國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

以數位語音處理技術解決異質視訊會議之同步問題

On Using Digital Speech Processing Techniques for Synchronization among Heterogeneous Teleconferencing Devices

林孝蒲

Hsiao-Pu Lin

指導教授：謝宏昀 博士

Advisor: Hung-Yun Hsieh, Ph.D.

中華民國 97 年 7 月

July, 2008

誌謝

2006 年 7 月 30 日，正式搬上台北，開始碩士班的生涯，很快的兩年就過去了。兩年的時間，快得令人覺得不可思議…轉眼間，台北各著名景點都差不多去遍了；轉眼間，也在台北看了兩次國慶煙火，煮了兩次湯圓。實驗室的生活雖然我一直到後半段才慢慢了解，但卻是組成我研究論文裡最重要的一段日子。

從一開始暑假的 short course 就體認到老師對我們的關心，而每一回懷著忐忑不安的心情去個咪時，老師都相當有耐心的，從無法想像的角度切入問題的核心點，每次的一語道破都讓我心頭一震，原來…這就是神的領域！！感謝老師這兩年來對我們的付出，讓我們能度過如此充實的研究所生活。

雖然研究室並不是我每天待最久的地方，但是每次一踏上這塊地板，都讓我有種溫馨的感覺，實驗室中的歡笑叫罵，為煩悶思緒提供紓解的管道。感謝 ricky、koka、csdog、dcd、bBorg 等多位學長的幽默，讓我們一開始就能融入實驗室生活；每當我遇到問題的時候，感謝蔡偉和廖學長總是撥出時間來幫我解決；感謝佑恩壯碩的肌肉，一直在我身邊警惕我鬆垮垮的肚皮；宗霖的垃圾話大全，還有最新的日劇和棒球戰況，讓實驗室總是能在很歡樂的狀態；五哥的熱心公益，擔當實驗室的大家長，使我們能醉心於研究而無後顧之憂；kct 的任勞任怨，radio 的山大王氣息，排球 boy 小豪邁，為我研究生活的後半段增添不少趣味；特別感謝 Beach Boy 均韋讓我在最後的最後能夠有機會去衝浪一下！Special thanks to Tuan and Prida. You guys give me chances to know better about Chile and Vietnam.

還有要感謝我的前室友小甜和毓傑和我一起住了兩年，也感謝小璧每次在我煩悶的時候聽我訴苦；辛苦小綠在這兩年間的忍耐，歷經多次分分合合的考驗，遠距離真的是相當難熬阿～最後多謝我的老爸老媽把我養這麼大，總是尊重和支持我讀書路上的選擇，也謝謝老哥在我忙得焦頭爛額時給我的精神鼓勵，還有孝諭總是抓我在忙的時間上來吃我的喝我的。

2008 年 8 月 7 日 林孝蒲

摘要

隨著通訊技術的蓬勃發展，參與語音會議的通話平台逐漸包含許多日新月異的通信裝置，諸如：傳統家用電話、行動電話、雙模手機或是透過閘道器轉接的網路電話。為了擁有更好的會議經驗，在此異質語音會議中，有能力使用視訊裝置的參與者，可能會在語音會議之上同時建立一個包含部份與會者的視訊會議，此時經由PSTN網路傳送的語音與經由IP網路傳送的影像之間便有了不同步的問題。傳統的同步演算法，由於多是針對單一網路下的語音與影像同步，因此不適合用在此種應用下。

在本篇論文中，我們提出了一個使用端的語音與影像同步架構，並將問題等化為PSTN語音與IP語音之間的同步。首先，我們採用基於互相關係數的時域演算法，並發現此方法對於受到噪音或混音等干擾而失真的語音，有其效能上的限制，所以我們尋求透過數位語音處理技術，將語音訊號中不容易受到干擾的特徵取出，作為同步演算法的設計基礎。我們將語音辨識中常用的MFCC特徵加入同步演算法中，發現MFCC在處理受到壓縮與封包遺失的失真語音時，皆能達到不錯的效能；然而由於MFCC先天上的限制，並不能有效使用在混入多個與會者的語音同步中。有鑑於此，我們利用不同語者的語音在頻譜上會分散開來的特性，設計了基於語音頻譜的同步演算法，並發現此演算法較能抵抗雜訊以及其他語音的干擾。從效能評估的結果中，我們發現使用數位語音處理技術來解決PSTN語音以及IP影像之間的同步問題，能有較低的運算複雜度，並較能對抗語音的失真而得到不錯的同步效能。

ABSTRACT

As the popularity of multi-functional telephony devices grows, traditional audio conference now may involve heterogeneous teleconferencing devices, including POTS phone, dual-mode smart phones, pocket PCs, and so on. Among these conferencing devices, some may have the capability of accessing IP networks and supporting video conferencing with peer devices in the audio conference so as to have better conferencing experience. In this scenario, it becomes necessary to synchronize between audio streams, traversed the PSTN network, and video streams, traversed the IP network. While related work has investigated the problem of audio/video synchronization, their scenario is limited to the synchronization within homogeneous network, hence they cannot be applied in the target scenario.

Therefore, in this thesis we propose an end-to-end framework for audio/video synchronization. We then simplify the problem as one that requires only synchronization between PSTN and IP audio streams. We first employ a time-domain algorithm based on cross correlation and identify its ineffectiveness in synchronizing distorted audio streams, due to noises or packet losses. Hence, we seek to extract distortion-tolerant audio features by Digital Speech Processing techniques for synchronization. We apply MFCC in the synchronization algorithm and obtain respectable performance for audio streams distorted by codec and packet losses. However, MFCC is inherently vulnerable to overlapping speakers. Therefore, we leverage the sparsity of speeches in spectrograms to design the spectrogram-based synchronization algorithm, and achieve favorable performance for speech mixtures and noisy speech. Evaluation results show that using DSP techniques is helpful in solving the synchronization problem across PSTN audio streams and IP video streams in terms of accuracy and robustness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Ever since the invention of the first radio wave communication device decades ago, telecommunication techniques have developed from simple telegram to telephony service which transmits the actual voice from remote speaker. People all over the world are easily connected by widely-distributed telephone lines. This has led to the prosperity of telecommunication industry. Many convenient and appealing services are promoted by the telecom companies to attract users. One of the most interesting services is the multi-party talk service for audio conferencing. With the capability of conferencing over telephony systems, distant conferees have no need to commute from afar to the same conference site, and thus enhancing the efficiency of communication.

On the other hand, as the modern communication technology evolves, telephony service is available on various types of platforms to make communication almost ubiquitous. Beside of traditional POTS phones, 2G/3G mobile phones, satellite phones, dual-mode smart phones, and even IP phones with voice gateway have already come into the market. Therefore, when an audio conference is held, conferees may attend the conference through various kinds of telephony devices, as illustrated in Figure 1. This heterogeneity of teleconferencing devices suggest heterogeneous capabilities. For example, 3G Smart phones, pocket PCs, and IP phone with voice gateway are capable of both IP network access and PSTN network access while others are not. This extra capability of dual-network access has inspired an interesting research area as discussed in [1] which suggests that traditional telephony service can benefit from the capability of IP network access. This is an important inspiration to this thesis which is explained later.

On the other hand, the defect of conferencing through telephony system in comparison to face-to-face conference is that people cannot see the real-time image of others which sometimes may be helpful while discussion. This defect seems to be inevitable for devices without video capturing and transmission capability, however, for devices with video function, the conference experience could be improved if real-time image is available. This has led to the motivation of holding video conference only among the capable devices atop audio conference. Note that usually video conference

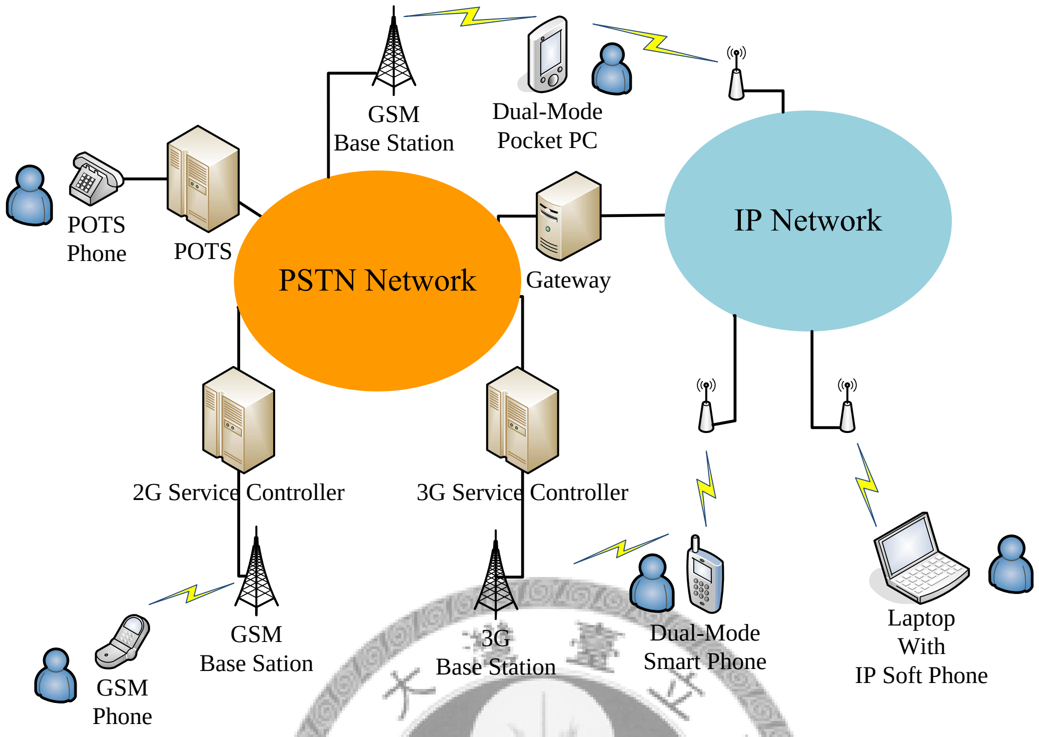

Figure 1: Conferencing over heterogeneous network

contains both real-time image and audio, however, since the audio conference is already in charge of the audio transmission, audio stream in the video conference part should be silenced to avoid echo.

Nevertheless, video conference service is not yet provided by most telecom companies. The well-known available video conference services are mostly provided through the IP network. Many research efforts are made in this area to provide better video conference structure such as [2–4]. Although some 3G telecom service provider, for example [5], claims that video calls are available through 3G system, video call service is still restricted to one-to-one calls. Multi-talk service is only available for audio. The video conference service provided for enterprise only provides a solution for 3G user to connect to enterprise video conference server through internet, and thus limiting the application to enterprise users only.

Therefore, holding the video conference atop audio conference through the IP network where video conference is easily and already supported seems to be the most appealing solution. When an audio conference is held, the conferees that are equipped with video transmission and display functions can decide to hold a video conference involving only the capable conferees. An important feature of this conferencing scenario is that *the audio conference is held through PSTN network*[1] *while the video conference is held through the IP network.*

Since video and audio conferences are held over different networks, heterogeneity in network environment may lead to different delays, jitters, and so forth. However, this distributed conference structure makes it inherently impossible to control and re-synchronize audio and video before reception since there is no central coordinator in the topology. Hence at the receiver side, video and audio are very likely to be asynchronous, and thus results in a perceptually unpleasant conference experience. The conventional research on audio/video synchronization might not be suitable here since they mostly consider only audio and video streams over the same network, whereas now the audio streams and video streams follow completely different protocols. It's hard for them to communicate and negotiate the time information with each other using protocol design.

Traditional audio/video synchronization research focuses on the conferencing in homogeneous network, therefore the timing information is not an issue. However, in the proposed scenario, since the timing information is corrupted due to the heterogeneous network, an algorithm which can re-establish this information according to the audio and video streams should be considered. Related work on lip synchronization provides a possible direction. Nevertheless, the time-consuming video processing and the vulnerability to interferences properties of lip synchronization imply the unsuitableness of this direction. Therefore, a different synchronization algorithm needs to be considered.

Thereby, we propose an end-to-end synchronization framework which requires no infrastructure supports to solve this problem. By taking advantage of the IP network, we simplify the problem to the synchronization between PSTN and IP audio streams. When the synchronization between these two streams is achieved, the timing information between PSTN audio and IP video can be easily derived.

To address this audio synchronization problem, we first propose a time-domain cross-correlation algorithm and depict the insufficiency of this time-domain algorithm. Hence, inspired by the research in Digital Speech Processing (DSP) [6], another MFCC-based algorithm is proposed. However, after the evaluation on the robustness of this algorithm, we point out the defects of this MFCC-based algorithm. Another spectrogram-based algorithm is thus proposed to address those defects.

[1] In the following text, PSTN is referred to as the general term for all traditional telephony networks, including POTS, GSM, UMTS, and so on.

The chapters in this thesis is organized as following. In the following chapters, the detailed background of the research scenario is first stated. Some related papers are reviewed and discussed in the following part. Previous works mostly provide solutions relying on the common time stamps stamped on both video and audio streams. The synchronization design considers how to design a protocol to control media flow so as to avoid playback buffer underflow or overflow. However, since in the proposed scenario, video conference and audio conference are held on heterogeneous networks, difference in network protocol makes these two streams difficult to negotiate the timing relationship through common time stamps. Therefore, recovering the timing information through the transmitted content is the main focus. Several related papers on synchronizing human speech which directly use video and audio contents for synchronization are reviewed. Nevertheless, several essential drawbacks implies the unsuitableness of this direction.

Next, in Chapter 3 we propose an end-to-end synchronization framework and simplify the problem into audio synchronization, which directly compare the transmitted audio streams from both networks. Preliminary measurement on the asynchrony between audio from PSTN network and video from IP network is conducted. Although the result only shows the delay difference of a simple environment, the well-known time-varying network characteristic of IP network suggests that the received audio and video streams may be asynchronous. Under this framework, the challenges of audio synchronization is discussed in the following part.

In Chapter 4, time-domain-based synchronization algorithm which uses time domain cross correlation on two audio streams is examined. Evaluation of performance shows the insufficiency of using only time domain characteristics of audio in that the timing structure may be easily corrupted by interferences and other distortions on audio. Although the performance can be improved by including more samples in the cross correlation, the resulting computation time may be so large that the synchronization algorithm may not be reactive to the network dynamics. Hence, *exploiting other representations of speech through Digital Speech Processing (DSP) techniques* that better characterize the speech of interest is considered.

Inspired by the speech recognition research area, a widely-used DSP feature called Mel-Frequency Ceptral Coefficient (MFCC) is first examined in Chapter 5. We found a patent for a similar application which proposes a simple synchronization algorithm based on MFCC. Since that solution focuses more on the design of synchronization framework rather than the synchronization algorithm for locating synchronization point, the effectiveness of using MFCC for synchronization against the aforementioned challenges is analyzed. The analysis reveals that MFCC might be fragile to interference of other speakers included in the PSTN network.

Therefore, in Chapter 6, we adopt another DSP feature, the spectrogram, of speech which is better tolerable to interferences and noises, as suggested by many blind speech separation research. The advantage of using spectrogram for the design of synchronization algorithm is described and evaluated. Then an algorithm based on the merits of time-frequency representation is proposed, followed by performance analysis in terms of the robustness to interference and other sources of waveform distortion.

From the performance evaluation, we conclude that the MFCC-based synchronization algorithm is more robust to codec distortion and packet loss while the spectrogram-based algorithm is more robust to AWGN noise and overlapping speakers. The measurement of computation time required for both MFCC-based and spectrogram-based algorithms is only of the order of a few milliseconds. Therefore, these algorithms can be better reactive to network dynamics.

CHAPTER 2

BACKGROUND

Before we step into the discussion of the synchronization algorithm in the proposed scenario, more insights into the heterogeneous teleconferencing structure is given in advance to better motivate the synchronization issue. Several related papers are reviewed in the field of audio/video synchronization.

2.1 Heterogeneous Teleconferencing Scenario

The heterogeneous teleconferencing scenario is illustrated in Figure 2. As shown in Figure 2, five conferees are attending the PSTN audio conference while only three of them, which are the dual-mode pocket PC, dual-mode smart phone, and the laptop with IP soft phone,are the participants of the IP video conference.

While the audio conference is usually controlled and maintained by the audio conference server belonging to a telecom company, the video conference is not restricted to this centralized structure. The IP video conference can be held by either centralized or distributed structures. In the following part, we intend to give an overview of how these conferencing structures work so as to acquire more in depth understanding of the heterogeneous teleconferencing structure.

2.1.1 Audio Conference Architecture

Audio teleconferencing services involve the use of computer-controlled electronic equipment known as an audio teleconference bridge (bridge). A bridge is similar to a telephone exchange PBX switch in that many telephone lines may be connected to it to accommodate either incoming or outgoing calls. Unlike a PBX connection, a conference may be established through the bridge. The bridge permits simultaneous speaking by all participants, eliminates clipping, filters out the echo of each participant's own speech, equalizes sound volume and clarity, and permits both dial-out and dial-in connections so that participants may join the conference either by a call from the teleconference operator or by dialing a prearranged number. For each conferencing end, the received voice is the mixture of all the other conferees' voice.

The network where a conference bridge lies usually determines the type of this

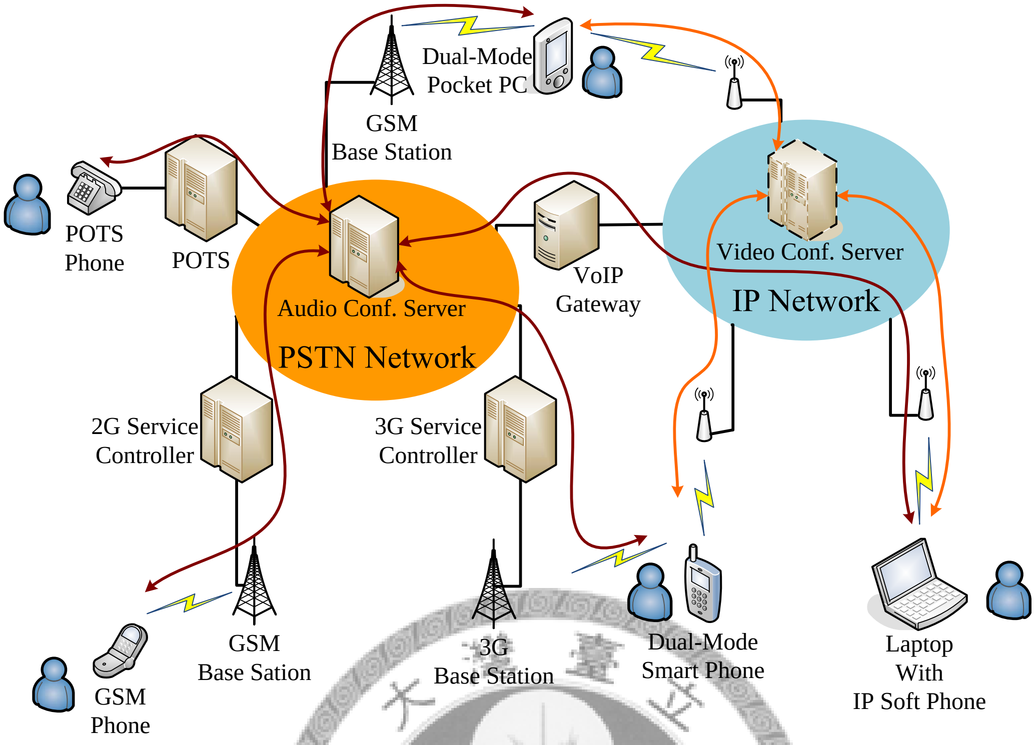

Figure 2: Audio and video conference held over heterogeneous network

conference bridge which could be a PSTN bridge, an IP bridge, or even a hybrid bridge. For example, [7] has announced a three protocol audio conferencing bridge which can support traditional telephones, internet connected phones, and SIP devices in the same conference call. Therefore, the audio conference server could locate at both PSTN and IP networks. However, these sophisticated conferencing bridges are usually aimed at enterprise clients. For general public telephone users, usually the only available audio conference server might be the one provided by the traditional telephone service company.

However, in the traditional conferencing architectures, Conference Service Providers (CSPs) use circuit-based TDM audio conference bridge equipment to integrate conferencing application logic, TDM interfaces, and audio mixing circuitry into a single piece of proprietary networking equipment. Therefore, the end users are unable to interfere the audio process that are manipulated by the telecom company. When the audio arrives at the bridge, the processing time of bridge adds additional delay to this audio stream, and then is sent to the receiver. Since the bridge processing time are usually the same and the PSTN network is relatively stable, the end-to-end delay doesn't varies a lot.

Note that in the PSTN audio conference architecture, the audio bridge bridge locates at the CSPs which is out of reach of general public. Any attempts to participate in the intermediate audio process is unlikely to be available.

2.1.2 Video Conference Architecture

On the other hand, for the video conference, most conferencing applications utilize the open structure of IP network which makes the conferencing architecture more flexible than the PSTN network. In [8], the authors collect and conclude the recent experiments and reassessment of practical implemented video conference systems. These systems can be roughly categorized into two categories which are the centralized and the distributed categories.

2.1.2.1 Centralized Structure

In a centralized video conferencing structure, usually a central coordinator called the Multipoint Control Unit (MCU) which is in charge of the video process is required. All the video streams from different conferees converge on this MCU. Then the MCU may first decompress all the received video streams. For each specific conferee to whom all the other video streams are destine, the MCU may re-compress the required video streams depending on his bandwidth, display resolution, and other capabilities. The authors in [9] provide an experimental analysis to support video adaptation over an extremely large range of display requirements. This decompress and recompress process are usually managed by a transcoder. In other words, the MCU may decompress all the video streams and then compress the required video streams for each specific conferee according to their capabilities.

The MCU usually has larger bandwidth than a regular participant so it can receives all participants' video signals and disseminates them after properly processed. Since all the video streams gather at the MCU, the synchronization of audio and video signals among conferees can be done by the MCU. The computation load is mostly on the MCU, so the end system can be relaxed from considering the computation capability. In order to make everyone in the video conference can see everyone else, the MCU usually merges all the received video streams into one single video stream where every participant occupies a certain location in the output video frame. The video combination problem is another research issue in the field of video conferencing, as in [10, 11]. For the audio streams, a mixing process as performed in the audio bridge is applied. The timing relationship between the combined video and audio stream is updated while combination. These audio and video mixtures are recovered by this updated timing information at the receivers to achieve synchronized playback.

However, the centralized schemes have their inherent defects. (1) Although the MCU has larger bandwidth than regular participants, but still its bandwidth is limited. Besides, since the MCU is in charge of the transcoding process for all the conferees, the complexity may increase rapidly as the number of participants increases. From this point of view, the scalability of the MCU is bad. (2) Since each conferee has to transmit to the MCU even though they might be near each other, it requires long round-trip delays for resource allocation and cannot react to fast changing conditions in both communication channel and video content. (3) The high cost and management complexities make MCUs suitable only for larger business applications. (4) When the MCU is down, no conference can be hold.

2.1.2.2 Distributed Structure

While a centralized structure is usually adopted for enterprises which have more privacy and security concerns, it is not available for most general public. Therefore, for general video conferencing service, a distributed conferencing structure is usually applied. Instead of centralized control, system designers can realize conferencing systems with a distributed fashion by utilizing receiver-driven layered multicasting algorithms and/or multihop forward error coding (FEC) transcoding to respond to time varying and heterogeneous channel conditions.

Since multiple streams are exchanged among multiple users, these streams may share the same transmission path. A dynamic resource allocation for each stream with awareness of other coexisting streams in the same path is more efficient than a static allocation. In [12], the authors explore the multi-stream diversity to provide better video quality and study how to perform cross-layer multi-stream error protection in a distributed manner.

The most critical problem for a distributed architecture is the limited bandwidth. Unlike the centralized architecture, where MCU has higher bandwidth than an ordinary node, the distributed architecture requires each conference node transmits its video to all the other nodes. Therefore, many bandwidth saving techniques are proposed to leverage this problem, including application layer multi-cast and a request-for-viewing system, as stated in [13, 14]. Note that in this distributed structure video from different conferees follows different network path to the receiver, the audio/video synchronization is handled for each stream.

In conclusion, the video conference architecture can be categorized into centralized or distributed categories. The received video streams is usually a combination of all the other conferees' in a centralized structure while video streams are individually received. No matter what video architecture is applied, in the proposed scenario, the audio conference and the video conference are held in different network. Since the audio and video conference is unlikely to be coordinated at a central device, we focus on solving the asynchronism problem of audio and video at the receiver side in an end-to-end manner.

2.2 Related Work

In a typical video conferencing system, audio and video signals are captured periodically at the source, fragmented into media data units (MDUs), packetized and transported in real time to the destination in separate streams. To faithfully recover the original form of the audio/video presentation, both the temporal ordering among the MDUs in a stream and the relative temporal relationship among streams need to be maintained. In other words, video conferencing applications require both "intra-stream synchronization" and "inter-stream synchronization". A common solution to this problem is to use a receiving buffer, which can smooth out the delay variations for each stream at the destination. By comparing the object timestamps as suggested in MPEG standard [15], the received MDUs are first placed into a buffer temporally, and then decoded and presented according to a predetermined fixed timeline. In the following part, we review several related papers in synchronizing audio/video streams.

2.2.1 Conventional IP Audio-Video Synchronization

In most IP synchronization schemes, receiving buffer is essential as afore-mentioned. The receiving buffer size determines the resistance of synchronization control scheme to network delay jitters. Larger buffer size makes the scheme more resistant to large network jitters. However, increase in buffer size also increases the delay before playback. For real-time services like teleconferencing, large delay may decrease the interactivity of conference. Therefore to make a compromise, most research endeavors propose adaptive buffering schemes for synchronization control as in [16–20]. Authors of [16, 17] proposed an adaptive buffering scheme by piecewisely equalizing the end-to-end delays of multimedia objects in order to suppress the synchronization phase distortion with minimal trade-in of buffering delays. The generating time of multimedia objects are time-stamped by the common local sample clock which is the synchronization source shared by all outgoing media streams. At the receiver side, objects along different streams are scheduled to playback according to their playback clocks to which a control mechanism using time-stamps as reference is employed.

Recent research schemes usually utilizes the timestamps included in the RTP (Real-time Transport Protocol) packets which starts with a random number and steps forward by sampling period to identify the matching audio and video slices for synchronization playback. Authors in [21] proposed an adaptive transmission scheme to ensure the continuous and synchronous playback of audio and video streams. Their proposed adaptive scheme is composed of three stages, namely, (1) dynamic reordering mechanism, (2) decoding-recovery mechanism, and (3) adaptive synchronization mechanism. The first two stages reorder the out-of-order packets and recover the possible lost packets by proper algorithm according to the network status. The third stage adaptively adjusts the queueing length to resist inter-arrival jitters and variances of the end-to-end transmission delay. They claim that their adaptive synchronization algorithm is able to control the queuing length precisely to eliminate the time-based skew between the audio and video streams and minimize the end-to-end delay.

In [18–20], the authors proposed an adaptive delay and synchronization scheme that (1) directly incorporates the quality requirement of the application into the parameters of the algorithm, (2) calculates the synchronization errors in real-time, (3) piecewisely adjusts end-to-end delay by controlling the virtual local clock to adapt to the network delay variation, and (4) gracefully recovers the synchronization if synchronization error occurs. This scheme monitors the synchronization errors and estimates the delay jitters among adjacent Media Data Units in real-time to compensate for the delay jitters. While [18] focus mainly on the synchronization for real-time streaming multimedia applications, [19, 20] concentrate on the audio/video conferencing application. By maintaining a virtual clock according to the playback time and arrival time at the receiver side, the synchronization control scheme can adjust the clock to match the QoS requirement. In order to reduce the computation load of synchronization while computing the correct match from the received RTP and RTCP SR (Sender Report), authors in [22] proposed an efficient decision rule for calculating the playback time without floating point operations.

To sum up, the above mentioned research mostly utilizes adaptive buffering schemes for synchronization control. The basic control, which consists of appending synchronization information (timestamps, sequence number, etc.), is essential for all algorithms. However, in the proposed scenario, audio conference is controlled by the telephony company while the video conference is held on the open IP network. Audio signals from all the conferees are sent to the mixer owned by the company and then these signals are mixed and sent to other conferees. There is no timestamps of specific speaker in the mixture. Therefore, conventional ways of audio/video synchronization to match video and audio timestamps is not applicable.

Authors in [23] propose an interesting synchronization methodology that requires no timestamps for audio/video synchronization. In this methodology, audio data is embedded within the corresponding video frames by means of high bitrate information hiding techniques. On receiving the video frames at the receiver, the embedded audio data is extracted and played along with the host video frame. Nevertheless, in the proposed scenario, the received audio stream might be a mixture of speakers from different ends, it is impossible to embed the mixture in advance within the video frames.

In conclusion, conventional synchronization control schemes usually depends on the common timestamps on the audio/video streams for inter-stream synchronization. The timing information can be recovered by these timestamps. Conventional IP audio/video synchronization schemes mainly aim at providing an adaptive transceiving scheme to accommodate the varying network delay and jitter which may disorder the receiving packets or even incur packet drops.

2.2.2 Lip Synchronization

As described in the previous subsection, conventional works on audio/video synchronization mostly require the *timing information* of audio and video to be appended in the media streams. However, in the proposed scenario, even though the timing relationship between audio and video can be stamped on both streams, the audio conference server in the PSTN network might destroy this information while mixing the audio streams from different conferees. Therefore, the receiver can only observe a multiple-source mixture with no individual timing information of specific conferee. Hence faithfully recovery of audio/video playback according to the timestamps is infeasible.

Since the timing information attached at the sender is discarded while transmission, we think of another direction of retrieving the timing relationship between audio and video streams at the receiver side. Since the only available useful information from the audio conference is the audio stream itself, the most possible method of recovering the timing relationship might be comparing the audio content to the IP video stream.

As shown in Figure 3, the video stream and audio stream leave the sender to different networks. While the video stream remains unmodified to the receiver, the

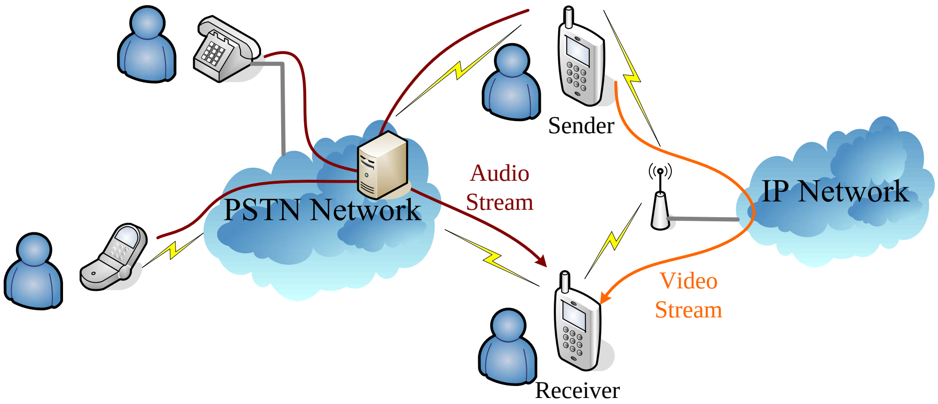

Figure 3: The concept of content comparison

audio stream may accumulate extra speakers' voice and arrive as a audio mixture. The concept of timing recovery by content comparison is to compare the received video stream containing only the sender's information to the received audio stream which might be a mixture.

Since it's the spoken speech that is of interest in the audio segments, the content comparison is related to the research area called lip synchronization which is a technical term for matching lip movements with voice, as stated in [24]. Among the multiple meanings of the term lip sync, it is referred to as the science of synchronization of visual and audio signals during post-production and transmission.

The lip sync techniques can be applied to many interesting applications such as automatic lip movement for animation characters as proposed in [25,26]. The authors extract audio features from the input speech and then use a pre-trained neural network model to map the pronounced speech to suitable visual lip movement and then show on the character's face. Since in this application the lip movement is pre-stored in the phoneme database, it requires less image processing load, and thus is claimed to be used for real-time application.

However, in the considering scenario, the lip sync is more related to the research area as in [27, 28] which is usually applied to lipreading, speech recognition, and audio/video synchronization. For all of these applications, the following issues should be addressed:

1. face localization,

2. facial feature localization (e.g. the eyes and the mouth),

3. lips modeling,

4. lips tracking and motion analysis,
5. identification and recognition.

On receiving each video frame, the human face should be first located, and then the facial parts are identified. From the partitioned facial components, the lip shape is analyzed and modeled. Combining consecutive video frames, the lip movement is characterized for further identification and recognition. Meanwhile, the human speech is segmented and characterized by speech features. Both the characteristics are fed to the audio-to-visual model to find the correct mapping for matching determination. Note that the matching determination is directly derived from the audio-to-visual model which might not necessarily use phonemic analysis. After the matching determination, the timing information could be recovered from the results. Nevertheless, the lip synchronization techniques have some inherent defects to be applied to the proposed scenario which is discussed later.

An important issue of lip synchronization to the proposed scenario is the computation load. Since most audio-to-video lip synchronization research focuses more on the off-line applications, the computation load introduced in the image processing stage is usually not the issue to be considered. However, in the proposed scenario, even though the time for obtaining timing information is not as critical as real-time applications, larger computation time may make the algorithm less reactive to the variation of network characteristics. After the computation of timing information, the network condition might have changed. Another issue is the large audio-to-visual model. In order to achieve better identification performance, training data should be used to establish the model which consequently require larger memory resource. Because in the proposed scenario the teleconferencing devices could be a dual-mode handset which has limited computation power and memory resource, the lip synchronization techniques might not perform well in this scenario.

Another problem occurs when the audio stream consists of multiple speakers' speech. Since the audio-to-visual model is trained by clear speech with the lip movement, speech mixture may confuse the mapping process, thus resulting in wrong judgments. This is the very problem that makes lip synchronization unsuitable for the proposed scenario. In conclusion, from the discussions in this section, we conclude that directly compare the audio segments to the video frames might not be applicable, either.

CHAPTER 3

A FRAMEWORK FOR AUDIO-VIDEO SYNCHRONIZATION

From the previous chapter, we conclude that (1) in the proposed scenario the asynchrony between audio and video stream could be a severe impact on conferencing quality, (2) traditional synchronization control schemes are not applicable for their reliance on appended timing information at the transmitter, and (3) recovering timing information by means of lip synchronization has its inherent challenges. In this chapter, we propose a synchronization framework based on the concept of direct comparison of transmitted contents from PSTN and IP networks. We extend this concept further to audio to audio synchronization and discuss the potential problems of using audio streams from both sides for synchronization.

3.1 Synchronization Framework

As concluded in the related work of lip synchronization, directly comparison of audio segments and video frames might not be suitable. Therefore, we try to simplify the problem so as to avoid the requirement of audio/video comparison. Instead of considering synchrony in video and audio streams, the problem can be solved by simply synchronizing audio streams from different networks. This concept of simplification is elaborated in the following paragraphs.

3.1.1 Concept of Simplification

Inspired by the afore-mentioned related work [23], adding audio information in the video stream is attractive because the timing information between this appended audio and the video streams is easily derived. At the transmitter side, not only video frames but also audio information related to the current audio stream are sent through the IP network to the receiver as shown in Figure 4. This direct timing relationship between appended audio and video streams can be achieved by either timestamping the audio information or embedding the hashed audio information in the video packets as suggested in [6].

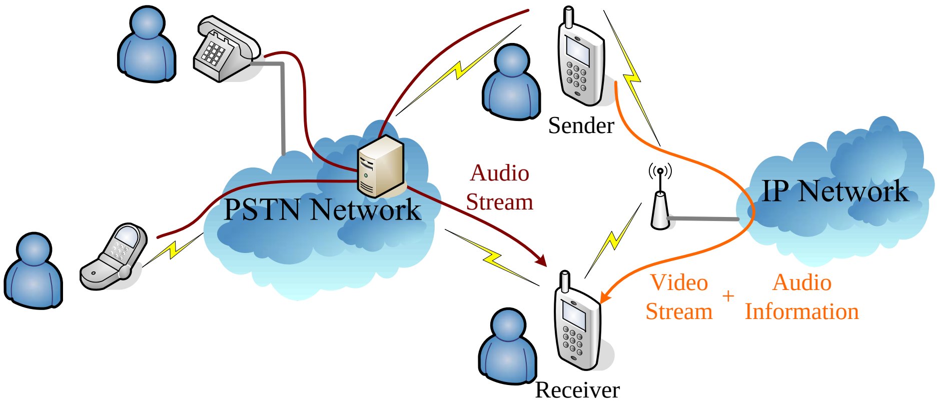

Figure 4: The concept of problem simplification

As a result, after the extra audio information from the IP network is received at the receiver side, it can be used to determine the timing relationship with the PSTN-audio stream. Consequently, together with the timing relationship between the extra information and video frames, the synchrony between IP-video and PSTN-audio streams can be achieved.

3.1.2 Proposed Framework

Based on the concept of audio synchronization, we propose a synchronization framework as shown in Figure 5. In Figure 5, the audio conference is held by the audio conference server located in the PSTN network. Conferees attend this audio conference via various kinds of teleconferencing devices, including a traditional POTS phone, a GSM phone, a dual-mode pocket PC, a dual-mode smart phone, and even a laptop using IP soft phone. On the other hand, the video conference is held only among the dual-mode pocket PC, the dual-mode smart phone, and the IP soft phone. The video conference could be supported by either a central video conference server or in a distributed manner.

For the conferees of video conference, since the audio stream and video stream are from different network, we propose an *end-to-end audio/video synchronization module* equipped with them so as to accommodate the asynchronism between audio stream and video stream. This audio/video synchronization module recovers the timing relationship between audio stream and video stream based on an audio synchronization scheme without any help from the audio or video conference server. Therefore this end-to-end approach can *recover the timing relationship without modifying the network infrastructure.*

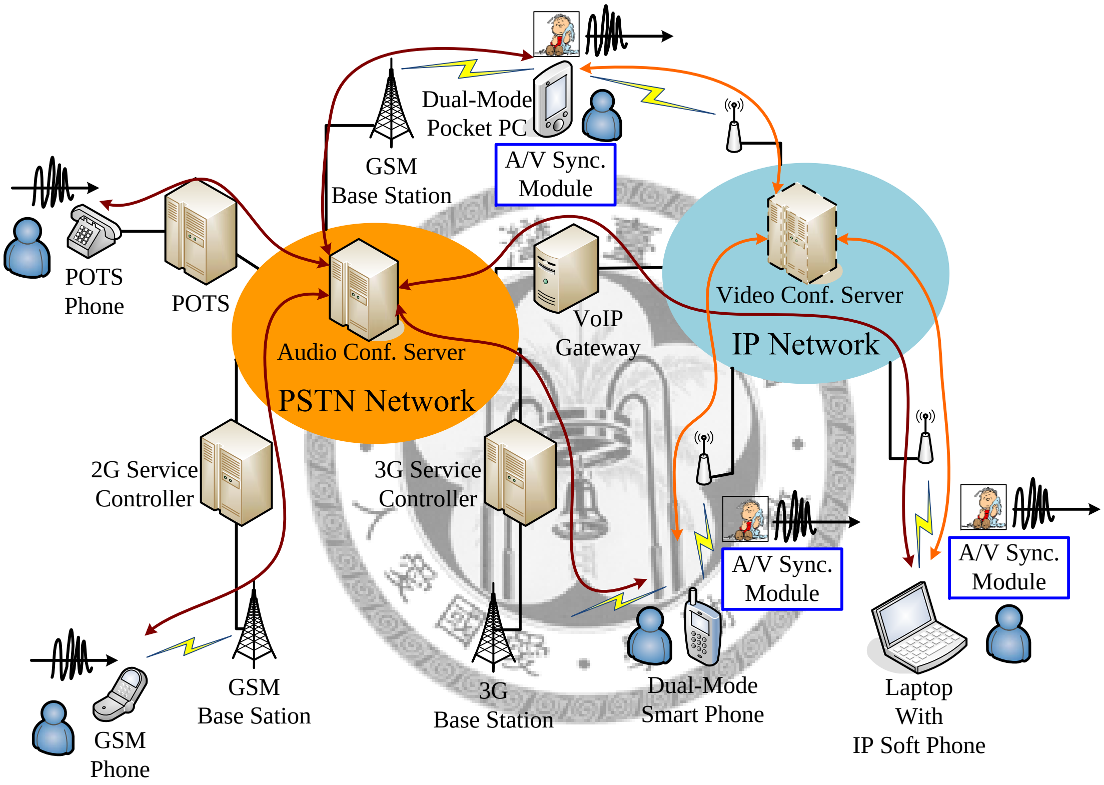

Figure 5: The proposed synchronization framework

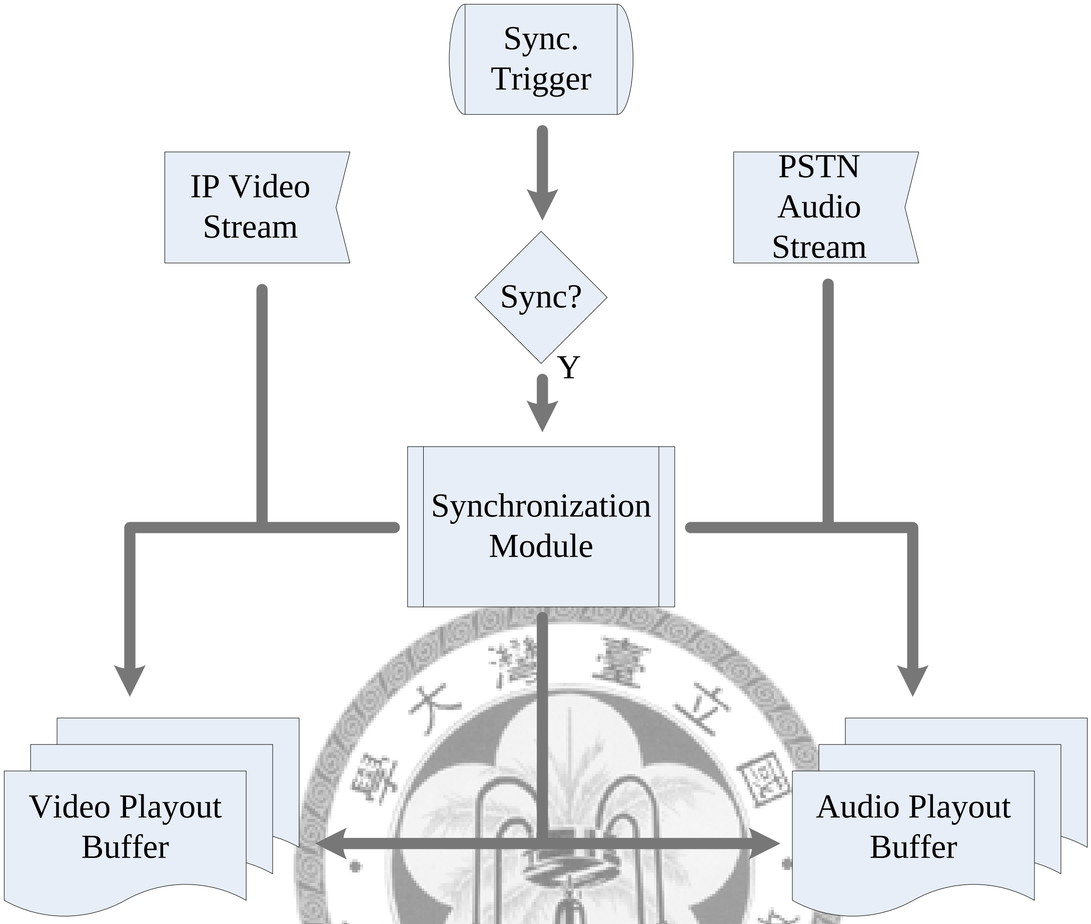

Figure 6: Detailed block diagram of the synchronization module

A more detailed block diagram of the proposed synchronization module is shown in Figure 6. When the conferee receives the video stream from IP network and audio stream from PSTN network, both of them are fed into two buffers which are the playback buffer for audio/video playback and the additional buffer inside the synchronization module for synchronization determination. We design the synchronization framework as trigger-driven since once the audio and video streams are synchronized, there is no need to waste computation power on the synchronization algorithm if the network condition doesn't varies a lot.

Whenever synchronization is triggered, the synchronization module feeds the pre-queued data inside the buffers to the synchronization algorithm which compares the above data to determine the matching point of these two streams. Note that data are pre-stored in the buffers so as to eliminate the data collecting time while synchronization is triggered. The synchronization algorithm is the core of synchronization module. It locates the matching point of these two streams and then uses this information for timing recovery. The resulting timing relationship is fed forward to the playback buffer where common synchronization control schemes can be applied.

Additionally, whether the received audio/video streams need to be synchronized depends on the synchronization trigger which can be determined by simply periodically or by network statistics obtained in the RTP packets. For example, according to the network statistics in IP network, if the delay or jitter exceed a certain threshold, the trigger can conclude that the video and audio might be asynchronous, thus triggering the synchronization. Further discussion on the design of the synchronization trigger is out of the scope of this work, so it is not included afterwards.

In conclusion, the proposed approach to synchronization is try to directly compare the received contents from PSTN and IP network for timing recovery. In the following context, first a measurement of asynchronism of the proposed scenario is performed so as to motivate the requirement of synchronization. Then the content used for comparison in the framework is discussed in the following part.

3.2 Asynchronism Measurement

Before the discussion of synchronization algorithm, the first question is that whether the asynchronism between audio and video really affects the conference experience. In other words, if no synchronization is performed, will the audio and video streams be so asynchronous that conferees may feel uncomfortable? In this section, several experiments are conducted to evaluate the degree of effect of asynchronism between PSTN audio and IP video in the proposed scenario, in order to motivate this research. The results reveals that conferees might feel perceptually uncomfortable in terms of delay difference and variance between audio and video streams.

3.2.1 Experiment Setup

In order to inspect how severe the asynchronism between audio and video may be, a simple testbed, as shown in Figure 7, is set up to measure the delays of audio and video traversing over different networks. According to the difference of delays of two sides, whether this asynchronism causes perceptual awareness can be determined.

On the one hand, for measuring the end-to-end delay of audio conference, using common one-to-one phone call is not enough since conferencing might include extra delays while processing the multi-end audio streams. We use the multi-party call service provided by CHT Telecom to hold an audio conference. Three conferees

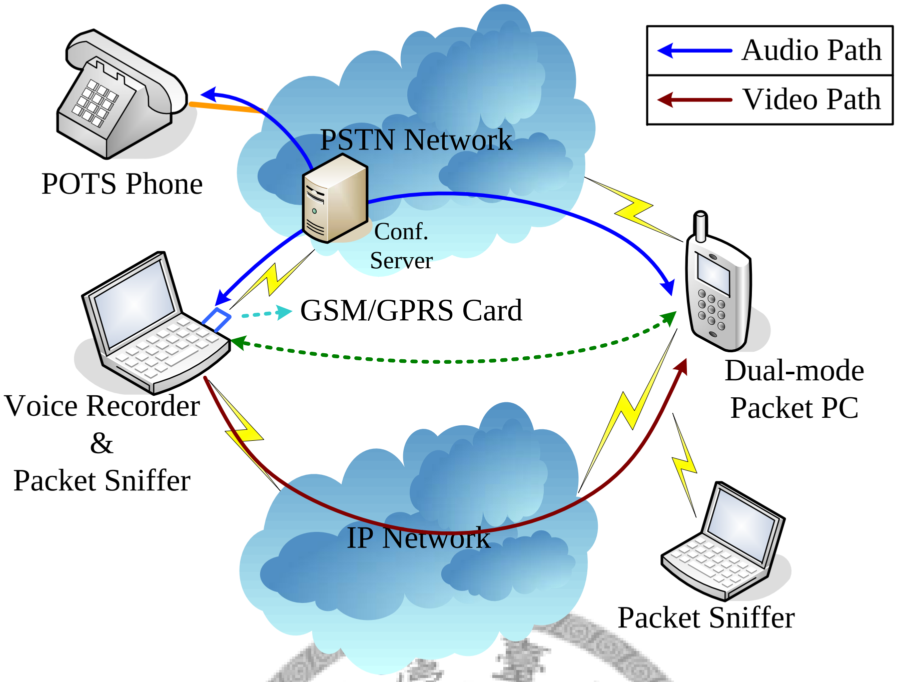

Figure 7: Testbed setup for audio conference delay measurement

which are a POTS phone, a 3G dual-mode pocket PC, and a laptop equipped with BenQ GSM/GPRS/WLAN PCMCIA card are attending this conference.

When the audio conference is held, the POTS phone is kept silence. The speaker pronounces a beep sound to the microphone on the dual-mode handset. The beep sound goes through both the PSTN network as the audio input from the dual-mode pocket PC and the open space (the dotted line) to the the laptop. Then these two waveforms are recorded using GoldWave [29] at the same laptop. In that way, the difference in time of these two waveforms which represents the end-to-end delay can be obtained manually, as shown in Figure 8. In Figure 8, there is a chain of input and output sets. Within each input and output set, the differences of waveform start point represents the end-to-end audio delay.

For the video stream delay measurement, the main objective is to inspect on the delay of video transmitted along the IP network. Therefore, for simplicity we make the laptop perform as a video streaming server and the pocket PC as a streaming client. Both the streaming server and client use wireless connections for network access, so as to simulate the situation for handsets. While the streaming server sends out QCIF format video to the client, the laptop simultaneously sniffs on the stream and record the transmitted packets. At the client side, another packet sniffer is included to sniff on the receipted packets. From the recorded time differences of the sniffed packets,

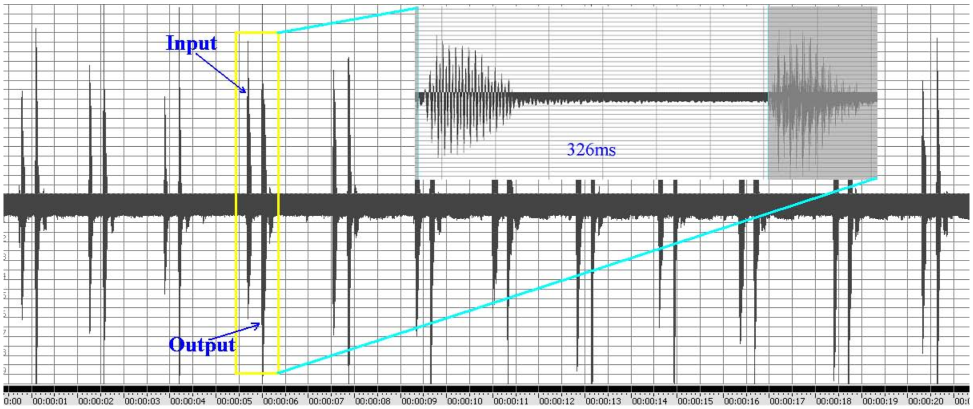

Figure 8: The recorded audio input and output from the audio conference

we can compute the delay of the video stream.

Note that, since the packet sniffing is performed on different laptops, the time clock might be asynchronous. Therefore, before the sniffing process, these two laptops have to be synchronized by Network Time Protocol (NTP), proposed by [30]. Although the NTP might not be able to perfectly synchronized these two laptops, authors in [31] claim that the synchronization error is only few milliseconds in LAN's and a few tens of milliseconds in WAN environment.

Another aspect of this video measurement is that we only consider one video stream to the receiver. Since more conferees in the video conference in a distributed manner may increase the traffic load, the end-to-end delay may also be increased due to the limited bandwidth.

3.2.2 Measurement Results

The measurement of audio conference delays is shown in Table 1 while the delay of video packets is shown in Figure 9. For the result of delay in audio conference, we first measure the delay of simple one-to-one calls as a comparison to the delay of 3-party calls. We can observe that in a 3-party call the delay is usually more than 400 ms, while the delay of a one-to-one call is usually between 300 to 400 ms. Although the average delay value may vary among different time of experiment, the delay variance is kept small during the same call.

Furthermore, the 3-party call usually results in longer delays than the one-to-one call. This additional delay might be introduced by the conference bridge that mixes the received audio and dispatches to the required speaker. Since the PSTN network is run and controlled by the telecom company for mainly audio transmission, the

Table 1: Measurement of audio conference delays

1 to 1 talk	Delay in ms					Average
First call	376	373	374	376	374	374.6
Second call	327	321	326	322	328	324.8
3-party talk						
First call	412	412	414	414	415	413.4
Second call	434	438	442	435	441	438.0

Table 2: Measurement of video delays

End Point	Delay in ms				Average
Lab to Lab	6	6	7	8	6.8
Library to Lab	25	31	24	28	27.0

network reveals a rather steady and QoS-guaranteed behavior. Therefore, the delay of audio conference is usually maintained at a certain level and the delay jitter is usually small.

On the other hand, the video delay is measured in two places. While the streaming server always stays in our lab and is connected to our own AP (named TONIC1), the streaming client is connected to the NTU campus AP either in our lab or in the library. The result is listed in Table 2 while Figure 9 shows one of the experiments. The results show that the delay is only around ten milliseconds while the streaming client is connected to the NTU AP near our lab, and also the variation in delays low. This may be due to the low traffic loads of these two APs and the short distance between them since they are in the same building which implies fewer intermediate routers are traversed. However, for the delay measured when the streaming client is in the library, the delay value is larger and there are several spikes indicating large delay variation. Since there are more WLAN users in the library and the longer distance between APs requires more intermediate routers to traverse.

The difference of measured audio and video delay is about 300ms. According to [32] which claims that ±160ms of asynchronism between audio and video is approximately the threshold for human awareness, a 300ms skew in synchronization is large enough for human awareness. Therefore, the synchronization algorithm is required even when this simple environment.

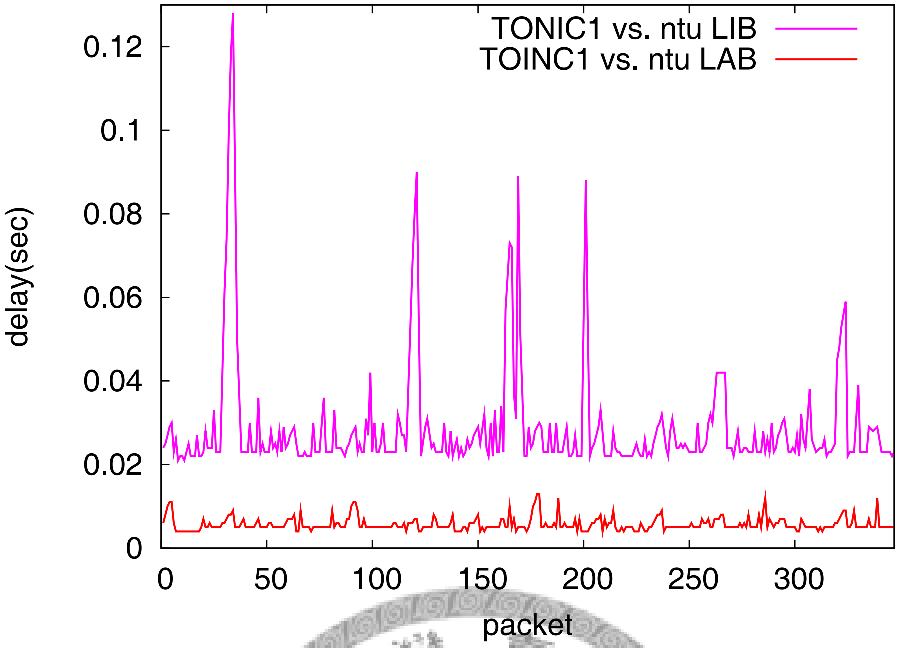

Figure 9: Delay of video packets

Although the video delay value is small which may not be the case for practical scenario, these measurement has revealed the key characteristics of IP network. When the traffic traverse through a long distance, the growing number of intermediate routers and the included traffic loads from other users may add uncontrollable delays to the traffic. The situation becomes even worse when the link involves wireless connections since wireless transmission could be severely affected by the current environment.

To compensate for the delay variation of IP video packets, playback buffers are usually applied to absorb the delay jitter. However, if the network dynamic exceeds the capability of playback buffer, buffer underflow or overflow may occur and thus the playback time may be adjusted. Since the PSTN audio is completely ignorant to this adjustment, the asynchronism may become larger. As a result, we conclude that synchronization is necessary to provide a better conference experience.

3.3 Challenges of Audio Synchronization

Nevertheless, when the problem is simplified to audio synchronization, several challenge might appear which might affect on the performance of audio synchronization. We individually elaborate on the challenges in the following paragraphs.

3.3.1 Distortion by Voice Codec

In order to reduce the bandwidth requirement and to increase the robustness to lossy channel, the audio signals are usually encoded before transmission, especially when the channel contains wireless connections. The widely used voice codec in PSTN network for GSM and UMTS voice is the Adaptive Multi-Rate codec while in VoIP application various types of codec such as G.723 and G.729 can be chosen from.

Since voice signal is considered to contain a lot of redundancy, the design of voice codec usually remove the redundancy to a certain level, thus resulting in a lossy compression. Since different network uses different voice codec, the discarded components might be different, therefore, the recovered audio waveform might be different. Difference in audio waveform might introduce a potential problem in the process of audio matching.

3.3.2 Distortion by Noise

In practical conversation, more or less noise is included in the audio stream. The noise could be due to different sources such as thermal noise, and environmental noise. No matter how the noise is induced, it does affect on the audio waveforms. However, for the conferee whose audio/video streams are to be synchronized, if the noise is introduced at the conferee, the effect of noise on comparison might be less since both audio streams contain this noise. On the other hand, if the noise is introduced by other conferees in the audio conference, this noise adds additional difference to these two audio streams.

3.3.3 Interference by Other Conferees

As illustrated in Figure 4, as the audio stream go through the PSTN network, multiple speech sources might be accumulated by the conference server to form a mixture and then is dispatched to the conferees. Therefore, the receive audio stream from the PSTN network could be a mixture of multiple speakers while the audio information from the IP network contains only the information of one specific speaker.

The synchronization algorithm may take the audio information of the specific speaker to find a correct match in the audio stream from PSTN network. If the multiple sources in the audio mixture are well-partitioned, namely, there is no overlap in time between different sources, then the synchronization still can perform well since the audio of the specific speaker is undistorted. However, in practical audio conference, speakers might strive for making statements while keen discussion. Consequently, it is very likely that multiple speeches may overlap in time. The waveform of the specific speaker might be distorted by the overlapping speeches and thus put a challenge to the audio synchronization algorithm.

3.3.4 Packet Loss in Wireless Connection

It is well-known that the unstableness of wireless connection may incur packet loss. Meanwhile, in real-time applications, packet loss might not only be due to the loss in transmission channel but also be in consequence of out-of-date packets. Therefore, packet loss is very common for audio streams via wireless connection. In [33], many algorithms that try to conceal the lost packets in the audio stream so as to recover the original audio.

For speech signals, due to the characteristic of speech waveforms, usually the concealing algorithm simply duplicates the previous received packet to fill in the lost packet. As long as the gap is small, this algorithm can achieve an acceptable quality to human hearing system. However, this packet duplication still distorts the original waveform. Therefore, the synchronization algorithm might be confused to the loss-concealed audio. Note that for the wireless connections in the path of PSTN audio stream, the packet loss are usually small as a result of the wide coverage of base station and the under-controlled PSTN network. Hence in the following discussion, we focus on the loss of wireless connection in IP audio.

3.3.5 Reactiveness to Network Dynamics

The synchronization may not require the capability of realtime process since the synchronization is not triggered all the time. However, if the synchronization module spends too much time on the computation, when the timing information is obtained, it might be stale and useless. The situation becomes even worse if the network environment is highly dynamic. Therefore, the required computation time for the synchronization module is also an important issue in designing the synchronization algorithm.

Regarding to Figure 10, when the synchronization module is triggered at $T_{trigger}$, the speech segments in the audio buffers (X(t) and X(t)+Y(t)) are fed to the synchronization module. The required computation time for synchronization module to obtain time shift τ is defined as the time period T_{sync} specified in the figure.

To evaluate on the reactiveness of an algorithm, we apply the tic and toc functions provided in MATLAB [34]. At the beginning of the synchronization module, tic

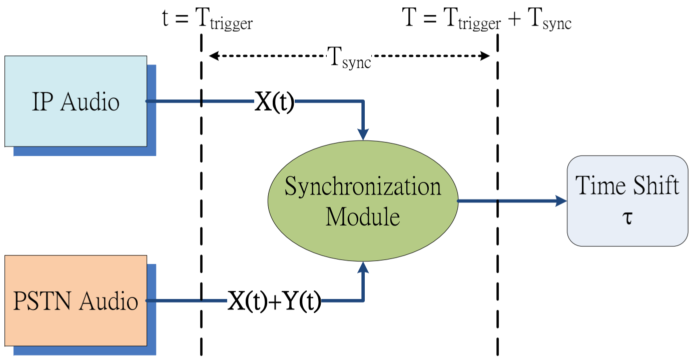

Figure 10: Synchronization computation time

function is set to start a stopwatch. Then after the synchronization process completes, toc is set to stop the stopwatch and then store the current elapsed time in toc, hence, from toc we can obtain the computation time T_{sync}. Nevertheless, since the computation time is processor dependent, different processor may results in different computation times. Therefore, the total flop count of the synchronization algorithm is also provided to show the relationship between flops and computation time.

In conclusion, we have *simplified the synchronization problem to simple audio synchronization.* While the receiver receives the two audio data, the synchronization algorithm should be able to locate the matching point in the audio so as to recover the timing information. Additionally, as described in the subsection above, the synchronization algorithm should also conquer the afore-mentioned challenge. Therefore, in the latter chapters, we focus on the discussion of possible means of synchronization algorithm and their pros and cons against the challenge.

CHAPTER 4

SYNCHRONIZATION BASED ON CROSS CORRELATION

In the previous chapter, we have concluded that the synchronization problem can be simplified to the synchronization of the PSTN audio stream to the appended audio information in the IP video stream. This audio information come along with the video stream could be a complete audio stream as in common video calls. Then intuitively, the synchronization algorithm can be simply comparing the waveform of these two audio streams and searching for a matching point where this two streams are synchronized.

Note that this audio stream arrived via IP network consists of only the speech of the specific speaker who generates the video stream, while the other audio stream from the PSTN network could be a mixture of multiple conferees. Therefore, aforementioned challenges may arise. In this chapter, we first examine on the effectiveness of using time domain features of audio waveform for synchronization against the challenges. Then we adopt the time domain cross-correlation to determine the similarity between these two audio streams. Larger correlation coefficient indicates higher similarity between the comparing audio segments, and thus suggests larger probability to be a correct match point.

4.1 Time-domain Audio Features

As previously suggested, extra audio stream could be added to the IP traffic flow. After these two audio streams are received, they are stored in the additional buffers inside the synchronization module. When the synchronization module is triggered, a segment of the IP audio stream is chosen to compare to the PSTN audio waveform. The size of the PSTN audio buffer restricts the search range to be compared. Every comparison chooses one segment of the audio waveform within the search range. The objective of synchronization is to locate the most similar segment of the PSTN audio to the IP audio segment. From this match we can determine the time shift between PSTN and IP audio streams. Thus PSTN audio and IP video streams could be synchronized, since the timing relationship between video and this IP audio stream can be reconstructed by conventional audio/video timestamps.

Although the waveform-comparing algorithm can use a simple square error of the waveform samples as metric, small distortion to the waveform might severely affects the effectiveness of this algorithm. Therefore, the synchronization algorithm should be able to recognize the important trends or patterns in the waveform so as to locate the time shifts. According to the conventional speech processing techniques, the commonly used time-domain speech features that are used to segregate auditory cues are the peak-to-peak period, pitch, and the envelope measurement, as suggested in [35].

Since the waveform is easily distorted by noises, to acquire the features, preprocessing stages, such as low-pass filtering and moving average, are usually applied. The peak-to-peak period and the pitch measurement are related to the determination of fundamental frequencies. Because of the quasi-periodicity of speech signals, in a small period of time the neighboring peaks might reveal the inverse of the frequency components. If the receiving speech is clear, peak-to-peak periods can usually serve as the metric to differentiate different auditory cues. However, the performance is seriously corrupted by the noises and interferences.

On the other hand, pitch is a relatively more reliable feature in obtaining frequency information. The commonly adopted method of pitch computation is the autocorrelation function. Many variations of autocorrelation is develop and adopted in the field of pitch extraction, such as [36]. Within a certain range, a windowed speech segment is chosen to compute the pitch information in that speech segment. Due to the quasi-periodicity of speech signals, the neighboring peaks in the autocorrelation result reveals the fundamental frequency of the selected segment. If the speech signal contains only single speech, the pitch detection technique can usually achieve proper accuracy for auditory cue segregation. Even for speech mixtures, the multi-pitch detection technique can still obtain fair performance, as long as the interference is of different fundamental frequency or small.

The envelope is determined as a short-time moving average of the signal energy, realized by low-pass FIR filtering of the squared signal. The filter order is chosen as a compromise between envelope smoothing and ability to follow fast energy changes on the boundaries of voiced/unvoiced parts of the speech signal. The shape of envelop represents how the speech segment is packaged, and thus can be used for speech segregation. For clean speech, the voiced/unvoiced parts and different speech segments can be distinguished from the envelop. Since the high frequency noises are filtered out by the low-pass FIR, energy envelop should be resistant to noises. However, if the speech is mixed with another speech, the envelop might reveal the shape of speech combination, and therefore is different from the original clean envelop.

Although the afore-mentioned time domain features may perform well in specific research fields, the performance might be affected in certain circumstances. However, as suggested in the pitch detection research field, the autocorrelation function seems to be robust to minor distortion of the original signal. Hence, we consider that the correlation function might be helpful in comparing the similarity between speech segments.

4.2 Cross-Correlation-Based Synchronization

Inspired by the research area of Correlation Pattern Recognition (CPR), we choose cross correlation in the discussion of time-domain synchronization because it is considered robust and general in the field of pattern recognition, whose main goal is to assign an observation into one of multiple choices, as described in [37].

4.2.1 Basics of Cross Correlation

Cross correlation which is widely adopted in many area tries to capture how similar or different a test object is from the specific object. The commonly used quantity of measuring cross-correlation similarity is the correlation coefficient, usually noted as r, which is defined as

$$r = \frac{\sum_i^N (x(i) - m_x)(y(i) - m_y)}{\sqrt{\sum_i^N (x(i) - mx)^2 \times \sum_i^N (y(i) - m_y)^2}}, \tag{4.1}$$

where $x(i)$ and $y(i)$ are the comparing objects and m_x and m_y are the mean of them.

The definition of correlation coefficient r shows that at sample i if $x(i)$ and $y(i)$ deviate from their own mean by a similar amount, the normalized product of their differences to means may results in a value near 1. The overall correlation coefficient is similar to the mean of the normalized products. This property implies that the correlation coefficient can faithfully represent the similarity of two signals. Therefore, If two comparing speech signals have similar waveforms, the correlation coefficient may acquire a value approaches 1. This characteristic of correlation coefficient could be applied as a metric in determining the similarity of two speech signals.

4.2.2 Cross-Correlation Synchronization Module

Based on the cross-correlation function, we design a synchronization module as illustrated in Figure 11. When audio signals are received from either IP network or

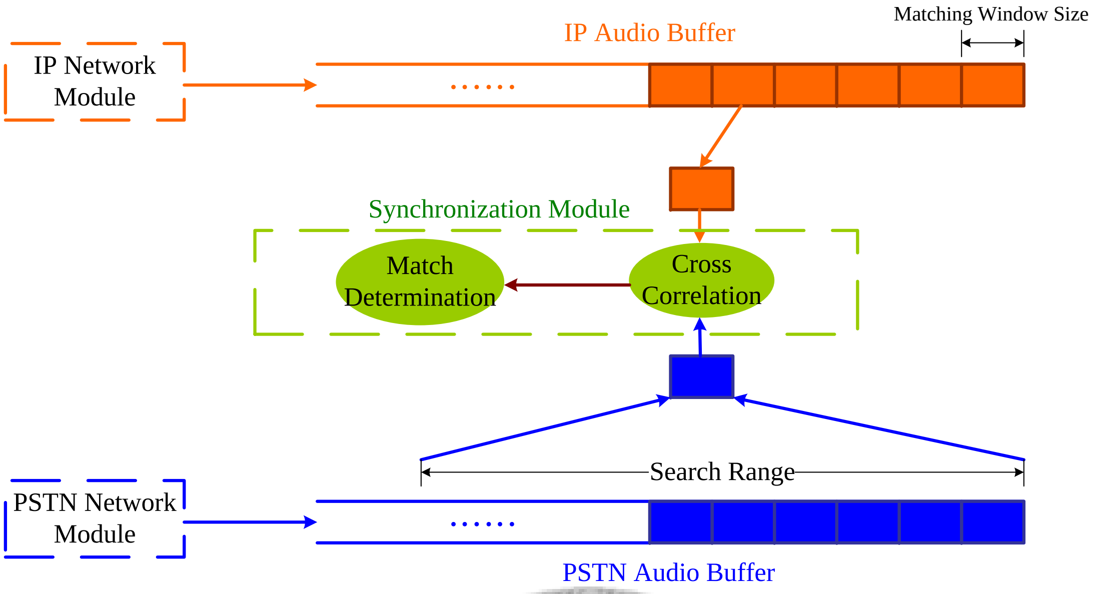

Figure 11: Synchronization module using cross correlation

PSTN network, they are first fed into respective audio buffers. When the synchronization process begins, a matching window of specific size is chosen from the IP audio buffer and then sent to the synchronization module. Accordingly, the synchronization module will iteratively select a matching window from the PSTN audio buffer, from the beginning toward the end of search range, to cross-correlate with the matching window from the IP audio buffer. The correlation coefficient obtained for these two windowed segments and the position of the PSTN matching window are recorded.

After each iteration, the matching window in the PSTN audio buffer shifts by a certain search step while the matching window in the IP audio buffer remains still, until the end of the search range. In the end iteration, the matching window with highest correlation coefficient is found at the match determination stage, and then is referred as the matched window. Thus the synchronization point is accordingly set.

4.3 Design Issues

Note that in the synchronization module, the matching window size and the search step size are not yet determined. However, different size settings may affect on the accuracy of similarity determination against waveform distortions. Therefore, in the following parts, the effect of parameter settings if examined.

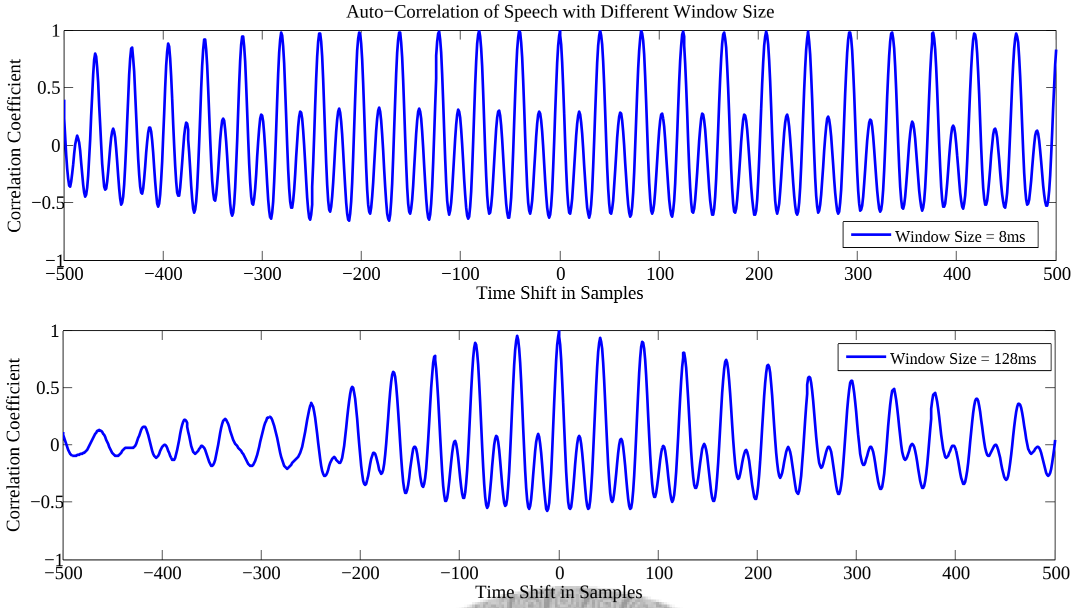

Figure 12: Auto-correlation with different matching window sizes

4.3.1 Matching Window Size

Practically, while synchronizing two audio inputs, we can only have limit length of signal for cross correlation, regarded as the matching window. Signal of length of the matching window from the IP audio buffer is cross-correlated with the matching windows within the search range of the PSTN audio buffer.

The first issue arises in that *the length of matching window may affect the accuracy of synchronization judgement.* Figure 12 shows the autocorrelation of a speech signal with window sizes of 8ms (64 samples) and 128ms (1024 samples). Since speech signals are considered quasi-periodic, the neighboring waveforms may seem similar to each other, as shown in Figure 13. Therefore, the resulting correlation coefficient might achieve peaks at the quasi-periods. This characteristic results in the high correlation coefficients around the zero-shift point.

The situation becomes even worse when the matching window size is small since the containing signal information is less. Therefore the correlation coefficient still remains at a high level at large shifts– over 400 samples (about 50ms) with 8ms window size for example. When the original speech is unclean, noisy or interfered, the correlation coefficient at the zero-shift point may easily be diminished to a level lower than other peaks around the zero point, and thus resulting in wrong synchronization.

Therefore, in order to acquire distinguishable peak at the zero shift and suppress

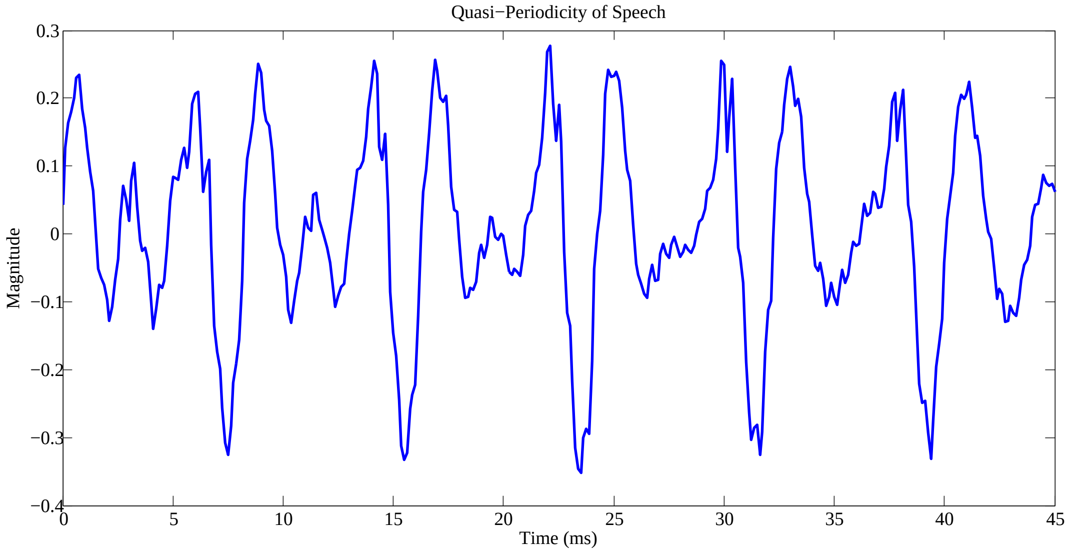

Figure 13: Quasi-periodicity of speech

the coefficient at neighboring shift so as to achieve better robustness against waveform distortions, using a larger matching window seems to be one solution to the above issue.

4.3.2 Search Step Size

Another parameter to consider is the search step. The search step represents the number of samples that the matching window step shifts at each iteration. Since the search step determines the possible positions of the matching window, different search step implies that different part of the speech within the search range is chosen for cross correlation. This implies that the the matching window might skip the correct matched window while searching within the search range. This phenomenon is referred as the matching window misalignment. As illustrated in Figure 14, the matching window starts at different positions of the PSTN audio buffer in each iteration, according to the search step. It is possible that the matching windows might not start at the same position as the correct matched window. For example, in Figure 14, the correct matched window lies within the third and the forth matching windows, instead of exactly the third or the forth matching window.

However, if the correlation coefficient obtained at the neighboring matching window, which covers the correct matched window, are still larger than the windows farther from the correct match, the determined synchronization point might be shifted from the correct point by an error bounded by he window size. Nevertheless, the

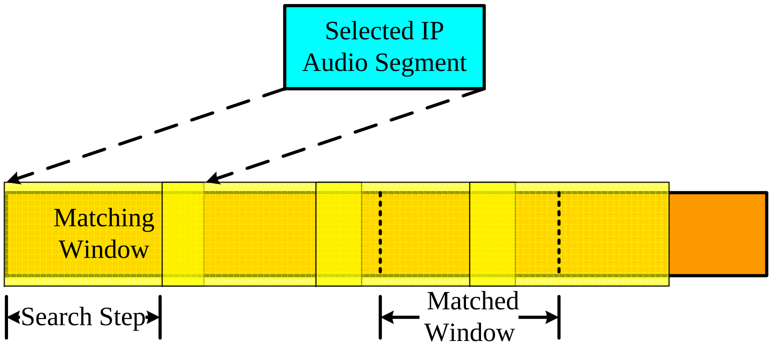

Figure 14: Concept of matching window misalignment

autocorrelation of speech shown in Figure 15 suggests that the correlation coefficient drops steeply when the time shift is only a few samples away from the zero point. Hence, choosing the segment with largest coefficient as the matched window may result in wrong judgment.

In order to overcome the matching window misalignment problem, the search step should be as small as possible. The safest search step to ensure that the correct matching window will be checked is of course the 1-sample step size which implies that every possible matching window within the search range is examined.

4.3.3 Short Conclusion

To sum up the discussion on design issues of using cross correlation as the synchronization algorithm, we conduct an experiment on the performance of synchronization using cross correlation with different matching window sizes and search steps. A speech is cross-correlated with its decoded version after G.729 codec which includes a time shift of 60 samples. The result is shown in Figure 16.

In Figure 16, we determine the accuracy of the algorithm to allow a ±10ms (80 samples) error range. It is shown that even larger window size can guarantee larger accuracy, however, for search step larger than 4 samples the accuracy may saturate and couldn't reach 100%, regardless of the window size. The reason to this inaccuracy is because the search steps larger than 4 samples we used are not factors of 60. Hence, the matching windows within the search range can never be aligned to the matching window to be compared. However, for search steps of 8 and 16 samples, since they are small relative to 60, they might have a larger probability to locate match points within the error range. Meanwhile, for step size larger than 16 samples, the accuracy is too low to be acceptable.

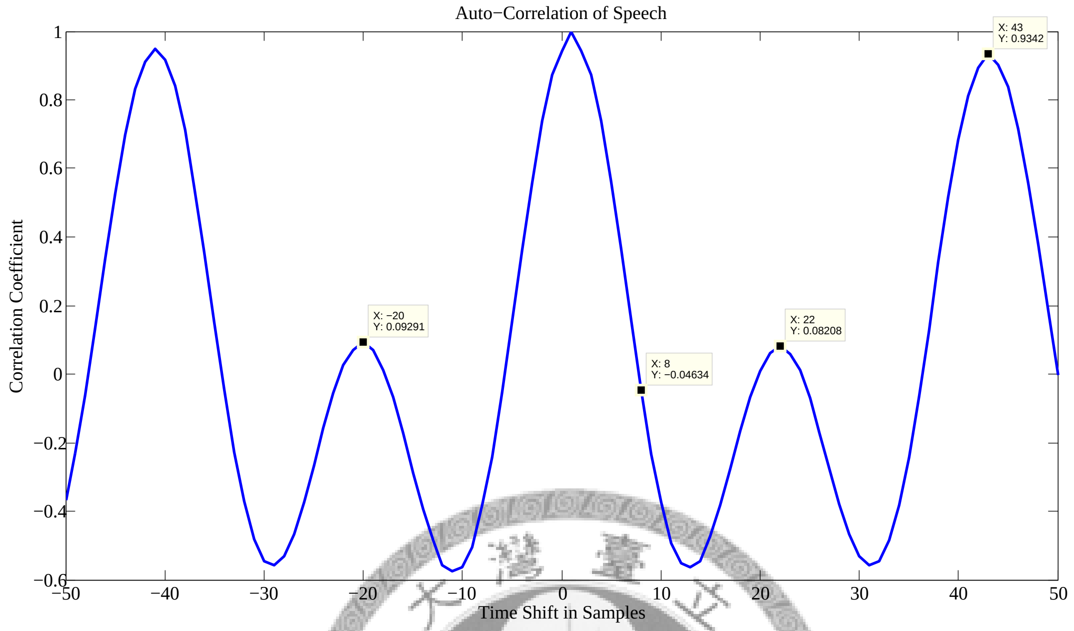

Figure 15: Auto-correlation of speech

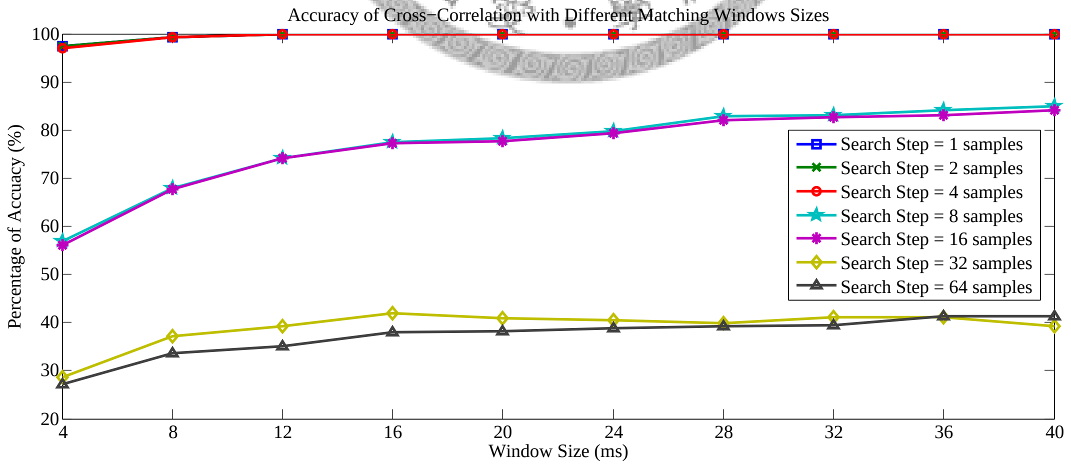

Figure 16: Accuracy of cross correlation with different settings

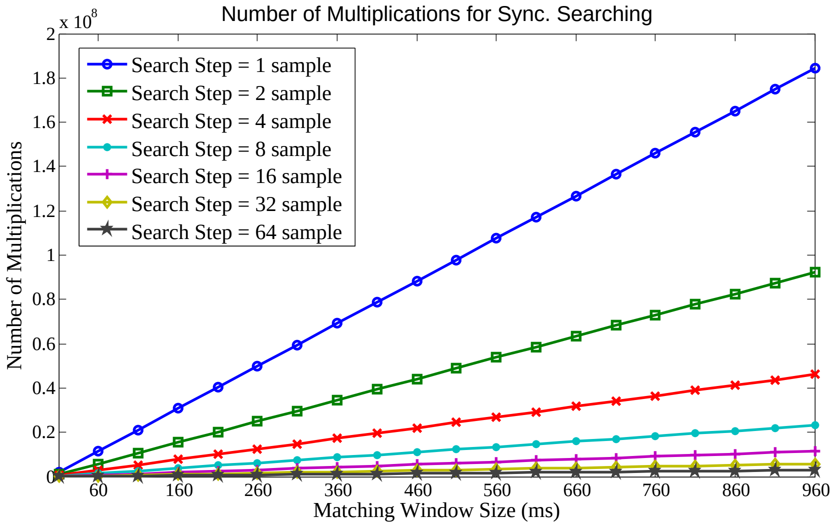

Figure 17: Number of multiplications required for synchronization

The result in Figure 16 suggests that larger window size and smaller step size can ensure better accuracy of the synchronization algorithm. However, larger window size implies more samples to be computed in the cross correlation while smaller step size implies more matching windows to be cross-correlated. The complexity might be an issue for the devices with limited computation power. Hence, we try to analyze the possible complexity for different window sizes and search steps.

For a search range of T samples, the counts of cross correlation to be computed C_{XCOR} with search step S can be determined as $C_{XCOR} = \lfloor \frac{T}{S} \rfloor + 1$. For each computation of correlation coefficient, according to Equation 4.1, the number of multiplications contains the multiplication in the numerator, the two squares in the denominator, and the multiplication of the two summations in the denominator. Besides, although the square root requires more computation, we treat it as a multiplication here for simplicity. Therefore, the amount of multiplications in each correlation is about $3N + 2$, where N is the number of samples in the matching window. The total number of multiplications for the complete synchronization point searching is

$$N_{Multiply} = (3N + 2) \times C_{XCOR} = (3N + 2) \times (\lfloor \frac{T}{S} \rfloor + 1). \tag{4.2}$$

Assuming that the search range is 1 second, the required number of multiplications for synchronization for different window sizes and search steps is shown in Figure 17. From Figure 17 we can observe that if a search step of 1 sample is chosen, the required multiplications grows fast as the window size increases. Although the 1-sample search step can ensure the correct matched window to be searched, a larger matching window

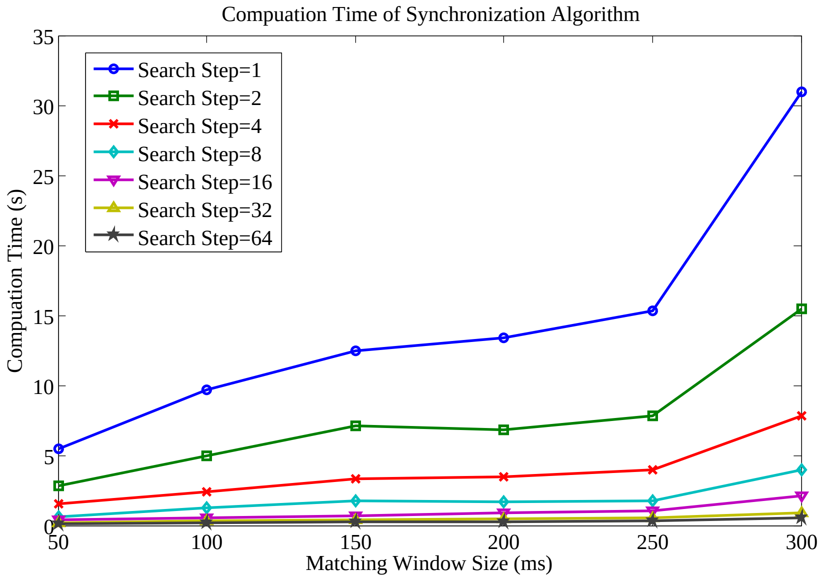

Figure 18: Computation time for synchronization using cross correlation

is essential for suppressing neighboring coefficient peaks as afore-mentioned. Hence, the number of multiplication in the synchronization algorithm might reach the order of 10^8.

Different CPU implementation might spend different CPU cycles for multiplication. Assuming 10 CPU cycles is required for one floating-point multiplication, for a 500 MHz CPU, the synchronization might spend a few seconds to learn the synchronization point, and thus making the synchronization module less reactive to network dynamics. To acquire a clue of how much the computation time is required, we use MATLAB to implement the cross-correlation-based synchronization algorithm and measure the required computation time by the tic and toc function. The synchronization process is performed on an laptop with single-core 1.73GHz CPU and 1GB RAM while the search range is set to 1 second.

The computation time is shown in Fig 18, along with the flop count in Figure 19. From Figure 18, we can observe that the required computation time for synchronization may grow to tens of seconds if the search step is small and matching window is large. The flop count in Figure 19 shows the same tendency. Note that the flop count analysis results in a similar order of numbers as in Figure 17 where only multiplications are considered. Therefore, while choosing the search step and matching window size, this issue should also be considered.

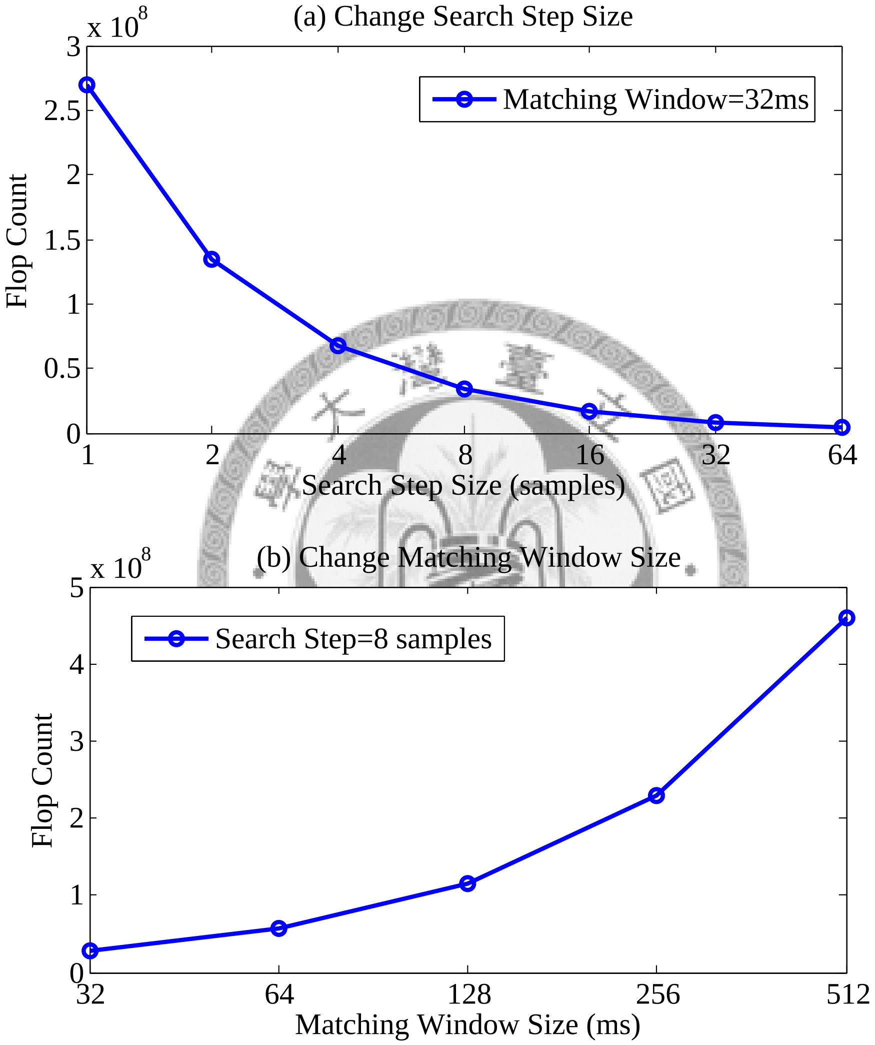

Figure 19: Flop count of cross correlation synchronization

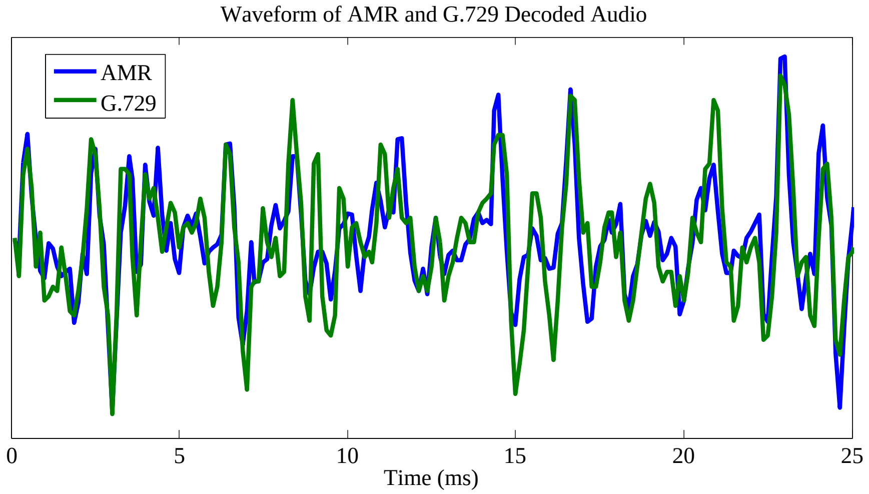

Figure 20: Decoded waveforms of speech signals

4.4 Performance Evaluation

In the previous chapter, the possible challenges for audio synchronization are elaborated. Whether cross correlation can overcome those challenges is discussed in the following parts.

4.4.1 Codec Distortion

To simulate the result of real audio signals, the audio signal should be encoded and decoded using common codec applied to 2G/3G phone or VoIP. For the traditional telephony system, Adaptive Multi-Rate (AMR) compression [38] is usually applied to compress the audio signal, while for the VoIP system, among the various voice codecs G.729 codec [39] is chosen for simulation here. Part of the decoded waveforms is shown in Figure 20. We can observe that although these two waveforms are different in temporal structure, they have similar variations in time. This is because different codec may neglect different time redundancies while the frequency characteristics are preserved.

Since the decoded waveforms have similar variations, the correlation should still maintain at high level. Figure 21 shows the correlation coefficient with different window sizes. Since the short-time temporal structure is modified by the codec, if a too small window is chosen the coefficient might have a low peak at the 0 shift, and thus is easy to be affected by other distortions. However, with larger window size, the correlation has a sharp peak at the 0 shift and quickly descends as expected.

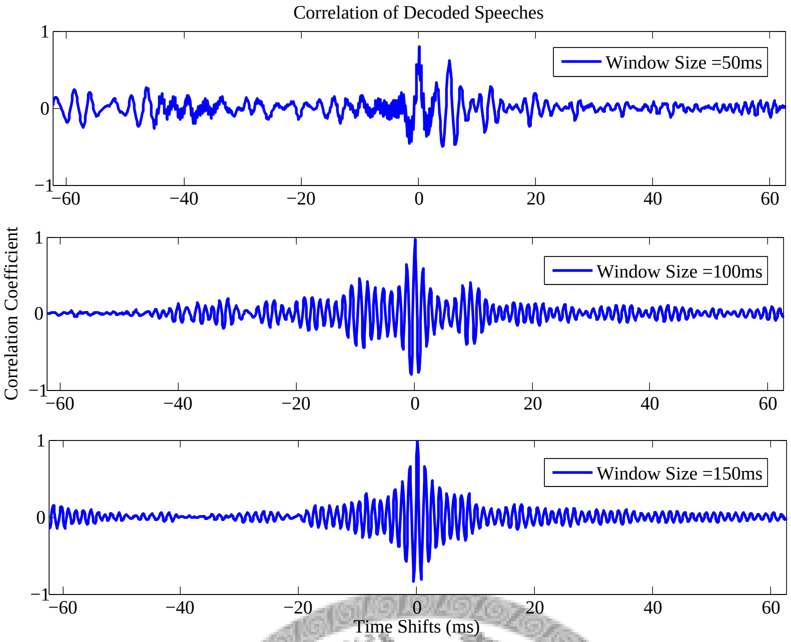

Figure 21: Correlation coefficients of decoded waveforms

4.4.2 Noise Distortion

Since the thermal noise is the most common source of noise, we focus our discussion on the distortion by thermal noise, which is usually modeled as an Additive White Gaussian Noise (AWGN). The variance (σ^2) of the AWGN determines the energy level of noise. By adding the AWGN with different energy level (σ^2) to the source speech, the effect on correlation (r) is shown in Figure 22. The upper two graph shows the waveform and correlation with ($\sigma^2 = 0.0001$). The correlation seems not to be affected by this small noise. However, as the noise energy increases, the correlation coefficient drops. Note that a noise with 0.01 energy level is almost the energy of lower volume parts in the source speech. Therefore, the effect of noise at this level is comprehensible.

Since the coefficient at the 0 shift is lowered by the noise, the accuracy of correlation might also be affected. Figure 58 shows the accuracy of synchronization when one of the source speech is noisy. Both the source speeches are encoded and decoded respectively according to previous discussion. By allowing an error range of $\pm 50ms$, we can observe that the accuracy of synchronization is lowered by the noise. Even though a larger window size is chosen, the accuracy is still limited to lower than 100%.

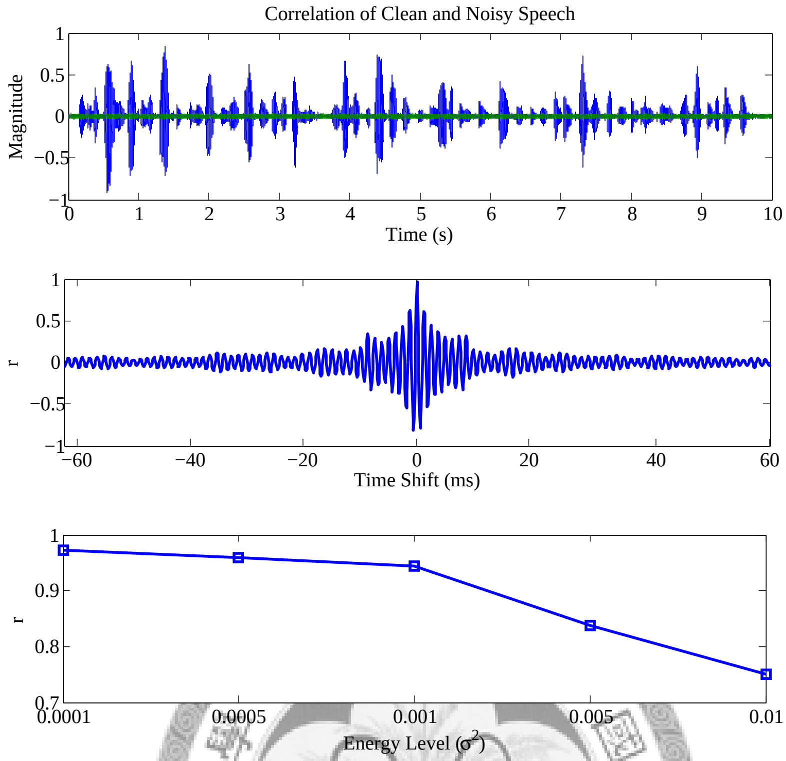

Figure 22: Effect of noise on correlation

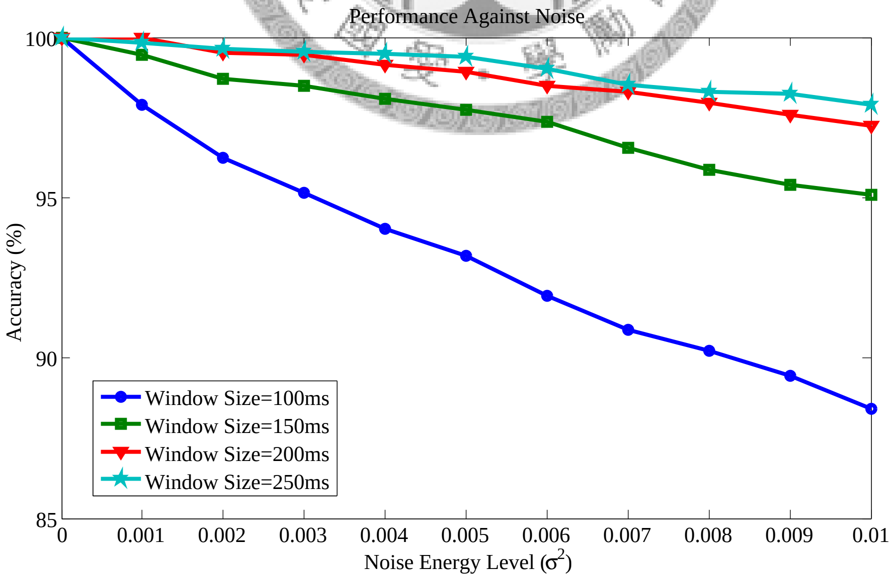

Figure 23: Performance of correlation against noisy signal

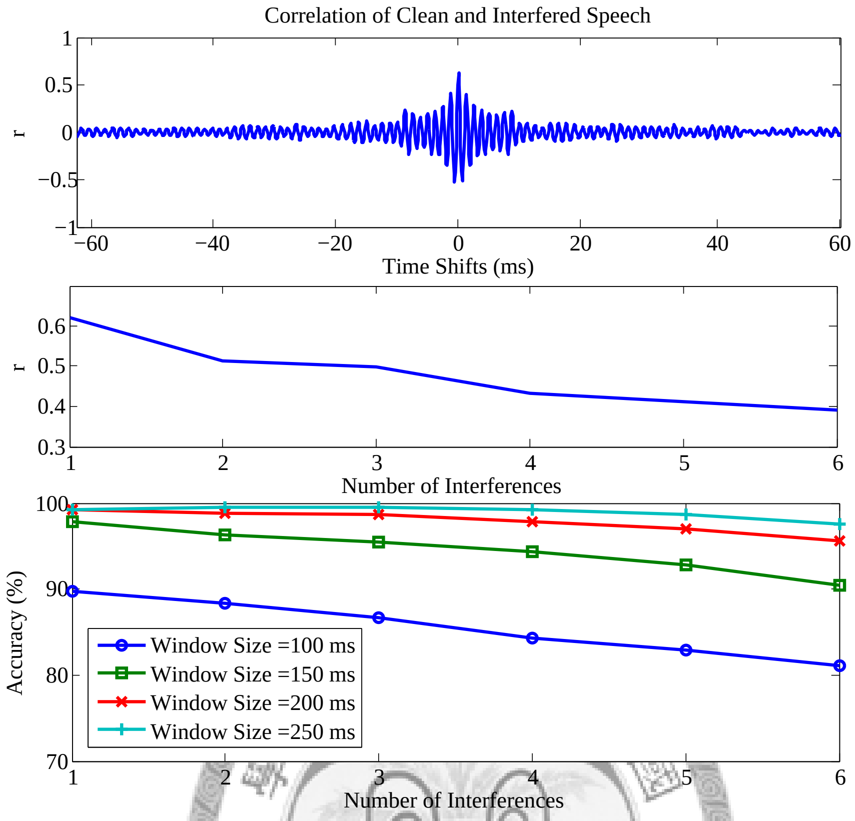

Figure 24: Performance of correlation against interferences

4.4.3 Overlapping Speakers

Another issue is that during an audio conference, there may be times when multiple speakers speak at the same time. Hence, the receiver may receive a mixed speech combining multiple speeches from the PSTN auido stream. In order to simulate this situation, the comparing AMR-decoded speech is mixed with other interfering speeches. Since the source waveform is severely distorted by the interference which has similar energy level, the correlation coefficient at the 0 shift point is apparently lower. However, unlike the AWGN noise adds noises at the same level on the overall speech, different speeches might not always maintain at the same high energy level. Therefore, the effect of interference might not be as severe as noise.

Figure 59 shows the effect of interference on correlation. From the upper two graphs, we can observe that the correlation coefficient is substantially lowered at the 0 shift point. This suggests that in this situation, the cross correlation may be vulnerable to other distortions. However, since the correlation value is also suppressed at other shifts, the performance while a 250ms window is applied remains high. But

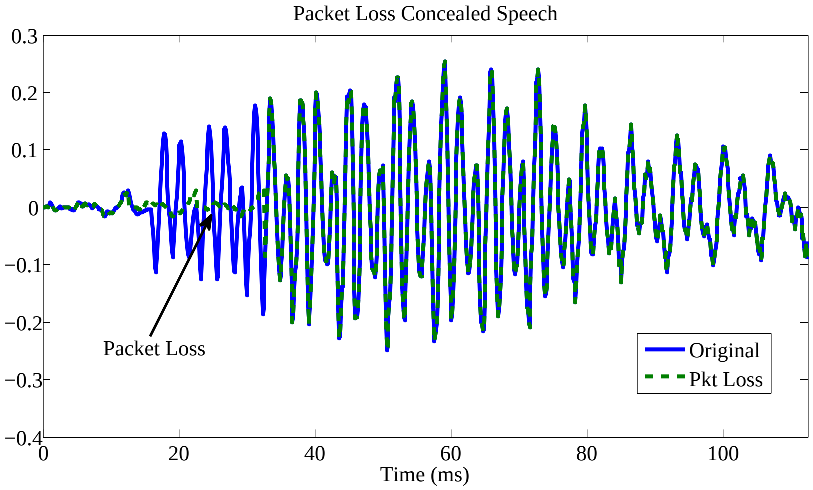

Figure 25: Packet-loss-concealed speech waveform

for small window size as 100 ms, since the coefficient at the 0 shift is lower than that of larger window sizes, it is more vulnerable to interference, and thus has lower accuracy than others.

4.4.4 Packet Loss

In practical application, the audio signal from the IP network may be severely affected by the channel condition. High error rate of the wireless connection and network congestion may cause packet loss at the receiver side. Although delicate packet loss concealment algorithms, according to [33], might be applied to compensate the effect of lost packets, most of the current applications use the simplest way which simply duplicates the packet in front of the lost packet. As shown in Figure 25, several packets, according to the loss rate, of the original G.729-decoded speech is lost and concealed by the duplication of the previous packet. The packet size is specified to contain 10ms speech. If more than three consecutive packets are lost, then the gap is filled with zeros, as suggested in [33].

Graph (a) in Figure 60 shows the performance against packet loss. Because of the duplication of the previous packet, the quasi-periodicity makes the coefficient less affected by the packet loss. Therefore, the performance can remain higher than 90% even though more than half of the packets are lost. However, the level of correlation is still decreased by the lost packets, and thus makes it vulnerable to other sources of distortions. Graph (b) to (d) simulate the situation that the IP audio is distorted by packet loss while PSTN audio suffers from other interferences. The accuracy is

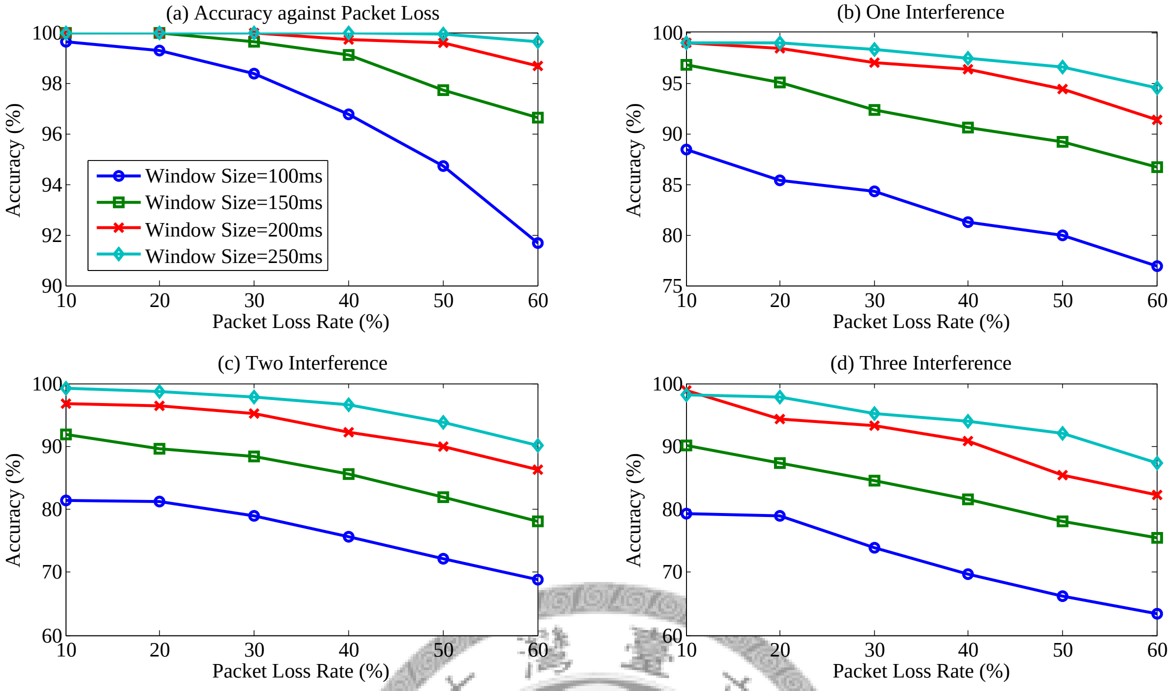

Figure 26: Performance of correlation against packet loss

severely affected by the additional interferences in the PSTN audio, even when the window size is set to 250 ms.

In consideration of practical situation, Figure 62 shows the performance of cross correlation when multiple sources of distortions occurs. We can observe that the performance is severely degraded by these extra distortions since cross correlation bases only on time-domain signal. Even though the distortion level is low, the performance is largely affected. Additionally, although increasing the matching window size can slightly improve the accuracy, the performance is limited.

4.4.5 Short Conclusion on Performance

From the above evaluations, we can observe that cross correlation can usually sustain minor distortion on the waveform. But *when the distortion level increases or multiple distortions are included, the performance rapidly drops.* Even large matching window size can't efficiently improve the performance. Additionally, to ensure high accuracy of the synchronization algorithm, the cost of computation time, as shown in Figure 18, may increase to tens of seconds. If the network is so stable that the synchronization algorithm is not frequently triggered, the computation time for cross correlation is not so important. Therefore, the search step and window size can be

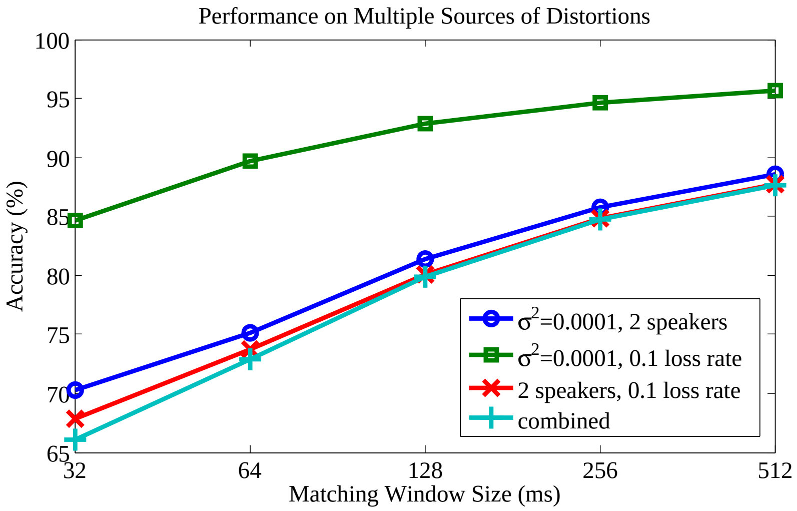

Figure 27: Performance of correlation against multiple sources of distortion

choose to the required value to ensure accuracy. However, if the network environment is highly dynamic, the synchronization algorithm should be able to respond to the varying network characteristics. Hence, then cross correlation may not be reactive to this network dynamic. This has lead to *a trade-off between high accuracy and high reactiveness to network dynamics.*

In conclusion, cross correlation is vulnerable to practical distortions because it only considers the time-domain signal. The performance is limited. The main reason that makes cross correlation time-demanding is that short time waveform is easily corrupted by distortions, and thus large windows should be applied. On the other hand, other time domain algorithms seem not as robust as the cross correlation which can directly respond to the similarity of waveforms. Therefore, in order to make the synchronization algorithm robust and less sensitive to short-time distortions, audio features in other domains, which can best characterize the behavior of speeches within a certain duration, might need to be considered.

CHAPTER 5

SYNCHRONIZATION BASED ON MFCC

In the previous section, we have concluded that using cross correlation for the design of synchronization algorithm may be vulnerable to practical speech distortions. When multiple sources of distortions are included in the speeches, the performance is limited. Considering the correlation formula, the reason might be rooted in the comparison of each sample which might already be distorted and thus requiring more samples to extract the trend of waveform variation. Hence, if a representation of speech can extract several essential characteristics which might not easily be distorted, then potentially it should be robust to distorting sources.

Therefore, using other speech representations which might transform the original speech into other domains is the focus of this chapter. This has lead to the field of Digital Speech Processing (DSP) techniques. In this chapter, we adopt the commonly used audio feature in speech recognition as the representation of speech for similarity comparison. The robustness of this audio feature is evaluated in a way similar to the previous chapter.

5.1 MFCC-Based Synchronization

In the synchronization module, one speech segment from the IP audio buffer is used to search for a correct match in the PSTN audio buffer. The representation should be able to recognize the correct speech segment among the search range. This concept is analogous to the research of speech recognition which segments the speech signal and search in the database for a match to this segment. As suggested by any speech recognition research endeavors, Mel-Frequency Cepstral Coefficients (MFCC) is the most widely used representation of speech signals in that it generally can obtain better accuracy at relatively low computational complexity. Therefore, whether MFCC is good enough as the representation for synchronization comparison is first discussed.

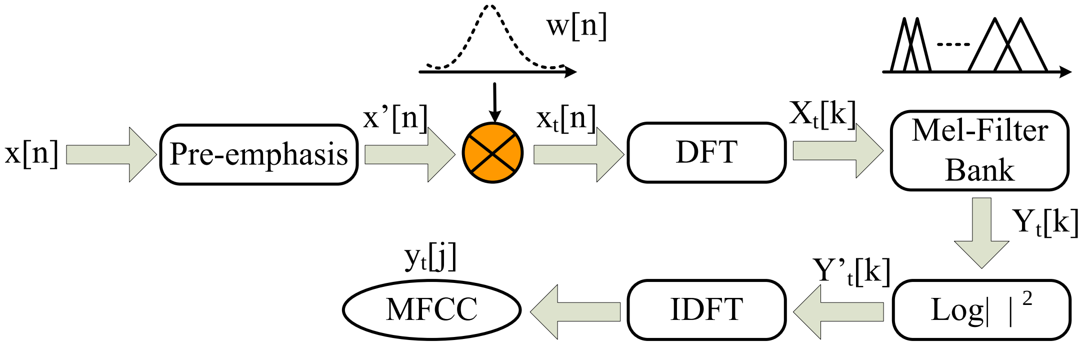

Figure 28: Block diagram for MFCC acquisition

5.1.1 Basics of MFCC

The process of acquiring the MFCC from a certain speech is illustrated in Figure 28. The block diagram mainly involves six stages which are introduced herein in order.

Pre-emphasis For a speech signal $x[n] = u[n] * g[n]$, where $u[n]$ represents lung excitation and $g[n]$ represents vocal tract response, the first stage into MFCC is to be pre-emphasized. Due to the physiological characteristics of the speech production system, high frequency components are attenuated while speaking. High frequency formants may have small amplitude with respect to low frequency formants. However, human hearing system is sensitive above the 1kHz region of spectrum. Pre-emphasis performs as an high pass filter which can emphasize the high frequency part to compensate the attenuation from speech production.

$$x[n] \Rightarrow H[z] = 1 - az^{-1} \Rightarrow x'[n] = x[n] - ax[n-1] \tag{5.1}$$

Windowing After pre-emphasis, $x'[n]$ is then fed to a windowing function $w[n]$ to obtain successive and overlapping frames $x_t[n]$. Windowing is needed because theoretically, spectral evaluation approaches are in general for stationary signals which only holds within short time intervals for voice signals. This is the so-called short-time stationary. Usually each frame length ranges between 10 20ms. Frame shift determines the length of time between successive frames. The most widely used window shape is the Hamming window for its narrow main lobe and low side lobes.

$$w[n] = \begin{cases} 0.54 - 0.46cos\frac{2\pi n}{L-1}, & \text{n=0,1,......,L-1;} \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

DFT and Mel-filter-bank Processing In this stage, spectrum $X_t[k]$ is obtained by feeding $x_t[n]$ to an L-point Discrete Fourier Transform. After that, $X_t[k]$ is then sent to a M-filter Mel-filter-bank. Each filter with different central frequency may filter the input according to the frequency and use different triangular function to get the weighted sum of filtered spectral components $Y_t[m]$, where $m = 0, 1, ..., M - 1$. The filter-bank processing simulates human auditory system which has high resolution at lower frequencies and the awareness of pitch is proportional to the logarithm of frequencies. Within the bandwidth of each filter-bank, human perception can't identify the differences in frequency. This bandwidth is referred to as the critical band.

$$X_t[k] = DFT\{x_t[n]\}, \tag{5.3}$$

$$Y_t[m] = \sum_{k=f_i}^{f_{i+1}} w_i \bullet X_t[n], \tag{5.4}$$

where w_i is the weight of the triangular weighting function.

Logarithmic Operation Since phase information is not important for human perception but signal energy is, the squared absolute value of $Y_t[m]$ is used. And then because logarithm can compress the dynamic range of values like human hearing system and make a convolved noise additive, logarithm is operated on the output of squared absolute value of $Y_t[m]$.

$$Y_t'[m] = Log(|Y_t[m]|^2) \tag{5.5}$$

IDFT After the operation of logarithm, the lung excitation $u[n]$ and vocal tract response $g[n]$ are now added together in log-spectral domain.

$$log|X[k]| = log|U[k]| + log|G[k]| \tag{5.6}$$

Since the log-power spectrum is real and symmetric, inverse DFT reduces to a Discrete Cosine Transform (DCT) which produces highly uncorrelated features y_t. The components of excitation is now separated from vocal tract response since excitation changes much faster than vocal tract. The feature extracted from vocal tract response is easily separated by choosing only the first J components which is usually set as 13. Each of these 13 components is usually referred to as the MFCC bin.

$$y_t[j] = \sum_{m=0}^{M-1} Y_t'[m] cos[j(m - \frac{1}{2})\frac{\pi}{M}], j = 0, 1, ..., J - 1 < M \tag{5.7}$$

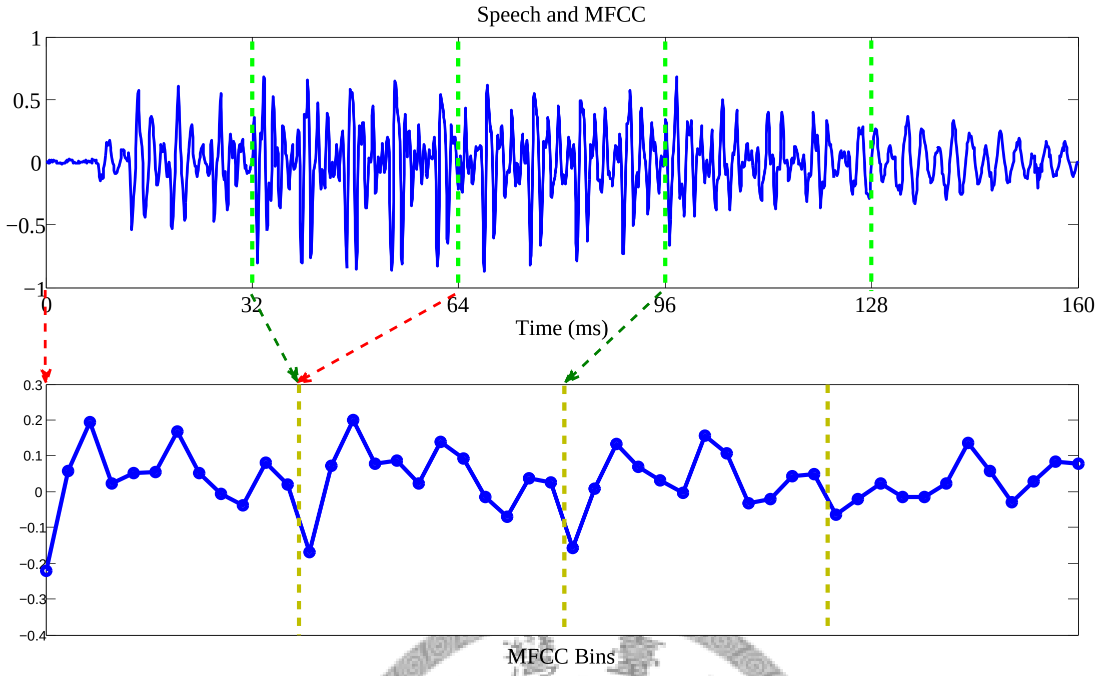

Figure 29: MFCC of a period of speech

In conclusion of the process for MFCC, the main objective is to extract the human vocal tract response which is directly responsible for human perception. On the other hand, since the lung excitation is more like an impulse to the vocal tract, the exact excitation information is not necessarily to be preserved. Note that due to the DCT operation, the slower changing characteristics of vocal tract is kept at smaller MFCC bins.

Figure 29 shows the corresponding MFCC value of a certain speech. The value is normalized to the first bin which represents only the signal energy and thus is omitted in the algorithm. Since the analysis window is half-overlapped in length, the resolution of MFCC columns is only half of the analysis window. From this figure we can observe that the MFCC value is usually within ±0.2. When the time-domain signal looks similar, the resulting MFCC values are also similar. Based on this characteristics of MFCC, next we will discuss about the synchronization algorithm.

5.2 Synchronization Algorithm Design

In this section, we discuss the design of synchronization algorithm for the synchronization module. Inspired by [6], we slightly change the similarity metric in the [6] and use this metric for searching of synchronization point. Generally speaking, this similarity metric simply use the sum of absolute differences of the MFCC bins from different audio sources while a large difference is penalized by adding additional penalty value to this metric. Details of the synchronization algorithm is shown in the following part.

5.2.1 Mathematical Analysis

Assume $x_u[n]$ as the unmixed speech signal from the IP audio buffer, $x_c[n]$ as the speech signal from other conferees, $x_n[n]$ as noise, and $x_m[n] = x_u[n] + x_c[n] + x_n[n]$ as the received mixed speech from the PSTN audio buffer. Since the first three steps of MFCC process are linear, $x_m[n]$ processed after the third step is $Y_m[m] = Y_u[m] + Y_c[m] + Y_n[m]$, where $Y_u[m]$, $Y_c[m]$, and $Y_n[m]$ are the outputs of the third step, obtained by individually feeding $x_u[n]$, $x_c[n]$, and $x_n[n]$ to MFCC.

Thus, the MFCC value of $x_u[n]$ and $x_m[n]$ can be represented as the following equations:

$$\begin{aligned} y_u[j] &= \sum_{m=0}^{M-1} log[|Y_u[m]|^2]cos[j(m-\frac{1}{2})\frac{\pi}{M}] \\ &= \sum_{m=0}^{M-1} 2log[|Y_u[m]|]cos[j(m-\frac{1}{2})\frac{\pi}{M}] \end{aligned} \tag{5.8}$$

$$\begin{aligned} y_m[j] = \sum_{m=0}^{M-1} & 2log[|Y_u[m]+Y_c[m]+Y_n[m]|]cos[j(m-\frac{1}{2})\frac{\pi}{M}], \\ & j = 0, 1, ..., J-1 < M \end{aligned} \tag{5.9}$$

Subtract Equation 5.9 by Equation 5.8 may obtain

$$\begin{aligned} y_m[j] - y_u[j] &= \sum_{m=0}^{M-1} 2log[\frac{|Y_u[m]+Y_c[m]+Y_n[m]|}{|Y_u[m]|}]cos[j(m-\frac{1}{2})\frac{\pi}{M}] \\ &= \sum_{m=0}^{M-1} 2log[|1+\frac{Y_c[m]}{Y_u[m]}+\frac{Y_n[m]}{Y_u[m]}|]cos[j(m-\frac{1}{2})\frac{\pi}{M}] \\ &<= \sum_{m=0}^{M-1} 2log[1+|\frac{Y_c[m]}{Y_u[m]}|+|\frac{Y_n[m]}{Y_u[m]}|]cos[j(m-\frac{1}{2})\frac{\pi}{M}] \end{aligned} \tag{5.10}$$

In Equation 5.10, it reveals that $y_m[j] - y_u[j]$ is restrained by $|\frac{Y_c[m]}{Y_u[m]}|$ and $|\frac{Y_n[m]}{Y_u[m]}|$ which are like the inverse of "Mel-SIR and Mel-SNR". The higher Mel-SIR and Mel-SNR, the smaller value of $y_m[j] - y_u[j]$. If the inverses of Mel-SIR and Mel-SNR are small enough with respect to 1, the difference value will approaches 0.

From the above derivations, if the received mixed speech signal is in synchrony with the unmixed speech, the difference of the mixed MFCC of PSTN audio and the

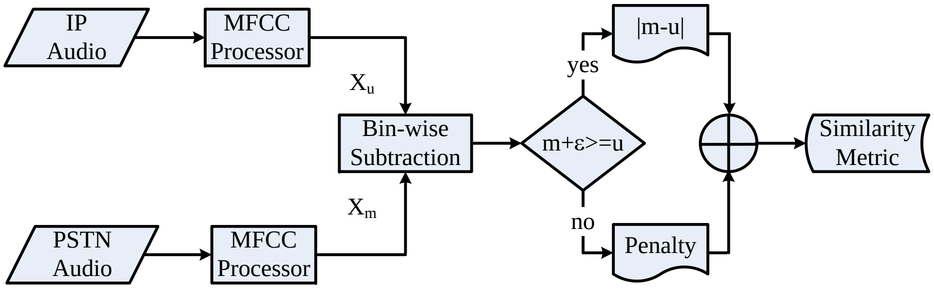

Figure 30: MFCC-based synchronization algorithm flow

unmixed MFCC of IP audio should be sufficiently small. Additionally, in order to further differentiate the similarity metric of the correct match from other shifts, a penalty value is assigned if the difference is so large that this matching window is very unlikely to be the correct window. The author of [6] claims that since the mixed MFCC contains the unmixed MFCC, the mixed MFCC should usually be larger than the unmixed MFCC. However, according to Equation 5.10, if $\frac{Y_c[m]}{Y_u[m]}$ or $\frac{Y_n[m]}{Y_u[m]}$ yield negative values, it's possible for the mixed MFCC to get a smaller value than the unmixed MFCC. Therefore, to compensate for this characteristic, the author of [6] suggests to include an error factor to ensure accurate judgement.

5.2.2 Similarity Metric

In order to determine the similarity of received speeches, for each MFCC bin in the MFCC of speech segment, the absolute difference of each mixed and unmixed MFCC bin (m and u) is calculated. As described previously, the similarity metric should includes a penalty value to further differentiate the correct match from other shifts so as to withstand distortions. Besides, an error factor which represent the fault tolerance is included to decrease possible error penalties.

The similarity metric is derived as follows:

$$B(m,u) = \begin{cases} |m-u|, & \text{if} \quad m+\epsilon >= u; \\ p, & \text{otherwise}, \end{cases} \tag{5.11}$$

where ϵ is the error factor, and p is a penalty value. For each bin in the MFCC, $B(m,u)$ is computed. Then $B(m,u)$ is summed up over each MFCC of matching window as the similarity value of that window. Note that the window size is determined by both the analysis window of MFCC computation and the number of analysis windows to be used for matching. The complete flow of MFCC-based synchronization algorithm is shown in Figure 30. After the MFCC bins of both IP and

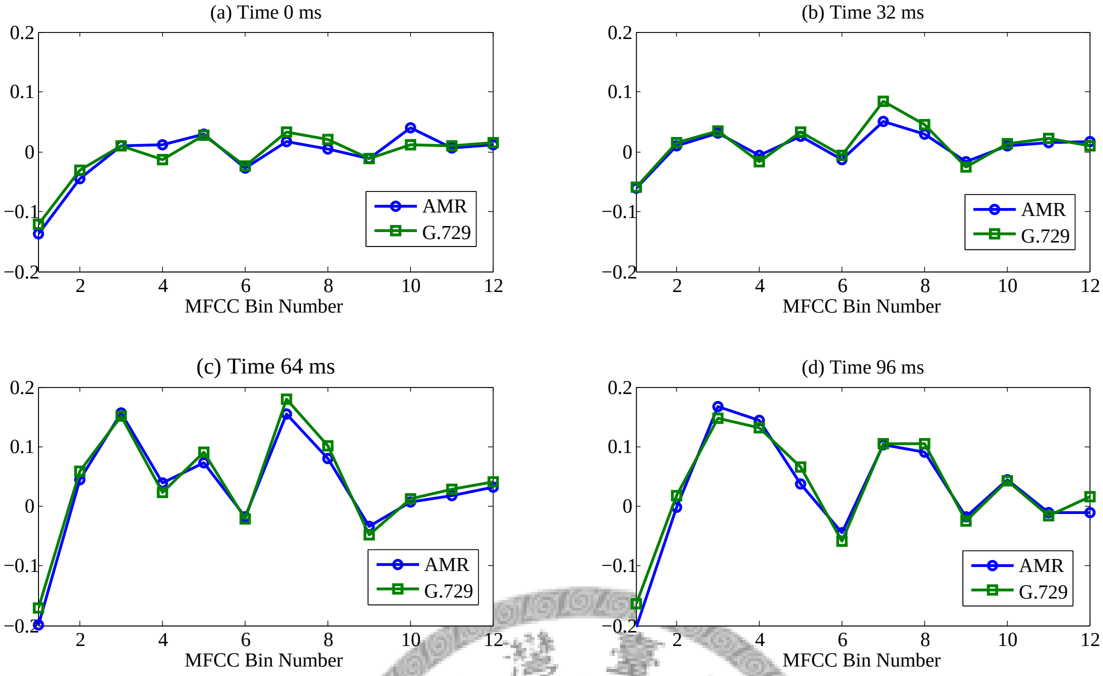

Figure 31: Effect of codec on MFCC

PSTN audio are extracted, X_u is subtracted from X_m. For each bin (u and m) after the subtraction, if m plus the error factor ϵ is larger than u, $|m - u|$ is summed into the similarity metric. Otherwise, a penalty value is summed, instead.

Note that since the MFCC acquired from an analysis window is a 13-entry column vector, the analysis windows used for matching is referred to as matching columns for simplicity. The author claims that the synchronized window should get the smallest similarity value. Therefore, within the search range, the matched window can easily be determined by taking minimum of the similarity metrics.

5.3 Performance Evaluation

We implement the algorithm using Matlab based on the MFCC code provided by [40]. Following the similar flow of discussion in the previous chapter, the evaluation of performance on distorted audio signals is discussed in this section. The size of the analysis window is set to 32ms in the evaluation, and the tolerable error is $\pm 32ms$ which implies a shift of 1 matching column.

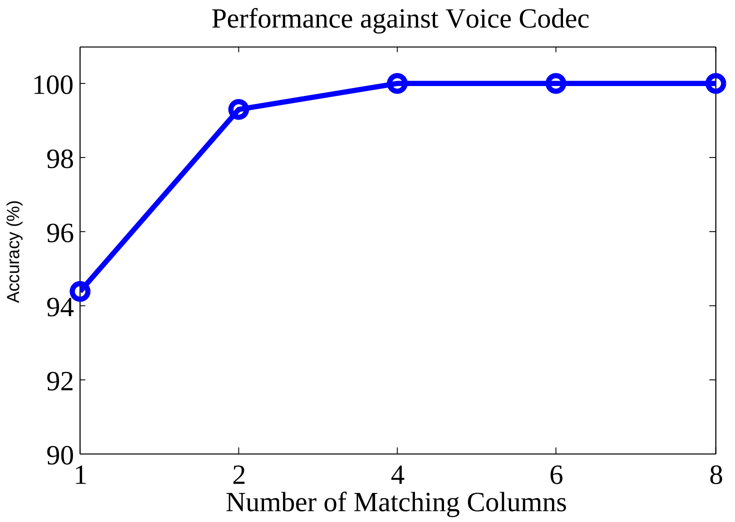

Figure 32: Performance of MFCC against codec distortion

5.3.1 Codec Distortion

As discussed in the previous chapter, different codecs may neglect different time redundancies and thus result in distinct waveforms. However, since the objective of redundancy removing is to achieve higher coding rate while the human perception is less affected, it should not severely influence the obtained MFCC. As shown in Figure 31, as expected, the MFCC bins of speeches after AMR and G.729 codec are similar, except for some small variation which is included by different redundancy removing criteria.

The performance shown in Figure 32 confirms the above inference. The percentage of accuracy can easily achieve 100% as the number of matching columns is larger than 2 which is 64ms in length. Therefore, we can conclude that the voice codec may not be an important issue if sufficient matching columns are used.

5.3.2 Misalignment of Analysis Windows

Unlike the search step in cross correlation, the a duration of audio signal is used to compute the MFCC value as a whole. While the MFCC is obtained, the windowing function might not take the same part of the audio segments for IP and PSTN network. Therefore, the obtained MFCC value might be different.

Figure 33 shows the effect of how misalignment affects the performance of MFCC. We manually include different delays in the PSTN audio so as to make it shifted away from the IP audio. We can observe that the performance may drop as the

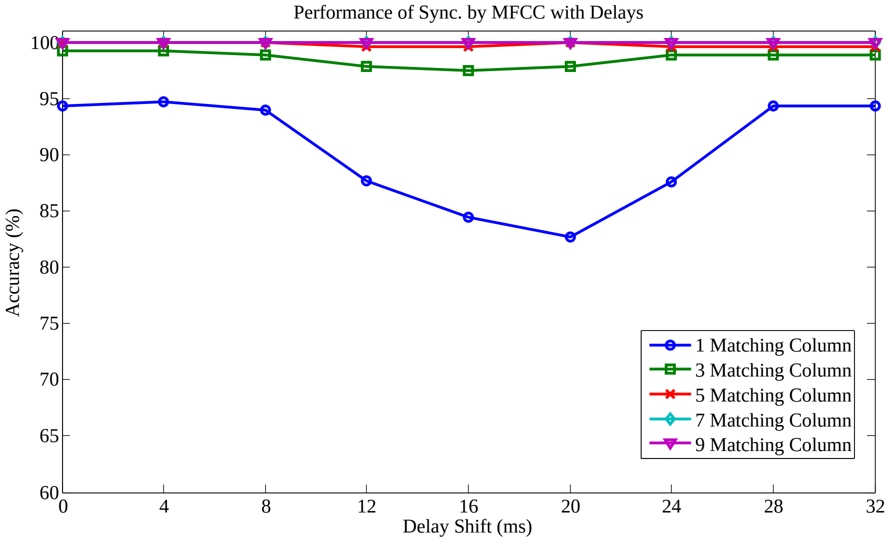

Figure 33: Effect of misaligned analysis window

PSTN audio is shifted away. However, as the time shift increases to near 32 ms, the percentage of accuracy increases again. This is because for a time shift around half of the analysis window size, the MFCC metric is more different than small shifts. When the time shift is near 32ms that implies the MFCC value is more similar to the next matching column. Hence the correct matched column moves to the next one which is still correct.

Note that although using only one matching column may severely suffer from the performance drop due to the analysis window misalignment, if more matching columns are used, for example 5 columns, the percentage of accuracy can achieve approximately 100%.

5.3.3 Noise Distortion

Similar to the discussion in cross correlation, we use AWGN noise for evaluation. Figure 34 shows that the MFCC bins might be distorted by the additional noise. However, we can observe that the noise mostly affects the lower MFCC bins. This might because the AWGN noise is spread through the entire spectrum which implies that the noise energy somewhat equally distributed in the spectrum with few variation. Therefore, after the DCT stage, the noise energy is kept in low MFCC bins as the low variation part of vocal tract response.

However, since the higher MFCC bins are not severely affected, if more matching columns are applied to compensate the effect on low MFCC bins, the performance

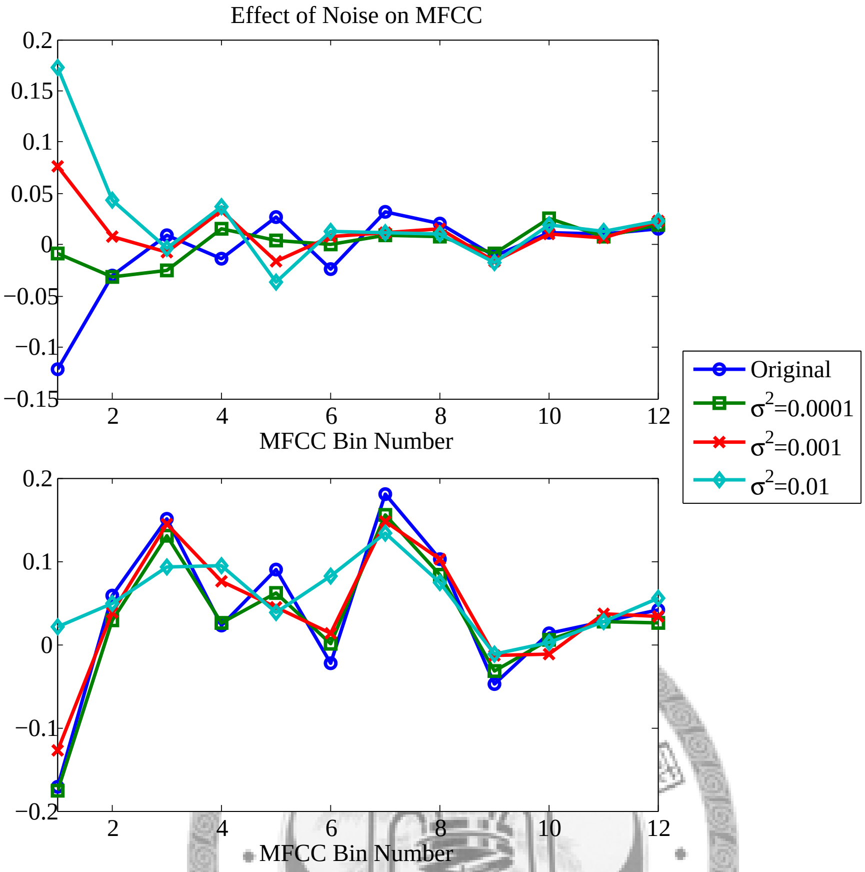

Figure 34: MFCC with different noise level

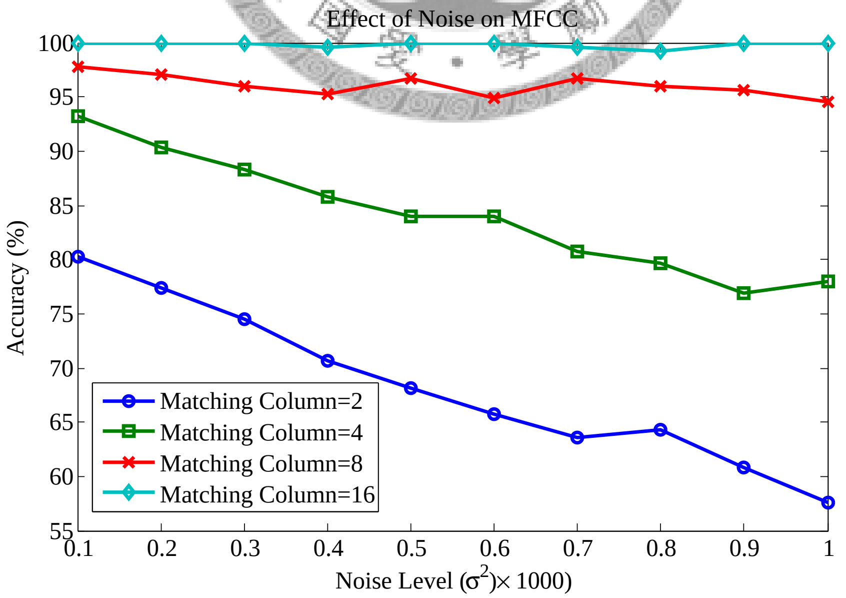

Figure 35: Effect of noise on MFCC performance

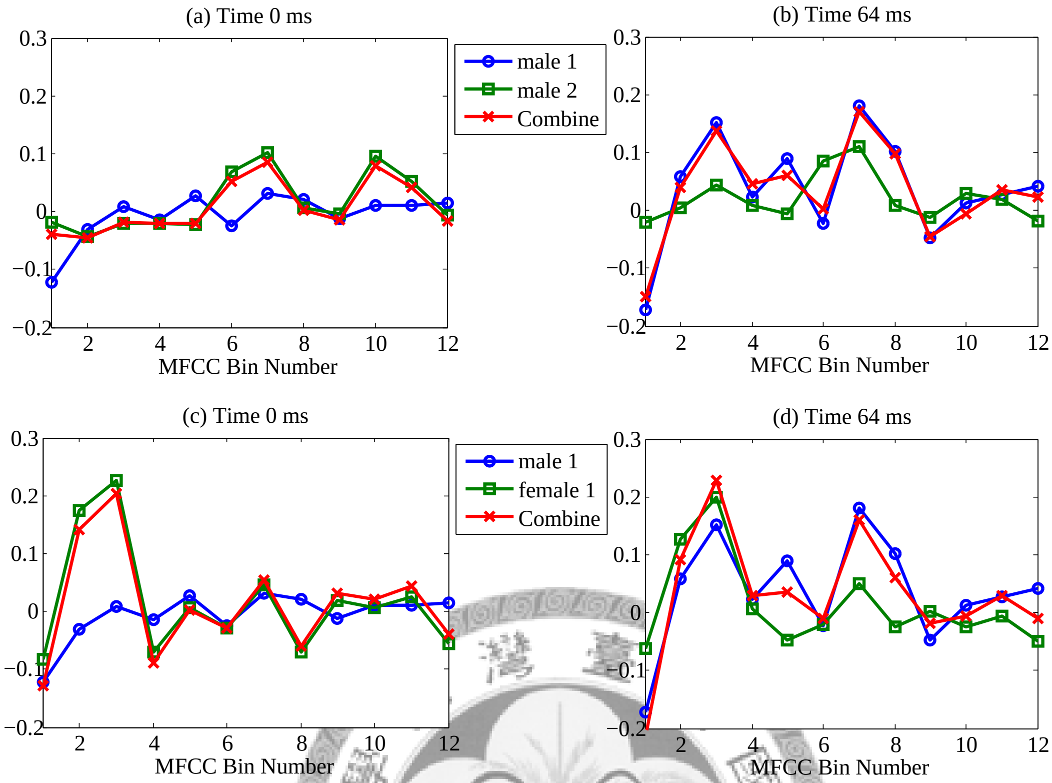

Figure 36: Effect of interference on MFCC

should not be too bad. As expected, Figure 35 shows that when only 2 matching columns are applied, the percentage of accuracy rapidly drops as the increment of noise energy. Nevertheless, if more matching columns are applied, for example 16 columns, the MFCC synchronization algorithm can usually achieve near 100% accuracy.

5.3.4 Overlapping Speakers

To verify the effect of overlapping speaker in the PSTN audio on the MFCC bins, we try different combination of speeches to compare the variation of MFCC bins. In Figure 36, we combine different gender speaker into the PSTN audio. The result shows that the MFCC bins of speech combination seems to follow the MFCC of either speeches in the combination. However, if the comparing IP audio is not the same as the one that the MFCC of combination follows, then the similarity metric may yield a large value at 0 shift point. Most of all, which speaker that the MFCC of combination may follow is difficult to predict in advance.

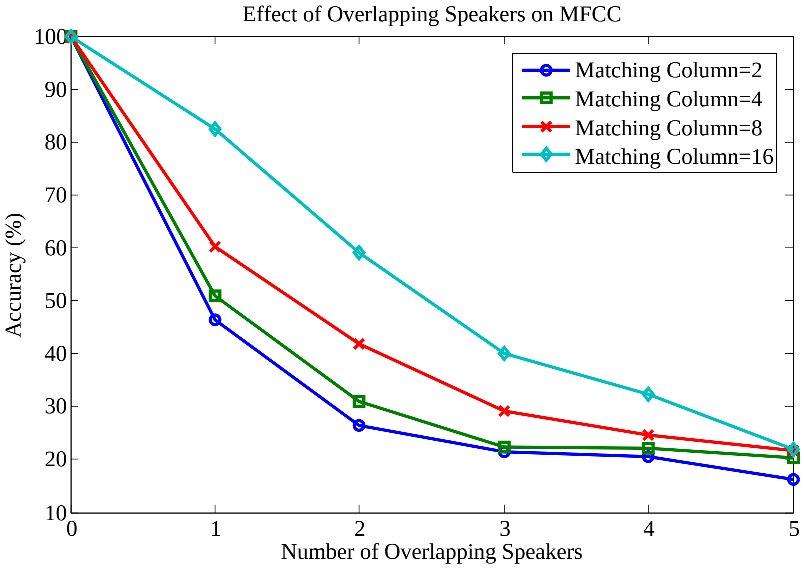

Figure 37: Performance against overlapping speeches

Additionally, according to Equation 5.10, there seems to be no apparent relationship between the unmixed MFCC and mixed MFCC that can be used to determine whether the mixed MFCC follows the unmixed speech. Therefore, it's hard to filter out the unmixed MFCC which is not the one that mixed MFCC follows so as to remove the cause of high similarity metric. This may severely corrupt the performance of MFCC synchronization.

As shown in Figure 37, once the PSTN audio contains an interference from other speaker, the percentage of accuracy suddenly drops, even though 16 matching columns (512ms in length which is half the size of search range) are applied. As the increment of speakers inside the mixture, the MFCC bins are so distorted that it can hardly differentiate the correct match and other shifts. Therefore, we conclude that MFCC may not be robust to interferences in the PSTN audio.

5.3.5 Packet Loss

In consideration of the packet loss that might be included in the IP audio, we apply the same packet loss concealment method as previous chapter which simply duplicates the previous packet for the lost one. Since the speech is inherently quasi-periodic, the packet duplication might still preserve this quasi-periodicity. Therefore, when MFCC tries to characterize the response of vocal tract from the speech, the resulting MFCC bins may not be largely interfered, as long as the packet loss rate is

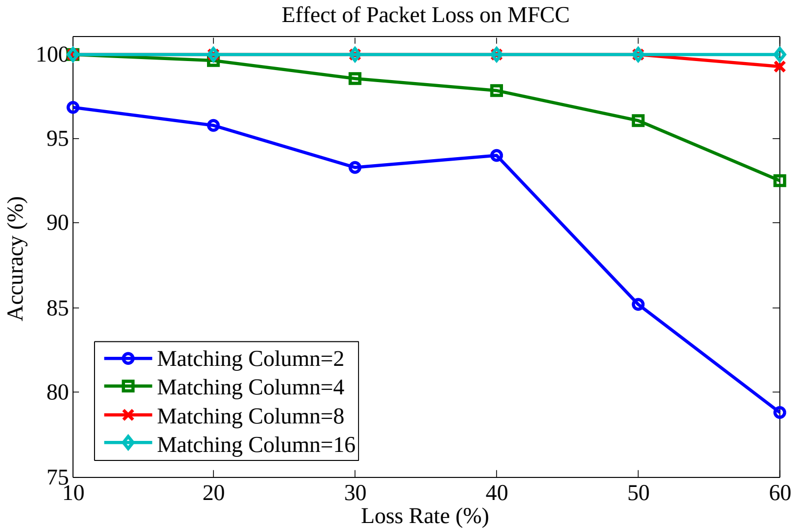

Figure 38: Performance against packet loss

not so large that the essential components of speech are lost.

Figure 38 reveals the robustness of MFCC against packet loss while the lost packet is concealed by the duplication of the previously-received packet. Although few matching columns might not have enough information to differentiate the correct match from other shifts, 4 matching columns may be quite enough for the common loss rate of less than 20%.

In consideration of practical situation, Figure 62 shows the performance of MFCC when multiple sources of distortions occurs. We can observe that since MFCC is vulnerable to overlapping speakers, whenever a speech is mixed, the performance is degraded. However, for non-overlapped speech, additional sources of distortion doesn't degrade the performance. In other words, if MFCC is robust to the sources of distortions, the combination of distortion doesn't largely affect the performance.

5.3.6 Short Conclusion on Performance

To sum up the afore-mentioned evaluations, using MFCC bins to find similarity for synchronization seems to be a good option in that it is robust against many kinds of source of waveform distortions, as long as sufficiently large matching columns are applied. However, in the evaluation of performance against overlapping speakers in the PSTN audio, we observe that the percentage of accuracy is severely corrupted by the additional speeches. Besides, even though large matching columns are applied the performance doesn't show major improvement. This may be a problem since in

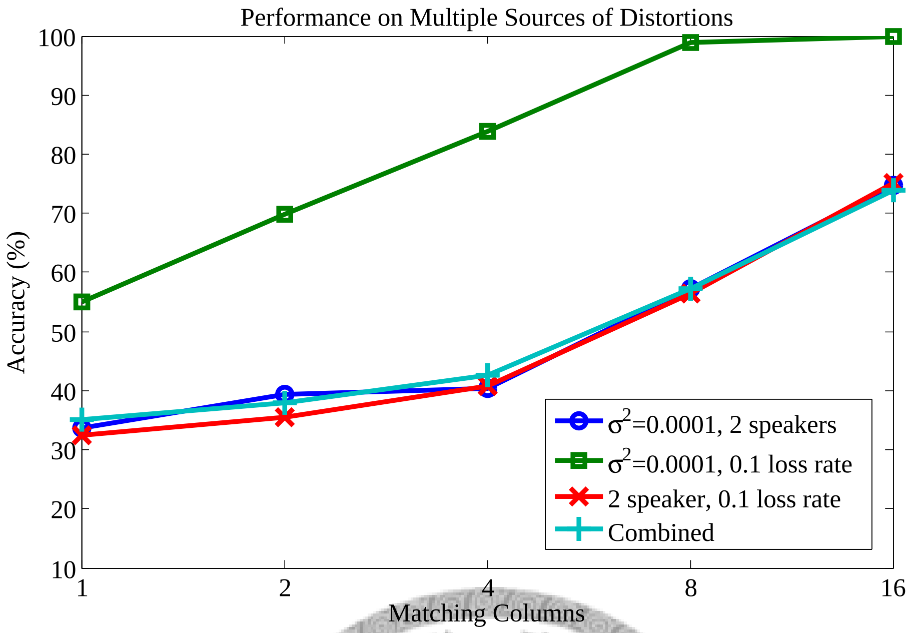

Figure 39: Performance against multiple sources of distortions

a practical conference, especial during a keen discussion, many speakers may speak at the same time, and thus the PSTN audio is likely to be a mixture of multiple speakers. Therefore, a different speech representation which can somewhat separate the speakers inside the speech mixture should be considered.

CHAPTER 6

SYNCHRONIZATION BASED ON SPECTROGRAM

As concluded in the previous chapter, synchronization based on MFCC may suffer from the performance corruption while PSTN audio is a mixture of conferees. The reason to the performance corruption is that MFCC bins are easily distorted by other speech sources, and additionally, it is hard to filter out the distorted matching column by inferring from the unmixed and mixed MFCC. Therefore, in this chapter, we try to discover a different representation of speech such that different sources of speaker can somehow be separated which is related to the research field of speaker separation. By surveying the research in speaker separation, we discover the advantage of simply using spectrogram for synchronization. After the analysis of advantage of spectrogram in separating different speakers, we propose a synchronization algorithm based on spectrogram. Then similar evaluation of performance on waveform distortions to previous chapters is included.

6.1 Spectrogram-Based Synchronization

Although speaker separation is not a new topic in the field of digital speech processing, the discussion background is so different that this research can hardly be used in this scenario. In [41–43], their algorithms require all the speeches are supervised so as to construct masks, bases, or decomposition matrices for further computation. Since the receiver has no way to know all the individual speeches, these solutions are not applicable. Even though we can obtain all speeches somehow, the modeling processes for masks, bases, and matrices are accurate only when enough audio received. It implies that at the beginning of conference, the models are not good enough.

In [44] and [45], speech separation can be done without supervised audio. However, authors of [44] use EM algorithm to estimate multi-pitch model. This cannot support real-time separation. Although in [45] the authors claim that their algorithm can achieve real-time processing with high-performance DSP architectures, this DSP architecture is not available for common mobile handsets. Furthermore, their algorithm assumes there are only two speakers and the volumes are sufficiently different.

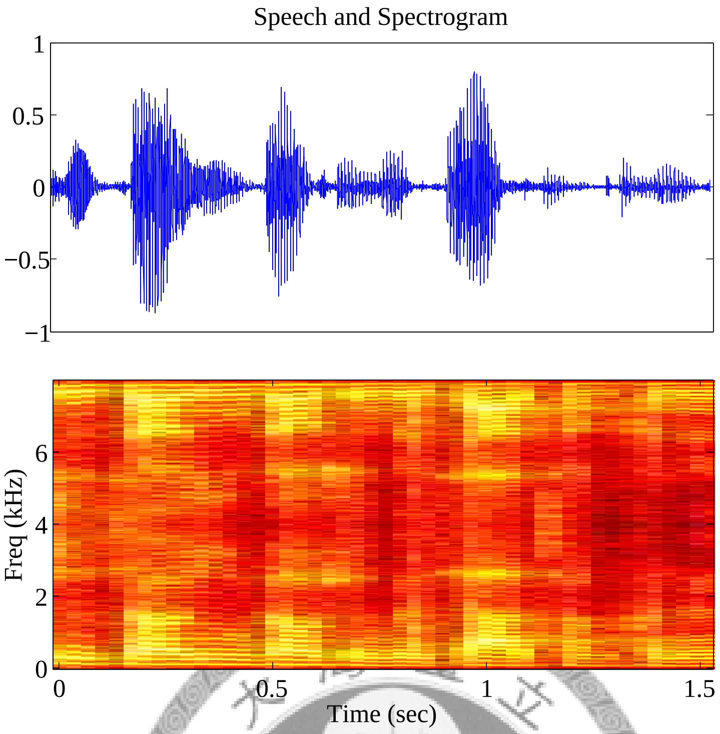

Figure 40: Spectrogram of a period of speech

This may not be the case all the time.

As concluded in the previous chapter, the previous algorithm based on MFCC is not a suitable solution to synchronization since MFCC considers only the factors that mostly related to human perception, and thus omitting other information of speech which might be useful in synchronization. Inspired by the human auditory system, research on Computational Auditory Scene Analysis (CASA) [46] deals with verification and segregation of audio segments to imitate how audio is processed in human auditory system. In the field of CASA and even common speech processing [47], audio signals are usually manipulated in the time-frequency (T-F) domain which demonstrates the frequency distribution within short time interval. Frequency distribution may differ in consequence of different speakers or different words. Even within the same word, different syllables may reveal different frequency distribution. This implies that from the frequency distribution, audio from different speakers or different syllables pronounced by the same speaker is distinguishable in T-F domain. Therefore, we develop an algorithm based on these T-F domain features for synchronization.

An example of spectrogram of speech is illustrated in Figure 40. The speech is divided into half-overlapping analysis windows. Each analysis window is transformed to one spectrum by Fourier transform. Different shapes in time-domain may results in different distributions in the spectrogram. This implies that the spectrogram can somewhat represent the time-domain characteristic in the frequency domain.

6.1.1 Sparsity on Spectrogram

To cope with the insufficiency of synchronization algorithm based on either cross correlation or MFCC for speech mixture, an individual-speaker-identifiable feature should be used. Joujine et al. [48] and Roweis [49] have noted that a speech signal is sparsely distributed in a high-resolution T-F representation and ,as a result, different speech utterances tend not to overlap in individual T-F units. This observation leads to the property of orthogonality between different speech utterances, which is often referred to as *Window-Disjoint Orthogonality (W-DO).* The orthogonality assumption holds well for mixtures of speech and other sparsely distributed signals, but is not valid for speech babble.

6.1.1.1 Concepts of Approximate W-DO

Perfect W-DO should satisfy that each frequency bin at a certain window is contributed by single speaker. It can be represented by Equation 6.6.

$$\begin{aligned}&X(\tau,\omega)\times Y(\tau,\omega)=0\\&\text{where } X(\tau,\omega) \text{ and } Y(\tau,\omega) \text{ are the spectrogram of different speakers}\end{aligned}\tag{6.1}$$

Many blind speech separation research endeavors are based on the approximate W-DO of speech, such as [50–55], while [56] has analyzed the effect of approximate W-DO. This sparsity in spectrogram from different speaker provides a helpful tool to separate speech mixture and also suggests a better feature than MFCC in synchronization algorithm design. A more concrete concept of approximate W-DO of speech can be presented by Figure 41.

The upper two spectrograms in Figure 41 shows the T-F distribution of original speeches. As expected, the male speech locates most of its energy at low frequency bins, while the female speech reveals a more spread distribution in spectrogram. To illustrates the concept of approximate W-DO, the lower spectrogram shows the square-rooted multiplication of the upper two spectrograms along with the spectrum for a certain analysis window. In comparison with the original two spectrograms, the square-rooted multiplication (geometric mean) exhibits a relative low magnitude distribution at most frequencies. That means if one speech is large at a certain frequency then the other speech is likely to be small. This implies that these two

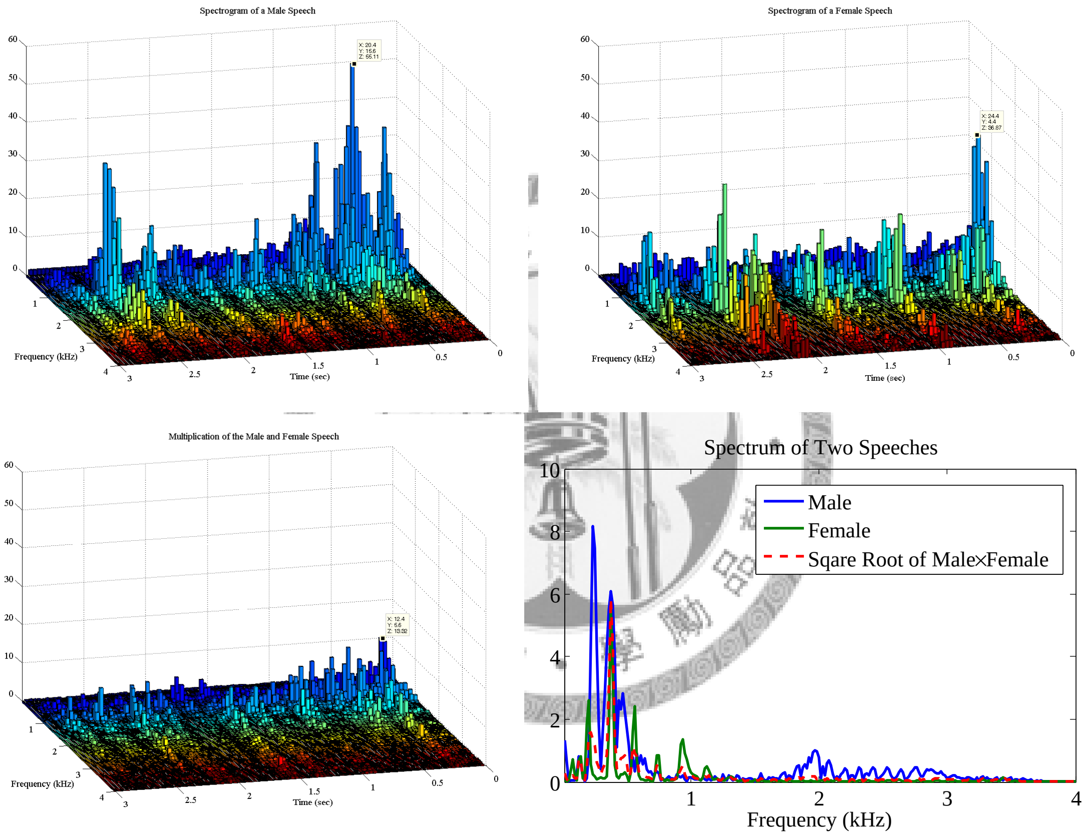

Figure 41: Spectrogram of a male, a female speech, and their multiplication

speeches tends not to have high magnitude at the same T-F bin which is exactly the spirit of approximate W-DO. However, solely from the spectrogram multiplication, it is hard to tell the degree of the accuracy of W-DO. In the next part, a quantity that measures the accuracy of W-DO is presented, along with a discussion of the influence of different parameter settings on W-DO.

6.1.1.2 Measurement of W-DO

Authors in [50] presents a suitable metric for W-DO measurement. Ideally, perfect W-DO ensures that for each T-F bin in the mixture spectrogram, the energy is solely contributed by a single speaker. However, approximate W-DO suggests that for speech signal, it is likely that more than one speech source may contribute to the same T-F bin while only one or few of them may provide significant large energy. Hence, a more accurate W-DO can be interpreted in two fold: If each T-F bin is assumed to belong to one significant speech,

1. the total preserved energy of the speech of interest should be approximately equal to the total original energy of it.

2. the remaining energy of other speeches should be as low as possible.

For this reason, two factors are defined: (1) the preserved-signal ratio (PSR) and (2) the signal-to-interference ratio (SIR). Before the definition of PSR and SIR, the significance criterion is first defined as

$$\Psi_j^x \equiv \begin{cases} 1, & \text{if } 20log(\frac{\hat{s_j}(\tau,\omega)}{\hat{y_j}(\tau,\omega)} \leq x); \\ 0, & \text{otherwise,} \end{cases} \quad \text{where } x \text{ is the masking threshold.} \tag{6.2}$$

Here, x is called masking threshold because Ψ masks out other speeches by a x dB threshold, and j represents the index of source speech. When the source energy is larger than the interference energy by x dB in a specific T-F bin, it is considered significant in this T-F bin.

For a specific significance criterion Ψ_j^x, PSR is defined as

$$\begin{gathered} PSR_{\Psi_j^x} \equiv \frac{||\Psi_j^x(\tau,\omega)\hat{s_j}(\tau,\omega)||^2}{||\hat{s_j}(\tau,\omega)||^2} \\ \text{where } ||f(\tau,\omega)||^2 \equiv \sum_\tau \sum_\omega |f(\tau,\omega)|^2 \end{gathered} \tag{6.3}$$

which suggests that $PSR_{\Psi_j^x} = 1$ if Ψ_j^x preserves all the original source energy.

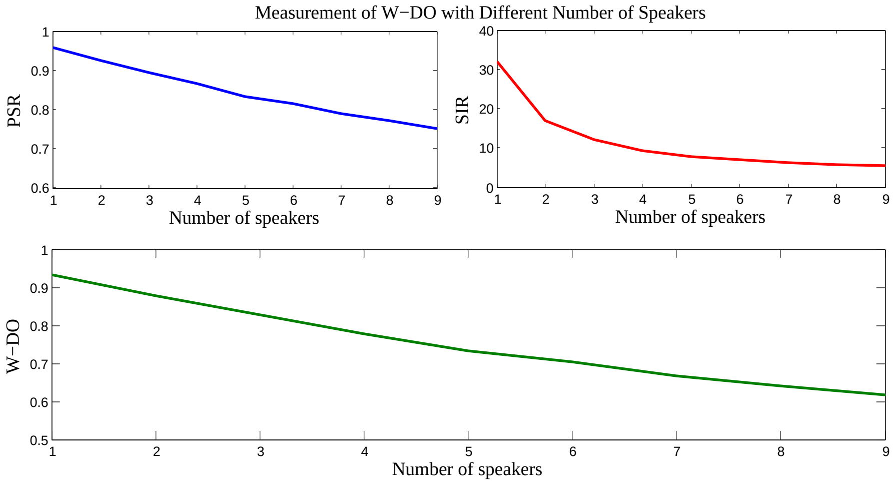

Figure 42: WDO measurement with different number of speakers

To define SIR, the interference y_j is defined beforehand as

$$y_j(t) \equiv \sum_{k,k \neq j} s_k(t) \tag{6.4}$$

which is the time-domain summation of speeches other than s_j. Thus SIR with significance criterion Ψ_j^x can be defined as

$$SIR_{\Psi_j^x} \equiv \frac{||\Psi_j^x(\tau,\omega)\hat{s_j}(\tau,\omega)||^2}{||\Psi_j^x(\tau,\omega)\hat{y_j}(\tau,\omega)||^2}. \tag{6.5}$$

Obviously, larger $SIR_{\Psi_j^x}$ indicates less remaining interference after Ψ_j^x.

Combining $PSR_{\Psi_j^x}$ and $SIR_{\Psi_j^x}$ into one measure of approximate W-DO, metric $WDO_{\Psi_j^x}$ is defined as

$$\begin{aligned} WDO_{\Psi_j^x} &\equiv \frac{||\Psi_j^x(\tau,\omega)\hat{s_j}(\tau,\omega)||^2 - ||\Psi_j^x(\tau,\omega)\hat{y_j}(\tau,\omega)||^2}{||\hat{s_j}(\tau,\omega)||^2} \\ &= PSR_{\Psi_j^x} - \frac{PSR_{\Psi_j^x}}{SIR_{\Psi_j^x}}, \end{aligned} \tag{6.6}$$

meaning the normalized difference of remaining source energy and interference energy after Ψ_j^x. For perfect W-DO, $WDO_{\Psi_j^x} = 1$ which implies that $PSR_{\Psi_j^x} = 1$ and $SIR_{\Psi_j^x} \rightarrow \infty$ so different sources are completely disjoint. The better approximate W-DO this speech can achieve, the more closer to 1 $WDO_{\Psi_j^x}$ is.

Since W-DO relies on the sparsity distribution of different speeches, when the number of speakers in the speech mixture, it is much more likely that different speakers may all largely contribute to the same T-F bin, and thus declining the accuracy of

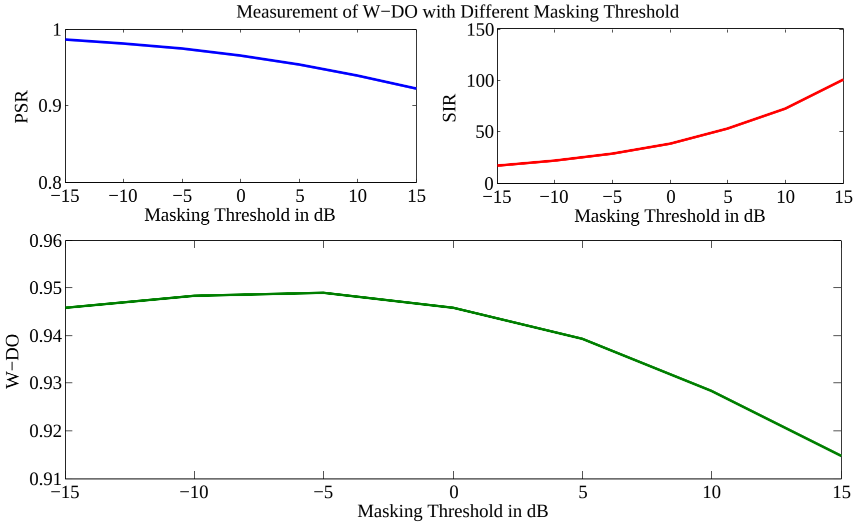

Figure 43: WDO measurement with different masking thresholds

approximate W-DO. Figure 42 shows the WDO measure with regards to different number of speeches in the interference y_j. As expected, WDO decreases as y_j contains more and more speakers as a result of declining PSR and SIR. This indicates that it is more difficult to tell a specific speaker from others since they are too disorderly to tract. Fortunately, in our scenario, the number of interfering speaker is unlikely to have a large value. For a mixture containing less than five speakers, the W-DO measure could be larger than 0.75. This means that deducting the effect of interference, still three-forth of the original energy is retained.

However, the accuracy of approximate W-DO is not only affected by the number of speakers in the mixture. Different parameter settings at the significance determination stage and even the STFT may also induce different W-DO measures. For significant determination, the masking threshold directly influence on both PSR and SIR. As shown in Figure 43, larger masking threshold suggests that less T-F bins would be chosen to be significant, and thus the preserved energy of original source is less. Therefore PSR is lower. On the other hand, for the chosen T-F bins, the source signal is much greater than the interference so SIR is accordingly larger. This is a trade-off to W-DO since going either side does not benefit both. The W-DO measure in Figure 43 reveals that a moderate masking threshold is required to achieve higher approximate W-DO.

On the other side, the windowing process may also affect the accuracy of approximate W-DO as discussed in [57]. Larger window size may increase the frequency resolution, however, it may also destroy the W-disjoint orthogonality among speeches.

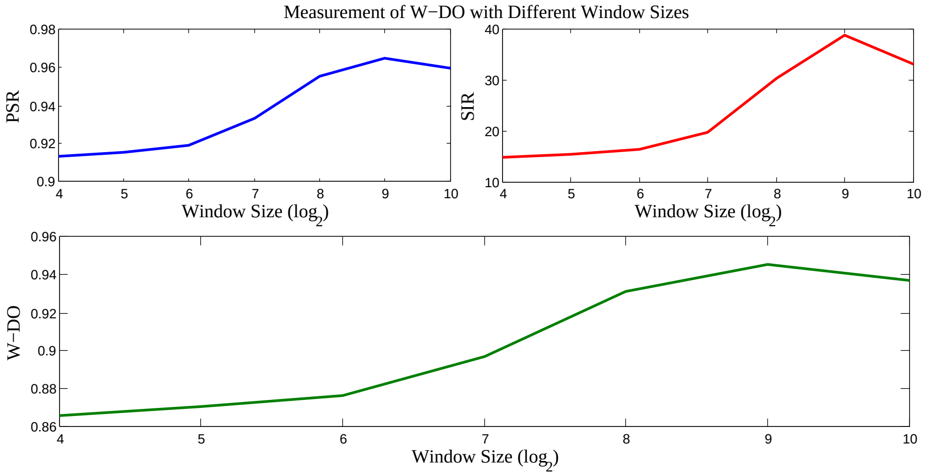

Figure 44: WDO measurement with different window sizes

On the contrary, small window size may not contain enough time information to represent the quasi-periodicity of speech as shown in the previous chapter. There is no optimal method of choosing the best-suited window size, nevertheless, Figure 44 shows that the best window size to achieve the highest WDO is 512 samples (64ms for 8k sampling rate). This is consistent with the work in [50]. So, we assume that an analysis window size of 64ms is the preferable one for high W-DO.

6.1.1.3 Significance in Mixture Spectrum

Since we have introduce the concept of W-DO in spectrogram, the next question to consider is how to apply the concept of W-DO to our synchronization algorithm. The objective is to infer from the unmixed IP audio and the mixed PSTN audio that which frequencies contains mostly of the unmixed IP audio. Figure 45 shows the spectrum of two speeches and their combination. When one speech is silence or almost silence, the spectrum of combination is of course dominated by the voiced one. However, if two speeches are both voiced, we can observe that the combination is usually dominated by either one of the speeches as a result of approximate W-DO. This implies that if we can infer from the unmixed IP audio and the mixed PSTN audio that which one dominates a certain frequency. Then by comparing only those frequencies that the unmixed IP audio dominates, the effect of speech mixture on the synchronization should be lessen.

Concluded from the discussion in this subsection, T-F representation of speech can lessen the indistinguishableness of speech mixture. Since in our scenario the unmixed speech can be transmitted through the IP network, exploiting the uncorrupted T-F

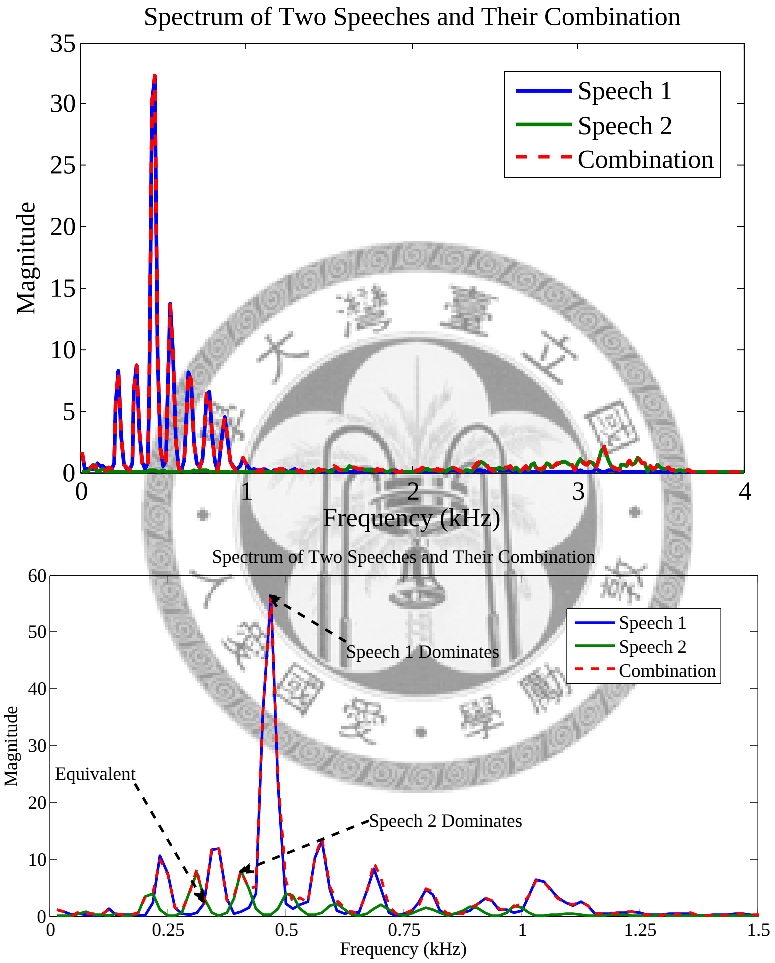

Figure 45: Spectrum of mixture is usually dominated by certain speech

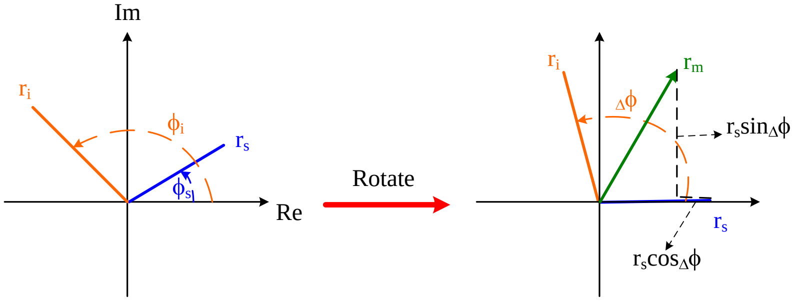

Figure 46: Complex number summation in the complex plane

Table 3: List of parameters

Notation	Meaning
$\hat{a}$	T-F representation of a after STFT
$\hat{a}(\tau,\omega)$	Specific bin of $\hat{a}$ at time τ and frequency ω
$r_{a(\tau,\omega)}$	Magnitude of $\hat{a}(\tau,\omega)$
$\phi_{a(\tau,\omega)}$	Phase angle of $\hat{a}(\tau,\omega)$
${}_{\Delta}\phi_{\tau,\omega}$	Phase difference of two sources at time τ and frequency ω
α	$\frac{r_i}{r_s}$
η	Bound of $\frac{r_m}{r_s}$

bins for synchronization determination might be able to increase the performance of synchronization algorithm for mixed speech from the PSTN network.

6.2 Synchronization Algorithm Design

Based on the previous discussions of merits on spectrogram for synchronization, we proposed an algorithm structure that utilizes the speech spectrogram to learn the synchronization point in PSTN-audio stream.

6.2.1 Significance Determination

The most important unit in our synchronization algorithm is the significance locator which is designed to improve the robustness of our algorithm against the effect of simultaneous speakers. Based on the fact that it is easier to distinguish a specific source $\hat{s}$ from the mixture $\hat{m}$ when $\hat{s}$ is larger than the interference $\hat{i}$, if we can somehow manage to determine the significance of $\hat{s}$ in $\hat{m}$, then the synchronization would be simply matching the significant frequency bins with $\hat{s}(\tau,\omega)$ received from the IP network.

To clarify the determination of significance merely from $\hat{m}(\tau,\omega)$ and $\hat{s}(\tau,\omega)$, further discussions are stated in the following paragraphs. Source $\hat{s}(\tau,\omega) = |r_{s(\tau,\omega)}|\angle\phi_{s(\tau,\omega)}$ and interference $\hat{i}(\tau,\omega) = |r_{i(\tau,\omega)}|\angle\phi_{i(\tau,\omega)}$ are shown in the left coordinate of Figure 46. To simplify the notation, $r_{s(\tau,\omega)}$ is simply referred to as r_s if no further statement, and so are $r_{i(\tau,\omega)}$ and $r_{m(\tau,\omega)}$. The list of used parameters are shown in Table 6.2.1.

Without loss of generality the coordinate can always be rotated so as to make the source lie on the horizontal axis. Thus the angel between $\hat{s}(\tau,\omega)$ and $\hat{i}(\tau,\omega)$ is ${}_\Delta\phi_{\tau,\omega}$ which is simply noted as ${}_\Delta\phi$. Therefore, r_m is derived as

$$\begin{aligned} r_m^2 &= (r_s + r_i cos({}_\Delta\phi))^2 + (r_i sin_\Delta\phi)^2 \\ &= r_s^2 + 2r_s r_i cos_\Delta\phi + r_i^2(cos^2_\Delta\phi + sin^2_\Delta\phi) \\ &= r_s^2 + 2r_s r_i cos_\Delta\phi + r_i^2. \end{aligned} \tag{6.7}$$

It is shown in Equation 6.7 that the magnitude of mixture is dependent not only on the magnitudes of source and interference but also on the relative phase angle between them.

Assuming that $r_i = \alpha r_s$ where $0 \leq \alpha$, Equation 6.7 becomes

$$\begin{aligned} r_m^2 &= r_s^2 + 2\alpha r_s^2 cos_\Delta\phi + \alpha^2 r_s^2 \\ &= r_s^2(1 + 2\alpha cos_\Delta\phi + \alpha^2) \end{aligned} \tag{6.8}$$

If r_s dominates r_m,

$$\begin{aligned} &\Leftrightarrow \alpha \text{ should be small} \\ &\Rightarrow r_m \to r_s \\ &\Leftrightarrow \frac{r_m}{r_s} \to 1. \end{aligned} \tag{6.9}$$

From Equation 6.9, we can see that if r_s dominates r_m, $\frac{r_m}{r_s} \to 1$. The reverse is not necessarily be true since $\frac{r_m}{r_s} \to 1$ is only a sufficient condition of r_s dominating r_m. However, the sparsity (approximate W-DO) on spectrogram of speech signals suggests that usually r_s and r_i are largely different from each other. In other words, when r_s is closer to r_m ($\frac{r_m}{r_s} \to 1$), it is most likely that α is very small $\Leftrightarrow$ r_s dominates.

Using the above relation, we design the similarity metric as the normalized absolute error between r_s and r_m ($|\frac{r_m - r_s}{r_s}|$). In general, the value of similarity metric can be derived as

$$\begin{aligned} r_s - r_i \leq r_m \leq r_s + r_i \Rightarrow -\alpha r_s &\leq r_m - r_s \leq \alpha r_s \\ \Rightarrow -\alpha &\leq \frac{r_m - r_s}{r_s} \leq \alpha \\ \Rightarrow |&\frac{r_m - r_s}{r_s}| \leq \alpha. \end{aligned} \tag{6.10}$$

It can be seen that the similarity metric is bounded by α. If only those frequency bins where r_s dominates are chosen, the similarity metric may obtain relatively low values since those bins have small α values.

In order to determine whether r_s is significant in a certain frequency bin, $\frac{r_m}{r_s}$ is first examined. If $\frac{r_m}{r_s}$ is less than a certain bound η above 1, this bin is chosen for similarity comparison since the possible α values ranges only from 0 to 1, and thus are potentially significant. For those frequency bins at the correct synchronization column, since bins with relatively small r_s values to r_m values are filtered out, the remaining bins are usually dominated by r_s. Therefore the similarity metric may usually obtain small values bounded by α. However, for those frequency bins at other columns, the comparing r_m values are not composed of r_s values, so the similarity metric will not be bounded and thus may obtain quite large values. Hence the correct synchronization column can be located by choosing the one with smallest average similarity metric value.

Note that possible α values are affected by the bound η. While larger η may also choose less significant bins for comparison, smaller η values may reduce the number of chosen frequency bins. However, more number of chosen frequency bins may lessen the effect of occasional α values near 1. This implies that the more significant frequency bins are chosen, the more accurate this algorithm can achieve.

6.2.2 Synchronization Module

Based on the afore-mentioned significance determination, we design a synchronization module which utilize the spectrogram for synchronization, as shown in Figure 47.

When the synchronization is triggered, a segment of speech from the IP audio buffer is chosen and sent to the *FFT processor*. The size of the speech segment depends on the specified matching window size. Similar to the notation in MFCC, the number of analysis windows inside a matching window is simply referred as the matching columns since the spectrogram is like a set of column vectors while each column contains the frequency distribution of certain analysis window. Meanwhile, within the search range, the PSTN audio is sent to the FFT processor for later use in synchronization.

Inside the synchronization module, both the spectrograms from IP audio and PSTN audio of the size of matching window are first sent to the *silence filter* which determines whether this matching column is silence. This silence filter is required since if the IP spectrogram is silence, then using this column for synchronization

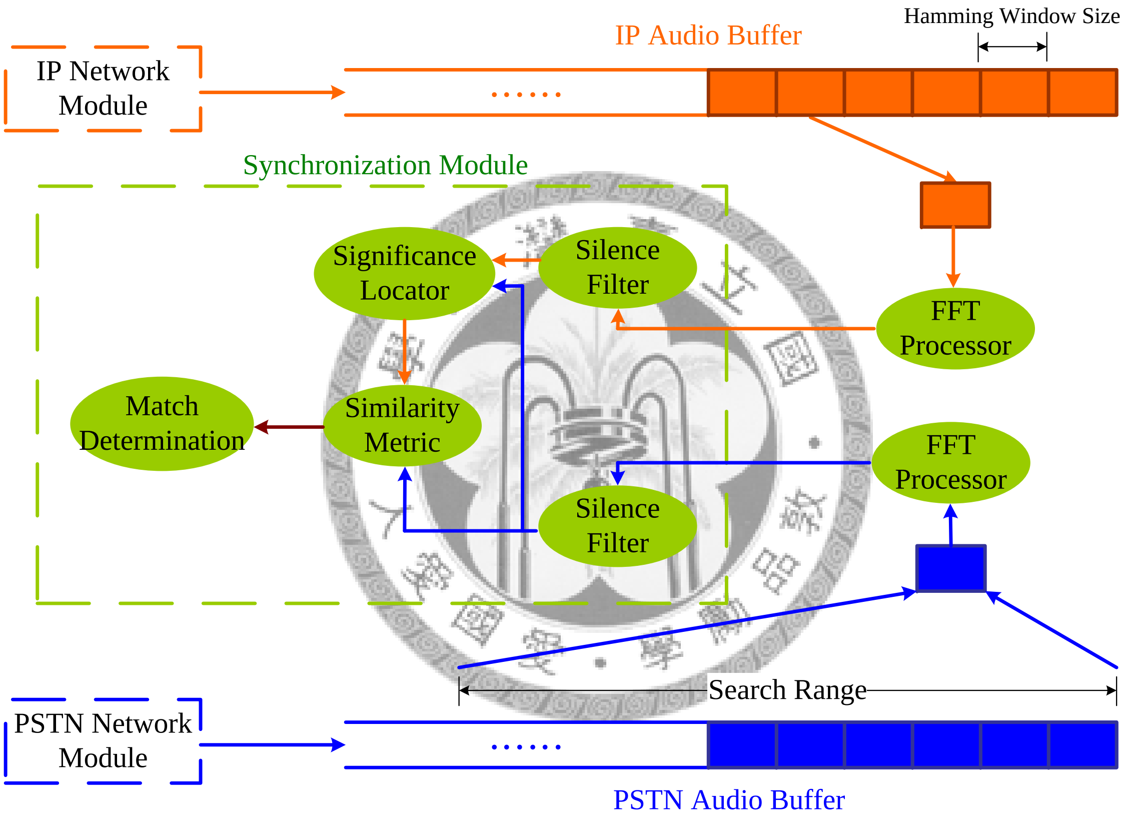

Figure 47: Synchronization framework using proposed algorithm

determination is easily affected by noises. On the other hand, because only none silence parts of IP audio are chosen for comparison, removing the silence parts in PSTN audio can reduce the computation for comparing these silence parts.

After the silence filter stage, the *significance locator* infer from these two spectrograms to determine which frequencies are possibly dominated by the IP audio. The significance locator is designed as previously stated. Then this potential dominating frequency information is sent to the *similarity metric*, together with the IP and PSTN matching columns. The similarity metric stage computes the mean absolute difference of these two matching columns at only the frequencies depicted by the significance locator.

The same process is iteratively performed for each matching windows within the search range. Each iteration records the acquired similarity metric for its matching window. After the iterations, the matched window is determined by the matching window with the minimum similarity metric.

6.3 Performance Evaluation

In this section, the synchronization algorithm is implemented using Matlab to evaluate the effectiveness and robustness of it. Again, the evaluation on performance focuses on the discussion of different kinds of distortions on spectrogram. Note that in the following evaluation, the analysis window used in FFT is 64ms with 32ms overlap. Therefore, the acquired spectrum columns are 32ms away from their neighboring columns

6.3.1 Codec Distortion

Since the spectrogram faithful represents the frequency distribution of the speech signal, when the speech signal is modified, the spectrogram is consequently modified, too. Therefore, the resulting spectrograms for different voice codecs might be different. Figure 48 shows the difference between the spectrums after different voice codecs for a certain analysis window. It is observed that at some frequencies the AMR and G.729 codec may result in different values. Since the differences may not be large enough to be eliminated by the significance locator, they are all included in the similarity metric and thus might induce error judgments.

Fortunately, the error induced by the voice codec can be lessen by sufficiently large matching columns, for example 6 columns which is 192ms in length, as shown in Figure 49. Therefore, the spectrogram-based synchronization algorithm can overcome

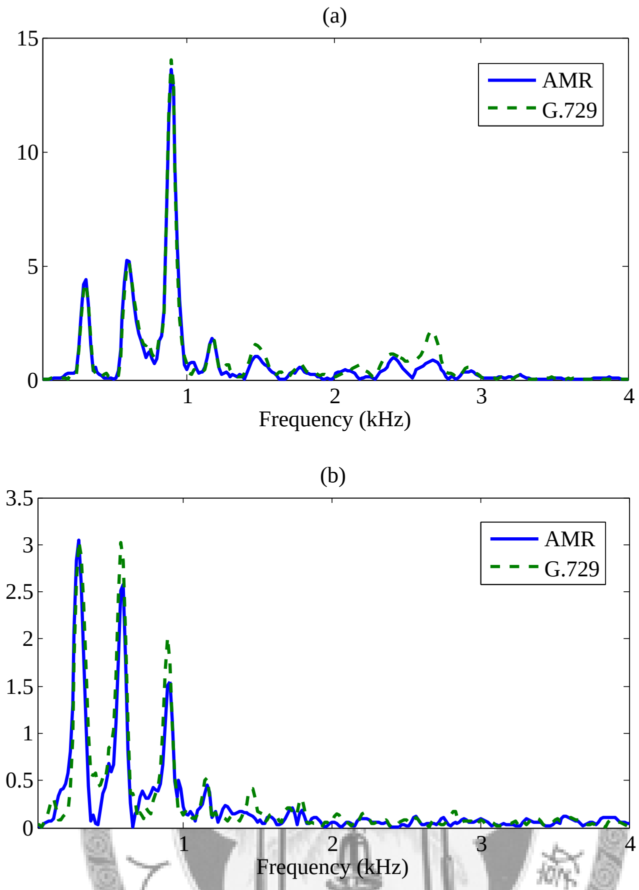

Figure 48: Effect of codec distortion on spectrum

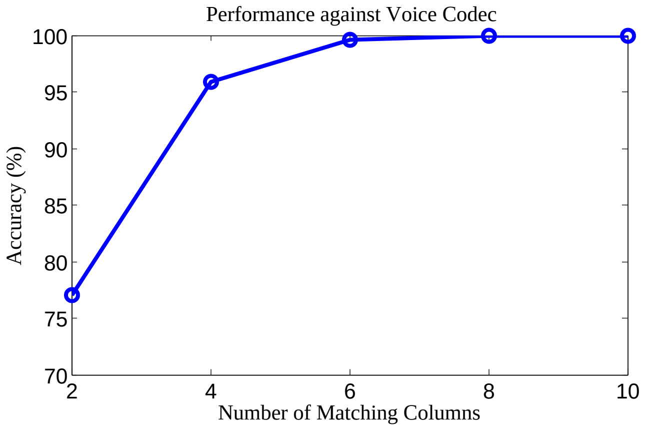

Figure 49: Performance of spectrogram against codec distortion

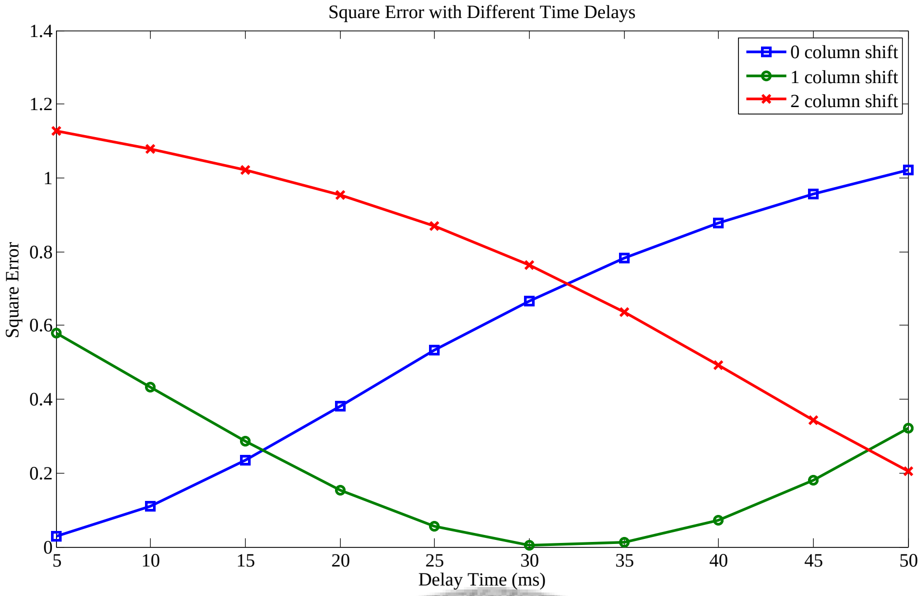

Figure 50: Effect of window misalignment

the distortion introduced by voice codecs.

6.3.2 Misalignment of Analysis Windows

In order to measure the effect of analysis window misalignment, we define a simple normalized square error metric, called SE, which is defined as

$$SE = \frac{||\hat{s}(\tau, \omega) - \hat{s}(\tau - \Delta, \omega)||^2}{||\hat{s}(\tau, \omega)||^2} \tag{6.11}$$

where Δ is the time shift.

SE actually represents the normalized error energy among neighboring columns. The more the neighboring columns differ from the column of interest, the larger SE value is obtained. Figure 50 shows the SE between the spectrogram with time shifted analysis windows and the unshifted columns. As the time shifts away from 0, the SE comparing to the 0-shifted column starts to grow while the SE comparing to the 1-shifted column starts to decrease. These two SE lines cross at the time shift of around 16 ms. Since each column is 32ms away from its neighbor, we can observe that the SE value comparing to the 1-shifted column has its minimum at the time shift of around 32ms.

This observation implies that when the time shift is near 0 or the integer multiple of 32ms, the determined matched column is highly probable to be chosen as the nearest column. However, when the time shift is around 16ms, the spectrogram becomes so

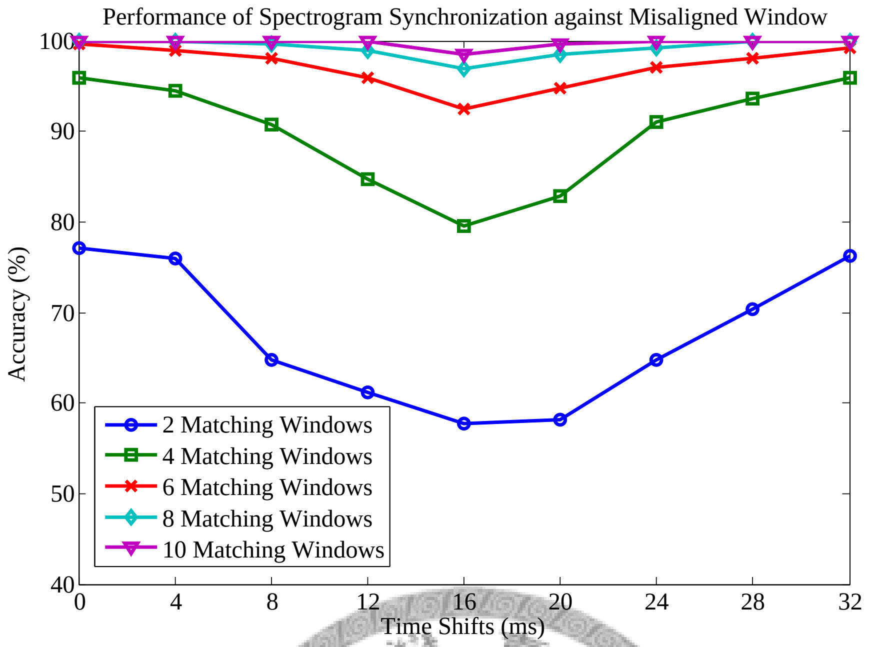

Figure 51: Performance of spectrogram for misaligned windows

obscure that error judgments may occur. The performance shown in Figure 51 is consistent with the inference from Figure 50. Although few matching columns might not obtain satisfying percentage of accuracy, when the number of matching columns increase to over 6 columns (which is 192ms in time), the achievable accuracy is more than 90%.

6.3.3 Noise Distortion

Again, the AWGN noise is applied on the speech to observe the influence of noise on spectrogram. Since the AWGN noise is ideally equally distributed throughout the spectrum, the effect on each frequency component may be small. As long as the energy of the clean speech is large, the influence of noise is negligible, as shown in Figure 52. Since the original speech energy level is high in this analysis window, the influence of noise is relatively small. However, this is not always the case since a common speech signal inherently contains both high energy parts and lower energy parts. Therefore, the performance might still be affected.

Figure 53 shows the performance against different noise levels. When the noise energy level increases, the percentage of accuracy fluctuates with slight tendency to decrease. Comparing to the performance acquired from MFCC algorithm, the spectrogram is less sensitive to the additional noise. The reason is because the noise energy is distributed on the entire spectrum, therefore the effect on each frequency

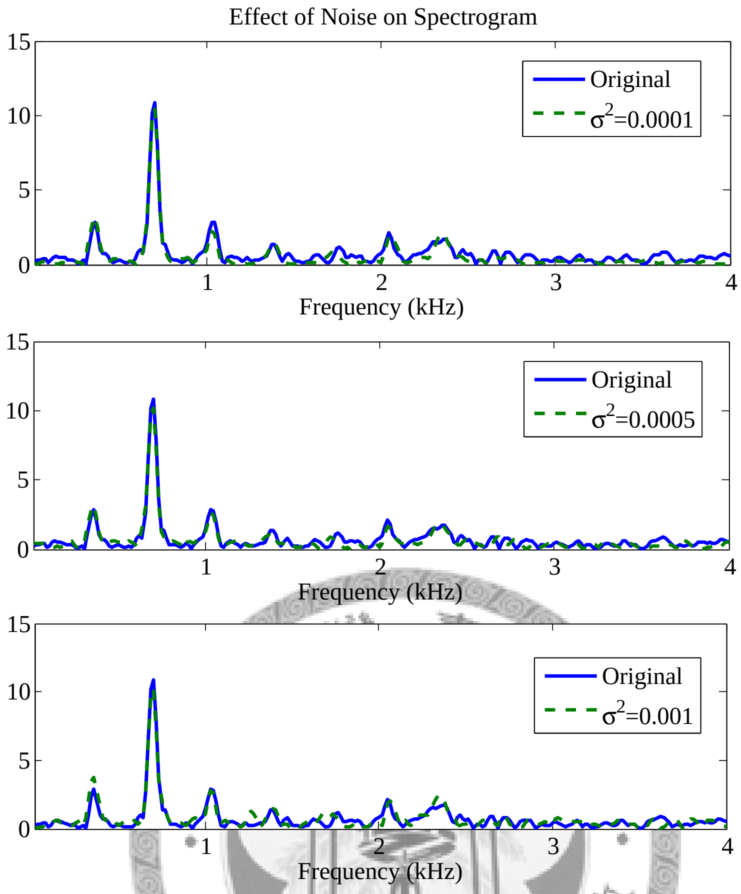

Figure 52: Effect of noise on spectrogram

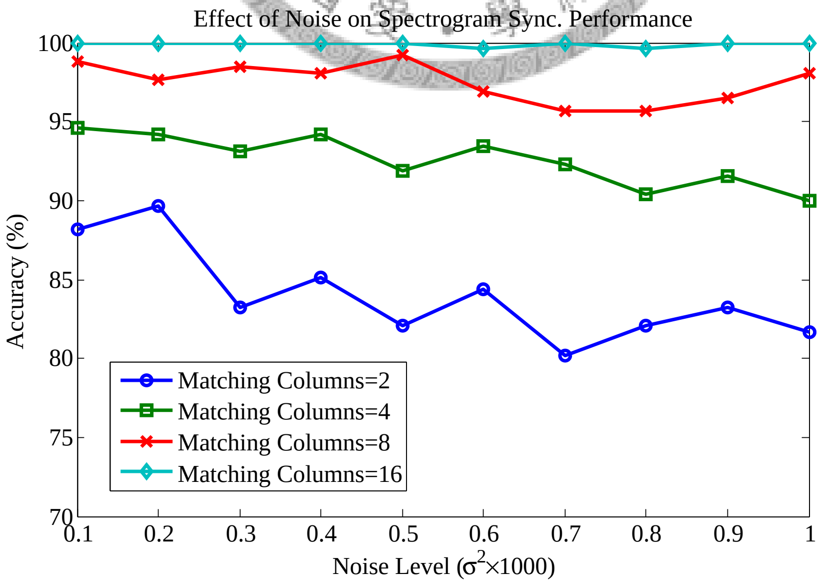

Figure 53: Performance of spectrogram against noise distortion

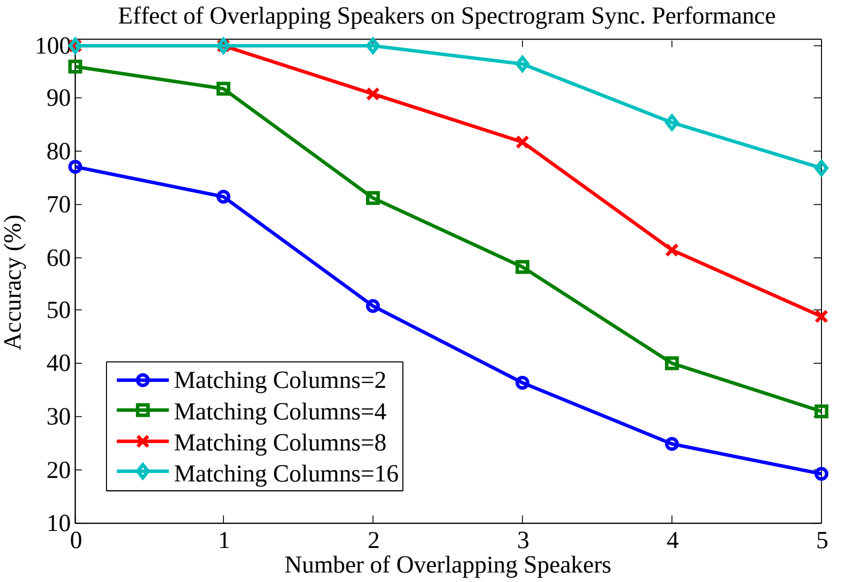

Figure 54: Performance of spectrogram against overlapping speaker

component is less. On the other hand, the process of computing MFCC bins may collect the distributed energy and then cast it to the low MFCC bins, hence the performance is easier to be corrupted.

6.3.4 Overlapping Speakers

Since the motivation of this spectrogram-based synchronization algorithm is to compensate the insufficiency of MFCC-based synchronization against overlapping speaker, this algorithm should outperforms the MFCC-based algorithm. The effect of additional speakers on the spectrogram is discussed in the early parts of this chapter, so we directly put the performance here, as shown in Figure 54.

From Figure 54 we can observe that the percentage of accuracy still drops as the number of interfering speakers increases. This is because more speakers in the mixture may corrupt the sparsity between different speeches, thus affecting the performance. Therefore, too many speakers may exceed the capability of this spectrogram-based algorithm. However, comparing to the performance of MFCC-based algorithm, this spectrogram-based algorithm can usually achieves better accuracy when there are overlapping speakers in the PSTN audio. Additionally, if we only consider less than two interfering speakers, this spectrogram-based algorithm can achieve more than 90% accuracy if 8 or more matching columns are applied.

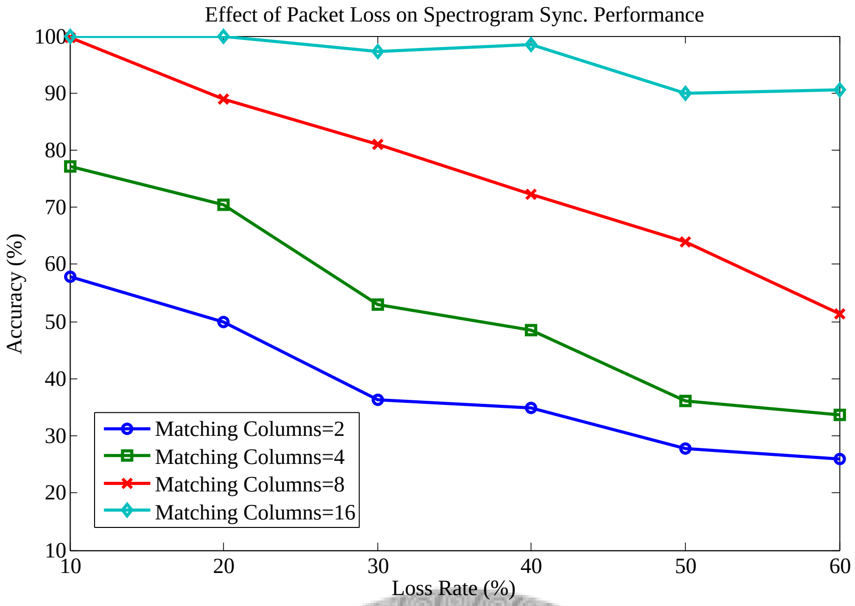

Figure 55: Performance of spectrogram against packet loss

6.3.5 Packet Loss

Since the packet loss may result in different speech waveforms, it also introduces variation to frequency components in the spectrum. Therefore, the resulting spectrum might be different from the original one. Additionally, the adopted packet loss concealment method simply duplicates the previous received packet, regardless of the phase of the waveform. It may result in a discontinuous joint at duplicated packet and thus introduce high frequency components to the spectrogram.

The performance of spectrogram on the packet lost IP audio is shown in Figure 55. The achievable accuracy for a certain loss rate and matching columns is lower than that of MFCC-based algorithm. Besides, even a large number of matching columns is applied, for example 16 columns which is 512ms in time, the performance still can't ensure 100% accuracy for high loss rates. However, a loss rate as high as 50 or 60% is impractical since the concealed audio may be unacceptable for human perception. If we only consider a loss rate less than 20%, 6 matching columns can still ensure 100% accuracy.

In consideration of practical situation, Figure 62 shows the performance of cross correlation when multiple sources of distortions occurs. Since the spectrogram-based algorithm is more vulnerable to packet losses, the performance of distortions including packet loss is worse than the non-packet-lost one. However, the performance is always improved as the number of matching columns increased.

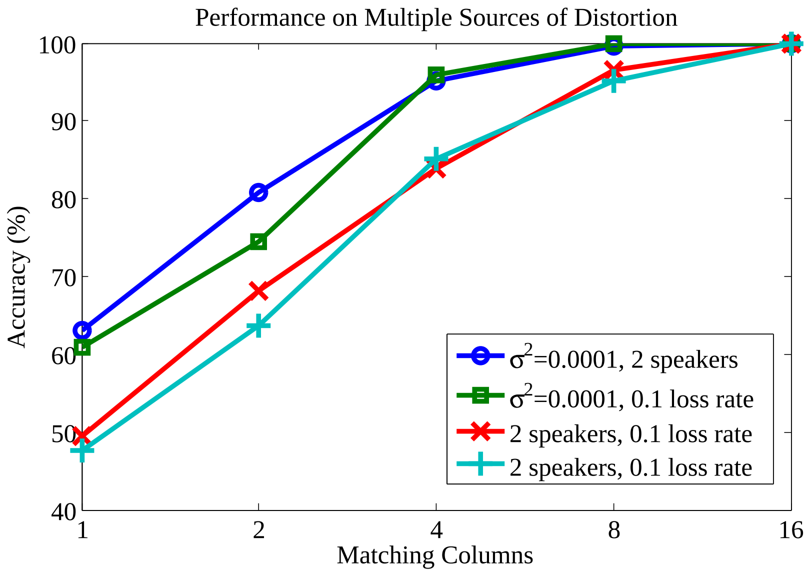

Figure 56: Performance against multiple sources of distortions

6.3.6 Short Conclusion on Performance

In conclusion, directly using spectrogram for comparing the similarity among the matching windows might be affected by waveform distortions since the spectrogram is transformed directly from the waveform. Therefore, since the packet loss concealment algorithm may include additional high frequency distortions to the spectrogram due to the discontinuity of waveform, the spectrogram is affected by packet loss. However, for an AWGN noise, since the noise energy is spread through the entire spectrum, the influence on each frequency component is less, comparing to that of the MFCC bins. On the other hand, for overlapping speakers, this algorithm can usually achieve acceptable performance as long as the sparsity holds which implies that the number of overlapping speakers should be small. Figure 62 reveals the robustness of this spectrogram-based synchronization algorithm, as long as a large enough matching window is applied. Even when multiple sources of distortions are added to the speeches, this algorithm can achieve excellent accuracy.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis, we investigate on the audio/video synchronization challenges for a heterogeneous teleconferencing scenario. Due to the heterogeneity in teleconferencing devices, video conference might only be able to be hold among some of the conferees in the audio conference. Therefore the audio stream and video stream may traverse through different kinds of networks to the receiver. Since the timing relationship may be corrupted by the conference server, synchronization algorithms should be applied to somehow recover this timing information.

7.1 Performance Comparison

We have proposed an audio synchronization framework in chapter 3 to address this synchronization problem in the heterogeneous teleconferencing scenario. Based on the framework, we have proposed three different types of synchronization algorithm and evaluated on the performance of each synchronization algorithm against possible sources of waveform distortion.

7.1.1 Codec Distortion

Figure 57 shows the performance of the afore-mentioned three synchronization algorithms against the codec distortion. It can be observed that the cross-correlation-based synchronization algorithm is highly robust to the voice codec since the general waveform trend is not severely distorted by the voice codec. On the other hand, for the spectrogram-based algorithm, it requires a 256ms matching window to achieve 100% accuracy due to the direct distortion on spectrogram by codecs. As for the MFCC-based algorithm, since the MFCC has extracted the important vocal tract responses as features, it should not be severely affected by the codec.

7.1.2 Noise Distortion

When it comes to noisy speeches which is usually the case in practical environment, as shown in Figure 58, the MFCC-based algorithm seems to be vulnerable to noises. In order to ensure high accuracy, the matching window should be larger than 256ms.

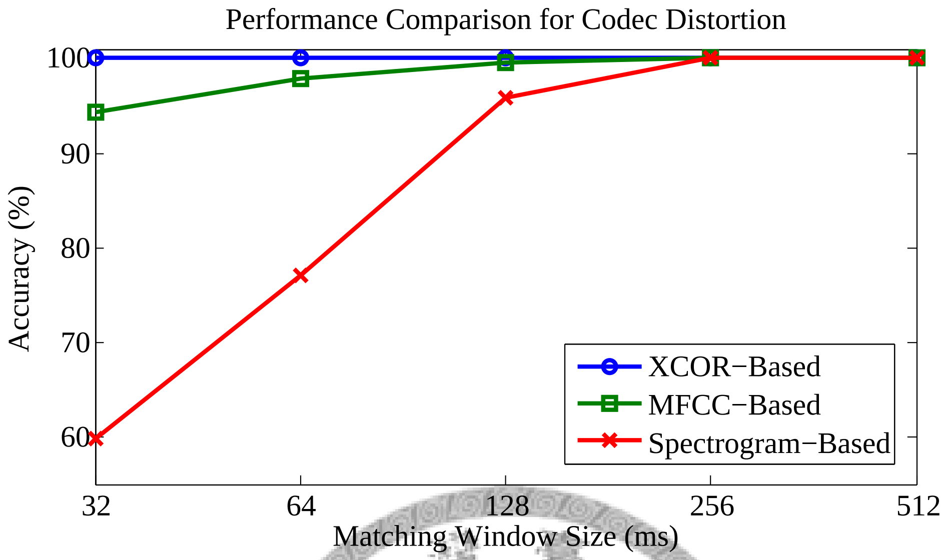

Figure 57: Performance comparison for codec distortion

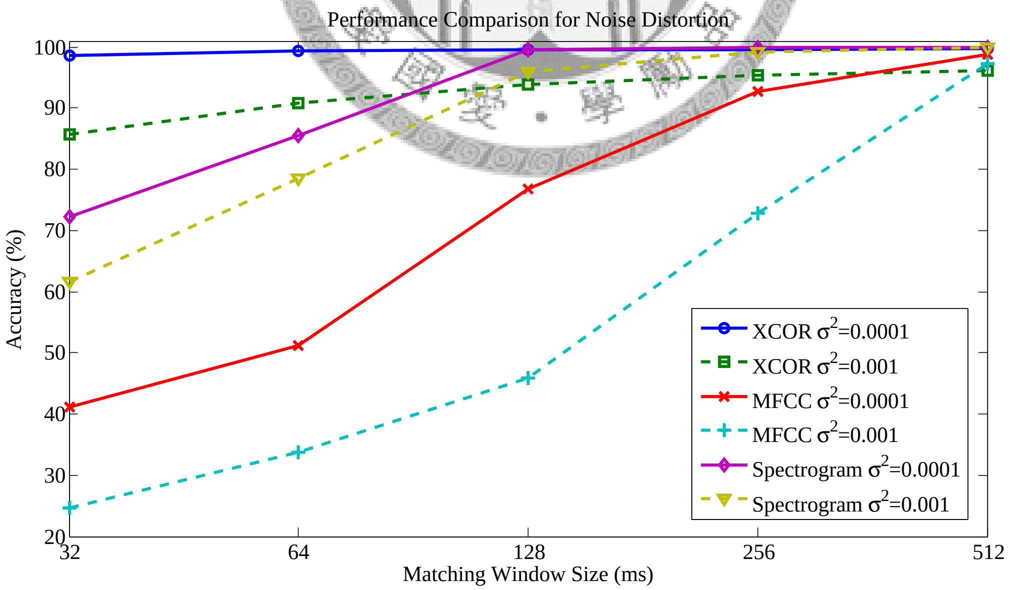

Figure 58: Performance comparison for noise distortion

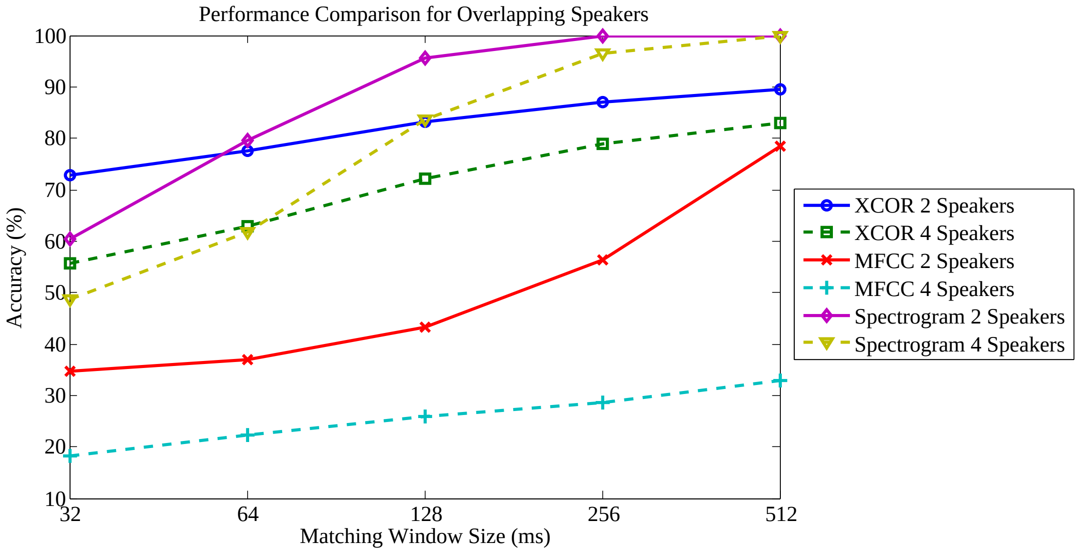

Figure 59: Performance comparison for overlapping speakers

However, the spectrogram-based algorithm, in comparison to the codec distortion, doesn't seem to be affected by this additional noise. This is because the introduced AWGN noise is spread all over the spectrum, therefore the effect on each frequency component is small.

As to the cross correlation algorithm, although the performance is able to achieve more than 85% accuracy even for a 32ms window, enlarging the matching window doesn't do much good on the performance. This is because this algorithm utilize only the time information. When the matching window is enlarged, more noises are included, too.

7.1.3 Overlapping Speakers

Since the spectrogram-based algorithm is designed so as to overcome the effect of overlapping speakers, it can outperform other algorithms for most cases, as shown in Figure 59. Note that the notation "2" speakers includes the speaker of interest, as well as "4" speakers. The performance of spectrogram-based algorithm can achieve larger than 90% accuracy even for 4 speakers if a 256ms window is applied.

On the other hand, while the cross-correlation-based algorithm steadily improves its performance to an acceptable level by applying a larger window, the MFCC-based algorithm is so corrupted that the accuracy is usually less than 75%. Especially when more interfering speakers are included in the PSTN audio, the performance is terrible.

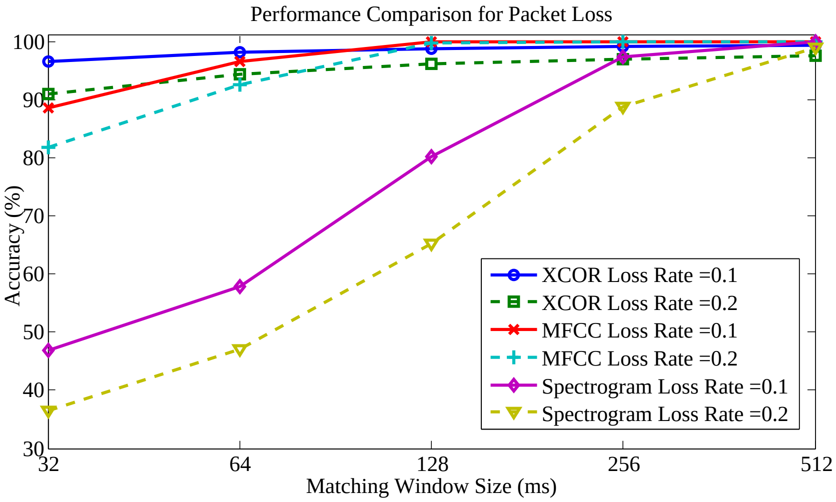

Figure 60: Performance comparison for packet loss

7.1.4 Packet Loss

If the IP audio suffers from packet loss, the spectrogram-based algorithm is severely affected since the loss-concealed audio has modify the waveform and consequently the spectrogram, as shown in Figure 60. However, for the other two algorithms, since the packet-loss concealment algorithm doesn't change much on the waveform trend or the speech feature, they seem to be tolerant to this kind of distortion.

Nevertheless, if at the same time the PSTN audio includes other overlapping speakers, the performance of different algorithms is shown in Figure 61. It can be observed that the cross-correlation-based algorithm now seems to be seriously affected by these two sources of waveform distortion. This has shown the vulnerability of using only time domain information which is easily distorted for synchronization.

Now, if we combine all the above sources of distortions, the performance of each algorithm is shown in Figure 62. We can observe that the spectrogram-based algorithm outperforms the other two. Synchronization algorithm based on cross correlation is limited in its performance by multiple sources of distortions as previously stated. As for the MFCC-based algorithm, since it is vulnerable to overlapping speakers, the performance is the worst among these three. However, we know that whenever there is no overlapping speakers, performance of MFCC is better than that of spectrogram. Generally speaking, although sources of distortion may affect on the performance for all kinds of synchronization algorithms, larger matching window usually seems to be a favorable solution to increase the accuracy of synchronization. However, larger matching window suggests more samples to be compared in each iteration. The increased

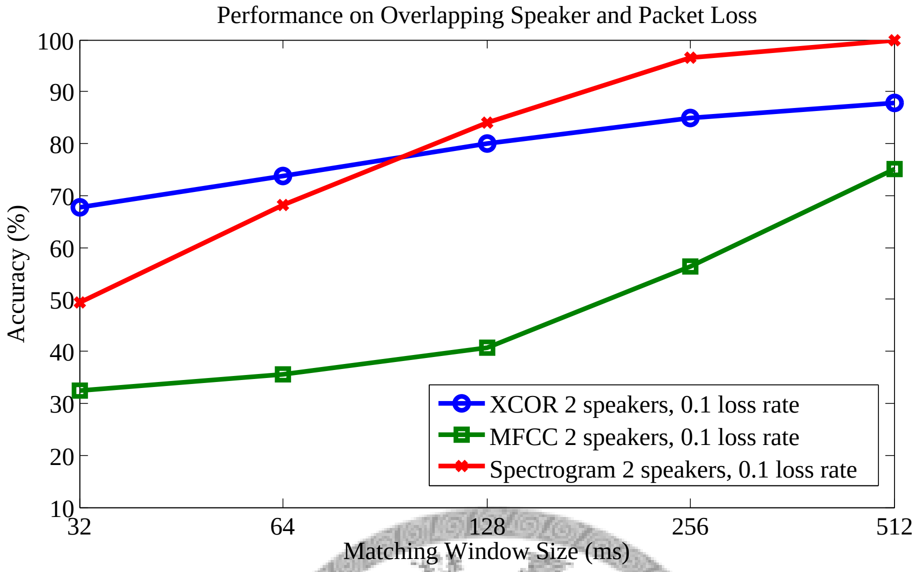

Figure 61: Performance on both overlap and packet loss

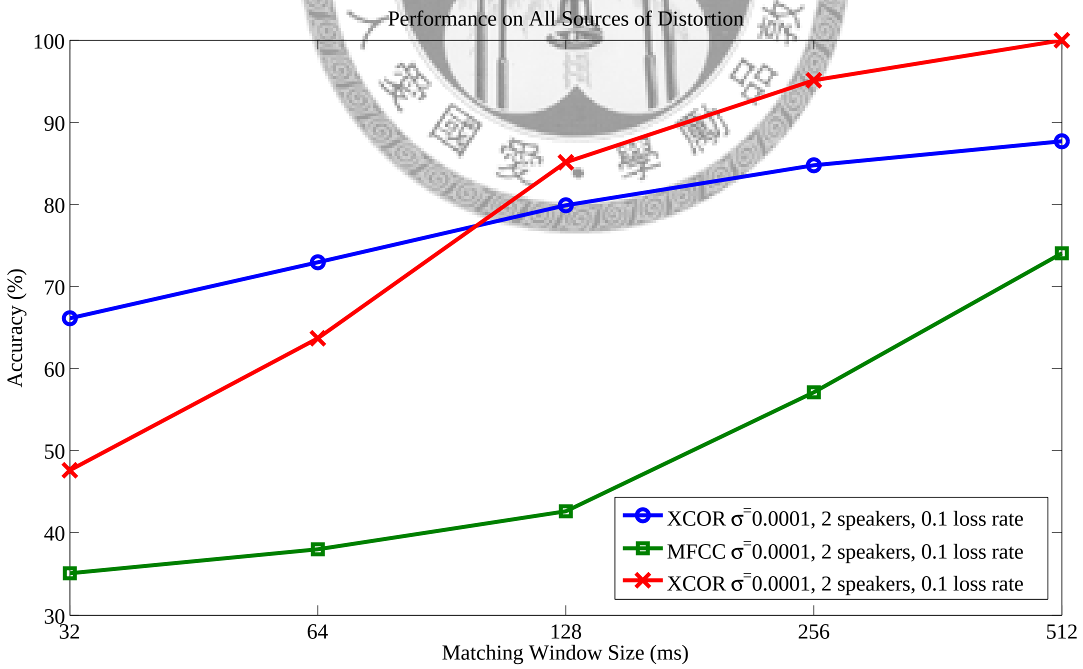

Figure 62: Performance on combined distortions

Table 4: Required computation time of different algorithms

Matching Window Size (ms)	32	64	128	256	512
Cross Correlation	4.4s	5.9s	9.5s	15.6s	31.0s
MFCC	14.7ms	16.2ms	17.8ms	34.3ms	40.9ms
Relative to XCOR	0.334%	0.275%	0.187%	0.220%	0.132%
Spectrogram	19ms	22.8ms	36.7ms	68.4ms	116.5ms
Relative to XCOR	0.432%	0.386%	0.386%	0.438%	0.376%

Table 5: Required computation time of stages in MFCC and spectrogram

Matching Window Size (ms)	32	64	128	256	512
MFCC Conversion	14.6ms	16.0ms	17.6ms	27.8ms	34.1ms
MFCC Comparison	0.1ms	0.1ms	0.2ms	6.5ms	6.8ms
Percentage of Conversion	99.3%	98.8%	98.9%	81.0%	83.4%
Spectrogram Conversion	13.9ms	13.4ms	16.6ms	27.6ms	36.2ms
Spectrogram Comparison	5.2ms	9.4ms	20.1ms	40.7ms	80.4ms
Percentage of Conversion	73.2%	58.8%	55.2%	40.4%	31.1%

computation time for the synchronization module might affect the reactiveness to network dynamics.

Table 4 has listed the required computation time of different synchronization algorithms when matching window of different sizes is applied. The required computation times for MFCC and spectrogram based algorithms are different from that of cross correlation by the order of two because cross correlation spends too much time on full search. Table 5 has revealed that for MFCC-based algorithm, the computation spends most of the time on MFCC conversion. On the other hand, for the spectrogram-based algorithm, the percentage of computation for comparison increases with the matching window size.

The above discussion suggests the trade-off between accuracy and computation time. Figure 63 shows the relationship of computation time and the achieved accuracy in the situation of Figure 61. It can be observed that higher accuracy could be traded from longer computation time. However, for the MFCC and Spectrogram based algorithms, the required computation time is only a few milliseconds, while the computation time for cross-correlation-based algorithm is larger by three orders. Although cross-correlation seems to be robust for most cases, the long computation time may destroy the reactiveness to network dynamics.

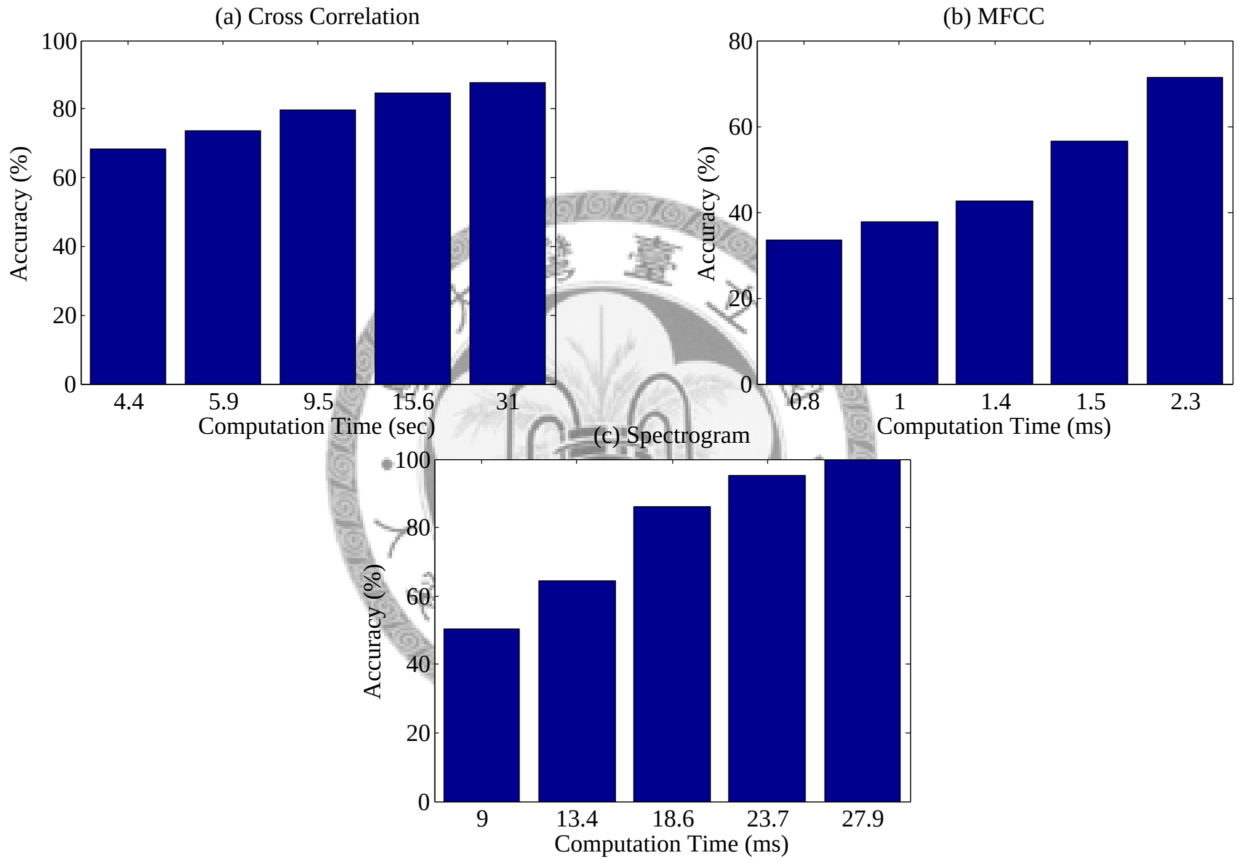

Figure 63: Achievable accuracy for certain computation time

7.2 Conclusions

The evaluation results shows that the time-domain cross correlation reveals an appealing performance for its robustness against single low level distortions. However, when the distortion level increases or multiple sources of distortions are included, the performance is degraded and limited even when larger matching windows are applied. On the other hand, the other two types of synchronization algorithms which are based on the features extracted by the DSP techniques have their own robustness to certain kinds of distortions.

However, these DSP features still have their limitations. The MFCC-based algorithm reveals its robustness against different voice codecs and packet loss, while the spectrogram-based algorithm shows more robustness to noise and overlapping speakers. This complementary relation of MFCC-based and spectrogram-based algorithm might suggest a future direction of optimizing the performance by utilizing both features.

Additionally, since the target scenario may involve multiple users in the video conference, this synchronization algorithm should be applied to all these video conference users one by one. For the latter users to be synchronized, the interferences of the former users to the PSTN audio can be subtracted in advance to achieve higher performance.

In conclusion, to address the proposed synchronization problem, using DSP techniques may be an appealing solution in terms of synchronization efficiency and robustness. As long as the timing information can be accurately acquired by the synchronization module, the PSTN audio can be synchronized with the IP video, according to this information. Since our work mainly focuses on the synchronization framework and the preliminary analysis of synchronization algorithms based on features extracted by DSP techniques, practical implementation and algorithm optimization could be possible future directions.

REFERENCES

[1] H.-Y. Hsieh, C.-W. Li, S.-W. Liao, Y.-W. Chen, T.-L. Tsai, and H.-P. Lin, "Moving toward end-to-end support for handoffs across heterogeneous telephony systems on dual-mode mobile devices," *Elsevier Computer Communications, Special Issue on End-to-End Support over Heterogeneous Wired-Wireless Networks, article in press*, April 2007.

[2] J. Grudin, E. Steven, and Poltrock, "Videoconferencing: Recent experiments andreassessment," *in Proceedings of the 38th Annual Hawaii International Conference on HICSS '05*, pp. 3–6, January 2005.

[3] C.-W. Lin, Y.-C. Chen, and M.-T. Sun, "Dynamic region of interest transcoding for multipoint video conferencing," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 982– 992, Octobor 2003.

[4] Akkus, I. Civanlar, and M. Ozkasap, "Peer-to-peer multipoint video conferencing using layered video," *Signal Processing and Communications Applications*, pp. 1–4, April 2006.

[5] VIBO Telecom Inc., "Taiwan 3g- vibo telecom." Online Available at: http://www.compression-links.info/

[6] Cutler, Ross G. (Duvall, WA, US) and Bridgewater, Alan L. (Issaquah, WA, US), "Audio/video synchronization using audio hashing," no. 20060291478, December 2006. Online Available at: http://www.freepatentsonline.com/20060291478.html

[7] Octave Communications, "Octave communications." Online Available at: http://www.octave.in/menu.htm

[8] S. E. Poltrock and J. Grudin, "Videoconferencing: Recent experiments and reassessment," *IEEE*, 2005.

[9] J. Huang, W.-C. Feng, J. Walpole, and W. Jouve, "An experimental analysis of dct-based approaches for fine-grain multi-resolution video," *Multimedia Systems*, pp. 513–531, January 2006.

[10] K.-T. Fung, Y.-L. Chan, and W.-C. Siu, "Low-complexity and high-quality frame-skipping transcoder for continuous presence multipoint video conferencing," *IEEE Transactions on Multimedia*, no. 1, pp. 31–46, February 2004.

[11] C.-W. Lin, Y.-C. Chen, and M.-T. Sun, "Dynamic region of interest transcoding for multipoint video conferencing," *IEEE Transactions on Circuits and Systems for Video Technology*, no. 10, pp. 982–992, October 2003.

[12] M. Chen, G.-M. Su, and M. Wu, "Robust distributed multi-point video conferencing over error-prone channels," *2006 IEEE International Conference on Multimedia and Expo*, pp. 1149–1152, July 2006.

[13] M. R. Civanlar, O. Ozkasap, and T. Celebi, "Peer-to-peer multipoint videoconferencing on the internet," *Signal Processing: Image Communication*, pp. 743–754, May 2005.

[14] I. E. Akkus, M. R. Civanlar, and O. Ozkasap, "Peer-to-peer multipoint video conferencing using layered video," *Signal Processing and Communications Applications, 2006 IEEE 14th*, pp. 1–4, Signal Processing and Communications Applications, 2006 IEEE 14th 2006.

[15] MpegTV, "Mpeg.org." Online Available at: http://www.mpeg.org/

[16] C. Liu, Y. Xie, and M. J. Lee, "Multipoint multimedia teleconference system with adaptive synchronization," *IEEE JSAC*, pp. 1422–1435, September 1996.

[17] Y. Xie, C. Liu, M. J. Lee, and Y. N. Saadawi, "Adaptive multimedia synchronization in a teleconference system," *ACM/Springer Multimedia Systems*, no. 4, pp. 326–337, 1999.

[18] H. Liu and M. E. Zarki, "A synchronization control scheme for real-time streaming multimedia applications," *in Proceedings of 13th Packet Video Workshop*, April 2003.

[19] ——, "On the adaptive delay and synchronization control for video conferencing over the internet," *in Proceedings of ICN 2004*, March 2004.

[20] ——, "An adaptive delay and synchronization control scheme for wi-fi based audio/video conferencing," *Springer Science + Business Media*, pp. 511–522, May 2006.

[21] C.-C. Kuo, M.-S. Chen, and J.-C. Chen, "An adaptive transmission scheme for audio and video synchronization based on real-time transport protocol," *IEEE International Conference on Multimedia and Expo*, pp. 525–528, 2001.

[22] C. Kim, K. deok Seo, W. Sung, and S. heung Jung, "Efficient audio/video synchronization method for video telephony system in consumer cellular phones," *ICCE '06 Consumer Electronics*, pp. 137–138, January 2006.

[23] M. Yang, N. Bourbakis, Z. Chen, and M. Trifas, "An efficient audio-video synchronization methodology," *IEEE International Conference on Multimedia and Expo*, pp. 767–770, July 2007.

[24] WIKIPEDIA, "Lip sync." Online Available at: {http://en.wikipedia.org/wiki/Lip_sync}

[25] G. Zoric and I. S. Pandzic, "A real-time lip sync system using a genetic algorithm for automatic neural network configuration," *IEEE International Conference on Multimedia and Expo*, pp. 1366–1369, July 2005.

[26] ——, "Automatic lip sync and its use in the new multimedia services for mobile devices," *in Proceedings of the 8th International Conference on Telecommunications, 2005*, pp. 353–358, June 2005.

[27] W.-N. Lie and H.-C. Hsieh, "Lips detection by morphological image processing," *in Proceedings of ICSP '98*, pp. 1084–1087, 1998.

[28] D. F. McAllister, R. D. Rodman, D. L. Bitzer, and A. S. Freeman, "Lip synchronization of speech," 1998.

[29] GoldWave Inc., "Audio editing software- GoldWave." Online Available at: http://www.goldwave.com/

[30] D. L. Mills, "Network time protocol (version 3) specification, implementation, and analysis," *RFC 1305*, 1992.

[31] S.-M. Jun, D.-H. Yu, Y.-H. Kim, and S.-Y. Seong, "A time synchronization method for ntp," in *RTCSA '99: Proceedings of the Sixth International Conference on Real-Time Computing Systems and Applications.* Washington, DC, USA: IEEE Computer Society, 1999, pp. 466–473.

[32] R. Steinmetz, "Human perception for jitter and media synchronization," *IEEE Journal on Selected Areas in Comm.*, no. 1, pp. 61–72, January 1996.

[33] H. Manhaiem, R. Silon, G. Fartuk, and S. Refael, *Packet Loss Concealment Techniques and Algorithms.*

[34] The MathWorks, Inc, "The MathWorks- MATLAB and Simulink for technical computing." Online Available at: http://www.mathworks.com/

[35] H. Boril and P. Pollak, "Direct time domain fundamental frequency estimation of speech in noisy conditions," *European Signal Processing Conference 2004*, pp. 1003–1006, September 2004.

[36] T. Shimamura and H. Kobayashi, "Weighted autocorrelation for pitch extraction of noisy speech," *IEEE Transactions on Speech and Audio Processing,*, pp. 727 – 730, Octobor 2001.

[37] B. V. K. Kumar, A. Mahalanobis, and R. D. Juday, *Correlation Pattern Recognition.* Cambridge University Press, 2005.

[38] VoiceAge Corporation, "Open amr initiative- amr codec." Online Available at: www.voiceage.com

[39] YUVSoft Corp. and Graphics and Media Lab, "Ultimate compression resources catalog." Online Available at: http://www.compression-links.info/

[40] DSPRelated.com, "MATLAB code- Mel Frequency Cepstral Coefficients." Online Available at: http://dsprelated.com/

[41] A. M. Reddy and B. Raj, "Soft mask methods for single-channel speaker separation," *IEEE Trans. Audio, Speech, Lang. Process.*, no. 6, pp. 1766–1776, August 2007.

[42] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, no. 1, pp. 1–12, January 2007.

[43] B. Raj and P. Smaragdis, "Latent variable decomposition of spectrograms for single channel speaker separation," *IEEE Workshop on Applications of Signal Process. to Audio and Acoustics*, October 2005.

[44] D. Chazan, Y. Stettiner, and D. Malah, "Optimal multi-pitch estimation using the em algorithm for co-channel speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, 1993.

[45] D. P. Morgan, E. B. George, L. T. Lee, and S. M. Kay, "Cochannel speaker separation by harmonic enhancement and suppression," *IEEE Trans. Audio, Speech, Lang. Process.*, no. 5, pp. 407–424, September 1997.

[46] D.-L. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications.* Hoboken, New Jersey: John Wiley and Sons, 2006.

[47] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 5th ed. San Diego, CA: Academic Press, 2003.

[48] A. Jourjine, S. Richard, and O. Yilmaz, "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*, pp. 2985–2988, June 2000.

[49] S. T. Roweis, "Factorial models and refiltering for speech separation and denoising," *In Proceedings of Eurospeech*, pp. 1009–1012, September 2003.

[50] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, no. 7, pp. 1830–1847, JULY 2004.

[51] Z. Shan, J. Swary, and S. Aviyente, "Underdetermined source separation in the time-frequency domain," *ICASSP*, pp. 945–948, 2007.

[52] A. Aissa-El-Bey, K. Abed-Meraim, and Y. Grenier, "Underdetermined blind separation of audio sources from the time-frequency representation of their convolutive mixtures," *ICASSP*, pp. 153–156, September 2007.

[53] L. T. Nguyen, A. Belouchrani, K. Abed-Meraim, and B. Boashash, "Separating more sources than sensors using time-frequency distributions," *ISSPA*, pp. 583–586, August 2001.

[54] S. Rickard, R. Balan, and J. Rosca, "Real-time time-frequency based blind source separation," pp. 651–656, May 2001. Online Available at: http://citeseer.ist.psu.edu/rickard01realtime.html

[55] P. Bofill and M. Zibulevsky, "Blind separation of more sources than mixtures using sparsity of their short-time fourier transform," *in Proc. Int. Workshop Independent Component Anal. Blind Source Separation*, pp. 87–92, June 2000, helsinki, Finland.

[56] S. Rickard and O. Yilmaz, "On the approximate W-Disjoint Orthogonality of speech," *ICASSP*, pp. 13–17, May 2002.

[57] M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, and Y. Kaneda, "Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones," *Acoust. Sci. Technol.*, no. 2, pp. 149–157, February 2001.