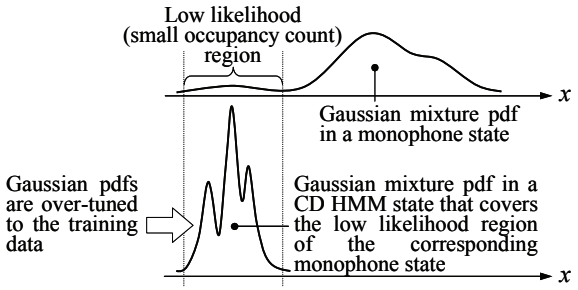

Fig. 3. Example of unreliable CD HMM state.

4. SPEECH RECOGNITION EXPERIMENTS

We evaluated the above three acceleration techniques in large vocabulary spontaneous speech recognition experiments.

4.1. Experimental Setup

An HMM-based female acoustic model was trained using 100 hours of speech data consisting of 120k spontaneous utterances by 55 female speakers. It had 2,000 states (consisting of 90 monophone states and 1,910 CD HMM states) and each state had 16-mixture Gaussian pdfs with diagonal covariance parameters. A 30k vocabulary size word trigram language model was trained using human transcribed text consisting of 1.1M word of spontaneous speech. The baseline speech recognizer was VoiceRex [8] which employed a standard Viterbi beam search with a two-pass decoding strategy. The three acceleration techniques described in Sections 2 and 3 were implemented on VoiceRex. The evaluation speech data consisted of 714 spontaneous utterances by 17 female speakers (42 utterances per speaker), who were different from the 55 female speakers who provided the acoustic model training data. The test set perplexity was 108 and the OOV rate was 0.8%.

As described in Section 2.2, there is concern that the acceleration performance of batch calculation (and also the combined technique) depends on the PC specifications. Thus, we conducted the experiments using four PCs with the different specifications shown in Table 1. Two were based on current standard Pentium CPUs and the other two were based on state-of-the-art Core 2 CPUs. All the PCs had Red Hat Linux 9 operating systems. In the following, we identify the PCs by their CPU types.

4.2. Experimental Results

Left hand side of Fig. 4 shows the real time factor (RTF) reduction rates of the three acceleration techniques from the baseline speech recognizer on each PC. In this figure, the RTFs were measured on the basis of the *total recognition time* (not only the acoustic likelihood calculation time). At each point, the RTFs were measured five times and averaged to reduce measuring errors. The baseline speech recognizer did not employ an acceleration technique, and its word accuracy was 75.61%.

Table 1. Specifications of the four PCs.

CPU type	Clock freq.	Cache size	Memory size
Pentium 4	3.6GHz	1MB	2GB
Pentium Xeon	3.6GHz	2MB	4GB
Core 2 Duo	2.4GHz	4MB	4GB
Core 2 Quad	2.4GHz	8MB	8GB

Techniques	Parameters			WACC	RTF Reduction rate [%]			
	β	α	γ	[%]	Pen4	Xeon	Duo	Quad
Baseline	—	—	—	75.61	—	—	—	—
Recycling	—	0.725	—	75.30	13.7	15.1	16.0	15.9
Batch calc.	7	—	—	75.61	16.3	19.7	11.5	10.8
			$-\infty$	75.39	32.4	36.0	29.4	28.6
			0.100	75.38	32.4	35.5	29.3	28.5
			0.200	75.43	32.2	35.5	29.3	28.5
			0.300	75.43	32.3	35.6	29.2	28.5
			0.350	75.47	32.3	35.3	29.3	28.3
			0.375	75.52	32.3	35.3	29.0	28.3
Combined	**7**	**0.825**	**0.400**	**75.54**	**32.3**	**35.8**	**29.3**	**28.3**
			0.425	75.44	32.3	35.5	29.1	28.3
			0.450	75.39	32.4	35.5	29.3	28.4
			0.500	75.41	32.3	35.7	29.4	28.5
			0.600	75.16	32.3	35.6	29.0	28.4
			0.700	74.69	32.1	35.4	29.3	28.4
			0.825	72.93	31.3	34.6	28.4	27.6

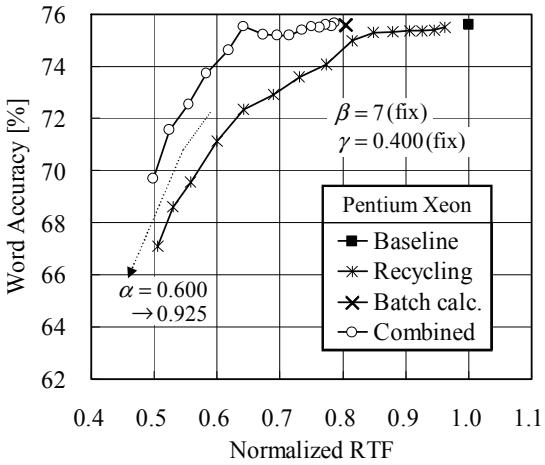

Fig. 4. (Left) RTF reduction rates of the three acceleration techniques on each PC. (Right) Normalized RTFs vs. word accuracies of the three acceleration techniques on Pentium Xeon.

With recycling, on condition that a word accuracy degradation of up to 0.5% from the baseline was allowed, the recycling coefficient α was increased from 0.600 in 0.025 steps, and was finally fixed at 0.725. With recycling, the RTF reduction rates for Pentium Xeon, Core 2 Duo and Quad are almost the same, while that for Pentium 4 is about 2% lower than those of the other three PCs. Thus, contrary to the expectation described in Section 2.1, it is revealed that the acceleration performance of recycling also depends on the PC specifications. However, the dependence is not so strong.

With batch calculation, the number of look-ahead frames β was fixed at 7 according to our preliminary experiments and [6]. In contrast to recycling, with batch calculation, it is confirmed that the RTF reduction rates depend heavily on the PC specifications as mentioned in Section 2.2. The RTF reduction rates for Core 2 Duo and Quad are about 9% lower than that for Pentium Xeon. We guess that this is because the costs of the fetching processes have been reduced by the use of several new technologies (e.g. Intel Smart Memory Access and Intel Advanced Smart Cache) in recent Core 2 CPUs [9].

With the combined technique, β was fixed at 7 as with batch calculation, and the two coefficients, the *potential activity* coefficient α and the *reliability* coefficient γ, were adjusted as follows: First, γ was fixed at $-\infty$. This meant that, the straightforward choice was fully performed in case (4) in Fig. 2. Then, α was adjusted with the same procedure employed with recycling as described above, and was finally fixed at 0.825. Next, γ was varied from 0.100 to 0.825 in 0.025 steps. If γ was set at 0.825 (i.e. equal to α), acoustic back-off was fully performed in case (4) in Fig. 2. We can confirm that word accuracy is steadily improved by adjusting γ, i.e. acoustic back-off prevents the word accuracy from degrading. The best γ is 0.400, and the degradation in word accuracy is only 0.07% from that of the baseline. With all the PCs, the combined technique further accelerates the speech recognition processes, and the RTF reduction rates range from 28% to 36%. The RTF reduction rates of Core 2 Duo and Quad are about 8% lower than that of Pentium Xeon. As described above, these performance degradations are caused by the PC specification dependence of batch calculation.

Right hand side of Fig. 4 shows the relations between the RTFs and word accuracies of the three acceleration techniques on Pentium Xeon (the RTFs were measured five times and averaged as before, and then normalized by the one of the baseline recognizer). In the experiments on which these figures are based, β for the batch calculation and combined technique was fixed at 7 as before, and γ for the combined technique was fixed at 0.400 (the best value). Then, α for recycling and the combined technique was varied from 0.600 to 0.950 in 0.025 steps. From this figure, we can again clearly confirm the good acceleration performance of the combined technique.

5. CONCLUSION AND FUTURE WORK

We proposed an efficient combination of state likelihood recycling and batch state likelihood calculation for accelerating acoustic likelihood calculation in an HMM-based speech recognizer. We introduced conditional fast processing and acoustic back-off strategies to our combined technique. And it realized a reduction of up to 36% in total recognition time from the baseline in large vocabulary spontaneous speech recognition experiments using four PCs with different specifications.

We believe that our combined technique will be further improved by implementing the following ideas: First, we could enhance the combination algorithm shown in Fig. 2 by making it work based on state likelihoods not only at the current frame but also for sevral preceding frames. For example, we could rate the monophone state likelihoods averaged for last $x(\geq 2)$ frames. Second, we could make our combination algorithm more flexible by adding other criteria. For example, we could divide case (1) in Fig. 2 into two cases (1) and (1') by introducing a third threshold, and in case (1'), we calculate CD HMM state likelihoods normally *without* batch calculating them for succeeding frames. Third, although acoustic back-off steadily prevented word accuracies from degrading, its effect will be increased by estimating the reliabilities of state likelihoods directly, as described in Section 3. Finally, of course, our technique will be further accelerated by combining it with other acceleration techniques.

6. REFERENCES

[1] K. Shinoda and K. Iso, "Efficient reduction of Gaussian components using MDL criterion for HMM-based speech recognition," *Proc. ICASSP*, 2002, pp. 869–872.

[2] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," *Proc. ICASSP*, 1993, pp. 692–695.

[3] Y. Komori, M. Yamada, H. Yamamoto, and Y. Ohora, "An efficient output probability computation for continuous HMM using rough and detail models," *Proc. EUROSPEECH*, 1995, pp. 1087–1090.

[4] S. Kanthak, K. Schutz, and H. Ney, "Using SIMD instructions for fast likelihood calculation in LVCSR," *Proc. ICASSP*, 2000, pp. 1531–1534.

[5] M. Afify, F. Liu, H. Jiang, and O. Siohan, "A new verification-based fast-match for large vocabulary continuous speech recognition," *IEEE Trans. on SAAP*, vol. 13, no. 4, pp. 546–553, July 2005.

[6] M. Saraclar, M. Riley, E. Bocchieri, and V. Goffin, "Towards automatic closed captioning: low latency real time broadcast news transcription," *Proc. ICSLP*, 2002, pp. 1741–1744.

[7] J. de Veth, B. Cranen, and L. Boves, "Acoustic backing-off as an implementation of missing feature theory," *Speech Communication*, vol. 34, pp. 247–265, 2001.

[8] A. Ogawa, Y. Noda, and S. Matsunaga, "Novel two-pass search strategy using time-asynchronous shortest-first second-pass beam search," *Proc. ICSLP*, 2000, pp. 290–293.

[9] Intel Corporation, http://www.intel.com