JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Research on Noise Reduction Algorithm Based on Combination of LMS Filter and Spectral Subtraction

Danyang Cao*, Zhixin Chen*, and Xue Gao*

Abstract

In order to deal with the filtering delay problem of least mean square adaptive filter noise reduction algorithm and music noise problem of spectral subtraction algorithm during the speech signal processing, we combine these two algorithms and propose one novel noise reduction method, showing a strong performance on par or even better than state of the art methods. We first use the least mean square algorithm to reduce the average intensity of noise, and then add spectral subtraction algorithm to reduce remaining noise again. Experiments prove that using the spectral subtraction again after the least mean square adaptive filter algorithm overcomes shortcomings which come from the former two algorithms. Also the novel method increases the signal-to-noise ratio of original speech data and improves the final noise reduction performance.

1. Introduction

During the speech communication, the speech signal is always inevitably contaminated by other invalid signals which belong to surrounding environment. Speech enhancement is an effective method to solve noise pollution. The goal of speech enhancement is to remove the noise components as much as possible, preserve the speech components and improve the quality of the speech signal finally [1]. After decades of explorations and experiments, researchers at home and abroad have achieved some basic research results based on time-frequency domain. Boll [2] assumed that original speech signal was not correlated with smooth additive noise, and proposed the spectral subtraction algorithm which had less calculation and was easier to implement. Tsoukalas et al. [3] proposed a speech enhancement method which used the auditory masking effect of ear to help block noise signals which contained less energy. Lim and Oppenheim [4] proposed wiener filtering algorithm to improve the quality of speech signal when the signal was stable for a short time. Widrow and Hopf [5] presented the least mean square (LMS) algorithm and used adaptive filter do best estimate to noisy signal when they were studying the adaptive theory. Based on wavelet theory, Donoho [6] put forward wavelet threshold algorithm to deal with the detail coefficients of the wavelet space and achieved noise reduction.

Because the variety of noises in the real environment have different characteristics, it is difficult to find

one universal speech enhancement algorithm which is suitable for eliminating all kinds of noise.

Based on the comprehensive consideration, we select spectral subtraction algorithm which has higher computation efficiency and LMS algorithm which has better noise reduction effect to conduct in-depth research. And we propose a new noise reduction algorithm based on the combination of LMS adaptive filter and spectral subtraction. The new algorithm uses self-adjustment and signal tracking to make up for the defects which come from original algorithms, while guaranteeing high computation efficiency. It can improve the signal-to-noise ratio (SNR) of original speech file and the accuracy in the speech recognition application.

2. Related Work

2.1 Introduction of LMS Adaptive Filter Algorithm

LMS algorithm is a typical algorithm among many adaptive filtering noise reduction methods. After each calculation, the self-adaptive filter can estimate the filter parameters at the next moment according to the mean squared error, minimum variance or the output filter at this moment [7]. Adjusting the estimate can be dynamically adapted to the characteristics of the changeable voice signal. It can also adjust the parameters at any time to optimize the performance and achieve the purpose of improving the SNR ultimately [8].

The LMS implements filter noise reduction based on the mean square error [9]. It generally includes the following five steps:

Step 1: Set $w(k)$ as the initial value of the filter, μ is the convergence factor to control the rate of convergence:

$$W(0) = 0 \qquad 0 < \mu < \frac{1}{\lambda_{max}} \tag{1}$$

Step 2: Do filtering and compute the actual output value of filter:

$$y(k) = W^T(k)X(k) \tag{2}$$

Step 3: Compute the error, and $d(k)$ is the filter's output value at the previous moment:

$$e(k) = d(k) - y(k) \tag{3}$$

Step 4: Compute filter's coefficients at the next moment:

$$W(k+1) = W(k) + \mu e(k)X(k) \tag{4}$$

Step 5: If k is not the last moment, then $k=k+1$ and repeat Step 2–Step 4. Otherwise, get the output value after filtering.

2.2 The Filtering Delay Problem in LMS Adaptive Filter Algorithm

The self-adjustment and signal tracking are necessary to LMS algorithm, thus it can achieve optimal filtering generally. There are also some problems still exiting in LMS algorithm. For example, some details

about the noise data can only be got by multi-channel acquisition. Otherwise, noise must be estimated firstly when only the single-channel acquisition is provided. Also if the distribution of spectrum in noise is not uniform, poor filtering will be led. The biggest problem is that many noises still exist in the initial part of speech file due to the filtering delay. Although modifying the parameters can make some improvements, the modified parameters will affect the filtering in the meantime. Among the practical applications, the self-adaptive filtering algorithm based on LMS is more common [10].

The comparison after LMS adaptive filter noise reduction algorithm to the speech "0" is as shown in Fig. 1.
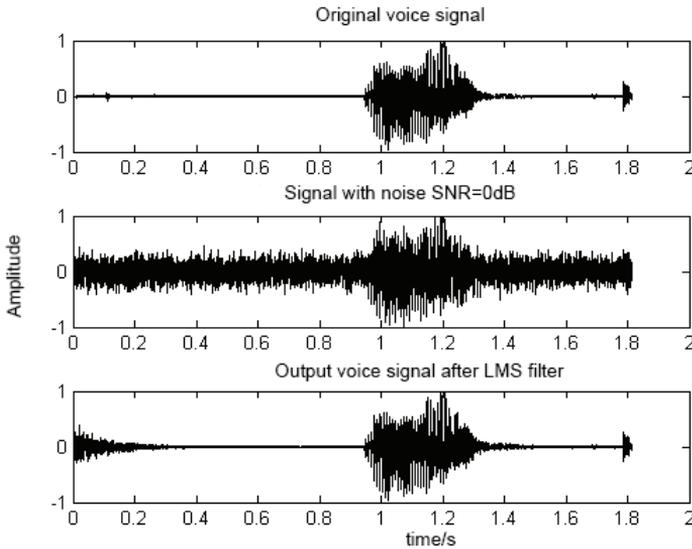

Fig. 1. Comparison of speech "0" after LMS filter noise reduction algorithm.

Three parts are contained in the Fig. 1. It includes the pure speech signal, speech signal containing noise and the output signal after LMS filtering. It can be seen that the noise reduction effect is very well after filtering. However, the delay problem causes a large amount of noise data to be retained in the initial position of the speech. Obviously, if important speech information exists in the initial position, it will cause blur and errors in the speech recognition.

2.3 Introduction of Spectral Subtraction Algorithm

Based on short-term spectral estimation, spectral subtraction algorithm is often used in the speech enhancement. Spectral subtraction algorithm assumes that the real speech signal is independent with additive noise from the environment. So just calculate the power spectrum of speech and average power spectrum of noise respectively and then do the subtraction with each other. Therefore, the spectrum of relatively pure speech signal will be got [11].

The traditional spectral subtraction algorithm generally includes the following steps. Supposed that the speech signal $x(n)$ will be changed to $x_i(m)$ at i^{th} frame after framing. And N is the number of frames. Firstly, do DFT (discrete Fourier transform) to each frame and get the frequency domain value. The formula is as follows:

$$X_i(k) = \sum_{m=0}^{N-1} x_i(m)\exp\left(j\frac{2\pi mk}{N}\right) \qquad k = 0,1,...,N-1 \tag{5}$$

After obtaining the frequency domain value of speech signal, calculate the amplitude and phase angle of each component. Amplitude and phase angle will be needed to restore the speech signal in the last. The amplitude is $|\hat{x}_i(k)|$, and the formula for calculating phase angle is as follows:

$$X_{angle}^i(k) = \arctan\left[\frac{lm(X_i(k))}{\mathrm{Re}(X_i(k))}\right] \tag{6}$$

Next, the number of frames in the initial part of speech file is assumed as *NIS*, and this part has ambient noise basically. Then the average energy of the noise segment can be calculated just as follows:

$$D(k) = \frac{1}{NIS}\sum_{i=1}^{NIS}|X_i(k)|^2 \tag{7}$$

The core of spectral subtraction algorithm is to subtract the average energy of the noise from the energy of the mixed speech. When the mixed speech energy is greater than or equal to a certain number of noise energy, they are subtracted with each other. When the mixed speech energy is less than a certain number of energy, a certain degree of compensation is got for the mixed energy. The formula is as follows:

$$|X_i(k)|^2 = \begin{cases} |X_i(k)|^2 - \alpha \times D(k) & |X_i(k)|^2 \geqslant \alpha \times D(k) \\ \beta \times D(k) & |X_i(k)|^2 < \alpha \times D(k) \end{cases} \tag{8}$$

After the above steps, using amplitude $|\hat{x}_i(k)|$ and the phase angle $x_{angle}^i(k)$ to do the IFT (inverse Fourier transform) can obtain the final output speech signal.

2.4 The Music Noise Problem in Spectral Subtraction Algorithm

Spectral subtraction algorithm is easy to be implemented and has less calculation. However, there are also some problems existing in the algorithm. For example, it does not analyze the distribution of the noise spectrum. So if the distribution is uneven, spectral subtraction algorithm will produce a lot of music noise during noise reduction. Because the algorithm is achieved by subtracting the average energy of the noise from the energy of the mixed speech, more noise is easy to be remained at the location where the noise is strong, and less noise will be compensated at the location where the noise is weaker. These residual noises constitute musical noise finally.

The comparison after spectral subtraction noise reduction algorithm to the speech "0" is as shown in Fig. 2.

It is shown from the Fig. 2 that spectral subtraction algorithm also has good noise reduction effect. And there are less noise remained in the initial part of speech file than LMS algorithm. But music noises remains in the entire speech file and causes that the overall noise reduction effect is slightly worse than the LMS algorithm. In order to reduce and remove music noise, scholars have come up with some other relevant algorithms to solve the problem, just like the probability spectrum subtraction [12], the multi-

template spectral subtraction [13], the weighting function spectral subtraction, and so on [14].
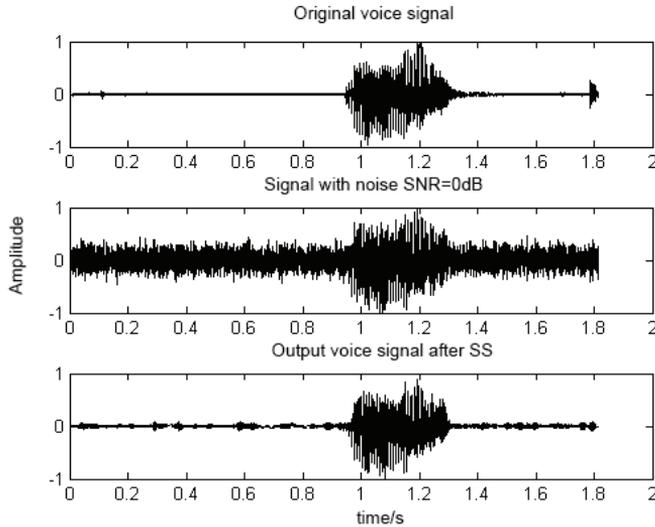

Fig. 2. Comparison of speech "0" after spectral subtraction noise reduction algorithm.

3. The Fusion of Algorithms

3.1 Introduction of LMSSS Algorithm

In order to deal with the music noise and filtering delay problem, we propose a least mean square and spectral subtraction (LMSSS) algorithm based on combination of other two algorithms. We first use LMS algorithm to reduce the average intensity of noise, and then add spectral subtraction algorithm to remove the noise in the initial position of speech file. Finally, we get the more stable and balanced speech enhancement algorithm, namely LMSSS algorithm.

The steps of LMSSS noise reduction algorithm are as follows. Do framing and mean normalization to the original speech signal and get the input speech signal x. Firstly, depended on a large number of experiments, select $M = 32$ $(M \neq 0)$, $\mu = 0.001$ $(0 < \mu < 1/\lambda\_max$) as the parameters. The M is the initial value of filter, and μ is the convergence factor. Then do the LMS filter and get intermediate output $temp\_signal$ at time k:

$$temp\_signal = [M(k) + \mu \times e(k) \times x]^T \times x \qquad (9)$$

And $e(k) = d(k) - y(k-1)$, $y(k-1)$ is the intermediate output at time $k - 1$, and $d(k)$ is the expected output.

After obtaining the LMS filtered output signal $temp\_signal$, repeatedly set α (subtracting factor) and β (gain factor) to compare the experiment results and select $\alpha = 4$, $\beta = 0.001$ as the best parameters. Then do DFT on the intermediate output signal $temp\_signal$ to obtain the amplitude $|x_i(k)|$ and phase angle $x_{angle}^i(k)$ of each frequency domain signal. And if the speech energy $|x_i(k)|^2$ is not less than α times average energy of the noise section e_noise, we get the final amplitude $|\hat{x}_i(k)|$:

$$|\hat{x}_i(k)| = \sqrt{|x_i(k)|^2 - \propto \times e\_noise} \tag{10}$$

Otherwise, we get the final amplitude $|\hat{x}_i(k)|$:

$$|\hat{x}_i(k)| = \sqrt{\beta \times e\_noise} \tag{11}$$

After above steps, we do the Fourier transform on the output amplitude $|\hat{x}_i(k)|$ and the phase angle $x_{angle}^i(k)$ to get final speech signal output y.

3.2 The Description of LMSSS Algorithm

Name of algorithm: Algorithm based on combination of LMS filter algorithm and spectral subtraction algorithm, also called LMSSS.

Input: Speech signal x and i^{th} frame's speech signal $x(i)$

Output: Speech signal y after noise reduction

Specific steps:

- Eliminate the direct current component of original speech signal and do normalization;
- Average the first *NIS* frames data as the estimated noise data *noise* ;
- Set parameters and do LMS filtering to get the filtered signal *temp _ signal* ;
- Set subtraction factor α and gain factor β in spectral subtraction algorithm;
- Do Fourier transform to the filtered speech signal and get frequency domain signal *y _ fft* ;
- Get the average energy *e _ noise* from the first *NIS* frames of frequency domain signal;
- Do spectral subtraction for each frame of frequency domain signal to get the result *temp _ y* ;
- Do Fourier transform to *temp _ y* and get the time domain signal *y* finally.

The pseudocode of LMSSS algorithm is shown as below.

Algorithm	Procedure LMSSS	
Input:	Speech signal: *x(i)*	
Output:	*y*	
1	*Double LMSSS (Double x[])*	
2	*{*	
3	*x(i) = x(i)-MEAN(x(i));*	Data preprocessing
	x(i) = x(i)/max(x)	Eliminate DC component, normalize
4	*noise = Average(x(1:NIS))*	Estimate the noise data
5	*M = 32, u = 0.001*	Set parameters
	h = adaptfilelms(M,u)	
6	*temp_signal = filter(h,x,noise)*	Do filtering
		Get *temp_signal* after filtering
7	*a = 4, b = 0.001*	Set parameters
8	*y_fft = fft(temp_signal)*	Get frequency domain signal *y_fft*
9	*e_noise = Average(y_fft(1:NIS))2*	Get the energy in first *NIS* frames

10	*temp_y = spec_sub(y_fft, e_noise)*	Do spectral subtraction
11	*y = ifft(temp_y)*	Restore the speech signal *y*
12	*return y;*	Output *y*
13	*}*	

3.3 The Analysis of LMSSS Algorithm

Spectral subtraction noise reduction algorithm will cause music noise, while LMS will cause bad noise reduction effect on the initial part of speech. Therefore, LMSSS first uses LMS algorithm, and then adds spectral subtraction to reduce remained noise again. After LMS noise reduction, the LMSSS reduces the average noise intensity to help latter added spectral subtraction algorithm avoid the music noise problem. At the same time, spectral subtraction algorithm can remove some noise which is from the initial part of speech file. The existing gain factor can ensure that the speech signal will not be affected too much during noise reduction. It can avoid adjusting filter parameters and deal with filtering delay problem. The comparison after LMSSS algorithm to the speech "0" is as shown in Fig. 3.

Fig. 3 shows that the filtered noise signal is obviously reduced again after doing spectral subtraction. And the music noise also gets removed finally. Therefore, this method increases the SNR and has outstanding effect.
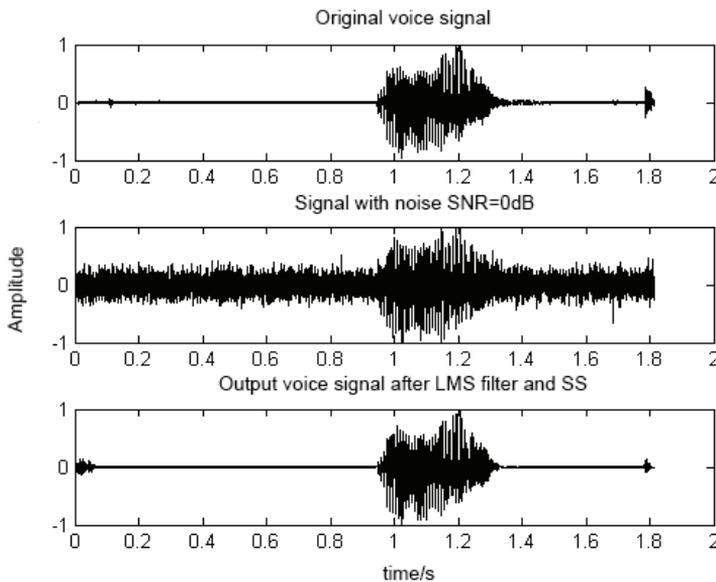

Fig. 3. Comparison of speech "0" after LMSSS noise reduction algorithm.

The advantages of LMSSS algorithm are (1) do not need to modify the parameters and avoid bad filtering effect; (2) deal with music noise problem which comes from spectral subtraction algorithm; and (3) combine two noise reduction algorithms and simplified the design and implementation of LMSSS algorithm.

At the same time, there are also some drawbacks existing in LMSSS algorithm. LMSSS algorithm performs two noise reduction processes, which increases computation and running time to some extent.

4. Experiments

4.1 Experimental Data

In order to ensure the accuracy in computing SNR, the experiments chose to artificially add a certain amount of Gaussian white noise to pure speech file. So the acquisition of experimental data was similar to the Section 3. All data came from 20 original pure speech files. And we set the experimental noise intensities range of 15 dB, 10 dB, 5 dB, 0 dB, -5 dB, -10 dB, and -15 dB.

Similarly, under each SNR, we added 1,000 white noises to each speech file, which meant we had 20,000 new speech files. At the same time, due to the set of seven kinds of different SNRs, the experiment had total 140,000 speech data to be used for verification.

4.2 Experimental Method

Firstly, perform LMS adaptive filtering and spectral subtraction on the experimental speech files, respectively and get the signal-noise ratios, SNR_1 and SNR_2. Then perform spectral subtraction on the LMS adaptive filtered speech files and get the final signal-noise ratio SNR_5. And we compared the noise reduction effects by recording different SNRs.

In order to further verify the reliability of the LMSSS algorithm, we also chose wiener filter algorithm and wavelet threshold algorithm to conduct noise reduction under the same experimental data. Wiener filter algorithm and wavelet threshold algorithm are also the excellent and classic algorithms in speech enhancement. So we selected $\alpha = 0.99$ as the smoothing parameter to do noise reduction and got SNR_3 as the final SNR. We selected Symlets6 as the wavelet basis and used the combination of hard and soft threshold to do noise reduction and got the SNR_4 as the final SNR. We recorded these new SNRs and analyzed the performance between them and LMSSS algorithm on speech enhancement.

4.3 The Results and Analysis of Experiments

The results are shown as the following Table 1. The second column is the original SNR, the third and fourth columns are the SNRs after LMS and spectral subtraction (SS) algorithms, respectively. The fifth and sixth columns are respectively the SNRs after wiener filter (WN) and wavelet threshold (WT) algorithms. And the last column is after the combination of LMS and spectral subtraction algorithms.

As we can see from the above Table 1, all algorithms can achieve the purpose of speech enhancement. With the reduction of original SNR, the LMSSS algorithm can improve the SNR better than the others. When the SNR is lower than 5 dB, LMSSS has better performance in speech enhancement than LMS and spectral subtraction. We can further illustrate the analysis in conjunction with Fig. 4.

In the Fig. 4, the abscissa is the SNRs before noise reduction, and its value is decreased from left to right to indicate that the intensity of the noise gradually increases. The ordinate is the SNRs after using noise reduction algorithm, and the value is gradually increased from bottom to top to indicate that the intensity of the noise gradually reduced. As we can see from the trend of curve, with the increase of noise intensity, the LMSSS algorithm achieves the highest SNR. The combination of LMS and spectral subtraction has better noise reduction effect than that of the other two algorithms.

Table 1. SNR values after three noise reduction algorithm

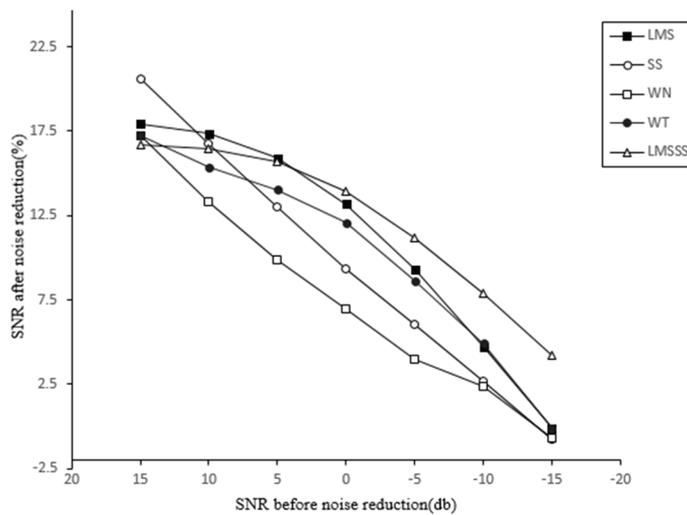|	Initial SNR (dB)	LMS (dB)	SS (dB)	WN (dB)	WT (dB)	LMSSS (dB)
1	15	17.9384	20.6200	17.2421	17.2312	16.6912
2	10	17.3406	16.7576	13.2935	15.3530	16.4608
3	5	15.9026	12.9901	9.8536	14.0289	15.6716
4	0	13.1954	9.3721	6.9636	12.0622	13.9622
5	-5	9.3054	6.0238	3.9639	8.6432	11.1432
6	-10	4.7390	2.6939	2.3632	4.9161	7.9161
7	-15	-0.1151	-0.7819	-0.7200	-0.1062	4.1819

Fig. 4. Comparison of SNR after all noise reduction algorithms.

Table 2. The increased SNR after five noise reduction algorithms

	Initial SNR (dB)	LMS (dB)	SS (dB)	WN (dB)	WT (dB)	LMSSS (dB)
1	15	2.9384	5.6200	2.2412	2.2312	1.6912
2	10	7.3406	6.7576	3.2935	5.3530	6.4608
3	5	10.9026	7.9901	4.8536	9.0289	10.6716
4	0	13.1954	9.3721	6.9636	12.5622	13.9622
5	-5	14.3054	11.0238	8.9639	13.1432	16.1432
6	-10	14.7390	12.6939	12.3632	14.9161	17.9161
7	-15	14.8849	14.2181	14.2800	14.8938	19.1819

The following will further compare the effects among these algorithms. Table 2 shows the differences in SNRs before or after the noise reduction.

Table 2 shows the SNRs get how many improvements by using these algorithms. And Fig. 5 is the trend curve of increased SNR after using these algorithms.

It can be seen from Fig. 5 that when the original SNR is lower than 5 dB, the LMSSS provides the highest SNR. When the SNR is lower than 0 dB, with the decrease of SNR, the lifting curve under LMS algorithm and wavelet threshold algorithm has been gradually flattened. The effects when only using spectral subtraction algorithm or wiener filter algorithm are always very low. The combination of LMS and spectral subtraction algorithm not only increases the SNR more, but also gives a still rising trend.
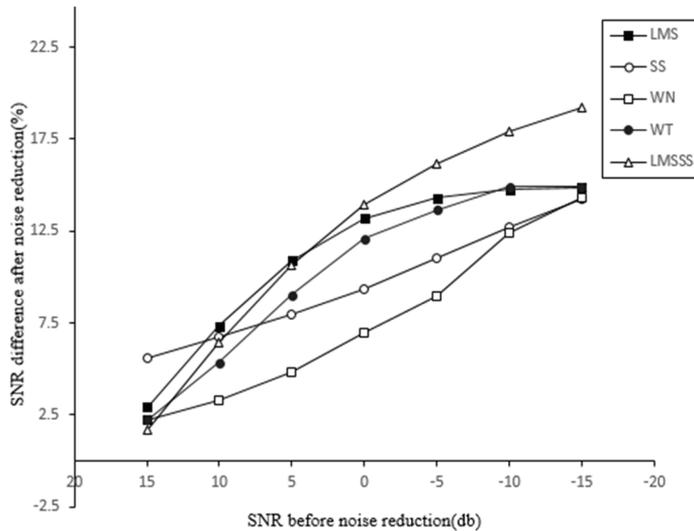

Fig. 5. Comparison of all algorithms for increased SNR.

As described above, in the noise environments which has low SNR (<5 dB), the LMSSS algorithm is not only better than that only use of LMS algorithm or spectral subtraction alone, but also more adaptable to the noise environment which has stronger noise.

Figs. 6–9 are the comparisons in the effects of the processed speech waveforms. Since the experimental data are so large, only a few typical diagrams are shown.
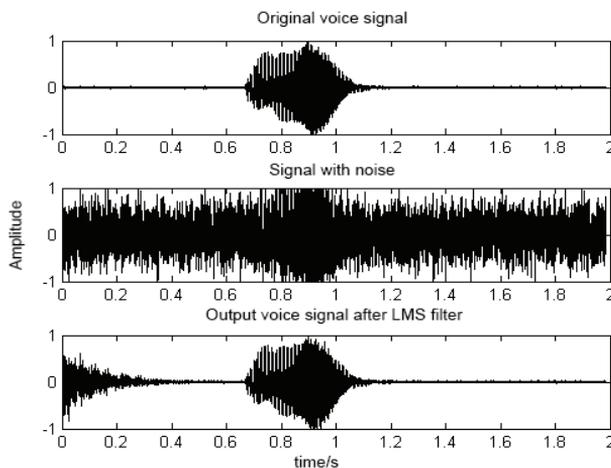

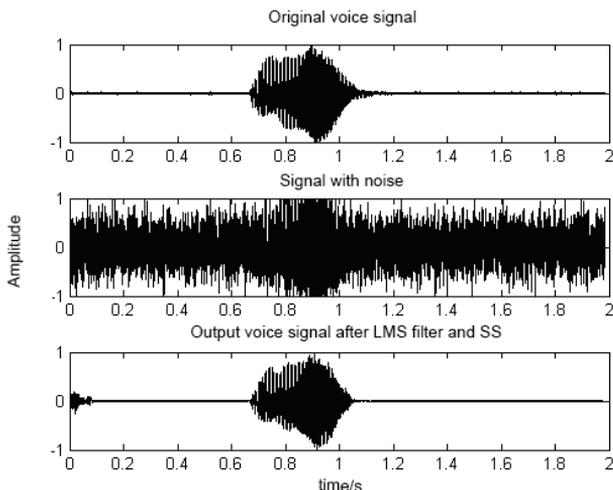**Fig. 6.** Output signal after LMS filter algorithm to the speech "1".

Fig. 7. Output signal after LMS filter and spectral subtraction algorithm to the speech "1".
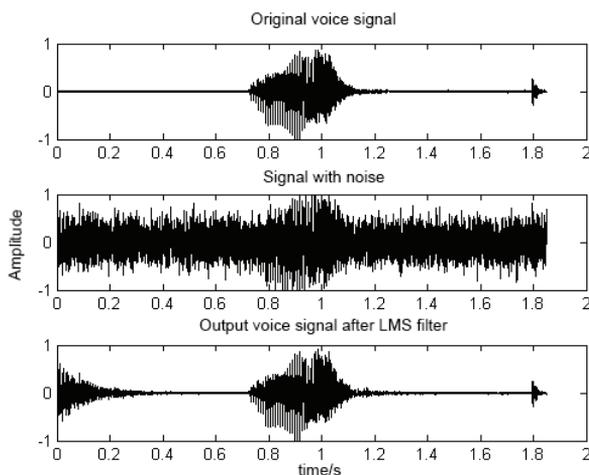

Fig. 8. Output signal after LMS filter algorithm to the speech "2".

5. The Application of LMSSS Algorithm in the Speech Recognition

The above experiments show that the LMSSS algorithm can effectively improve the SNR and achieve the purpose of speech enhancement. With the decrease of initial SNR, the LMSSS algorithm has better performance than the other two algorithms.

But in fact, the SNR is not the only quantitative analysis to the speech recognition and speech enhancing. Degree of damage to the original speech files after noise reduction is also one important factor. Hence, based on the comprehensive consideration about SNR and degree of damage in the output speech files after noise reduction, we designed experiments to verify that LMSSS algorithm is well in improving the accuracy rate of speech recognition.
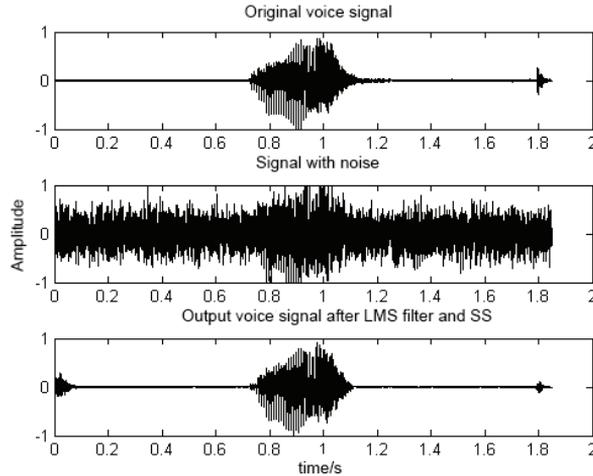
Fig. 9. Output signal after LMS filter and spectral subtraction algorithm to the speech "2".

5.1 Experimental Data

The experimental data were divided into A, B two groups, and both of them had the same speech contents. Group A was used for identification and group B for comparison. Each group could be divided into 13 smaller groups according to the different SNRs, A1, A2, ..., A13 and B1, B2, ..., B13, and the corresponding SNRs were 20 dB, 18 dB, 16 dB, ..., 4 dB, 2 dB, 0 dB, -2 dB, -4 dB. Each small group had 1,000 speech data which were added random Gaussian white noise.

5.2 Experimental Methods

Traditional speech recognition only includes four steps: preprocessing of speech data, voice activity detection, feature extraction, and template matching [15]. In order to verify the validity of the LMSSS algorithm, we made some changes in the combined LMSSS algorithm to compare with the previous LMS algorithm and spectral subtraction algorithm. And the recognition rate could be verified to be further improved. The speech recognition experiments can be divided into the following steps:

Step 1: Prepare the voice signal $x(i)$, including pre-emphasis, framing and windowing;

Step 2: Use the MFCC_COS algorithm to detect the endpoint value for the processed data, and obtain the starting point A and the end point B.

Step 3: Use the LMSSS algorithm (or LMS algorithm and spectral subtraction algorithm) to do noise reduction for the speech signal $x(i)$ and get final processed speech signal $y(i)$;

Step 4: Extract the MFCC characteristic parameters from the speech signal $y(i)$ to obtain the characteristic matrix R.

Step 5: Intercept the corresponding feature matrix R to get the feature matrix W which is corresponded to endpoint detection result according to the values of the starting point A and the end point B.

Step 6: Calculate the similarity between the feature matrix W and each feature matrix in the template library by using dynamic time warping (DTW) algorithm and output the final matching results.

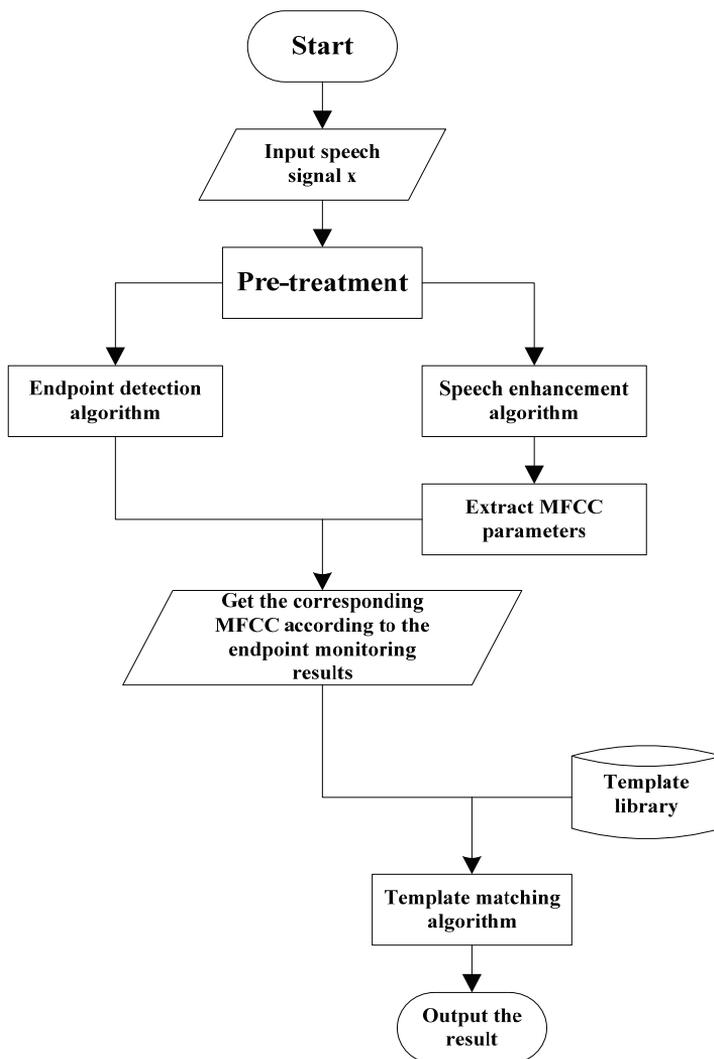The process of speech recognition can be represented by the following Fig. 10.

Fig. 10. Speech recognition process combining with LMSSS and MFCC_COS.

5.3 The Results and Analysis of Experiment

The accuracy rate of speech recognition is shown as the Table 3.

The comparison chart for accuracy rate is shown as Fig. 11.

As is shown from Table 3 and Fig. 11, the accuracy rate of speech recognition based on LMSSS algorithm is significantly higher than that of LMS algorithm and spectral subtraction algorithm alone. Although when the initial SNR is less than 0 dB, the accuracy rates are all dropped sharply. The LMSSS algorithm is still higher than that of LMS and spectral subtraction algorithm. In fact, the reason is that the addition of spectral subtraction algorithm compensates for the filtering delay problem which comes from LMS algorithm. And the LMS algorithm will focus on improving the filtering effect, and do not

need to reduce the filter parameters all the time. And the noise intensity of initial part of speech file can be reduced again by the spectral subtraction algorithm.

Table 3. Comparison in recognition rate under three algorithms

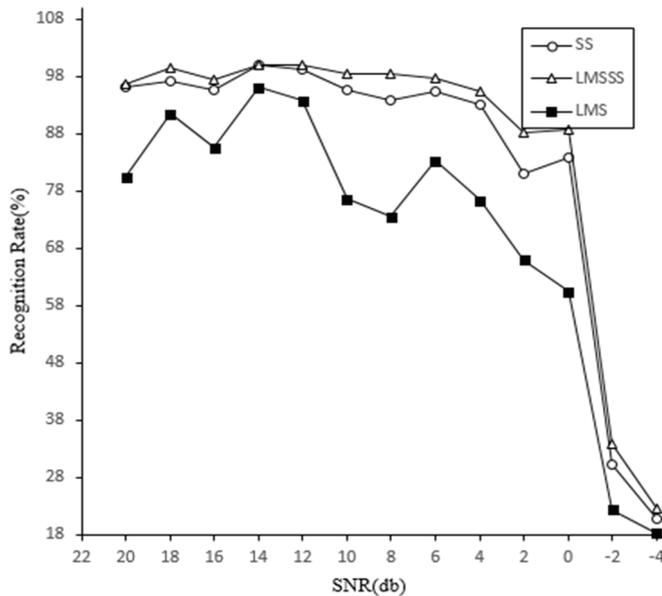|	Initial SNR (dB)	Recognition rate (%)		
		LMS	SS	LMSSS
1	20	80.4	96.2	96.7
2	18	91.5	97.1	99.5
3	16	85.5	97.3	95.5
4	14	96.1	100	100
5	12	93.7	99.1	100
6	10	76.6	95.7	98.4
7	8	73.5	93.8	98.5
8	6	83.3	95.4	97.6
9	4	76.3	93.1	95.4
10	2	65.9	80.9	88.2
11	0	60.4	83.8	88.7
12	-2	22.4	30.3	33.8
13	-4	18.3	20.8	22.6

Fig. 11. Comparison of accuracy rate under three algorithms.

Table 4 and Fig. 12 show the comparison of time consumption for 1,000 speech recognitions.

Shown from comparison of time consuming, LMSSS algorithm needs more calculation, and the 'Time' is almost doubled than that of spectral subtraction algorithm. But compared with the LMS algorithm, the computational costs are relatively close. If we just consider the time consuming, LMSSS algorithm will take 325.9069 seconds to do 1,000 speech recognitions when the SNR is 0 dB. It will take 0.326 seconds

on average in once speech recognition. The LMS algorithm is 0.345 seconds and spectral subtraction algorithm is 0.14 seconds. Considering one time speech recognition, the difference of time consumption is not so obvious. And the 0.345 seconds has included all operations, such as adding white Gaussian noise, preprocessing, frame division, endpoint detection, feature extraction, noise reduction, template matching based on DTW algorithm, and so on. So in the system with low real-time requirements, LMSSS can meet the needs for effective speech recognition. On the whole, although the LMSSS algorithm spends more time, its advantages in speech enhancement and recognition are more obvious.

Table 4. Time consumption for 1,000 speech recognitions

|	Initial SNR (dB)	Time consumption (s)		
		LMS	SS	LMSSS
1	20	367.8192	189.1658	391.8110
2	18	355.8416	178.2789	379.0001
3	16	361.0980	169.7724	368.2490
4	14	362.3975	175.4268	374.6448
5	12	364.0179	154.9321	376.1151
6	10	381.0179	158.9518	386.0327
7	8	359.3360	152.7651	375.0003
8	6	334.8769	151.5179	354.4190
9	4	339.6121	156.1250	364.1827
10	2	337.7034	145.0781	343.9932
11	0	325.9069	139.7418	345.5355
12	-2	323.4367	137.8965	343.9680
13	-4	320.6598	128.9805	344.9089
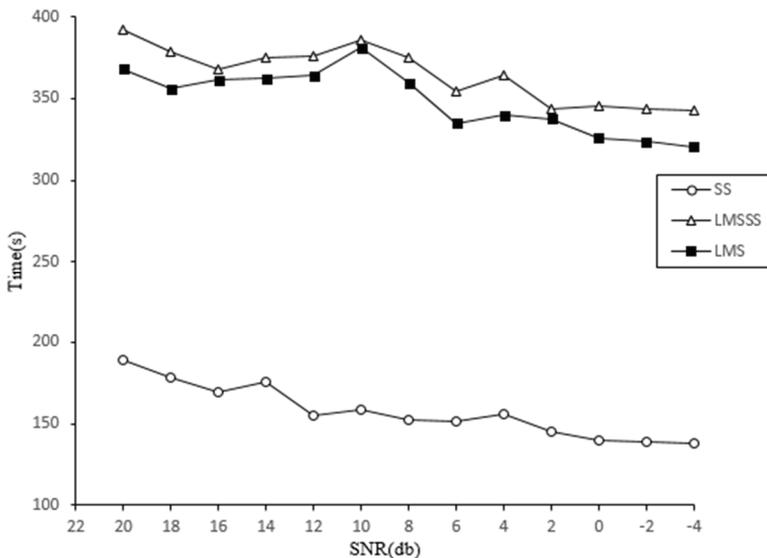
Fig. 12. Comparison of time consumption under three algorithms.

In the worst case analysis, as the original SNR continues to decrease, especially when the SNR is lower than 0 dB, the advantage of the LMSSS algorithm will gradually decrease or even be surpassed by the other two algorithms. LMSSS algorithm will bring more damage to original speech file's content after two consecutive noise reduction operations (LMS noise reduction and spectral subtraction noise reduction). Although the SNR will continue to be increased, the original speech has become very blurred. Also the LMSSS algorithm will take more time than the other algorithms when processing a large number of speech files in real time.

6. Conclusion

In this study, we expound the spectral subtraction algorithm which has higher computational efficiency and LMS adaptive filtering algorithm which has better noise reduction effect. Finally, we propose LMSSS speech enhancement method based on the noise reduction principle. The LMSSS algorithm adds spectral subtraction operation to the noisy speech after LMS processing, which simplifies the parameter adjustment of the filter and avoids music noise and filtering delay problem. In the speech recognition experiments, LMSSS algorithm gets the highest accuracy of speech recognition. Although the computational costs are larger, the final effects on speech enhancement are optimistic enough within the experimental range (-4 dB–20 dB).

Experiments show that LMSSS algorithm can improve the SNR of original speech better when the original SNR is lower than 5 dB. And when the SNR is in a certain range (0 dB–15 dB), LMSSS algorithm is more suitable for speech recognition, the effect of speech enhancement is also better than LMS algorithm, spectrum subtraction algorithm and more other commonly used algorithms.

Acknowledgement

References

[1] X. Wang, L. Li, and C. Liu, "Study of speech enhancement algorithm based on spectral subtraction," *Journal of the Staff and Worker's University*, vol. 2013, no. 6, pp. 85-87, 2013.

[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113-120, 1979.

[3] D. Tsoukalas, M. Paraskevas, and J. Mourjopoulos, "Speech enhancement using psychoacoustic criteria," in *Proceedings of 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, MN, 1993, pp. 359-362.

[4] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586-1604, 1979.

[5]   B. Widrow and M. E. Hoff, "Adaptive switching circuits," Stanford Electronics Labs, Stanford University, CA, Report No. TR-1553-1, 1960.

[6]   D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, 1994.

[7]   S. Thomas Alexander, *Adaptive Signal Processing: Theory and Applications*. New York, NY: Springer, 1986.

[8]   W. Xu, G. Wang, Y. Geng, F. Bai, and T. Fei, "Speech enhancement algorithm based on spectral subtraction and variable-step LMS algorithm," *Computer Engineering and Applications*, vol. 51, no. 1, pp. 213-217, 2015.

[9]   H. Chen and X. H. Qiu, "Research on speech enhancement of improved spectral subtraction algorithm," *Computer Technology and Development*, vol. 24, no. 4, pp. 70-76, 2014.

[10] A. D. Poularikas and Z. M. Ramadan, *Adaptive Filtering Primer with MATLAB*. Boca Raton, FL: CRC Press, 2006.

[11] Z. Song, *The Application of MATLAB in Speech Signal Analysis and Synthesis*. Beijing: Beihang University Press, 2013.

[12] J. Han, C. Wang, C. Lu, L. Zhang, W Ren, and Y. Ma, "Robust speech recognition system in noisy environment," *Audio Engineering*, vol. 2002, no. 1, pp. 27-29, 2002.

[13] J. Bai, L. H. Yang, and X. Y. Zhang, "An antinoise SVM parameter optimization method for speech recognition," *Journal of Central South University: Science and Technology*, vol. 44, no. 2, pp. 604-611, 2013.

[14] R. Wang and P. Chai, "A method for speech enhancement based on improved spectral subtraction," *Pattern Recognition and Artificial Intelligence*, vol. 16, no. 2, pp. 247-251, 2003.

[15] Y. Yang and W. Shi, "Implementation of adaptive filter on wave-generated magnetic noise based on LMS algorithm," *Journal of Jiangsu Institute of Education (Natural Science)*, vol. 27, no. 1, pp. 9-10, 2011.

Danyang Cao https://orcid.org/0000-0002-9779-9466

He received B.S. and M.S. degree in Computer Science and Technology from North China University of Technology in 2000 and 2006, respectively. He received his Ph.D. degree in Computer Application Technology in 2012 from University of Science Technology Beijing, China. He is currently an Associate Professor in the Department of Computer Science at the North China University of Technology, China. His research interests cover the fields of artificial intelligence, data mining.

Zhixin Chen https://orcid.org/0000-0003-4325-4783

She received B.S. degree in School of Computer from Northeastern University in 2016. She is currently a graduate student in North China University of Technology.

Xue Gao https://orcid.org/0000-0003-4056-9536

He received B.S. degree in School of Computer from North China University of Technology in 2014. He is currently a graduate student in North China University of Technology. His research fields cover data mining.