Voice Code Verification Algorithm Using Competing Models for User Entrance Authentication

Heungkyu Lee* and Hanseok Ko**

*Dept. of Visual Information Processing,
**Dept. of Electronics and Computer Engineering,
Korea University, Seoul, Korea
Hklee@ispl.korea.ac.kr, hsko@korea.ac.kr

Abstract. In this paper, we propose a voice code verification method for an intelligent surveillance guard robot, wherein a robot prompts for a code (i.e. word or phrase) for verification. In the application scenario, the voice code can be changed every day for security reasoning and the targeting domain is unlimited. Thus, the voice code verification system not only requires the text-prompted and speaker independent verification, but also it should not require an extra trained model as an alternative hypothesis for log-likelihood ratio test because of memory limitation. To resolve these issues, we propose to exploit the subword based anti-models for log-likelihood normalization through reusing an acoustic model and competing with voice code model. The anti-model is automatically produced by using the statistical distance of phonemes against a voice code. In addition, a harmonics-based spectral subtraction algorithm is applied for a noisy robust system on an outdoor environment. The performance evaluation is done by using a common Korean database, PBW452DB, which consists of 63,280 utterances of 452 isolated words recorded in silent environment.

1 Introduction

For surveillance task, a lot of manpower at the sentry is placed on duty to guard the premises against unauthorized personnel for 24 hours. To lessen the time and overload of human guards at post, an intelligent surveillance guard robot is desirable. The surveillance guard robot takes the role of detecting and authorizing a person entering into the perimeter of the secured area as well as passing the status warning. This system includes detection, recognition and tracking by using multiple sensors such as stereo cameras, IR cameras and array microphones. Under such an environment, a robot prompts for a code (i.e. word or phrase) for verification. In the application scenario, the voice code can be changed every day for security reasoning and the targeting domain is unlimited. Thus, the voice code verification system not only requires the text-prompted and speaker independent verification but also it should not require an extra trained model such as a filler or garbage model for an alternative hypothesis model in a log-likelihood ratio test (LRT). This is due to the memory limitation on an embedded DSP (Digital Signal Processing) hardware system that we developed.

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 610–622, 2004.

This paper is motivated by the task where the system does not need to know the speaker and has only to verify whether the uttered voice code is correct or not on a specific area. Mostly, confidence measure (CM) for this task is used to verify the uttered observation sequences after or during calculating the probability of a word *W* being recognized by an ASR system. Besides the utterance verification [1][2], a filler model or garbage model can be used for these purposes. However, most algorithms require the extra model trained for a garbage model or anti-model [3]. But a limited memory size of our proposed embedded system prevents the algorithm from using and storing the extra alternative hypothesis model. Thus, the method that does not require the extra trained model and the re-use of the acoustic model is investigated for the voice code utterance verification. Generally, a log-likelihood ratio test is applied to verify the utterance in this field of utterance verification where the verification step requires the alternative model for doing this task. To manage this problem, the anti-models that are re-usable from an acoustic model and can compete with a voice code model should be considered.

Our proposed system uses a two-pass strategy using a SCHMM (Semi-Continuous Hidden Markov Model)-based recognition [4] and verification step as in Figure 1. In the first pass, recognition is performed via a conventional Viterbi beam search algorithm that segments the test utterance into the *N*-best strings of phoneme hypotheses. In the second pass, voice code verification is performed. It computes a confidence measure that determines whether or not to reject the recognized voice code [5]. This paper is organized as follows. In Section 2, we describe the voice code verification method using sub-word based anti-models. In Section 3, we conduct the representative experiments. Finally, the conclusive remarks are presented and we discuss the results on performance of the proposed methods.

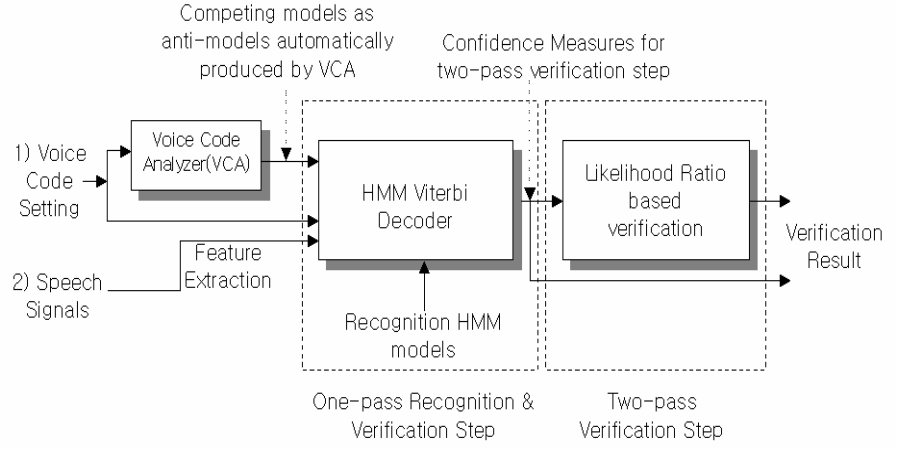


Fig. 1. The block-diagram of the voice code verification

2 Voice Code Verification

2.1 Competing Models as Anti-models

The a posteriori probability used for the likelihood normalization method in text-prompted speaker verification is given by

$$p(S_c, W_c / O) = \frac{p(O / S_c, W_c)\, p(S_c, W_c)}{\sum_i \sum_j \{p(O / S_i, W_j)\, p(S_i, W_j)\}} \approx \frac{p(O / S_c, W_c)}{\sum_i \sum_j p(O / S_i, W_j)} \tag{1}$$

where S_i is a speaker and S_c is the claimed speaker. W_i is a text and W_c is the prompted text. $p(S_i,W_j)$ is the simultaneous probability for speaker i and text j. $p(O/S_c,W_c)$ is the probability of the claimed speaker's HMM corresponding to the prompted text. In the voice code verification, S_c and S_i can be ignored because this is a speaker independent verification. Thus, the equation (1) can be simplified as

$$p(W_c / O) \approx \frac{p(O / W_c)}{\sum_j p(O / W_j)} \tag{2}$$

This is the same with the likelihood normalization method for utterance verification in a conventional ASR algorithm. In equation (2), W_c becomes the uttered word sequence, $p(O/W_j)$ is approximated by the summation of the n highest likelihoods by using the parallel phoneme HMM networks for all registered words. As a result, if the speaker information is ignored, the text-prompted verification technique is equal to the conventional ASR algorithms as a pattern classification problem using the maximum a posteriori (MAP) decision rule to find the most likely sequence of words as follows.

$$W_k = \arg\max_j L(O / W_j) \tag{3}$$

where $L(O/W_j)$ is the likelihood of an observation sequence O given word W_j. In a text-prompted verification, this is the time that the number of a given word, j is equal to one. At this time, when someone speaks a false word for a text-prompted verification, we cannot verify the uttered word because we have no normalized models to test a likelihood score. Thus, we need the models to increase the likelihood score more than the one of claimed voice code model when someone speaks the false voice code. But, we do not want the previously trained models (filler or garbage model) for likelihood normalization because the memory of our system is limited. To cope with these problems, we reuse the original acoustic model for the alternative hypothesis model. Alternative hypothesis models as anti-models can be made automatically through the analysis of phoneme information with respect to the prompted text word. In this paper, we propose the construction method of anti-models by using the statistical distance of phonemes against the voice code. This reused anti-model can be used for competing with the prompted voice code as follows.

$$W_k = \arg\max_j L\left(O / W_0, \overline{W_1}, ..., \overline{W_j}\right) \tag{4}$$

where W_0 is a prompted voice code, $\overline{W}$ is a competing model to be used for likelihood normalization, and the combination of anti-phonemes. W_k is a concatenation of syllable units that can be written as

$$W_k = S_1^k S_2^k \cdots S_N^k \tag{5}$$

where N is the number of a syllable. In addition, a syllable unit is a concatenation of context independent phoneme units that can be written as

$$S_k = P_1^k P_2^k \cdots P_M^k \tag{6}$$

where M is the number of the phoneme. Finally, this context independent phoneme unit is changed into a context dependent phoneme unit after anti-phoneme units are constructed. Then, the anti-phoneme units become the context dependent model.
As you see in equation (4), in the first pass, Viterbi algorithm is employed to find the most likelihood word W_k. In this step, prompted voice code is first verified as in given;

$$PVC = \begin{cases} true & if\ \mathrm{j} = 0 \\ false & \text{else} \end{cases} \tag{7}$$

If the verification result, PVC is true, the second pass to test a likelihood score is followed. In this sub-section, we describe the automatic construction method of anti-models that opposes to the statistical distance according to the manner and place of articulation, and tongue advancement and aperture. At first, the prompted voice text is automatically changed into a phoneme string, produced using a grapheme to phoneme (G2P) converter through the text analysis. Then, the following rules for construction of anti-models are applied. The voice code can be composed of a concatenation of a syllable, S that is the set of phonemes. A voice code, W_0 is expressed by

$$W_0 = \{S_1, S_2, ..., S_N\} \tag{8}$$

where N is the total number of syllable of a given voice code. At first, when a person says a similar word, this may result in a verification success. This occurs when any person says the word as follows.

$$\begin{aligned} \overline{W_1^1} &= \{\overline{S_1}, S_2, ..., S_N\}, \\ \overline{W_2^1} &= \{S_1, \overline{S_2}, ..., S_N\}, ..., \\ \overline{W_N^1} &= \{S_1, S_2, ..., \overline{S_N}\} \end{aligned} \tag{9}$$

where N is the number of anti-syllable models for the first method and the variable, $\overline{S}$ is the anti-syllable. This sometimes results in a verification success. Thus, we can use the equation (9) as anti-models to prevent the false acceptance through competing with a voice code model when a person says a similar password. The anti-syllable

model can be constructed using a concatenation of an anti-phoneme against each syllable unit as

$$\overline{S_N} = \left\{\overline{P_1}, \overline{P_2}, ..., \overline{P_M}\right\} \tag{10}$$

The criterion to select the anti-phoneme is to use the method to classify phonemes according to the manner and place of articulation, and tongue advancement and aperture as in Table 1, which is matched in order between phoneme and anti-phoneme. Table 2 depicts the matched phoneme set between Korean and English for your understanding. In this paper, we use the 44 phonemes set for Korean voice code verification. The anti-phoneme is chosen the one to one matching between phoneme and anti-phoneme. To make the anti-model of each syllable, the corresponding syllable in the prompted voice code is changed into an anti-syllable using the anti-phoneme according to the Table 1 after the text is changed into the phoneme list using a grapheme to phoneme converter, where it needs a parsing process to find the each syllable that is composed of consonant and vowel. In the Korean language, a syllable can be composed of "C+V", "C+V+C" and "V+C" where V is the vowel and C is the consonant. A Korean syllable can be classified into 9 groups as in Table 3. Using this rule, a given text is classified into the syllable lists [7][8].

Table 1. The anti-model production rules using statistical distance of phonemes

		Phoneme to Anti-phoneme	Standard
Consonant	Phoneme	ㄱㅋㄲㅇㄷㅌㄸㅅㅆㅈㅊㅉㄴㄹㅂㅍㅃㅁㅎ	Manner and place of articulation
	Anti-phoneme	ㅃㅂㅂㅂㄲㄱㄱㅃㅂㄲㄱㄱㄲㄲㄲㄱㄱㄱㅂ	
Vowel	Phoneme	ㅏㅑㅓㅣㅢㅝㅜㅠㅡㅔㅐㅟㅒㅖㅚㅛㅋㅕㅘㅞㅙ	Tongue advancement and aperture
	Anti-phoneme	ㅝㅜㅝㅏㅏㅏㅣㅏㅏㅕㅕㅏㅏㅣㅏㅏㅔㅣㅏㅏ	

Table 2. Matching table between Korean and English phoneme

Consonent						Vowel					
ㄱ	g	ㅋ	kh	ㅊ	ch	ㅏ	a	ㅛ	yo	ㅋ	yv
ㄴ	n	ㅌ	th	ㅈ	j	ㅓ	v	ㅠ	yu	ㅑ	ya
ㄷ	d	ㅍ	ph	ㅉ	jj	ㅗ	o	ㅒ	yae	ㅖ	we
ㄹ	r	ㅎ	h	ㅇ	ng	ㅜ	u	ㅖ	ye	ㅢ	eui
ㅁ	m	ㄲ	gg	ㅆ	ss	ㅡ	eu	ㅘ	wa	ㅝ	Wv
ㅂ	b	ㄸ	dd			ㅣ	i	ㅚ	we	ㅙ	we
ㅅ	S	ㅃ	bb			ㅐ	e	ㅟ	wi	ㅔ	e

Table 3. Korean syllable production rules

Syllable	Word production rules	Group	Group number	Comments
CV	CV/CV	CV/CV (PART1)	1	
	CV/CVC			
	CV/VC	CV/V (PART2)	2	
	CV/V			
CVC	CVC/CV	CVC/C (PART3)	3	
	CVC/CVC			
	CVC/VC	CVC/V (PART4)	1	It follows the rule part 1 according to the Korean utterance rule.
	CVC/V			
VC	VC/CV	VC/C (PART5)	4	
	VC/CVC			
	VC/VC	VC/V (PART6)	5	It follows the rule part 7 according to the Korean utterance rule.
	VC/V			
V	V/CV	V/CV (PART7)	5	
	V/CVC			
	V/VC	V/V (PART8)	6	
	V/V			

The second, when any person utters the similar word that includes all parts of a prompted voice code, it often results in a verification success. It would be the time that any person utters a false text as follows

$$\begin{aligned}
&\overline{W_1^2} = \{S_1, S_2, ..., S_N, \overline{S_{N+1}}\}, \\
&\overline{W_2^2} = \{S_1, S_2, ..., S_N, \overline{S_{N+1}}, \overline{S_{N+2}}\}..., \\
&\overline{W_M^2} = \{S_1, S_2, ..., S_N, \overline{S_{N+1}}, \overline{S_{N+2}}, ..., \overline{S_{N+M}}\}
\end{aligned} \tag{11}$$

where M is the number of anti-syllable models to compete with a given voice code model, and anti syllable, $\overline{S_{N+M}}$ is matched to its syllable, S_N. To prevent this case, we use the equation (11) as anti-models. The anti-syllable model also can be constructed using Table 1 or 2.

The third, when any person says a similar word that is some part of the password text, this also often results in a verification success. This happens when any person says a text as follows.

$$\begin{aligned}
&\overline{W_1^3} = \{S_1\}, \\
&\overline{W_2^3} = \{S_1, S_2\}, ..., \\
&\overline{W_{N-1}^3} = \{S_1, S_2, ..., S_{N-1}\}
\end{aligned} \tag{12}$$

where *N-1* is the number of anti-syllable models. To prevent this case, we use the equation (12) as anti-models. In addition, we can use anti-models contrary to the equation (12) as follows.

$$\begin{aligned}\overline{W_1^4} &= \{\overline{S_1}\},\\ \overline{W_2^4} &= \{\overline{S_1}, \overline{S_2}\},...,\\ \overline{W_{N-1}^4} &= \{\overline{S_1}, \overline{S_2},..., \overline{S_{N-1}}\}\end{aligned} \tag{13}$$

Finally, the following anti-model is applied.

$$\overline{W_1^5} = \{\overline{S_1}, \overline{S_2},..., \overline{S_N}\} \tag{14}$$

After these anti-models are constructed through the analysis of a given voice code, all anti-models are used for competing with a voice code model. These models would increase the likelihood score of anti-models while the likelihood score is reduced when someone speaks a false word or phrase.

2.2 Voice Code Verification Using Sub-word Based Anti-models

In the second pass, the voice code verification task is applied. Generally, a sub-word based utterance verification or out- of-vocabulary rejection method is based on a likelihood ratio test. The major difficulty with an LRT in utterance verification is how to model the alternative hypothesis, where the true distribution of the data is unknown and an alternative hypothesis usually represents a very complex and composite event. Given a decoded sub-word in an observed segment, we need a decision rule by which we assign the sub-word to either hypothesis H_0 or H_1. For the binary testing problem, one of the most useful tests for decision is the Neyman-Pearson Lemma. For a given number of observations, which minimizes the error for one class while maintaining the error for the other class constant, is a likelihood ratio test as follows.

$$LRT(X) = \frac{P(X/H_0)}{P(X/H_1)} = \frac{P(O_n/\lambda_n)}{P(O_n/\overline{\lambda_n})} \geq \eta \tag{15}$$

where H_0 means that the hypothesis is true and H_1 means that the hypothesis is false, λ is the sub-word model, $\overline{\lambda}$ is the anti-subword model, and X is the uttered input observation that the number of a sub-word is N as follows.

$$X = \{O_1, O_2,..., O_N\} = \{O_1^{t_1}, O_{t_1+1}^{t_2},..., O_{t_{N+1}}^{t_N}\} \tag{16}$$

The sub-word alignment and log-likelihood value are obtained on a log domain through the Viterbi segmentation. For the normalization of likelihood ratio, an average frame log-likelihood ratio (LLR), *R(n)* is defined as

$$R_n = \frac{1}{l_n}\left[\log P(O_n/\lambda_n) - \log(O_n/\overline{\lambda_n})\right] \tag{17}$$

The dynamic range of a sub-word based likelihood ratio is higher. This can affect the overall performance. One way to limit the dynamic range of the sub-word confidence measure is to use a sigmoid function of the form.

$$R_n = \frac{1}{l_n}\left[\log P(O_n/\lambda_n) - \frac{1}{nBest}\sum_{m=1}^{nBest}\log(O_n/\overline{\lambda_m})\right] \tag{18}$$

In this equation (18), dynamic range of sub-word based log likelihood ratio is high. This can affect to the overall performance. One way to limit the dynamic range of the sub-word confidence measure is to use a sigmoid function of the following form.

$$U_n = \frac{1}{1+\exp(-\alpha\times(R_n-\tau))} \tag{19}$$

where τ and α are location and weighting parameters. The log confidence score has a slope of α when the log likelihood score is less than zero.

2.3 Confidence Measure

For an effective voice code verification, we need to define a function to combine the results of sub-word tests. The confidence measure (CM) for an input utterance O can be represented as

$$CM(O) = f\left(CM_1, CM_2, ..., CM_N\right) \tag{20}$$

where *f()* is the function to combine the verification scores. This is defined as a function of their likelihood ratios. It can be considered as a joint statistic for overall word-level verification. The first confidence measure CM_1 is based on a frame-duration normalization, which is defined as follows:

$$CM_1 = \frac{1}{L}\sum_{n}^{N}\left(l_n * R_n\right) \tag{21}$$

where N is the total number of sub-words in the utterance, and L is the total number of utterance frames, $L = \sum_{n=1}^{N} l_n$. The second one CM_2 is based on a syllable segment-based normalization. It is a simple average of a log likelihood of all the syllables.

$$CM_2 = \frac{1}{N}\sum_{n}^{N} R_n \tag{22}$$

$$CM_3 = \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log R_n\right) \tag{23}$$

$$CM_4 = \frac{1}{N}\sum_{n=1}^{N} U_n \tag{24}$$

$$CM_5 = \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log U_n\right) \tag{25}$$

where equation (22) and (23) are the arithmetic and geometric means of the unweighted sub-word level confidence scores, and equation (24) and (25) are the arithmetic and geometric means of the sigmoid weighted sub-word score.

For every confidence measure, a specific threshold is set up. If its value is below the threshold, the candidate is discarded from the verification task. Thus, it results in a voice code verification failure.

3 Evaluation of Proposed System

3.1 Experimental Condition

For speech input to verify the uttered voice code, the sampling rate is 11KHz 16bit, and speech signals are analyzed within 125ms frame with 10ms lapped into 26th order feature vector that has 13th order MFCCs including log energy and their 1st and 2nd derivatives. A training data set consists of about 120,000 utterances of 6,000 isolated words set recorded in an office environment. In addition, we used a different speech corpus for testing the data set, which is PBW452DB, Korean Common DB. It consists of 63,280 utterances of 452 isolated words recorded in a silent environment.

3.2 Experimental Results

We applied an utterance verification technique using an *N*-best alternative hypothesis model for likelihood normalization in an LRT. In our previous work [6] on utterance verification, several utterance verification methods are simulated. Our method, the Bayesian fusion technique showed the performance higher than any other methods. However, we applied the 5-best technique that is easy to implement and has a low computing time on a DSP board.

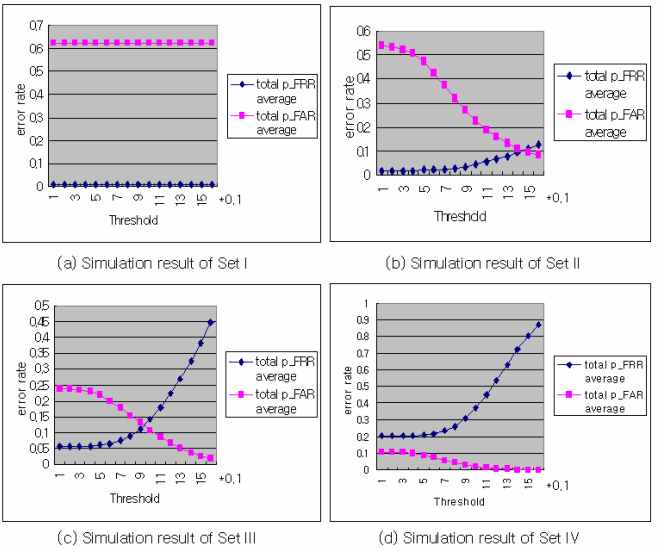
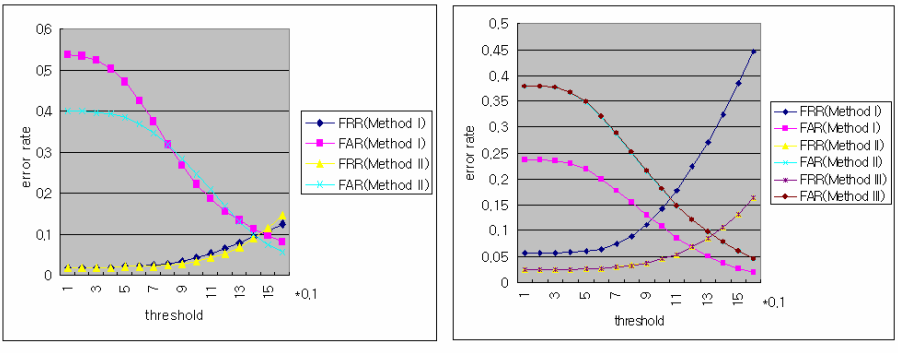


Fig. 2. Simulation results of the second approach using statistical distance of phonemes

For a voice code verification simulation, an anti-model making routine for likelihood normalization against a log-likelihood score of voice code is implemented as in Section 2. The simulation is done using the total anti-models as in equations (9), (11), (12), (13) and (14). At first, we evaluated by using four categories of the set. Set I is the anti-model set using equation (14), set II is the anti-model set using equations (11) and (14), set III is the anti-model set using equations (11), (12) and (14) and set IV is the anti-model set using equations (9), (11), (12) and (14). As shown in (a) of Figure 2, set I has a high FAR while FRR is low. In set II, the EER is 0.09. However, it cannot cope with various situations as described in Section II. Thus, we did extra simulations about set II. The (b) of Figure 2 is the time that we use M=1 of equation (11). When we use M=3 (method II), FRR and FAR are improved as in Figure 3, (a).



(a) Comparison result of the set II using method I, II (b) Comparison result of the set III using method I,II, III.

Fig. 3. Comparison results using anti-models set

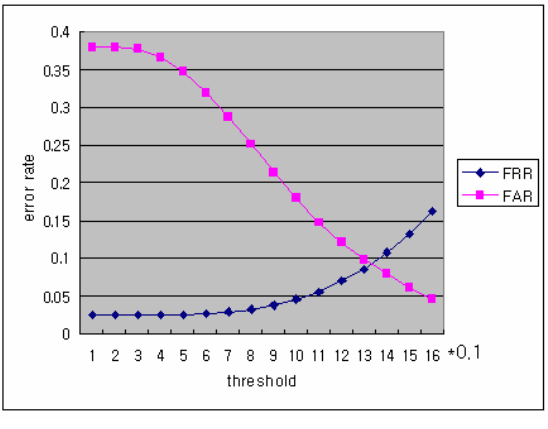


Fig. 4. Final result using all anti-models

Using the result of (a) in Figure 3, we combined the anti-models using equations (12) and (13). In (b) of Figure 3, Method I is the method using equations (11), (12) and (14). Method II is the method using equations (11), (13) and (14). Method III is the method using equations (11), (12), (13) and (14). Method II and III showed the similar result. However, method III is a bit improved and also can cope with the any utterance of people very well. Finally, to method III of set III, we combined anti-models using equation (9) as in Figure 4. The curve shape and result of final method is similar with method III of Figure 3, (b). But the final result is a bit improved than method III of Figure 3, (b). In addition, the EER is 0.08. This result is improved by 16% than the one of utterance verification result in our previous work [6].

This system usually is utilized on an outdoor surveillance region. Thus, this requires a noise robust voice code verification to cope with not only environmental noise, but also other white noises. To resolve this problem, a harmonics-based spectral subtraction algorithm [9] is applied for preprocessing the noise. First, experiment is conducted by the Aurora 2 evaluation procedure under a continuous digits recognition tasks. Test sets are reproduced using TIDigits of which the entire speech data are down-sampled to 8 Khz and various realistic noises are added artificially. The feature vector order is 39 and is composed of 13 order static MFCC (c1-c12+log energy), its derivatives and accelerations. For comparison, a spectral subtraction algorithm and nonlinear algorithm are evaluated as in Table 4. As you can see in Table 4, the HSS showed that the proposed algorithm is more robust than other algorithms. A notable advantage of the proposed scheme is that it does not require an exact SNR estimate in various noise conditions.

Next, HSS is applied to the voice code verification experiment, which is also conducted using PBW452DB. The simulation result is shown in Table 5. In a babble noise environment, EER did not show the rapid decrease of EER. However, in white noise, EER showed a rapid decrease of EER. But, It brought about a 40% relative improvement than when there was no harmonics-based spectral subtraction algorithm.

Table 4. Word accuracies on Aurora2 mis-matched training/testing condition(%)

	Baseline	SS	NSS	HSS
Baseline	60.06	77.89	78.20	80.59
CMN	71.16	78.56	78.73	82.00

Table 5. EER of voice code verification under babble and white noise environments

		EER(Equal error Rate)			
		Clean	5dB	10dB	15dB
Clean DB	FRR	0.076738	-	-	-
	FAR	0.109624	-	-	-
Babble noise	FRR	-	0.088653	0.088069	0.089238
	FAR	-	0.096128	0.095923	0.095385
White noise	FRR	-	0.518268	0.376042	0.221871
	FAR	-	0.362295	0.320828	0.218299

4 Discussions and Conclusions

The key point is to use the competing models that are anti-models using a statistical distance of phonemes. This idea is due to the fact that the alternative model always follows the same state as the target model. Thus, if we can do the modeling of the alternative hypothesis very well, we thought that the voice code verification task could be solved by competing against each other without extra trained models such as filler or garbage models. As you saw the simulation result, we know that the use of a lot of anti-models degraded the detection probability while the use of a few anti-models degraded the false acceptance rate. Thus, the proper number of anti-models that can compete with the voice code model should be used. In addition, outdoor noise is an important issue that should be considered. Under this condition, the speaker verification rate is very low and also the voice code verification rate is the same. Even though we applied a harmonic-based spectral subtraction algorithm, some other algorithms should also be used for compensating the verification rate on an outdoor environment where wind or rain noise exists, and so on.

As a result, our proposed method for a text-prompted and speaker independent verification provided a voice code verification function without an extra trained model such as filler or garbage models for likelihood normalization through the reuse of a general acoustic model. In experiment, the performance evaluation is done by using a common Korean database, PBW452DB, which consists of 63,280 utterances of 452 isolated words recorded in a silent environment. The result is improved by 16% higher than the result of utterance verification result. In addition, simulation result showed that the performance is higher under noisy environment than in any other algorithms when we applied the harmonics-based spectral subtraction algorithm compared to general spectral subtraction and nonlinear spectral subtraction.

Acknowledgements

This work was supported by grant No. 2003-218 from the Korea Institute of Industrial Technology Evaluation & Planning Foundation.

References

[1] Hui Jiang, Chin-Hui Lee, "A new approach to utterance verification based on neighborhood information in model space," Speech and Audio Processing, IEEE Transactions on , Volume: 11, Issue: 5, Sept. 2003.

[2] Tomoko Matsui, Sadaoki Furui, "Likelihood normalization for speaker verification using a phoneme- and speaker-independent model," Speech Communication 17(1995) 109-116.

[3] Bing Xiang, Berger, T. "Efficient text-independent speaker verification with structural Gaussian mixture models and neural network," Speech and Audio Processing, IEEE Transactions on, Volume: 11, Issue: 5, Sept. 2003, pp447-456.

[4] X. Huang, A. Acero and H. Hon, ***Spoken Language Processing***, Prentice Hall PTR, 2001

[5] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," IEEE Trans. Speech Audio Processing, vol. 9, Mar. 2001.
[6] Taeyoon Kim and HANSEOK KO, " Uttrance Verification Under Distributed Detection and Fusion Framework" , Eurospeech 2003, pp. 889~892, Sep, 2003.
[7] Hansang Park, B.A., M.A., "Temporal ans spectral Characteristics of Korean Phonation Types," Doctor of philosophy degree thesis, The university of Taxas at Austin, August, 2002.
[8] Willian J. Hardcastle and John laver, "The Handbook of Phonetic Sciences," Blackwell publishers Ltd, 1997.
[9] Jounghoon Beh and Hanseok Ko, "A Novel Spectral Subtraction Scheme For Robust Speech Recognition: Spectral Subtraction using Spectral Harmonics of Speech," ICME 2003, III 633 ~ III 636, Jul, 2003