Evaluation of HMM-Based Feature Compensation Applied to a Finite-State Grammar Speech Recognizer

Akira Sasou[1], *Hiroaki Kojima*[1], *Shuichi Itabashi*[1,2], *Kazuyo Tanaka*[1,3]

[1]National Institute of Advanced Industrial Science and Technology (AIST)
[2]National Institute of Informatics (NII)
[3]University of Tsukuba, Institute of Library and Information Science
{a-sasou,h.kojima,s.itabashi,kaz.tanaka}@aist.go.jp

Abstract

In this paper, we describe a Hidden Markov Model (HMM)-based feature-compensation method. The proposed method compensates for noise-corrupted features using the output probability density functions (pdfs) of clean acoustic HMMs provided to the recognizer in advance. In this way, the proposed method achieves model-based feature compensation without any extra parameters. In compensating for the features, the output pdfs are adaptively weighted according to forward path probabilities. Because of this, the proposed method can minimize degradation of feature compensation accuracy due to temporary changes in the noise environment. We applied the proposed feature compensation to a finite-state grammar speech recognizer and evaluated it by conducting hundred-word recognition experiments in noisy environments. The experimental results indicate that, compared with the baseline performance, the proposed feature compensation method achieved a 12.06 % improvement in accuracy on an overall average.

1. Introduction

One approach to realize noise-robust speech recognition is to apply feature compensation of noise-corrupted speech during front-end processing[1, 2]. Segura et al. demonstrated the effectiveness of the feature-compensation method based on a Gaussian Mixture Model (GMM). This method estimates the distortion from a noise-corrupted speech feature based on the GMM and generates a compensated speech feature by subtracting the estimated distortion from the corrupted speech feature[3]. The GMM is trained in advance with clean-speech features. In the feature-compensation process, the GMM adapts to the noisy environment by using several frames prior to an utterance, which are expected to be noise-only frames. Because the adapted GMM is well matched to noise-corrupted speech features, the method can adequately compensate for the corrupted speech features in a stationary noise environment. However, because the noise characteristics change in a non-stationary noise environment, mismatches between the adapted GMM and a current noise-corrupted speech feature occur. Because of such mismatches, the post probabilities of the Gaussians, which should not contribute to feature compensation, are over-estimated. This degrades the feature-compensation accuracy.

To achieve robust speech recognition in a non-stationary noise environment, K.Yao[4] proposed an effective method that separates the noise effects into stationary and residual, and estimates the residual noise parameters at each time with a sequential EM algorithm. However, the computation cost is very high because the method needs a number of iterations at each time in order to converge the residual noise parameters. Recently, in order to reduce the computation cost, a particle filter-based sequential estimation method has been adopted in some methods, e.g. [5, 6].

We proposed a Hidden Markov Model (HMM)-based feature-compensation method[7]. The proposed method assumes that the non-stationary noise can be divided into a stationary component and some temporary, changing components. The effects of the stationary component are eliminated by adapting the output pdfs of the HMMs to the stationary component of the noise in the same way as the GMM-based methods. The proposed method can eliminate the degradation of feature-compensation accuracy due to temporary, changing components by evaluating the post probability of each Gaussian, taking into account forward path probabilities. Even if the current noise characteristics have temporarily changed, the proposed method can enhance the post probabilities of the Gaussians, which should contribute to feature compensation. In [7], we applied the proposed feature-compensation method to a whole-word-model-based speech recognizer and showed the feasibility of the proposed method using the AURORA2 database[9]. We also demonstrated the effectiveness of the proposed method in a non-stationary noise environment such as a transient-pulse noise environment.

In comparison with methods described in [4, 5, 6], our method does not need to estimate the residual noise parameters using such a complex processing as the sequential EM algorithm or the particle filter-based method, and simply applies weights to the pdfs in order to compensate for the residual noise effects in evaluation of posterior probabilities of Gaussian distributions. The weights are evaluated based on the forward path probabilities of the Viterbi algorithm. Our method is thus very simple and can easily be implemented with an existing speech recognizer, which is one of our method's advantages. In this study, in order to confirm the feasibility of the proposed method under a more generalized scenario, we applied the proposed feature compensation to a finite-state grammar speech recognizer and evaluated it by conducting hundred-word recognition experiments in noisy environments.

2. Feature compensation based on HMM

Assuming that the speech and noise signals are uncorrelated, the filter bank energy (FBE) of the noisy speech $\mathbf{x}_b$ can be represented as a function of the clean speech $\mathbf{s}_b$ and the noise $\mathbf{n}_b$:

$$\mathbf{x}_b = \mathbf{s}_b + \mathbf{n}_b. \tag{1}$$

This relation yields the expression of the noisy speech $\mathbf{x}_l$ in the log-FBE domain:

$$\begin{aligned}\mathbf{x}_l &= \log(\mathbf{x}_b) \\ &= \mathbf{s}_l + \log\left[\mathbf{1} + \exp(\mathbf{n}_l - \mathbf{s}_l)\right] \end{aligned} \quad (2)$$

where the subscript l denotes that the expression is in the log-FBE domain. Similarly, the noisy speech $\mathbf{x}$ in the cepstral domain can be represented by:

$$\begin{aligned}\mathbf{x} &= \mathbf{C} \cdot \mathbf{x}_l \\ &= \mathbf{s} + \mathbf{g}(\mathbf{s}, \mathbf{n}) \end{aligned} \quad (3)$$

where $\mathbf{g}(\mathbf{s}, \mathbf{n})$ represents a distortion of the noisy speech in the cepstral domain, which is given by:

$$\mathbf{g}(\mathbf{s}, \mathbf{n}) = \mathbf{C} \cdot \log\left[\mathbf{1} + \exp\left\{\mathbf{C}^{-1} \cdot (\mathbf{n} - \mathbf{s})\right\}\right]. \quad (4)$$

Here $\mathbf{C}$ and $\mathbf{C}^{-1}$ denote the discrete cosine transform (DCT) matrix and its inverse transform matrix, respectively, and $\mathbf{s}$ and $\mathbf{n}$ are the clean speech and noise in the cepstral domain.

In the following, HMM parameters are estimated from a clean training data set, S denotes the entire set of HMM states, π_s denotes the initial probability of the sth state, and a_{jq} denotes the transition probability from the jth state to the qth state. The output pdf of the jth state $b_j(\mathbf{s})$ is given by:

$$b_j(\mathbf{s}) = \sum_{m=1}^{M} w_{jm} \mathcal{N}(\mathbf{s}; \mu_{jm}, \mathbf{\Sigma}_{jm}) \quad (5)$$

where M is the number of Gaussian distributions and $\mathbf{\Sigma}_{jm}$ is a diagonal matrix.

The proposed feature-compensation method assumes that a distortion of the noisy speech feature in the cepstral domain can be divided into a stationary and a non-stationary distortion components. The temporal trajectory of the non-stationary distortion component is assumed to be zero almost everywhere but temporarily changes. The stationary distortion component is absorbed by adding the estimated stationary distortion component to the expectation value of each Gaussian distribution in the output probability density functions (pdfs) of the clean speech's HMMs. The proposed method eliminates the degradation of feature-compensation accuracy caused by the non-stationary distortion component by evaluating the posterior probability of each Gaussian distribution, which is adaptively weighted based on the forward path probability.

The first step in the feature-compensation process is to generate a copy of the output pdf for each state. The second step is to evaluate the stationary distortion component, which is evaluated as the expectation value of the distortion vectors. The expectation value is evaluated using the N noise-only frames prior to each utterance. The third step is to adapt each Gaussian distribution in the copied clean speech pdfs to the noisy speech. We take into account only the expectation of each Gaussian distribution. The diagonal covariance matrix in each noise-adapted Gaussian distribution is assumed to be the same as the covariance matrix of the clean speech. The adapted Gaussian distribution is generated by adding the expectation value of the distortion vectors to the expectation value of the clean speech. The pdfs adapted to the noisy speech are used to evaluate the posterior probability of each noise-adapted Gaussian distribution. The original clean speech pdfs are used to evaluate the output probability of the compensated speech feature.

Let $\hat{\mathbf{x}}$ denote an element vector consisting of the part of an observed feature vector $\mathbf{x}$ that is to be compensated. The elements of the vector $\mathbf{x}$, except the vector $\hat{\mathbf{x}}$, remain the same. In our settings, the vector $\mathbf{x}$ has a 39-dimensional vector, which consists of a 13-dimensional base mel-frequency cepstral coefficient (MFCC) and its delta and delta-delta components, and the element vector $\hat{\mathbf{x}}$ corresponds to the base MFCC. In addition, let $\mathcal{N}(\hat{\mu}_{jm}, \hat{\mathbf{\Sigma}}_{jm})$ represent the Gaussian distribution of the clean speech $\hat{\mathbf{s}}$ corresponding to the element vector $\hat{\mathbf{x}}$ of the noisy speech.

The clean speech $\hat{\mathbf{s}}$ is assumed to conform to the Gaussian distribution $\mathcal{N}(\hat{\mu}_{jm}, \hat{\mathbf{\Sigma}}_{jm})$. Every noise-adapted Gaussian distribution $\mathcal{N}(\hat{\mathbf{m}}_{jm}, \hat{\mathbf{Z}}_{jm})$ is then given by:

$$\begin{aligned}\hat{\mathbf{m}}_{jm} &= E[\hat{\mathbf{x}}]_{\hat{\mathbf{s}},\hat{\mathbf{n}}} \\ \hat{\mathbf{Z}}_{jm} &= \hat{\mathbf{\Sigma}}_{jm}. \end{aligned} \quad (6)$$

where $E[\hat{\mathbf{x}}]_{\hat{\mathbf{s}},\hat{\mathbf{n}}}$ represents the expectation of $\hat{\mathbf{x}}$ with regard to both $\hat{\mathbf{s}}$ and $\hat{\mathbf{n}}$. The $\hat{\mathbf{n}}$ is the noise vector corresponding to the element vector $\hat{\mathbf{x}}$ of the noisy speech. The expectation $\hat{\mathbf{m}}_{jm}$ can be evaluated as follows. By applying a first-order Taylor series expansion around $\hat{\mu}_{jm}$ to Eq.(3) and evaluating the expectation with regard to the clean speech $\hat{\mathbf{s}}$, we can obtain the following formula:

$$E[\hat{\mathbf{x}}]_{\hat{\mathbf{s}}} = \hat{\mu}_{jm} + \mathbf{g}(\hat{\mu}_{jm}, \hat{\mathbf{n}}) \quad (7)$$

The actual distribution of the noise $\hat{\mathbf{n}}$ is not known. Thus, we evaluate the expectation of $E[\hat{\mathbf{x}}]_{\hat{\mathbf{s}}}$ with regard to $\hat{\mathbf{n}}$ as follows:

$$\hat{\mathbf{m}}_{jm} = E[\hat{\mu}_{jm} + \mathbf{g}(\hat{\mu}_{jm}, \hat{\mathbf{n}})]_{\hat{\mathbf{n}}} = \hat{\mu}_{jm} + \hat{\mathbf{d}}_{jm} \quad (8)$$

$$\begin{aligned}\hat{\mathbf{d}}_{jm} &= E[\mathbf{g}(\hat{\mu}_{jm}, \hat{\mathbf{n}})]_{\hat{\mathbf{n}}} \quad (9) \\ &\approx \frac{1}{N}\mathbf{C} \cdot \sum_{t=1}^{N} \log\left[\mathbf{1} + \exp\left\{\mathbf{C}^{-1} \cdot (\hat{\mathbf{n}}_t - \hat{\mu}_{jm})\right\}\right] \end{aligned}$$

where $\hat{\mathbf{n}}_t, (t = 1, \cdots, N)$, are noise vectors.

The proposed method utilizes each forward path probability $\alpha(s, t)$ of the best path to the sth state at time t for the feature compensation process. Every forward path probability $\alpha(s, t)$ is recursively calculated using the Viterbi algorithm as follows. First, the initial probabilities are set to $\alpha(s, 0)$.

$$\alpha(s, 0) = \log(\pi_s) \quad \text{for } \forall s \in S \quad (10)$$

For each time instant t and for each state s, the forward path probabilities are then calculated as follows:

$$\alpha(s, t) = \max_{j \in S} \alpha(j, t-1) + \log\{a_{js} \cdot b_j(\mathbf{y}_t)\} \quad (11)$$

where the vector $\mathbf{y}_t$ represents the compensated feature vector. At the last time instant T, the most likely final state is selected according to the following equation:

$$\log P_{\max} = \max_{s \in S_F} \alpha(s, T) \quad (12)$$

where S_F represents the set of final states.

Each compensated-feature vector $\mathbf{y}_t$ is evaluated according to the following procedures, which are executed for each time instant t before the forward path probabilities are updated using Eq. (11). First, using the previously calculated forward path probabilities, we evaluate weights $\alpha'(s, t-1)$ to be applied to the noise-adapted output pdfs of all the states.

$$\alpha'(s, t-1) = \exp\left\{\frac{\alpha(s, t-1)}{t}\right\} \quad (13)$$

K.Yao[8] adopted an approximation of the posterior probability of the state sequence, which is used for estimation of the residual noise parameters. The approximated posterior probability can be obtained by normalizing the joint likelihood of the observation sequence and state sequence with respect to the sum of those from all active partial state sequences in the recognition stage. Our method does not compensate for the non-stationary distortion component, only the stationary distortion component. Thus, in our method, the approximation formula adopted in [8] does not always give us correct posterior probability because of the remaining distortion in the compensated speech feature. In order to overcome this problem, we adopted the following approximation of posterior probability $P(j,m)$ of each noise-adapted Gaussian distribution $\mathcal{N}(\hat{\mathbf{m}}_{jm}, \hat{\mathbf{Z}}_{jm})$:

$$P(j,m) = \frac{\alpha'(j,t-1) w_{jm} \mathcal{N}(\hat{\mathbf{x}}_t; \hat{\mathbf{m}}_{jm}, \hat{\mathbf{Z}}_{jm})}{\sum_{s \in S} \sum_{n=1}^{M} \alpha'(s,t-1) w_{sn} \mathcal{N}(\hat{\mathbf{x}}_t; \hat{\mathbf{m}}_{sn}, \hat{\mathbf{Z}}_{sn})}. \tag{14}$$

The compensated element vector $\hat{\mathbf{y}}_t$ is given by:

$$\hat{\mathbf{y}}_t = \hat{\mathbf{x}}_t - \sum_{j \in S} \sum_{m=1}^{M} P(j,m) \hat{\mathbf{d}}_{jm}. \tag{15}$$

When the noise has no stationary component, such as transient pulse noises, the stationary distortion components are ideally equal to zero: $\hat{\mathbf{d}}_{jm} = \mathbf{0}$. Hence, the feature compensation of Eq. (15) becomes useless. In this case, we need to find another feature compensation formula without using $\hat{\mathbf{d}}_{jm}$. Eq.(15) can be reformulated to

$$\hat{\mathbf{y}}_t = \sum_{j \in S} \sum_{m=1}^{M} P(j,m) \left\{ \hat{\mathbf{x}}_t - \hat{\mathbf{d}}_{jm} \right\}, \tag{16}$$

where $(\hat{\mathbf{x}}_t - \hat{\mathbf{d}}_{jm})$ can be regarded as an estimate of the clean speech feature assuming that the clean speech feature was emitted from the mth Gaussian distribution in the jth state's pdf. The expectation vector of the Gaussian distribution can be also regarded as an estimate of the clean speech feature, because the expectation vector is the most probable one in the distribution. We thus obtain the following feature compensation formula.

$$\hat{\mathbf{y}}_t = \sum_{j \in S} \sum_{m=1}^{M} P(j,m) \hat{\mu}_{jm}. \tag{17}$$

The compensated feature vector $\mathbf{y}_t$ is finally obtained by combining the compensated element vector $\hat{\mathbf{y}}_t$ with the elements of the observed feature vector $\mathbf{x}_t$ except the element vector $\hat{\mathbf{x}}_t$.

3. Experiments

3.1. Experimental set-up

We first developed a normal decoder based on a finite-state, grammar-restricted, token-passing algorithm. The clean acoustic models consist of 7946 Phonetic Tied Mixture (PTM) triphones. Each triphone has three states, and the output pdf of each state has 64 Gaussian mixtures. The acoustic models were trained using clean speech data from the continuous speech corpus of Japanese Newspaper Article Sentences (JNAS) [10]. The sampling frequency of our speech recognition system is 8kHz, so we converted the sampling frequency of the JNAS speech data from 16kHz to 8kHz. Procedures for MFCC calculation are as follows. The frame length is set at 25 ms and the period at 10 ms. The Fast Fourier Transform (FFT) is calculated after pre-emphasizing by $1 - 0.97z^{-1}$. The inner products between the squared amplitudes of the FFT coefficients and the triangle windows of the mel filter bank are calculated to generate the mel filter-bank energy (mel-FBE) feature. The MFCC is then obtained by applying DCT to the natural logarithm of the mel-FBE feature. The MFCC is a 13-dimensional vector including the 0th coefficient. The delta and the delta-delta features are evaluated. Combining these features generates the 39-dimensional feature vector.

We then applied the proposed feature compensation to the normal decoder. The proposed method compensates the features according to the procedures described in Section 2. Because the feature-compensation process is embedded in the decoding process, these two processes end simultaneously. In addition, the two processes are executed using only the HMMs provided to the decoder in advance. Therefore, unlike the GMM-based methods, the proposed method needs no extra parameters to make the decoder robust against the effects of additive noises. However, one drawback of this method is the use of HMMs that have not been trained from the compensated features to calculate the output probabilities of the compensated features.

We used 492 Japanese words from the ETL speech database [11] in the evaluations. Since each VCV/CVC balanced word set contains text and speech waves as spoken by 10 male speakers, the test set actually contained 4920 words.

The speech recognitions were carried out using a dictionary file and a grammar file. The finite-state grammar defines a network of word categories. The grammar file written in EBNF is as follows.

```
$words = aNe- | aNgya | aNnyui |
         ... | zuijyu-;
( [silB] $words [silE] )
```

The word dictionary has a tree structure. Each word in the dictionary file was represented with the triphones. In addition to the decoder with the feature compensation, we conducted the same experiments using the normal decoder that has no feature compensation, in order to evaluate the baseline performance.

3.2. Evaluations in environmental noises

We used 13 noise sources from the ambient noise database collected by NTT-AT (airport, amusement facility, aircraft, construction site, exhibition hall, factory, lobby, office, restaurant, shopping mall, train, train station, and Street). These noise sources, recorded in actual noise environments, contain not only a stationary noise component but also many non-stationary noise components. We converted the sampling frequency of both noise and speech data to 8kHz and added them at seven SNR levels (Clean, 20dB, 15dB, 10dB, 5dB, 0dB, -5dB). In this experiment, the features of noise-corrupted speech data were compensated for by Eq.(15).

3.3. Evaluation in a transient pulse noise environment

In the experiment conducted in a transient pulse noise environment, we used wooden collision sound sources recorded in the Real World Computing Partnership (RWCP) Sound Scene Database [12]. We generated noise-corrupted speech data by adding a transient pulse every 125 ms to clean speech data. Only C0,ΔC0, and $\Delta\Delta$C0 of noise-corrupted features were compensated for by Eq.(17), because a transient pulse noise affects mostly only those coefficients.

Table 1: Accuracy averaged in a range from SNR20dB to SNR0dB.

	Arprt	Amusement	Acrft	Cnst st	Exhib.
Proposed	66.16	59.95	70.43	77.16	65.28
Baseline	56.96	48.74	56.46	67.77	47.38
	Fctry	Lobby	Office	Rstnt	Shp Mall
Proposed	60.97	66.39	76.43	60.45	56.00
Baseline	45.72	54.07	65.64	46.56	45.53
	Train	Trn Stn	Street	Pulse	Avg.
Proposed	90.87	66.99	75.93	53.27	67.59
Baseline	79.69	57.41	67.19	38.35	55.53

3.4. Experiment Results

The experiment results are presented in Figs 1 to 14. In each figure, a dotted line indicates the performance of the normal decoder and a solid line, that of the feature-compensation decoder. Table 1 represents the accuracies averaged in a range from SNR20dB to SNR0dB for each condition. The table illustrates that, compared with the baseline performance, the proposed feature compensation method achieved a 12.06 % improvement in accuracy on an overall average.

4. Conclusion

In this paper, we presented an HMM-based feature-compensation method that can minimize degradation of feature-compensation accuracy due to temporary changes in the noise environment by compensating for features based on distributions adaptively weighted according to forward path probabilities. We also applied the proposed feature compensation to a finite-state grammar speech recognizer and evaluated it by conducting word-recognition experiments in noisy environments. The experiment results revealed that the proposed method can improve the accuracy on an overall average by 12.06 % in comparison with the baseline system.

5. References

[1] S.F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. ASSP-33 27, pp.113-120, 1979.

[2] B.S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," J. Acoust. Soc. Am., 55, pp.1304-1312, 1974.

[3] J.C. Segura, "A. de la Torre, M.C. Benitez, A.M. Peinado, "Model-based compensation of the additive noise for continuous speech recognition. Experiments using the AURORA II database and tasks," Proc. of Eurospeech 2001, Vol.I, pp.221-224, 2001.

[4] K.Yao, B.E.Shi, S.Nakamura, Z.Cao, "Residual Noise Compensation by a Sequential EM algorithm for Robust Speech Recognition in Nonstationary Noise," Proc of ICSLP2000, 2000, pp.760–763.

[5] K.Yao, S.Nakamura, "Sequential Noise Compensation by Sequential Monte Carlo Method," Proc of NIPS'01, 2001, pp.1205–1212.

[6] M.Fujimoto, S.Nakamura, "Particle Filter Based Non-Stationary Noise Tracking for Robust Speech Recognition," Proc. of ICASSP2005, Vol.I, 2005, pp.257–260.

[7] A.Sasou, F.Asano, K.Tanaka, S.Nakamura, "HMM-Based Feature Compensation Method: An Evaluation Using the AURORA2," Proc. of ICSLP2004, pp.121-124, Oct. 2004.

[8] K.Yao, K.Paliwal, S.Nakamura, "Noise adaptive speech recognition based on sequential noise parameter estimation," Speech Communication 42, pp.5–23.

[9] D.Pearce, "Aurora project: Experimental framework for the performance evaluation of distributed speech recognition front-ends," ISCA ITRW ASR2000, Sep.2000.

[10] K.Itoh, M.Yamamoto, K.Takeda, T.Takezawa, T.Matsuoka, T.Kobayashi, K.Shikano, and S.Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," J. Acoust. soc. Japan (E), Vol.20, No.3, pp.199-206, March, 1999.

[11] K.Tanaka, S.Hayamizu, K.Ohta, "The ETL speech database for speech analysis and recognition research," Proc. of ICSLP-90, pp.1101-1104,1990.

[12] S. Nakamura et al, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," Proc of 2nd International Conference on Language Resources & Evaluation, Athens, June 2000.

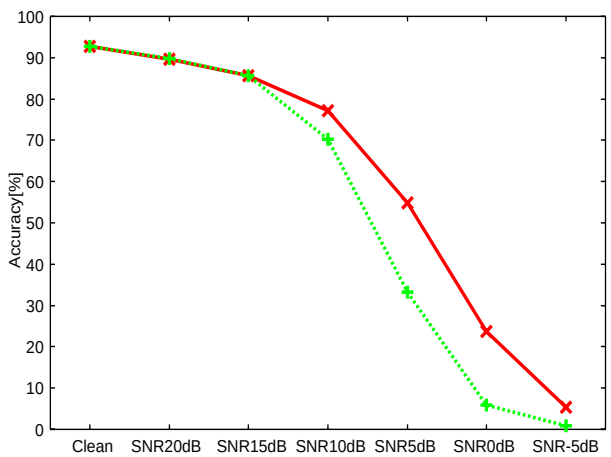

Figure 1: Airport

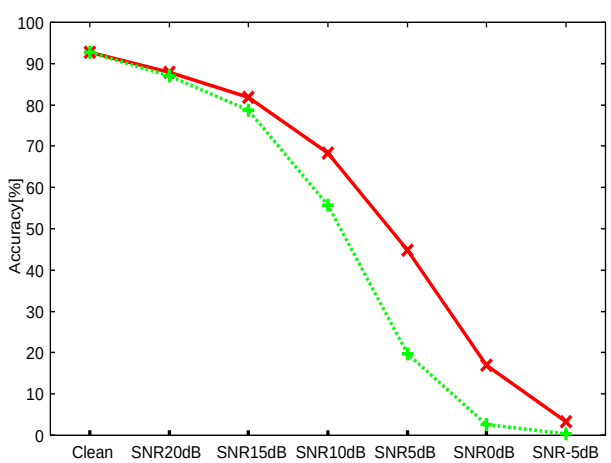

Figure 2: Amusement facility

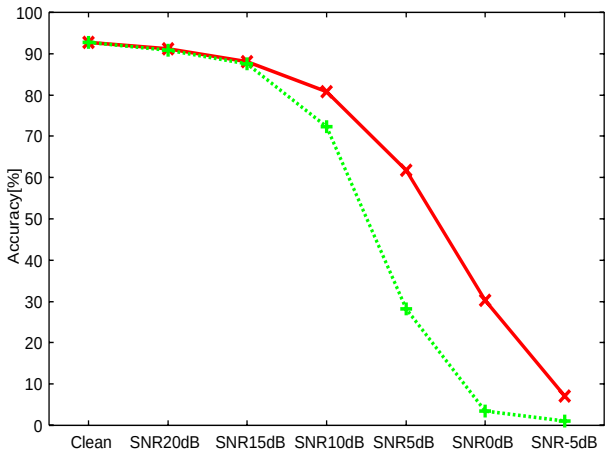

Figure 3: Aircraft

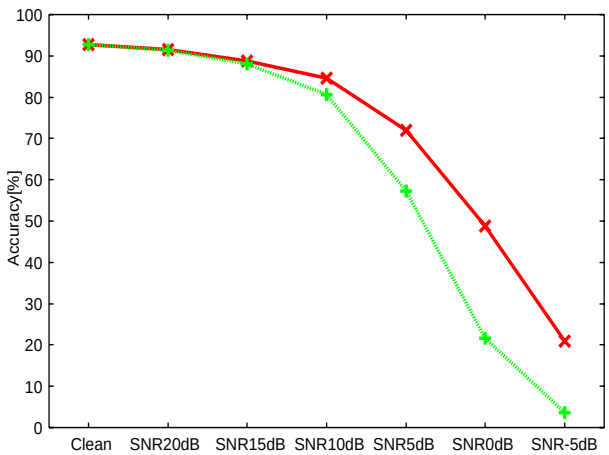

Figure 4: Construction site

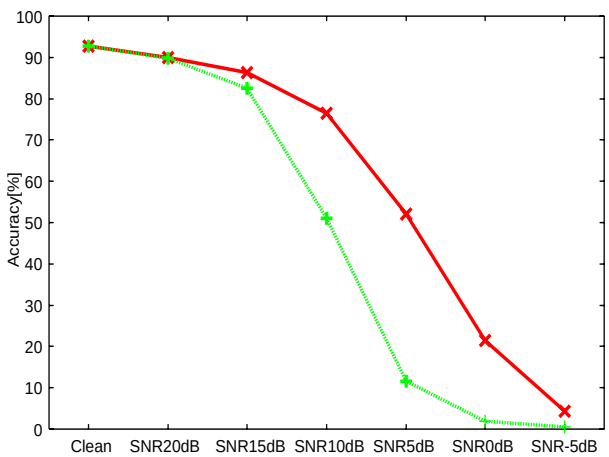

Figure 5: Exhibition hall

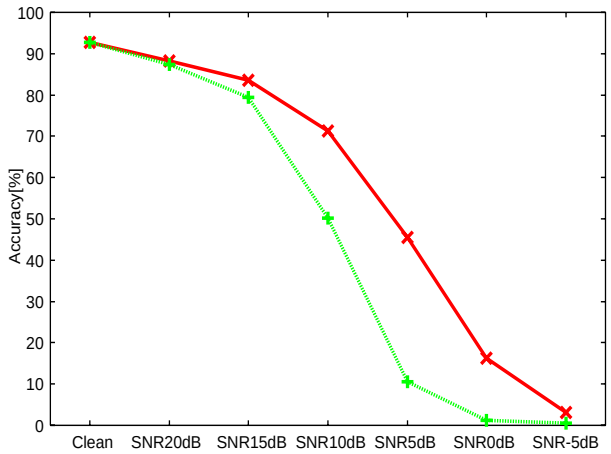

Figure 6: Factory

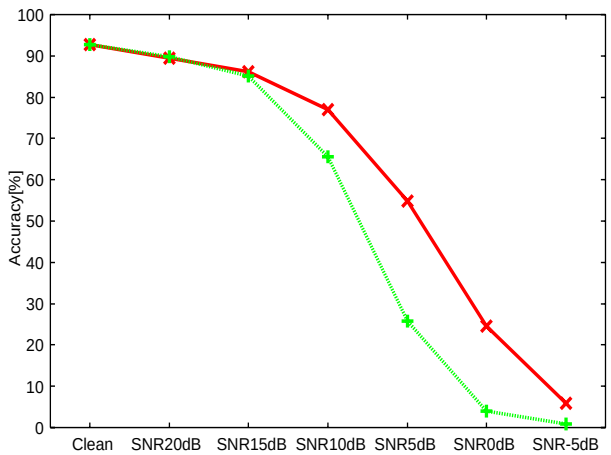

Figure 7: Lobby

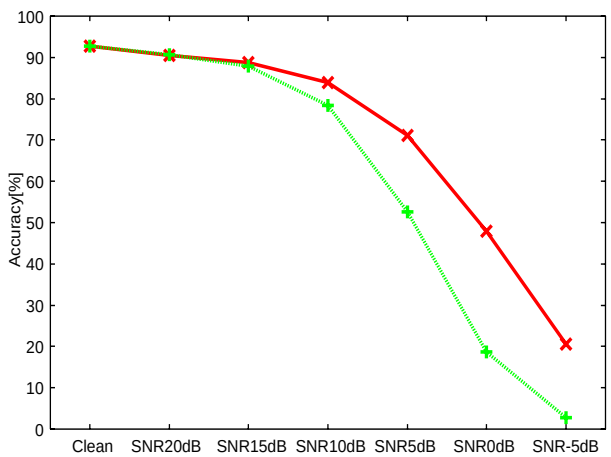

Figure 8: Office

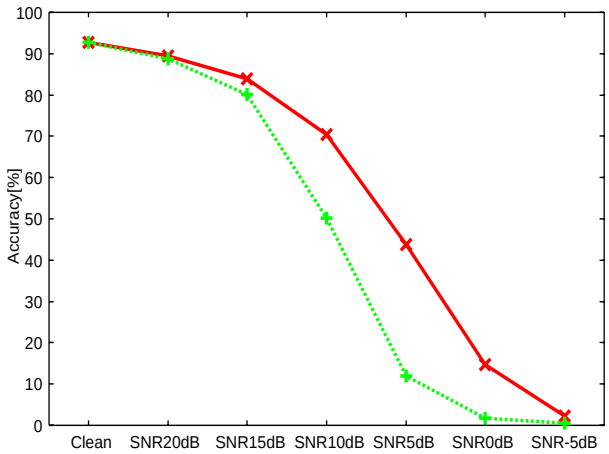

Figure 9: Restaurant

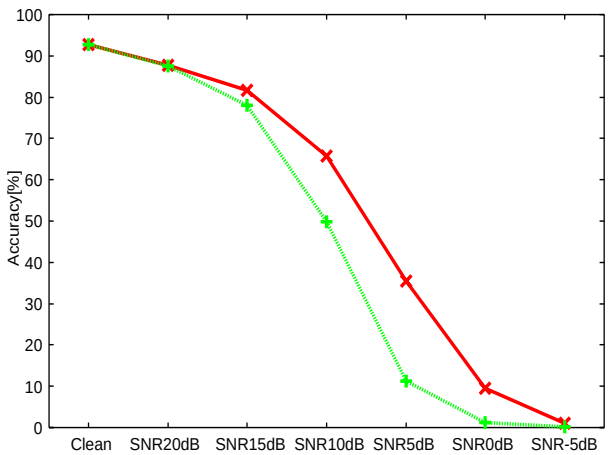

Figure 10: Shopping mall

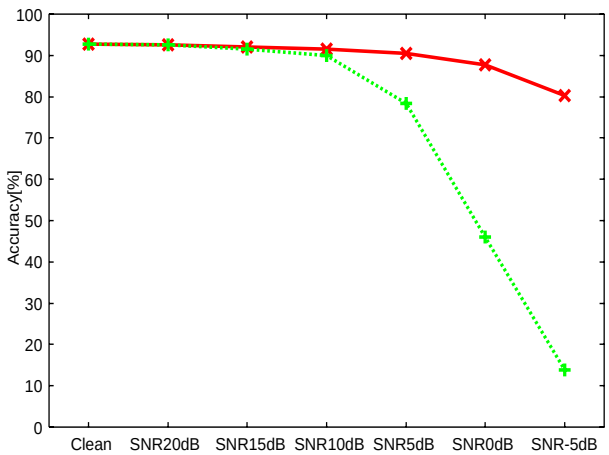

Figure 11: Train

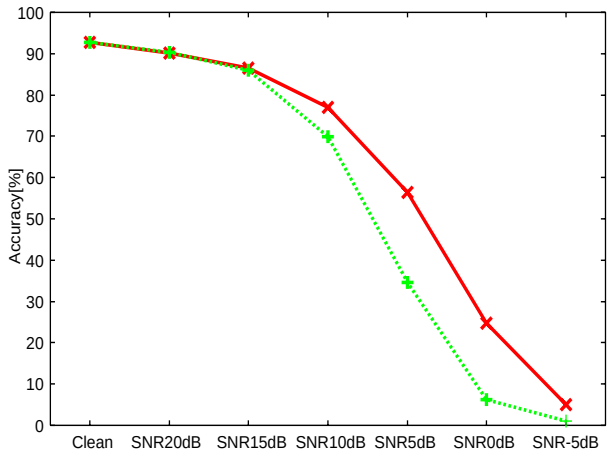

Figure 12: Train station

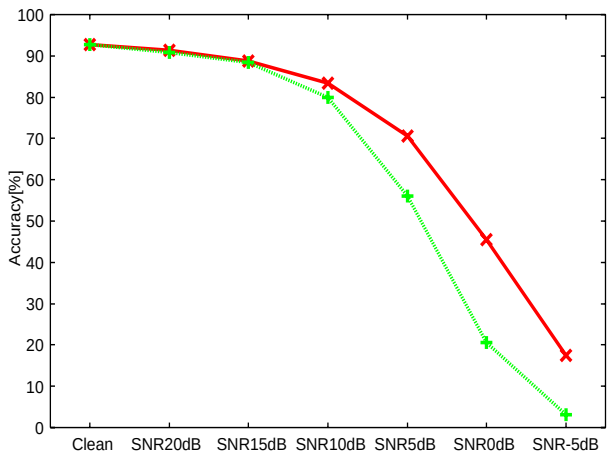

Figure 13: Street

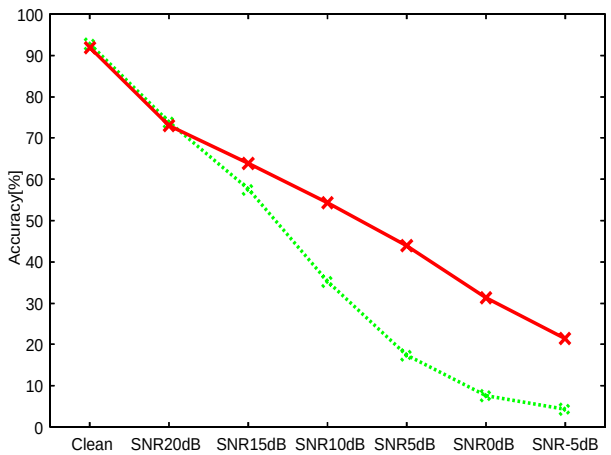

Figure 14: Pulse