On Using Digital Speech Processing Techniques for Synchronization in Heterogeneous Teleconferencing

Hsiao-Pu Lin[1] and Hung-Yun Hsieh[1,2]

[1] Graduate Institute of Communication Engineering
[2] Department of Electrical Engineering,
National Taiwan University,
Taipei, Taiwan, 106
hyhsieh@cc.ee.ntu.edu.tw

Abstract. As the popularity of multi-functional communication devices grows, traditional audio conferencing now may involve heterogeneous teleconferencing devices, including POTS phone, VoIP phones, dual-mode smart phones, and so on. During a multi-party audio conference involving heterogeneous devices, it is possible that a video conference is held concurrently involving a subset of devices capable of processing video streams for better the conferencing experience. In such a scenario, the need for synchronization between circuit-switched audio streams and packet-switched video streams arises. While the problem of audio-video synchronization has been extensively investigated in related work, existing solutions are limited to synchronization in packet-data networks and hence are not applicable in the target environment. In this work, we consider the problem of supporting such an overlay video conference among dual-mode phones. We first transform the audio-video synchronization problem into the problem of synchronizing circuit-switched and packet-switched audio streams. We then propose an end-to-end solution for audio synchronization that is transparent to the heterogeneous network protocol suites involved. We investigate synchronization algorithms based on digital speech processing using different acoustic features of the speech signal in the waveform, cepstrum, and spectrum domains. We evaluate the effectiveness of different algorithms under various impairments including codec distortion, line noises, packet losses, and overlapping utterances. Evaluation results show a promising direction for using DSP-based algorithms to address the synchronization problem across heterogeneous telephony systems.

Keywords: Overlay video conference, heterogeneous telephony device, dual-mode phone, VoIP.

1 Introduction

As modern communication technology advances, more and more devices have been made available for use with different telephony services, including POTS

N. Bartolini et al. (Eds.): QShine/AAA-IDEA 2009, LNICST 22, pp. 679–695, 2009.

phones, 2G/3G mobile phones, GSM/WiFi dual-mode smart phones, and even VoIP phones. A multi-party teleconference thus may involve conferees using such various types of telephony devices. Due to the disparity of device capability, however, it is possible that only a POTS audio conference can be held among such heterogeneous teleconferencing devices, despite the fact that the conferees with smart phones and pocket PCs may be capable of participating in a video conference.

To provide a better conferencing experiencing among capable devices while maintaining the audio conference, one feasible scenario is to hold the video conference atop the multi-party audio conference. Since the audio conference is held through the PSTN, a concurrent video conference based on IP thus is possible among IP-based devices such as dual-mode phones. In this way, while the audio conference involving participants with legacy POTS phones proceeds as usual, participants with dual-mode phones may still be able to leverage their hardware capability for face-to-face communications.

An important feature of such heterogeneous teleconferencing is that *the audio conference is held through the PSTN network while the video conference is held through the IP network.* Since audio and video conferences are held over different networks, heterogeneity in network environment may lead to different delays and jitters of the multimedia streams [1]. Audio and video streams thus are very likely to be asynchronous at the receiving side, potentially resulting in a perceptually unpleasant conferencing experience.

While synchronization of audio and video streams is a well-attended problem in the literature [2, 3, 4, 5, 6], conventional synchronization control schemes typically rely on the use of the common timestamp information on the audio and video streams for inter-stream synchronization, including adaptive buffer control and playout scheduling. Clearly, these schemes are proposed for operating in the IP network where it is possible to manipulate the packet header (e.g. injecting time-stamp information). They therefore cannot be used directly in the target scenario involving the circuit-switched PSTN telephony system with a very different suite of network protocols from the packet-switched IP telephony system.

To address the problem of synchronization across heterogeneous telephony systems, we therefore investigate solutions based on digital speech processing (DSP). The goal is to *avoid reliance on network protocols of the circuit-switched telephony system for providing synchronization tips.* Instead, we aim to explore the acoustic features inherent in the audio streams for synchronization at the receiving end. Acoustic features have the nice property that they can potentially prevail against different switching technologies as long as the audio streams are generated from the same utterance of the speaker. We investigate three DSP-based algorithms using acoustic features in the waveform, cepstrum, and spectrum domains. The performance of the three algorithms is evaluated against different sources of impairments including codec distortion, line noise, packet loss, and overlapping utterances in individual telephony systems. Evaluation

results show that utilizing digital speech processing techniques for synchronizing audio streams across heterogeneous telephony systems is promising.

The rest of this paper is organized as follows. Section 2 describes in details the target scenario, and how the problem of audio-video synchronization can be simplified into a synchronization problem across PSTN and IP audio streams. Sections 3, 4, and 5 present synchronization algorithms and their performance under different impairments in the waveform, cepstrum, and spectrum domains respectively. Finally, Section 6 compares the overall performance of the three algorithms and concludes the paper.

2 Synchronization in Heterogeneous Teleconferencing

In this section, we first describe the synchronization framework for supporting heterogeneous teleconferencing. We then discuss the challenges of achieving synchronization in the proposed framework.

2.1 Synchronization Framework

The scenario for overlay video conferencing atop multi-party audio conferencing is shown in Fig. 1. The audio conference is held by the audio conference server located in the PSTN network. Conferees attend this audio conference using various kinds of teleconferencing devices, including legacy POTS phones, GSM phones, GSM/WiFi dual-mode smart phones, and even laptops as IP soft phones. The video conference, on the other hand, is held among devices capable of accessing the IP network and processing video streams such as dual-mode pocket PCs and smart phones.

In such heterogeneous teleconferencing, it is necessary on dual-mode phones that the PSTN-based audio stream is synchronized with the IP-based video stream. While related work has proposed the concept of lip synchronization [7, 8] for audio-video synchronization, sophisticated processing algorithms such as face localization, lip modeling and tracking, and identification are required. We instead seek a solution without the need for processing the video stream in this paper.

Inspired by related work [6], we consider an approach that embeds audio information in the video stream for synchronization between PSTN audio and IP video streams. At the transmitter, each video frame is sent through the IP network to the receiver while carrying with it audio information (audio hash) of the current audio frame to be used by the synchronization algorithm. At the receiver, after the video stream is received from the IP network, the embedded audio hash can be extracted and used to determine the timing relationship of the IP video stream with the PSTN audio stream through audio synchronization. The problem of audio-video synchronization thus becomes the problem of synchronizing PSTN and IP audio streams.

To detail, when the dual-mode phone receives the IP video stream and PSTN audio stream, both streams are buffered for the purpose of synchronization and

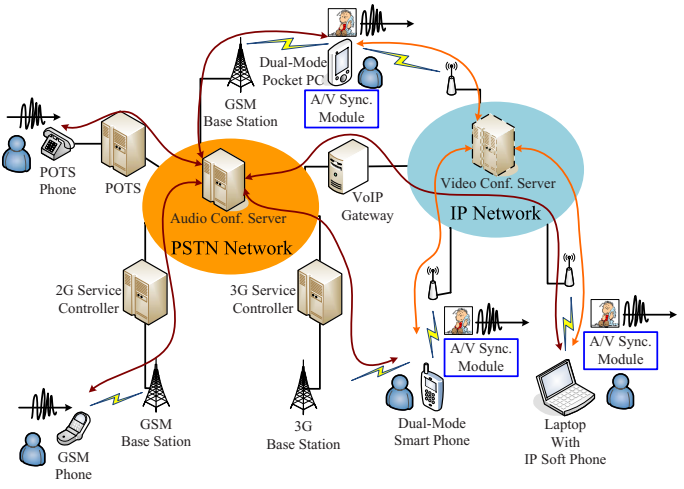

Fig. 1. Heterogeneous teleconferencing

playback. If synchronization between the two streams is necessary, the synchronization module at the receiver extracts the audio hash from the IP video stream and determines the timing relationship between the two audio streams. The resulting timing relationship is fed to the playback buffer where common synchronization control schemes can be applied. Note that whether the received audio and video streams need to be synchronized depends on the synchronization trigger that can be determined on a periodic or dynamic basis. Clearly, since *synchronization does not need to be performed for every video frame* but only when the dynamics of the two telephony systems change significantly, the requirement for real-time computation of the synchronization algorithm becomes less strict.

2.2 Challenges

To better support heterogeneous teleconferencing, it is important to synchronize circuit-switched and packet-switched audio streams with little or no reliance on circuit-switched network protocols for providing synchronization tips. Synchronization based on the inherent acoustic features of the concerned speech signal thus is one possible solution for achieving the goal. There are however challenges in such DSP-based approaches as we describe in the following:

Codec Distortion. To reduce bandwidth requirement, the audio stream is typically encoded before transmission. Different telephony services have been using different voice codecs such as the AMR (Adaptive Multi-Rate) codec in GSM and the G.723 or G.729 codec in VoIP. These voice codecs however result in a lossy compression and introduce different degrees of distortions to the original speech signal.

Packet Loss. In packet-switched networks, packets are subject to errors, delay and reordering. Such impairments may result in losses of audio frames in the packet-switched audio stream. While loss concealment algorithms have been used in VoIP to combat such impairments, nonetheless the speech signal is distorted, especially in wireless networks with bursty losses.

Line Noise. Similar to the transmission problem in IP telephony, circuit-switched audio streams may suffer from different types of noises incurred by the transmission line or user device. Such noise clearly impairs the speech signal and complicates the synchronization process.

Overlapping Utterances. In multi-party teleconferencing, the speech signal to be synchronized is "buried" inside a mixture of multiple speech signals uttered by other speakers (conferees). Different from the static, broadband noise incurred by the transmission line, interference incurred by overlapping utterances is more difficult to separate. The distortion thus incurred will impose great challenges on the synchronization algorithm.

Therefore, while a DSP-based algorithm operates directly on the speech signals and can allow for better transparency over voice switching technologies, it needs to be resilient to various distortions incurred by heterogeneous telephony systems on the speech signals. We present in the following three DSP-based algorithms for synchronization of circuit-switched and packet-switched audio streams.

3 Waveform-Based Synchronization

To determine the relative timing of two audio streams, the simplest way is to match the waveforms of the two speech signals after decoding the audio streams based on time-domain processing.

3.1 Basics of Cross Correlation (XCOR)

Cross correlation is widely used in many areas such as pattern recognition. It tries to capture how similar or different a test signal is from the specific signal. The commonly used similarity measure is the correlation coefficient, r, defined as

$$r = \frac{\sum_i^N (x(i) - m_x)(y(i) - m_y)}{\sqrt{\sum_i^N (x(i) - m_x)^2 \times \sum_i^N (y(i) - m_y)^2}}, \tag{1}$$

where $x(i)$ and $y(i)$ are the comparing signals and m_x and m_y are individual means. Therefore, if two comparing speech signals have similar waveforms, their correlation coefficient may reach a value close to 1 when the two signals are "aligned" in time. It thus can be used as a metric for determining the relative timing of two speech signals.

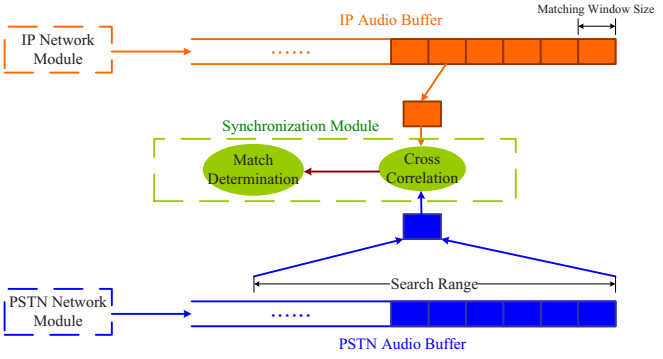

Fig. 2. XCOR synchronization module

3.2 XCOR Synchronization Module

Based on the cross-correlation function, we can design a synchronization module as illustrated in Fig. 2. Audio streams received from either circuit-switched (PSTN) or packet-switched (IP) networks are first stored in individual buffers. When the synchronization process begins, a window of speech samples (matching window) is selected from one buffer to iteratively match against a time-shifted window of speech samples of the same size in the second buffer. To reduce the computation complexity, the time shift can be limited to a *search range* if the maximum or approximate time offset of the two audio streams can be estimated beforehand. The correlation coefficient thus computed at each iteration is recorded against the time-shift value. At the end of iterations, the time shift that corresponds to the largest correlation coefficient is used as the relative time offset of the two audio streams.

3.3 Performance Evaluation

We evaluate the performance of the synchronization algorithm based on cross correlation for various types of impairments as mentioned in Section 2.2. Due to lack of space, however, we only present and discuss a subset of the results in this section.

Line Noise. The thermal noise is the most common source of noise so we focus our discussion on the distortion by thermal noise. We model the noise as the Additive White Gaussian Noise (AWGN), where the variance (σ^2) determines the energy level of noise. A noise with energy level 0.01 in our experiments corresponds to almost the lower-volume part in the source speech. Therefore, the effect of noise at this level is comprehensible. Since the noise reduces the correlation coefficient at the time shift of 0, the accuracy of correlation-based synchronization might also be affected. Fig. 3 shows the accuracy of synchronization when one of the source signal is impaired by noise. We encode one source

signal using the Adaptive Multi-Rate (AMR) codec (GSM audio stream), and the other signal using the G.729 codec (VoIP audio stream). We can observe that the accuracy of synchronization is lowered by the noise. Increasing the size of the matching window can potentially improve the accuracy, although at the cost of increased computation complexity.

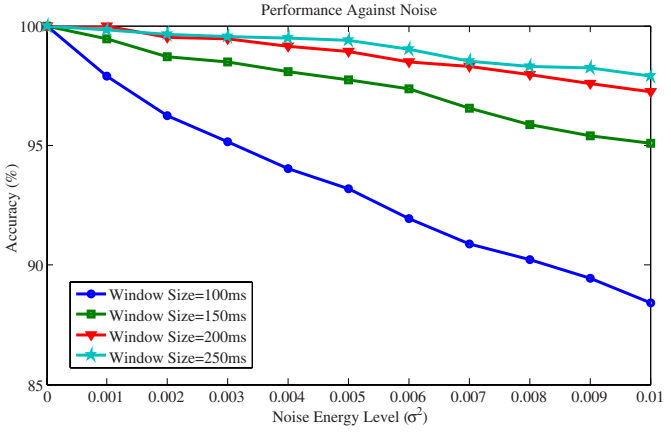

Fig. 3. Performance of XCOR against line noise

Overlapping Utterances. In a multi-party conference, it is possible that multiple speakers speak at the same time. Hence, the receiver may receive an audio stream with mixing utterances from different speakers. Fig. 4 thus shows the effect of overlapping utterances on correlation-based synchronization. From the top two sub-figures, we can observe that the correlation coefficient is substantially lowered at the point of 0 time-shift as the number of interfering utterances increases. For a small matching window such as 100 ms, the algorithm is more vulnerable to interfering utterances, and thus achieves lower accuracy compared to the case with a large matching window.

Combined Impairments. We combine all sources of impairments and evaluate the performance of correlation-based synchronization as shown in Fig. 5. We can observe that the performance is severely degraded as distortions add up. This is because cross correlation considers only the time-domain waveform, and different types of distortions change the waveform-similarity of the concerned speech signal in different ways. Even though the distortion level of individual impairments is low, overall the performance is still significantly affected. In addition, although increasing the matching window size can improve the accuracy, the performance improvement is limited considering the increase in computation complexity. Therefore, we can observe that cross correlation can sustain minor distortions on the waveform, but *as the distortion level increases or multiple distortions add up, its performance quickly drops.*

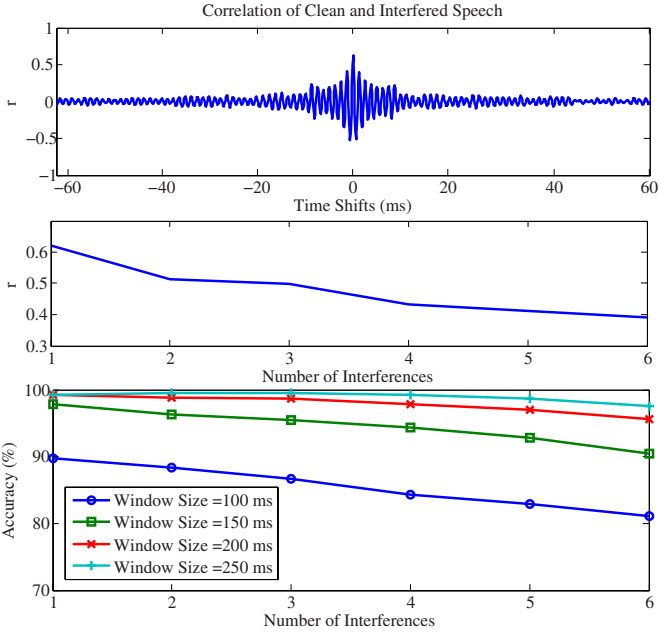

Fig. 4. Performance of XCOR against overlapping utterances

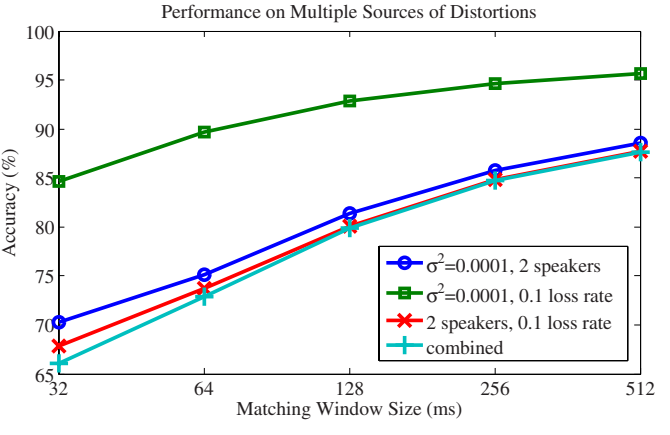

Fig. 5. Performance of XCOR against combined impairments

In conclusion, cross correlation is vulnerable to practical distortions because it considers only the time-domain waveform of the speech signal. Since the waveform is easily corrupted by distortions, a larger window size of speech samples should be used for synchronization, thus increasing computation complexity of the algorithm.

4 Cepstrum-Based Synchronization

The cepstrum of a signal is commonly used in digital speech processing applications such as voice identification and pitch detection. We thus investigate the use of cepstrum-based analysis for voice synchronization in the section.

4.1 Basics of MFCC

The cepstrum of a signal is obtained as the Fourier transform of the logarithm of the power spectrum of the signal ("spectrum of a spectrum"). It has the nice property that the *convolution* of two signals in the time domain is equivalent to the *addition* of their ceptra in the cepstrum domain. To capture the perception of the human auditory system to the speech signal, the Mel-Frequency Cepstral Coefficient (MFCC) is often used in cepstral analysis. The frequency warping introduced in the mel-scale transformation of the cepstrum allows a better representation of sound. Conventionally, for each window of speech samples (analysis window), a set of coefficients (MFCCs) is obtained as a column vector, and the index of each coefficient is referred to as the MFCC bin.

4.2 MFCC Synchronization Module

To use MFCC for audio synchronization, we first define the similarity metric for each MFCC bin similar to [9] as follows:

$$B(m,u) = \begin{cases} |m-u|, & \text{if } m+\epsilon \geq u; \\ p, & \text{otherwise,} \end{cases} \tag{2}$$

where m and u are the coefficients of the two comparing speech signals (referred to as mixed and unmixed signals) in the concerned MFCC bin respectively, ϵ is the error factor, and p is the penalty value. The penalty value is designed to differentiate the correct match at the time shift of 0 from other shifts. The error factor, on the other hand, is included for tolerance of minor faults such as those introduced by random noises. As shown in Fig. 6, different time shifts of one signal result in different signals to be compared against the other signal. For each

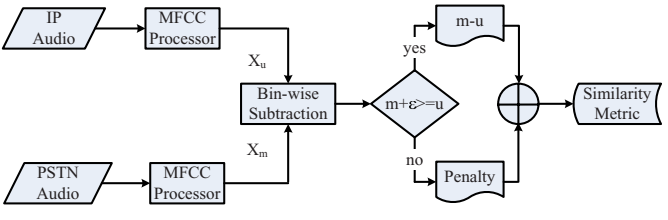

Fig. 6. MFCC synchronization module

comparing signal, a window of speech samples (matching window) consisting of multiple analysis windows is transformed into multiple columns of MFCCs. $B(m, u)$ is then computed for every MFCC bin in one column, and values over all MFCC bins over all columns are summed up as the similarity metric of the two comparing signals. Once the similarity metric for the two comparing signals (with different time shifts) is determined, the synchronization process can proceed.

4.3 Performance Evaluation

To evaluate the performance of the synchronization algorithm based on MFCC, we set the size of the analysis window to 32ms (corresponding to one matching column). The size of the matching window is varied depending on the number of analysis windows (matching columns) included for calculating the similarity of the two signals. Due to lack of space, we present and discuss only a subset of the results in the following.

Packet Loss. When packet losses occur in VoIP, conventional approaches apply packet loss concealment methods where the missing frame is filled using the previously received frame. Fig. 7 reveals the robustness of MFCC against packet losses when the lost frame is concealed by duplicating the frame in the previously received packet. Although a small number of matching columns might not be sufficient to differentiate the correct match from other shifts, the case with 4 matching columns performs reasonably well for a packet loss rate of less than 20%.

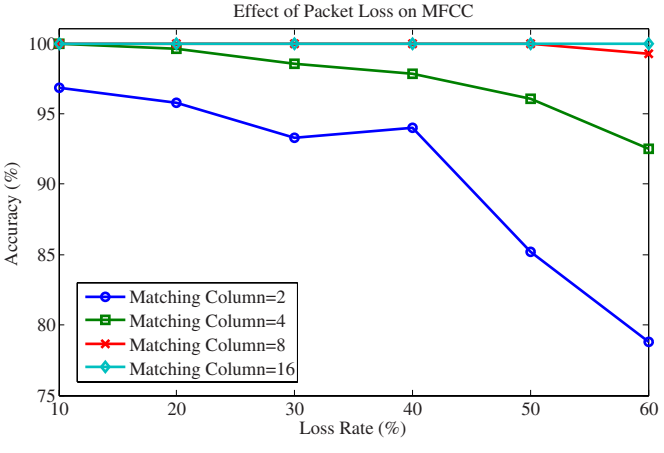

Fig. 7. Performance of MFCC against packet loss

Overlapping Utterances. As shown in Fig. 8, once the PSTN audio contains interfering utterances from other speakers, the percentage of accuracy drops significantly. Even for the case with 16 matching columns (about 512ms of matching window), the performance of the MFCC-based algorithm drops from 100% to about 60% as the number of utterances in the PSTN audio stream increases from 1 to 3. This is because as the number of speakers increases, the coefficients in each MFCC bin are heavily distorted and it becomes more difficult to differentiate the correct match from other shifts using the penalty value. Therefore, the MFCC-based synchronization algorithm also shows the problem of vulnerability to interfering utterances.

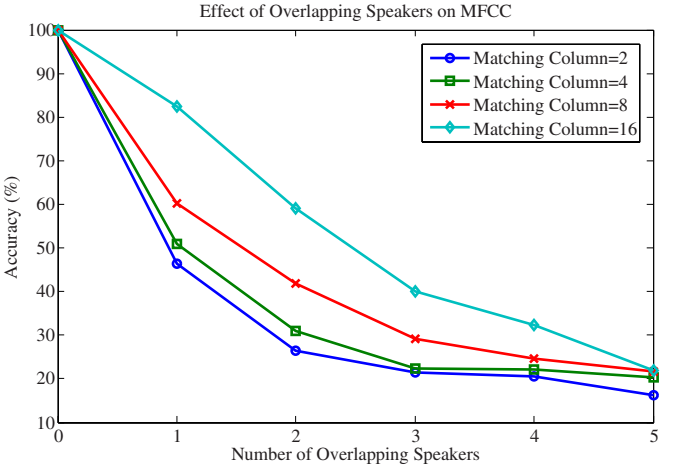

Fig. 8. Performance of MFCC against overlapping utterances

Combined Impairments. Fig. 9 shows the performance of MFCC when multiple sources of distortions occur. We can observe that since MFCC is vulnerable to overlapping utterances, whenever an audio stream involves mixed utterances, its performance degrades significantly. For the case with no overlapping speakers, however, MFCC performs reasonably well under additional sources of distortion if the size of the matching window is sufficient (about 256ms).

To sum up, as long as the size of the matching window is appropriately chosen, using the similarity of MFCC bins for synchronization seems to be a viable option since it is robust against many types of waveform distortions. However, its performance is severely degraded against overlapping utterances in the PSTN audio stream. In addition, using a larger size of the matching window is not effective in this case. This is a potential problem with MFCC since in a practical conference, especial during a keen discussion, many speakers may speak at the same time, and thus the PSTN audio stream is likely to be a mixture of utterances from multiple speakers.

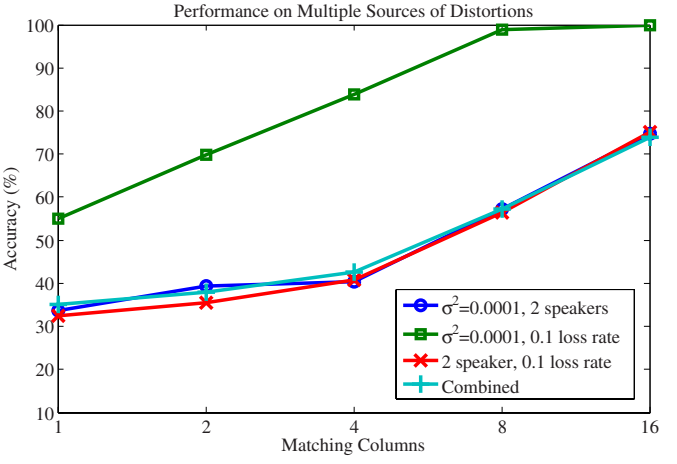

Fig. 9. Performance of MFCC against combined impairments

5 Spectrum-Based Synchronization

Waveform-based synchronization exploits the similarity of audio streams in the time domain. Spectrum-based synchronization, on the other hand, exploits the similarity in the frequency domain.

5.1 Basis of Spectrogram (SPGM)

Since the human speech varies with time, a direct transform of the speech signal into the frequency domain will lose many useful information. Instead, a short-time Fourier transform (STFT) is applied on the speech signal on a window-by-window (time frame) basis. The squared magnitude of the STFT over time thus obtained is the spectrogram of the signal, and it shows the variation of the spectral density of the signal over time.

While the frequency-domain representation of a signal ideally is equivalent to the time-domain representation, frequency-domain processing has several nice properties. An important property of the spectrogram that we explore in this paper is the property of *window-disjoint orthogonality (W-DO)* for human speech [10, 11]. It has been discovered that a speech signal is sparsely distributed in its time-frequency representation and, as a result, different speech utterances (except speech babbles) tend not to overlap (orthogonal) in the time-frequency plot such as the spectrogram. With the assumption of W-DO, the value of each frequency bin at a certain time frame can be considered as being contributed by a single speaker as we show in Fig. 10. Many blind speech separation techniques based on the approximate W-DO of speech have been proposed [12, 13]. The sparsity in spectrogram provides a helpful tool for voice synchronization *if the speech signal to be synchronized is "buried" inside a mixture of multiple speech signals.*

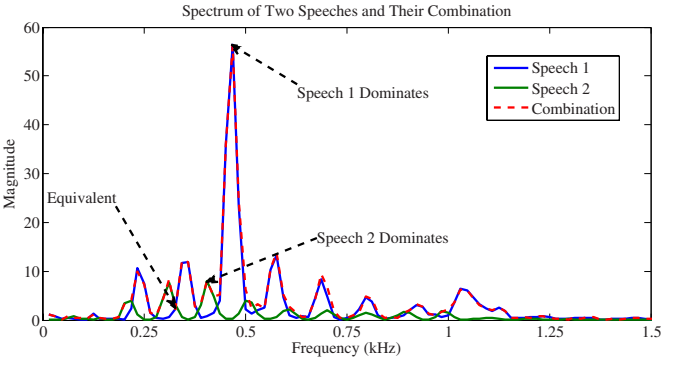

Fig. 10. Spectrum of mixture is usually dominated by one utterance

5.2 SPGM Synchronization Module

To explain the operation of the synchronization algorithm based on spectrogram, assume that the PSTN audio stream is mixed with multiple utterances, and the goal is to synchronize the unmixed IP audio stream against the mixed PSTN audio stream. Based on the concept of W-DO, some frequency bins in the spectrogram of the PSTN speech signal are dominated by the utterance of the concerned speaker, and hence they can be used for similarity measure against the IP audio stream. We can choose only those "significant" frequency bins for minimizing the disturbance of overlapping utterances in the synchronization process.

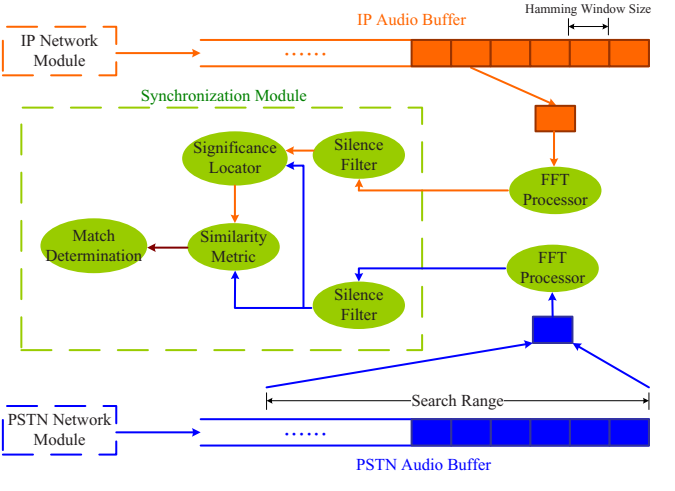

Fig. 11. SPGM synchronization module

The synchronization module based on spectrogram is shown in Fig. 11. When the synchronization process begins, a window of speech samples (unmixed signal) is selected from the IP audio buffer and sent to the *FFT processor* to compute the spectrogram. The speech samples of the PSTN audio buffer within the search range (of time shifts) is also sent to the FFT processor for generating the second spectrogram. The *silence filter* inside the synchronization module is used to exclude silence frames to avoid unnecessary and error-prone matching. For each possible time shift of the two spectrograms, the *significance locator* infers from the two spectrograms and determines which frequency bins should be included for calculating the similarity. Let $m(\tau, \omega)$ and $u(\tau, \omega)$ be the spectrogram values of the mixed and unmixed signals at time τ and frequency bin ω respectively. The metric $S(\tau, \omega) = |\frac{m-u}{u}|$ is then used for determining the significance of bin ω of frame τ. The similarity (dissimilarity) metric of the two comparing signals is calculated by summing the significance of all bins with $S(\tau, \omega) < \eta$ over all time frames (matching columns). The time shift that yields the lowest value is determined as the relative time offset of the two audio streams.

5.3 Performance Evaluation

To evaluate the performance of the synchronization algorithm based on spectrogram, we set the analysis window to 64ms with 32ms overlap. Each matching column thus represents a 32ms time-shift from adjacent columns.

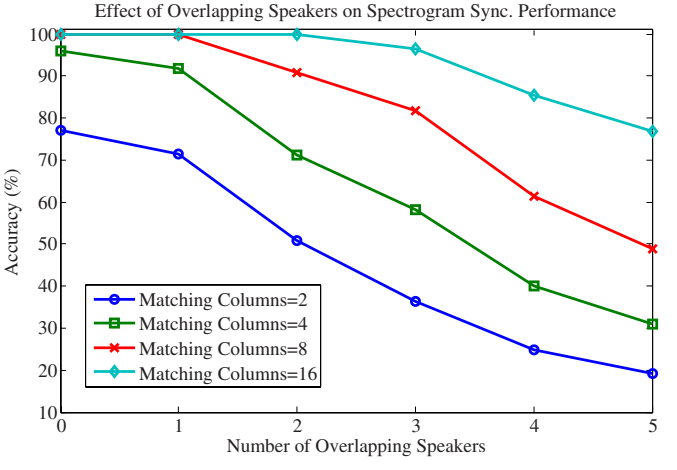

Fig. 12. Performance of SPGM against overlapping utterances

Overlapping Utterances. From Fig. 12 we can observe that the percentage of accuracy still drops as the number of interfering speakers increases. This is because the sparsity is somewhat compromised as the number of speech utterances

increases. However, comparing to the performance of MFCC-based algorithm, this spectrogram-based algorithm can achieve better accuracy. If we consider the case with two interfering speakers, the spectrogram-based algorithm can achieve more than 90% of accuracy if 8 or more matching columns are applied. The MFCC-based algorithm, on the other hand, achieves only 40% of accuracy as shown in Fig. 8.

Combined Impairments. Fig. 13 shows the performance of the spectrogram-based algorithm when multiple sources of distortions occur. The spectrogram-based algorithm shows performance benefits compared to the other two algorithms when the number of speakers increases. In addition, its performance is consistently improved as the number of matching columns increases.

In conclusion, for the case of overlapping utterances, the synchronization algorithm based on spectrogram achieves better performance compared to the other two algorithms due to speech sparsity. As for other impairments such as packet losses, however, the spectrogram-based algorithm does not show significant performance benefits over the other two.

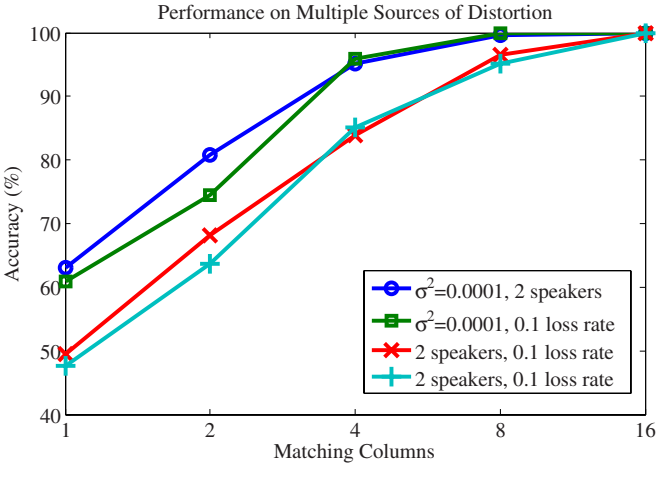

Fig. 13. Performance of SPGM against combined impairments

6 Conclusions

To compare the performance of the three algorithms, we show in Fig. 14 their performance under all sources of distortions as described in Section 2.2. The two audio streams are encoded using AMR and G.729 respectively, the packet loss rate is set to 10%, the variance of the noise is set to 10^{-4}, and the number of overlapping utterances is set to 2 and 4 for comparison. We can observe that the spectrogram-based algorithm outperforms the other two. The XCOR-based synchronization algorithm is limited in its performance by multiple sources

of distortions. As for the MFCC-based synchronization algorithm, since it is vulnerable to overlapping utterances, the performance is the worst among these three. However, without overlapping utterances, the MFCC-based algorithm can typically achieve better performance than the spectrogram-based algorithm. A hybrid algorithm combining multiple metrics may potentially be used for audio synchronization in future work.

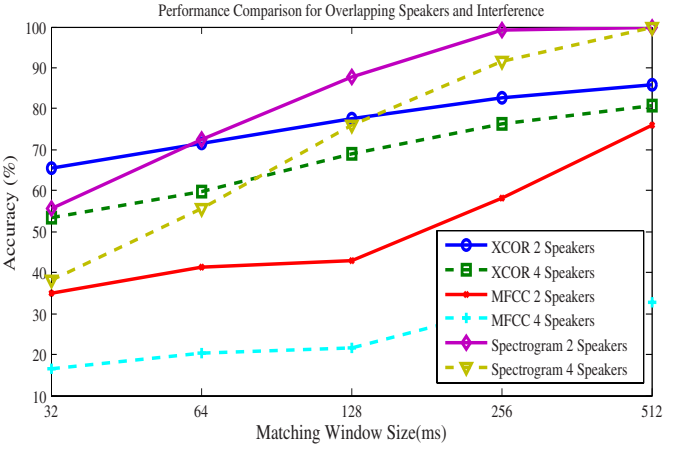

Fig. 14. Performance of the three algorithms against combined impairments

In conclusion, we have considered in this paper the problem of supporting video conferencing atop a multi-party audio conference with heterogeneous telephony devices. We have identified the importance of synchronizing the IP video stream and the PSTN audio stream in the target scenario. We have transformed the audio-video synchronization problem into the problem of synchronizing audio streams across heterogenous telephony systems. To address the synchronization problem between circuit-switched and packet-switched audio streams, we have proposed an end-to-end solution framework transparent to the heterogeneous network protocol suites involved. Under this framework, we have investigated three synchronization algorithms based on digital speech processing in the waveform, cepstrum, and spectrum domains. Evaluation results show that such DSP-based techniques are an appealing solution toward addressing the target problem.

Acknowledgment

This work was supported in part by funds from the Excellent Research Projects of the National Taiwan University under Grant 97R0062-06.

References

1. Hsieh, H.-Y., Li, C.-W., Lin, H.-P.: Handoff with DSP support: Enabling seamless voice communications across heterogeneous telephony systems on dual-mode mobile devices. IEEE Transactions on Mobile Computing 8(1), 93–108 (2009)
2. Liu, C., Xie, Y., Lee, M.J.: Multipoint multimedia teleconference system with adaptive synchronization. IEEE Journal on Selected Areas in Communications (J-SAC) 14, 1422–1435 (1996)
3. Xie, Y., Liu, C., Lee, M.J., Saadawi, Y.N.: Adaptive multimedia synchronization in a teleconference system. ACM/Springer Multimedia Systems 7(4), 326–337 (1999)
4. Kim, C., Seo, K.-D., Sung, W., Jung, S.-H.: Efficient audio/video synchronization method for video telephony system in consumer cellular phones. In: Proceedings of the ICCE 2006 Consumer Electronics, January 2006, pp. 137–138 (2006)
5. Liu, H., Zarki, M.E.: A synchronization control scheme for real-time streaming multimedia applications. In: Proceedings of 13th Packet Video Workshop (April 2003)
6. Yang, M., Bourbakis, N., Chen, Z., Trifas, M.: An efficient audio-video synchronization methodology. In: Proceedings of the IEEE International Conference on Multimedia and Expo., July 2007, pp. 767–770 (2007)
7. Lie, W.-N., Hsieh, H.-C.: Lips detection by morphological image processing. In: Proceedings of ICSP 1998, pp. 1084–1087 (1998)
8. Zoric, G., Pandzic, I.S.: A real-time lip sync system using a genetic algorithm for automatic neural network configuration. In: Proceedings of the IEEE International Conference on Multimedia and Expo., July 2005, pp. 1366–1369 (2005)
9. Cutler, R., Bridgewater, A.: Audio/video synchronization using audio hashing. Patent No. US 2006/0291478 A1 (December 2006)
10. Jourjine, A., Richard, S., Yilmaz, O.: Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), June 2000, pp. 2985–2988 (2000)
11. Rickard, S., Yilmaz, O.: On the approximate W-Disjoint Orthogonality of speech. In: Proceedings of ICASSP, May 2002, pp. 13–17 (2002)
12. Yilmaz, O., Rickard, S.: Blind separation of speech mixtures via time-frequency masking. IEEE Transactions on Signal Processing 52(7), 1830–1847 (2004)
13. Shan, Z., Swary, J., Aviyente, S.: Underdetermined source separation in the time-frequency domain. In: Proceedings of ICASSP, September 2007, pp. 945–948 (2007)