Auditory Representation Using Matching Pursuit and Sparse Coding for Speech Fingerprint

TRAN KIM DUNG

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

Auditory Representation Using Matching Pursuit and Sparse Coding for Speech Fingerprint

TRAN KIM DUNG

Supervisor: Professor UNOKI MASASHI

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science

September 2022

Abstract

Speech is one of the natural ways human beings communicate; however, natural speech is impossible for long-distance communication. Nowadays, global real-time communication over the internet has become a vital part of modern civilization. The internet is not only a telecommunication technology, it has become a world. Consequently, the risks related to speech in the virtual world become more complicated than just wiring or eavesdropping telephone lines. Actions should be taken to ensure that we do not become victims of this world. We need to deal with security issues when speech is used as a tool to control automated systems; spoofing and privacy when speech is used in communication; and properties protection and management when speech is used as commercial products.

One way to deal with these issues is using speech fingerprint. Speech signals are believed to convey unique features that can be used as a biometric security measure along with iris, fingerprint, and facial recognitions. The purpose of a speech fingerprint technique is extracting distinguishable features related to speaker individuality and linguistic content from speech signals and combining the features to create unique speech fingerprints. The speech fingerprints can then be used for speaker verification in security, anti-spoofing in communication, and properties protection of commercial products.

Current speech fingerprint techniques produce speech fingerprints in three basic steps. In step one, speech signals in time domain are projected into a time-frequency domain. In step two, patterns analysis methods are used to obtain important spatiotemporal features. In step three, hashing methods are used to combine the obtained features to create speech fingerprints. Step one is a challenging issue in speech coding. Various speech coding techniques and speech representation models have been proposed such as spectrograms using Fourier and wavelet transforms, auditory spectrograms using auditory filterbanks, spikegrams using matching pursuit algorithms, and auditory sparse representations using perceptual matching pursuit algorithms. The purpose of step one is emphasizing important acoustical features on a representation model. Step two is another challenging issue in pattern recognition. Speech signals are the natural carrier of information about speaker individuality, language, emotion, and so on. Obtaining unique and distinguishable features for speech fingerprints is the purpose of step two. Step three is also a challenging issue. Reducing the dimensions of features causes loss of information and thus, speech fingerprints become less distinguishable. Keeping high dimensions of features causes problems in storage, transferring, and searching. The purpose of step three is producing speech fingerprints that are accurate in matching and convenient to use.

Although the current speech fingerprint techniques in the literature can achieve high performance in various application. However, there is critical drawback is that they are driven by practical results; thus, actual speech fingerprints are either lesser important or not the main focus in their applications. The present study assumes that speech fingerprints are highly related to speaker individuality, and they are a part of the neural activity patterns of the auditory nerves. Based on this assumption, the purpose of the present study is extracting biologically accurate speech fingerprints. In pursuing this purpose, the first task aims to mimic the neural activity patterns to obtain speech fingerprints. Then in the second task, the uniqueness of the proposed speech fingerprints is verified. Finally in the third task, a speech fingerprint identification method is used to apply speech fingerprints in practical applications.

Emphasizing significant features of a speech signal in a signal representation model, e.g., spectrogram, spikegram, and auditory representation, is an essential task of a speech fingerprint technique. Previous studies have revealed that by mimicking the neural activity patterns (NAP) of the auditory periphery to obtain perceptual features of speech signals, the resultant auditory representation is beneficial to speech-coding and pattern-analysis applications in comparison with spectrogram and spikegram representations. This study proposes to use auditory representations in the process of creating speech fingerprints.

Many efforts have been spent on applying psychoacoustics to concentrate perceptual features on auditory representations to mimic the neural activity patterns generated by the auditory periphery to reproduce the amazing abilities of our hearing system. However, several limitations—using the Bark scale and gammatone basis—remain in the methods used for creating auditory representations. This study found that by mimicking: (1) the sparseness of NAP with a sparse coding technique, i.e., a matching pursuit algorithm, (2) the characteristic frequency of basilar membrane motion with an equivalent rectangular bandwidth scale, (3) the impulse response measured at the basilar membrane with a gammachirp function, and (4) auditory masking with a masking model, perceptual features in auditory representations could achieve similar perceptual evaluation scores, e.g., PEMP-Q and PESQ, while requiring the lowest number of non-zero elements in comparison with features in spectrograms and spikegrams.

Our hearing system has the ability to identify who is speaking, understand spoken language, recognize emotions, etc. simultaneously in very noisy conditions. This miracle is still a mystery to science. Contemporary knowledge divides our hearing system into the auditory periphery and the central auditory cortex. The auditory periphery is responsible for converting speech as sound pressures into NAPs at the auditory nerve and the central auditory cortex is responsible for cognitive functions. At the present time, due to the lack of equipment to obtain the real NAPs at the auditory nerve. Therefore, current speech analysis techniques can only be evaluated by using perceptual evaluation scores and pattern analysis methods. Because of these reasons, in the second task, the present study hypothesizes that there must be unique patterns that help the central auditory cortex identify who is speaking. Therefore, a landmark-based pattern analysis technique is used to combine the features on auditory representations. This technique is used to create a graph-like structure of perceptual features to mimic the neural activity patterns. Then, a uint32 function is used to convert the perceptual structures into hash sequences for fast indexing. Experimental results show that the perceptual structures of auditory representations are the most effective in identifying speakers.

In the last task, a deep hashing technique is used as a speech fingerprint identification algorithm. At first, the proposed speech fingerprint method is used to extract speech fingerprints from speech signals. Then, the extracted speech fingerprints are used as input features of a supervised deep learning algorithm. Then, the deep learning algorithm converted the speech fingerprints into binary hash sequences in the Hamming space. Finally, speaker identification and retrieval experiments are conducted to evaluate the effectiveness of the speech fingerprints and the identification algorithm. Experimental results show that the proposed method can achieve very high identification performance that is competitive to other contemporary state-of-the-art methods.

Keywords: auditory filterbank, gammatone/gammachirp, masking effect, perceptual features, spikegram

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Professor Unoki Masashi, my supervision. This research could not succeed without his patient supervision. Pursuit the Ph.D. is such a long journey. Along the journey, many new challenges happen almost every day. There are some days of exhaustion, burnout, and suffering. However, "giving up" has never been in our spirit, students in Unoki-lab. As I was trained to be an engineer before becoming a Ph.D. student, scientific research is, however, no longer an engineering project, so my paradigm could not convey good scientific research. Professor Unoki persistently motivates and encourages me to morph my view as *a researcher*. He is an expert in broad areas of research. His advice is fruitful when I was struggling. Needless to say, I have learned many things from him and will never forget anything. For example, effectiveness and good results are important; however, the intermediate stages during the process, research strategy, basic principle, and philosophy are much more important to understand/solve scientific issues; even negative results also need thorough consideration, he taught.

Secondly, I appreciate Professor Akagi Masato, my second supervisor, who always gives brilliant advice and consideration throughout my study. I would also like to express my sincere gratitude to Professor Okada Shogo, my minor research supervisor. Professor Akagi and Professor Okada always give me very kind suggestions. In addition, I also would like to extend my appreciation to Professor Dang Jianwu, Professor Yoshitaka Atsuo, Professor Lu Xugang, and Professor Sakti Sakriani for reading my dissertation, listening to my presentation, and giving constructive questions as well as useful comments.

Furthermore, many thanks go to my colleagues in the acoustic information science laboratory, especially Suradej Duangpummet and Kasorn Galajit. They make my JAIST's life to be enjoyable.

Contents

List of Figures

List of Tables

Abbreviations

ASR Auditory Sparse Representation

CSQ Central Similarity Quantization

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

DNA Deoxyribonucleic Acid

DS Damped Sinusoid

ERB Equivalent Rectangular Bandwidth

FFT Fast Fourier Transform

FT Fourier Transform

GB Gabor

GC Gammachirp

GT Gammatone

LPCCs Linear Prediction Cepstral Coefficients

LPCs Linear Prediction Coefficients

MFCCs Mel-Frequency Cepstral Coefficients

MP Matching Pursuit

NAPs Neural Activity Patterns

OMP Orthogonal Matching Pursuit

PCA Principal Component Analysis

PEMO-Q Perceptual Model Quality

PESQ Perceptual Evaluation of Speech Quality

PMP Perceptual Matching Pursuit

SNR Signal to Noise Ratio

STFT Short-time Fourier Transform

Chapter 1

Introduction

1.1 Speech Fingerprints

In the literature, speech fingerprints are generally regarded as binary sequences that can be used to identify music or speech signals. Various techniques have been proposed and they technically comprise of three parts: speech analysis, classifier, and hash function. The speech analysis part includes but not limited to expanding a speech signal into the time-frequency domain, obtaining cepstral coefficients, noise reduction, and feature enhancement, etc. The classifier part includes statistical analysis such as landmark-based pattern analysis and deep learning methods. The hash function part is used to produce binary sequences that includes a *sign* function, thresholding, and *max* pooling, etc.

In the present study, there is a different oppinion about the concept of speech fingerprints. There are many other types of features such as Mel-Frequency Cepstral Coefficients (MFCCs) and Linear Prediction Cepstral Coefficients (LPCCs) that can be used for the purpose of creating speech fingerprints and then for speaker verification. However, the chain of analyses applied to speech signals to obtain speech fingerprints is not related to human perception. As a result, the resultant speech fingerprints can only be regarded as unique identifiers of the speech signals, not as distinctive identifiers of the persons who produced the speech signals. If speech fingerprints are used as a bio security metric, they should be based on our physical bodies. Figure 1.1 shows examples of several unique body parts and their corresponding analysis techniques. It has been discovered that there are unique patterns in our Deoxyribonucleic Acid (DNA), irises, and retinal blood vessels. Measurements and algorithms have been design to target these unique patterns and use them in the process of identity verification. Therefore, an identity verification process based on speech should also take into account the biophysical properties of our bodies. A speech fingerprint is expected (in this study) to be the unique identifier of a person that is obtained from his/her speech. In security, a speech fingerprint can be used to identify a person along with other unique identifier such as fingerprints and facial patterns. With careful calculations, the unique body parts can be projected into distinctive mathematical objects in the linear space; thus, each person can have a virtual self which is as unique as the physical one.

During speech production process, speaker individualities are produced by our glottal sources and vocal tracts; and because we believe that speaker individualities are unique to each person, there exists a one-to-one mapping between the set of all people and the set of their speaker individualities. During hearing process, speaker individualities become an intrinsic part of the Neural Activity Patterns (NAPs) (see Appendix A.2 for

a brief description about NAPs); and it is assumed in the present study that there also exists a one-to-one mapping between the set of all speaker individualities and the set of all perceived speaker individualities. Thus, the unique NAPs associated with speaker individualities are speech fingerprints. In other words, each data point in the distribution of the perceived speaker individualities is a speech fingerprint. Based on this assumption and the belief that speaker individuality is unique, speech fingerprints can be one of the unique traits that can be used for identity verification.

In the present study, it is belived that a speech fingerprint is a tangible thing and speaker verification is a task. And maybe the whole effort of calculating speech fingerprints is for the sole purpose of speaker verification. However, assuming that we have a complete proof about the uniqueness of a speech fingerprint, when we can confidently state that there shall not be two humans possessing the same speech fingerprint; speech fingerprints should be recommended over other features for speaker verification. If speech fingerprints are used as a bio security metric, they should be based on our physical bodies.

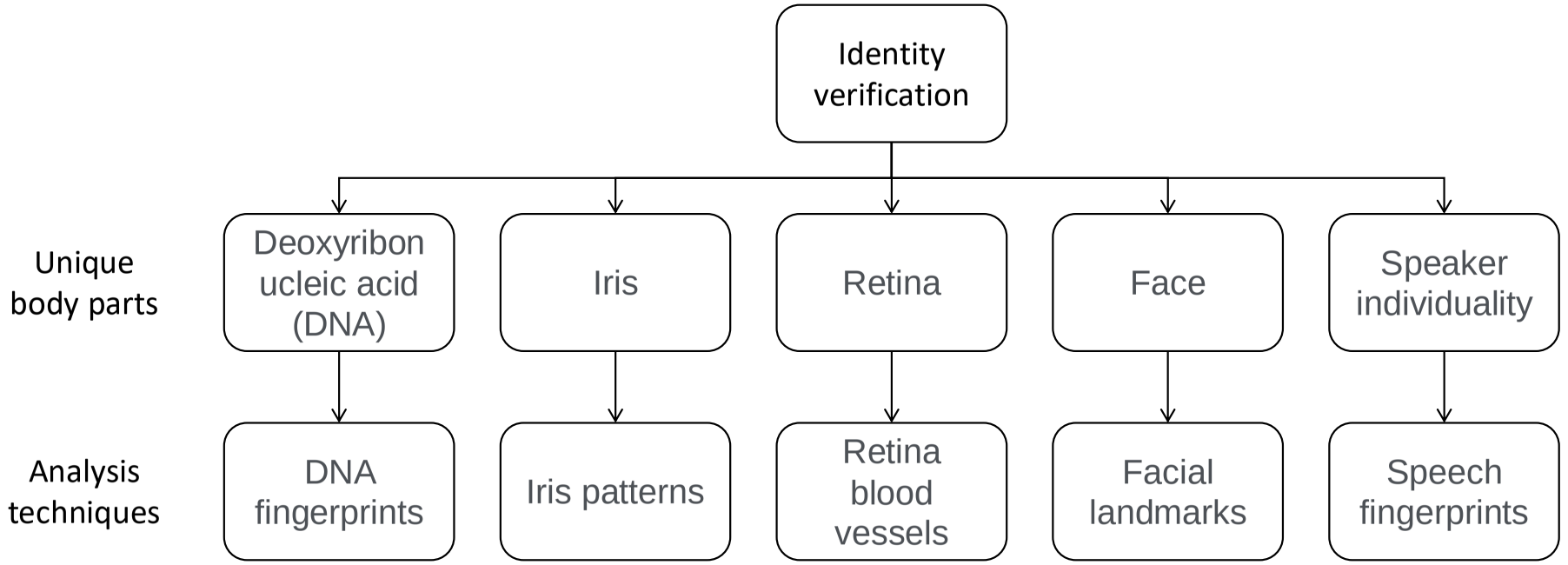

Figure 1.1: Unique body parts and their corresponding analysis techniques used in identity verification.

1.2 Significance

The application of speech fingerprints is speaker verification (identity verification using human speech) and other closely related tasks such as speaker recognition, identification, and retrieval. Comparing to many other methods used for speaker verification, the advantage of speech fingerprint is faster indexing time because comparing two binary strings should be faster than other real features such as i-/x-/d-vectors. The advantage of speech fingerprint should be significant in the near future when we conquer large dataset such as Voxceleb2 and start working with larger ones containing tens of thousands of labels and billions of speech utterances. Furthermore, when we have to deploy speaker verification system on small devices, speech fingerprint system should have an edge on computational capabilities and power consumptions.

Identity verification can be accomplished by using one of the unique traits of humans such as DNA, iris, retina, fingerprint, face, and speech. Although DNA analysis is by far the most accurate verification technique, it requires a lot of time, specialized equipment, expert knowledge, and a piece of us to perform an analysis. Thus, DNA analysis is neither convenient nor suitable for flash checks or daily applications. Other types of identity verification are iris, retina, and fingerprint analyses. These types of verification are undoubtedly more convenient than DNA analysis. Although these analyses are supported by complete proof of uniqueness, can be used for real time verification, and do not require a piece of us; they still depend on specialized cameras or scanners to perform the verification tasks and cannot be used for continuously verifying of identity. Examples of these disadvantages are holding online examinations or important meetings, in which, the claimed identities must be verified continuously from start to stop. In these situations, one-time verification is insufficient to ensure the continuous participations of the claimed identities. Perhaps face and speech analyses are competitively the most convenient techniques used for continuous, real-time, and daily verification applications. The advantages of these techniques have become more evident to-day due to the abundant of cameras and microphones on mobile devices such as smart phones, webcams, and headsets, etc. The ubiquity of human speech on the internet has made identity verification using speech fingerprints a must for security.

Our bodies are composed of flesh and bone, and to overcome this weakness, we build tools to enhance our abilities and enable new possibilities. At the present time, technologies have reduced some limitation of physical distances, our activities have grown from within local villages to global collaborations and soon will be multiplanetary colonies. Therefore, communicating over the virtual world has become a more reasonable solution than traversing long distances. Consequently, digital identity has become evidently necessary. Our faces and fingerprints—two of the unique body parts—have been used in identification documents such as passports, driving license, and identification cards, etc. Subsequently, by projecting our unique body parts into digital identities and using them in identity verification processes, we can increase the effectiveness of our global activities and open new opportunities.

In cyber physical systems, speech is used as a tool to control other systems, as a way to communicate with real and virtual people, and as properties. Therefore, security, content management, and digital properties protection are important issues. Furthermore, speech synthesis and voice conversion systems have been growing rapidly; thus, a counter measure is necessary to ensure security for systems that are dependent on speech.

An example about using speech as a tool to control other automated systems is Amazon

Alexa. Amazon Alexa is a speech analysis and synthesis system that allows users to use natural speech to make purchases, play music, and check delivery status, etc. In a recently published article [2], researchers reported that Alexa was available in more than 100 million households worldwide and in both Amazon and third-party products. Apart from the convenience brought to us by the speech system, the article reported that there were various ways to attack the speech system on both front-end devices and cloud-based back-end processing. Nevertheless, Alexa is just one speech system, there are numerous other IoT devices in our smart environments and much more sensitive areas such as banking, military, and politics, etc. Therefore, security is utmost important if we are about to use speech to control our information systems.

Automatic speaker identification, verification, and retrieval applications are essential parts of our information systems. However, active research in recording devices, speech enhancement, text-to-speech, voice conversions, etc. is rapidly increasing the risks of using such technologies to spoof human speech and to bypass security systems. Thus, speech spoofing has become a complicated security issue. A recent community-led challenge—ASVspoof—has been organized to promote the development of countermeasures to speech spoofing [3]. The logical access dataset released by the challenge was created by using 17 types of text-to-speech engines and voice conversion techniques. Totally, the dataset contains 19 types of spoofed speech. It means that the tools that can be used to fake someone speech are easy and many. Therefore, spoofing countermeasures are needed; otherwise, speech will become too dangerous to use.

1.3 Challenges

Researchers have believed that speaker individuality is unique to each person that is produced by the glottal source and vocal tract, carried by a speech signal, and perceived by the hearing system. However, most speech verification methods utilize traditional signal processing techniques and optimization algorithms rather than incorporating knowledge about biology and physiology into technologies. Very often, the attention of the proposed methods is shifted to verification results rather than to the faith in speaker individuality.

Identity verification using fingerprints is good example to explain this problem. We have complete proof to be confident that the minutiae of a fingerprint are what make the fingerprint unique. Therefore, we design algorithms to target the uniqueness, the minutiae, of the fingerprint instead of using Principal Component Analysis (PCA).

In the same manner, identity verification using speech fingerprints is a challenging problem. We have complete faith to be confident that the speaker individuality of a speaker is what make the speaker unique. Therefore, we should design algorithms to target the uniqueness, the speaker individuality, of the speaker instead of using MFCCs.

One way to mitigate the issues is using speech fingerprints. Different researchers have proposed different methods used for creating speech fingerprints. Regardless of the intricacy of the fingerprinting systems, their designs appear to have three basic steps [4]. In the first step, speech signals in time domain are usually transformed into time-frequencies domain. The next step is designing a suitable feature extraction method so as to collect unique features presented in the speech signals. The final step is compressing these unique features to create the desired fingerprints.

Three steps of a speech fingerprint technique do not seem to be many but each is a challenging research field and they have to support each other to form a thorough solution. Speech signal representation can be generalized into four categories that are

spectrogram, auditory spectrogram, sparse representation, and auditory sparse representation. Each type of coding has its unique characteristics and provides specific features on its representation model and the choices are left for the designers to decide depending on their ideas. Extracting relevent features from the representation models is a matter of pattern recognition problems. How feature extraction methods are designed relied heavily on subjective intentions and the ultimate goal is to obtain distinctive information that can help identify speech signals. The hashing techniques that are used to create speech fingerprints can be as simple as applying a hash function to generate hash chunks or complicated as converting the set of features from step two into another set of features. No matter how sophisticated they are, the final fingerprints are bounded to some requirements such as distance metrics, storage size, and indexing speed.

Ellis *et al.* [5] created an application used for audio fingerprints identification. Based on his work, an experiment was conducted to verify the effectiveness of applying an audio fingerprinting technique to speech. A data set consisting of eleven speech signals was created by a speaker speaking /Hello/ eleven times. Ten speech signals were used to create a database of speech fingerprints using this application. The other speech signal was used for the identification task. Testing results showed that the performance of this application was unreliable in speech although it worked very well in identifying music.

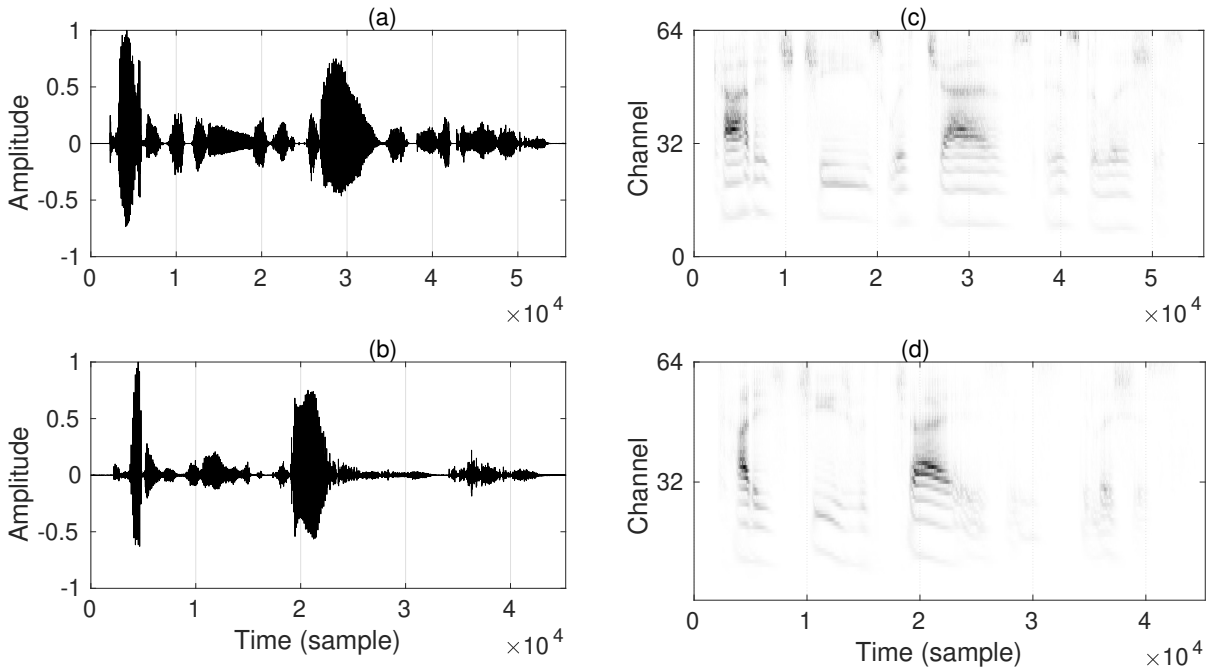

Figure 1.2: Example of signal shifting and phoneme scaling problems. Panel (a) and (b) are two speech signals produced by the same speaker speaking the same speech content at different speed. Panel (c) and (d) are spectrogram representation of (a) and (b), respectively.

Working with speech signals, one may have to pay close attention to their instability. Unlike music clips, which are controled by machines and copied from same sources, acoustic events of speech signals vary significantly even if they have the same content and are produced by the same speaker. Signal shifting is one of the challenging issues in recognizing patterns of speech signals. It is difficult to align two signals perfectly therefore the beginnings and the ends of speech signals are shifted. As a result, patterns on the corresponding representation model are also unaligned. Consequently, it is difficult to compare the similarities and differences among the features directly.

Another challenging issue is phoneme scaling. It is unlikely that a person can produce exactly the same speech signals even with the same linguistic contents. Perceptually, they may sound the same but technically, the durations of the phonemes are different. As a result, local patterns on the representation models are difficult to be compared directly.

Figure 1.2 shows an example of the signal shifting and phoneme scaling problems. Signals on panel (a) and (b) are produced by a same speaker speaking the same speech content at different speed. It can be seen that the signals are shifted; also, the durations of their compartments are uneven. When these signals are transform into time-frequency domain, the amount of acoustical features and their localization are different. Thus, it is difficult for speech fingerprint techniques to analyze the similarity and difference in speech signals.

The problem becomes more serious with the block-based coding technique because it is sensitive to signal shifting and phoneme scaling. This is the most common coding technique that is used in speech analysis. An arbitrary speech signal is divided into blocks by overlapping windows, and then, each block is transformed independently to create a representation model for the signal, known as spectrogram. Considering the fact that a speaker makes a different speech signal anytime he speaks something even when the speech content is the same. Therefore, applying block-based coding techniques to speech will result in having different spectrograms even when the speaker and the speech content are the same. These problems can be seen clearly in panel (c) and (d) of fig. 1.2. The durations of the prominent acoustical features and their localization are substantially different. An adaptive coding method, which is able to cope with the variations of speech, should be used in the process of creating speech fingerprints.

One theory of efficient auditory coding hypothesizes that the auditory periphery produces an efficient spike code that conveys the maximum amount of information about an input signal [6]. Also, according to current opinion on the sparse coding of sensory inputs [7] and theories of hearing [8], the auditory periphery emphasizes acoustical cues of a continuous speech waveform into neural activity patterns (NAPs) that are sent to the central nervous system, at which point, we are able to understand the speech waveform, e.g., linguistic and speaker individuality. Therefore, mimicking the ability of the auditory periphery to obtain such auditory representations would be beneficial for speech fingerprint techniques.

Previous studies revealed that sparse representation outperforms spectrogram in speech signal representation because sparse coding is data-driven and able to capture the underlying structures and adapt to the variations of speech. Subsequently, the underlying structures can be emphasized by the non-zero elements and are said to be the "geometric" information of the signals [9]. In addition, incorporating psychoacoustic principles into sparse representations can make the underlying structures become perceptual structures. Thus, the perceptual structures can be used to mimic the abilities of the auditory system and improve the performance of speech fingerprint techniques, consequently.

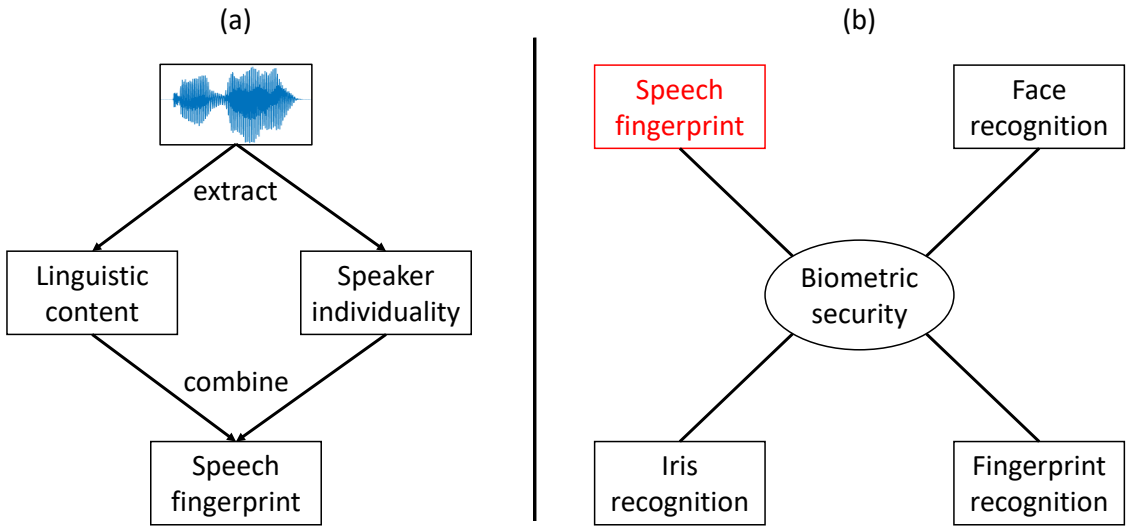

Figure 1.3: Speech fingerprints and biometric security. Panel (a) shows the important features of a speech signal—linguistic content and speaker individuality—in the process of creating speech fingerprints. Panel (b) shows some unique traits of humans including speech fingerprint that can be used in biometric security.

1.4 Motivation and Research Goals

The problem can be solved by using biologically accurate speech fingerprints because it is the recommended way. Identity verification using fingerprint and facial verification is based on biologically accurate features such as minutiae on a fingerprint and facial landmarks on a face. In the same manner, identity verification using speech fingerprint should also be based on biologically accurate features.

Speaker individualities are carried by speech signals, projected onto the auditory nerves, and inherent in neural activity patterns (NAPs). Therefore, auditory sparse representations are derived from speech signals to mimic the NAPs. In general, our minutiae are on our fingers, our facial landmarks are on our face, and our speaker individualities are on our auditory nerves.

To say that using biologically accurate speech fingerprint is the recommended way for speaker verification could be a false claim, to argue, because there are many other nonbiologically accurate ways to deal with the problem of speaker verification. A counterargument is that we use MFCCs as a feature, and deep learning as a classifier for speaker verification because we have insufficient knowledge about the uniqueness of speaker individuality. Perhaps it is this study which has insufficient knowledge about the uniqueness of speaker individuality. Once, as loyal to our faith in speaker individuality, we discover that there are unique minutiae on the NAPs, just like the unique minutiae on the fingers, we should redesign our algorithms to focus on biologically accurate speech fingerprints.

As shown in Fig. 1.3 (b), humans possess many unique traits such as fingerprints, irises, facial features, and speech fingerprints. Tremendous efforts spending on researching speaker individuality, speaker verification, and speech recognition have formed the belief that speech can also become a measurement in biometric security. Regardless of the wide variations of speech, it is a matter of fact that we can recognize the spoken words and distinguish the voice of a speaker from others. Therefore, it is believe that speech signals must have contained information about linguistic content and speaker individuality as depicted in Fig. 1.3 (a). These features are essential to the process of creating speech fingerprints; by extracting and combining them, the accuracy and reliability of speech fingerprint can be greatly improved.

In real life, speech is one of the natural communication tools of humans. The auditory system transforms speech waveforms into neural activities, and information, such as linguistic information, speaker individuality, and emotions, can be recognized. Our hearing system can perform complicated listening tasks, such as speaker identification, speech recognition, and sound localization, in extremely noisy environments, such as in the case of the well-known cocktail party effect. Inspired by the amazing abilities of our hearing system, researchers have been studying the characteristics of the auditory system and utilizing the collected knowledge to improve the performance of various techniques and applications, e.g., hearing aids in health care, speaker recognition in automation, and anti-spoofing in security.

There are numerous direct and indirect studies about speech fingerprints. However, the direction of existing studies is influenced by practical purposes such as speaker identification, verification, and retrieval. As the result, the actual speech fingerprints are often neglected. Perhaps this is because of the lack of a clear definition of what speech fingerprint is. Defining speech fingerprint is not the ambition of the present study; however, curiosity often causes obsessions with answers. A long time ago, when we looked at our fingerprints, those lines on the surface of the fingertips combining with sweat to

increase friction strength when grabbing objects, we did not know that they were one of the most reliable identification systems. Today, we have found that each person possesses unique fingerprints that can be used as a biometric security measure.

The word fingerprint may cause some misunderstandings. When we apply ink on our fingers and stamp on a piece of paper or when we press our fingers on an electrical scanner, we obtain physical or digital images or "prints" of our fingers. Meanwhile, the actual fingerprints are always present on our fingers. Unlike fingerprints, speech fingerprints are not always present and which part of our body accommodates speech fingerprints is unclear. Based on the fact that we can identify different speakers, it is reasonable to believe that there exist speech fingerprints somewhere in the auditory pathway. At this point, the present study assumes that the auditory nerves are where speech fingerprints reside and speech fingerprints are only appeared at the present of human speech. The auditory nerves cary NAPs to the auditory cortex; apperantly, the NAPs contain many information including speech fingerprints as well as linguistic, emotion, age, etc. Some of the natural responses to the assumption are:

- How to extract speech fingerprints from the auditory nerves? This question is important because we cannot pull out the auditory nerves and stamp them on a piece of paper to get speech fingerprints.
- Are the speech fingerprints unique to each person? This question is important because the uniqueness of speech fingerprints is essential if they were to be used as a biometric security measure.
- How to compare two speech fingerprints? This question is important because it is related to the applications of speech fingerprints.

The first goal of the present study is constructing an algorithm to approximate the NAPs of the auditory nerves. Speech fingerprints are assumed to be contained in the NAPs of the auditory nerves; therefore, the present study aims to mimic the NAPs by using psychoacoustic principles, auditory filterbank, and sparse coding. The second goal is verifying the uniqueness of the speech fingerprints. This step is important because the calculated speech fingerprints are approximations of the real ones. This is a challenging task because equipment is unavailable for obtaining the real NAPs of the human ears so that we can compare the real NAPs with the computed speech fingerprints. Thus, we conduct a speech analysis/synthesis experiment and a pattern matching experiment to evaluate the important information encoded by speech fingerprints. The third goal is constructing a suitable identification system to evaluate the uniqueness and usefulness of speech fingerprints. At this point when the first and second goals are achieved, uniqueness of the calculated speech fingerprints are proved, the next step is constructing an effective identification algorithm. One of the main applications of speech fingerprints is identifying speakers in a large dataset. Therefore, speech fingerprints are used in a speaker identification and retrieval experiment.

1.5 Contributions

Biologically accurate features are proposed to use for the process of extracting speech fingerprints. More specifically, Equivalent Rectangular Bandwidth (ERB) scale, Gammachirp (GC) kernel, and masking effect are used in the speech analysis process towards

speech fingerprints extraction. Three experiments were conducted to evaluate the proposed method.

The first experiment is an analysis/synthesis experiment. The proposed method is used to derive auditory sparse representation from speech signals to mimic the NAPs. So, the question is: how close is the calculated NAPs—the auditory sparse representation—to the real NAPs of the auditory nerves? We do not have the real NAPs to compare with yet; therefore, we can only do the analysis/synthesis experiment to evaluate the perceptual qualities—Perceptual Model Quality (PEMO-Q) and Perceptual Evaluation of Speech Quality (PESQ) scores—of the proposed auditory sparse representation. Experimental results show that the proposed method is the most effective to mimic the NAPs in comparison with other types of features.

The second experiment is a pattern analysis experiment. Speaker individuality is believed to be unique to each person and the purpose of the proposed method is obtaining speaker individuality. So, the question is: Do the proposed Auditory Sparse Representation (ASR) contain unique patterns? Landmark-based pattern analysis is used for this purpose. Experimental results provide strong evidence that the proposed ASR contain unique patterns in comparison with other types of features.

The third experiment is about finding an effective speech fingerprint identification system for speaker verification/identification/retrieval. Assuming that the proposed ASR is very similar to the NAPs, it contains unique speaker individuality, and there are currently seven billion speaker individualities of living people and more to come, not to mention those of deceased people and artificial intelligence. So, the question is: What could be an effective speech fingerprint identification system to do speaker verification in this gigantic dataset? Globally and in real-time? A deep hashing—Central Similarity Quantization (CSQ)—is used for this purpose. Identification results show that CSQ is highly effective in Voxceleb2 dataset. The identification results can also be strong evidence about the unique patterns conveyed by the proposed ASR.

1.6 Organization of thesis

The remainder of this dissertation is organized as follows. Chapter 2 describes the related work to creating speech fingeprints including audio fingerprints and various kinds of representations of speech signals. Chapter 3 goes into detail on how our proposed method calculates speech fingerprints from speech signals. Chapter 4 elaborates our experiments to evaluate important information encoded by speech fingerprints. Chapter 5 describes an experiment used for verifying the uniqueness of speech fingerprints by using a landmark-based pattern analysis application. Chapter 6 describes another experiment used for verifying the uniqueness and usefulness of speech fingerprints by using a deep hashing method. Finally, Chapter 7 states our conclusions.

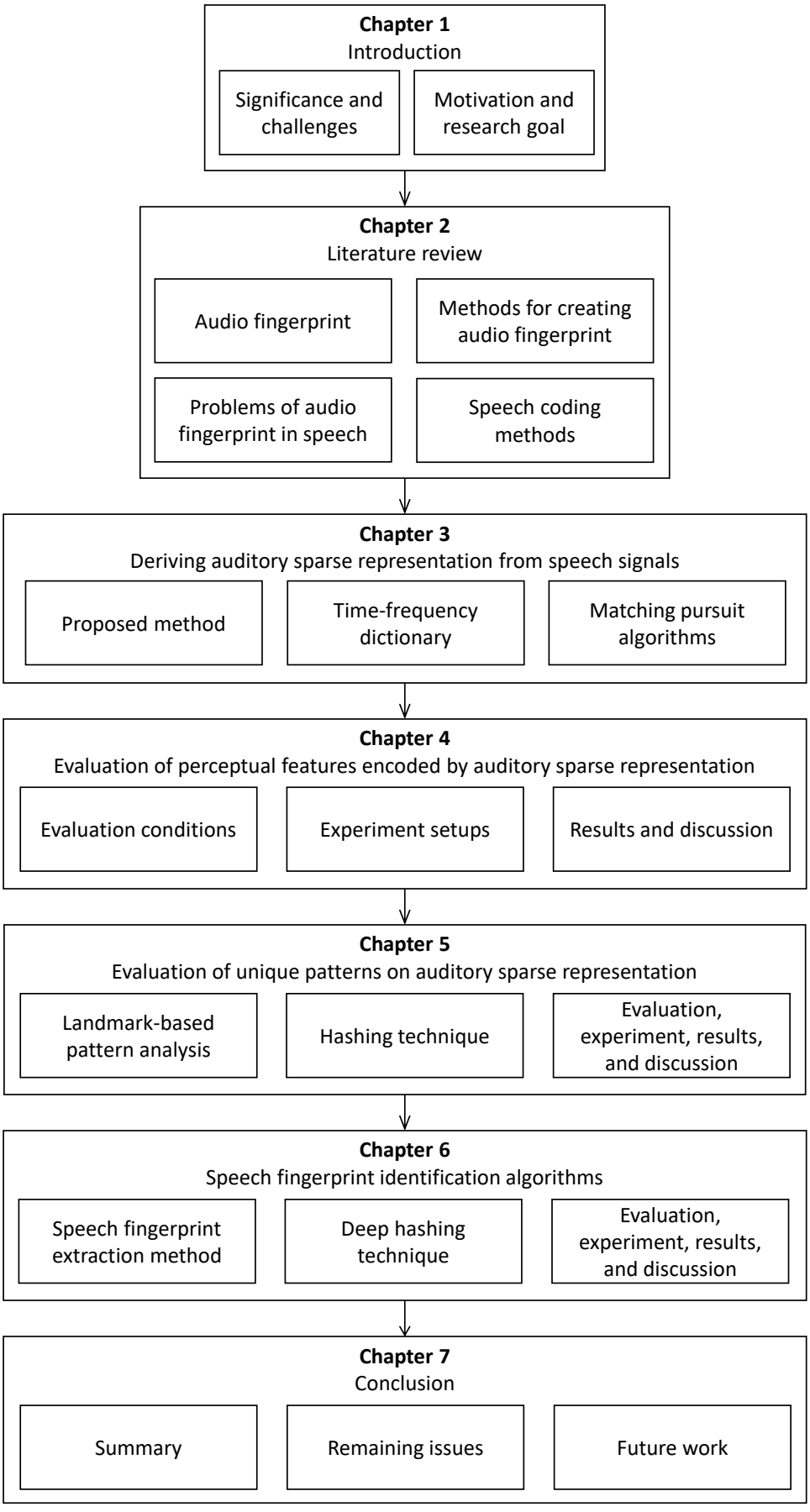

Figure 1.4: Organization of this dissertation

Chapter 2

Literature Review

2.1 Summary

In this chapter, a survey about audio/speech fingerprint methods is presented. The purpose is to learn about what audio/speech fingerprint is, its importance, how they are created, the related issues, and its applications, etc. Furthermore, an experiment is conducted to evaluate the performance of one of the most popular audio fingerprint techniques with speech signals. Experimental results and analysis are presented. Furthermore, a study about current techniques used for decomposing speech signals is presented.

2.2 Audio/Speech Fingerprint Methods

2.2.1 Audio fingerprints

Audio fingerprinting is the process of analyzing audio signals to obtain their unique features, then combining these unique features to create compact representatives for the audio signals [10]. Unlike audio watermarking, which is the process of embeding metadata into audio signals, audio fingerprinting processes the waveforms of audio signals directly to capture their inherent characteristics so as to produce distinguishable information.

In the digital era, when music, speech, and videos are digitalized, uploaded, and transferred in the Internet, audio fingerprints play a significant role in the content identification tasks as media can be quickly identified. According to a report [11], illegal usage of digitalized media have dealt serious damages to our economy; content creators can use audio fingerprints to prevent unauthorized broadcasting of their assets. Applications can be developed to help people identify music quickly using audio fingerprinting techniques; this efficiency can boost productivities and convenience in our lives.

Audio fingerprinting is a very complicated technique. To achieve the effectiveness and efficiency that fit for the demands of content-identification applications, the technique must be accurate and reliable enough to distinguish one audio signal from others. It has to be fast in response to timing requirements. In reality, we do not usually compare two original audio signals together; therefore, the fingerprints must be durable against attacks such as distortions and signals shifting. In addition, audio fingerprints have to be small in size storage requirements and searching performance [4].

Figure 2.1 illustrates a simple application of audio fingerprinting technique that can be used for identifying unknown audio signals. The application can be divided into two processes. The main objective of the first process is creating a database of audio

fingerprints. In this process, a fingerprints extraction unit analyzes audio signals to create their audio fingerprints and these fingerprints are kept in a database for future references. The main purpose of the second process is identifying unknown audio signals. In this process, the fingerprints extraction unit analyzes the unknown audio signals to determine their fingerprints, then these unknown fingerprints are looked up in the database of audio fingerprints by a search algorithm. Once their matches are found, the unknown audio signals can be identified and their additional information can be retrieved. Among the various stages, the fingerprints extraction unit is the heart of this application and will be discussed carefully in the following sections.

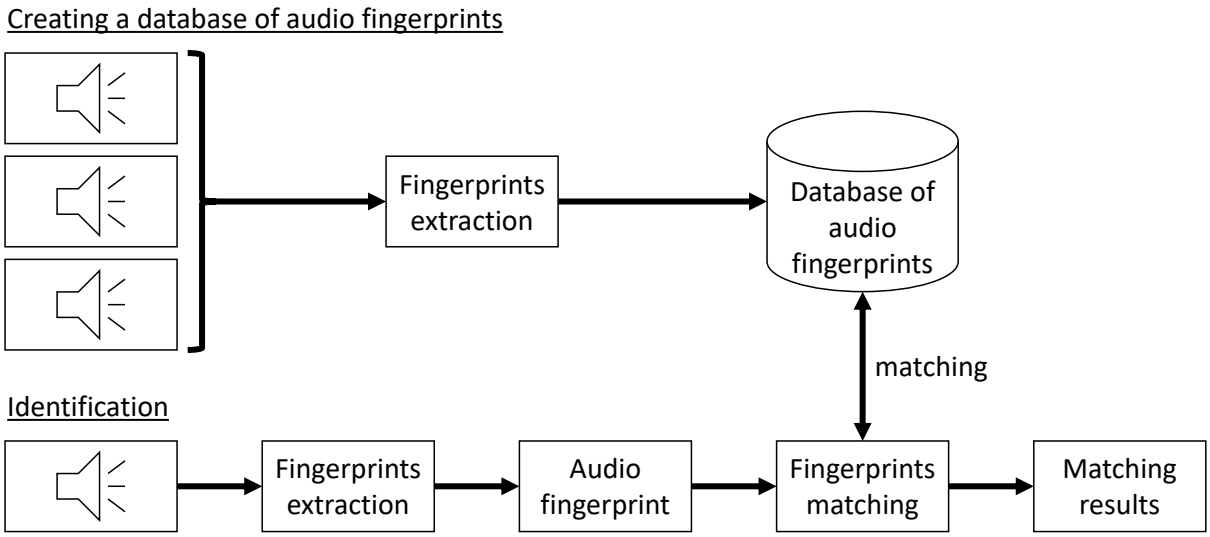

Figure 2.1: Application of audio fingerprints used for content identification.

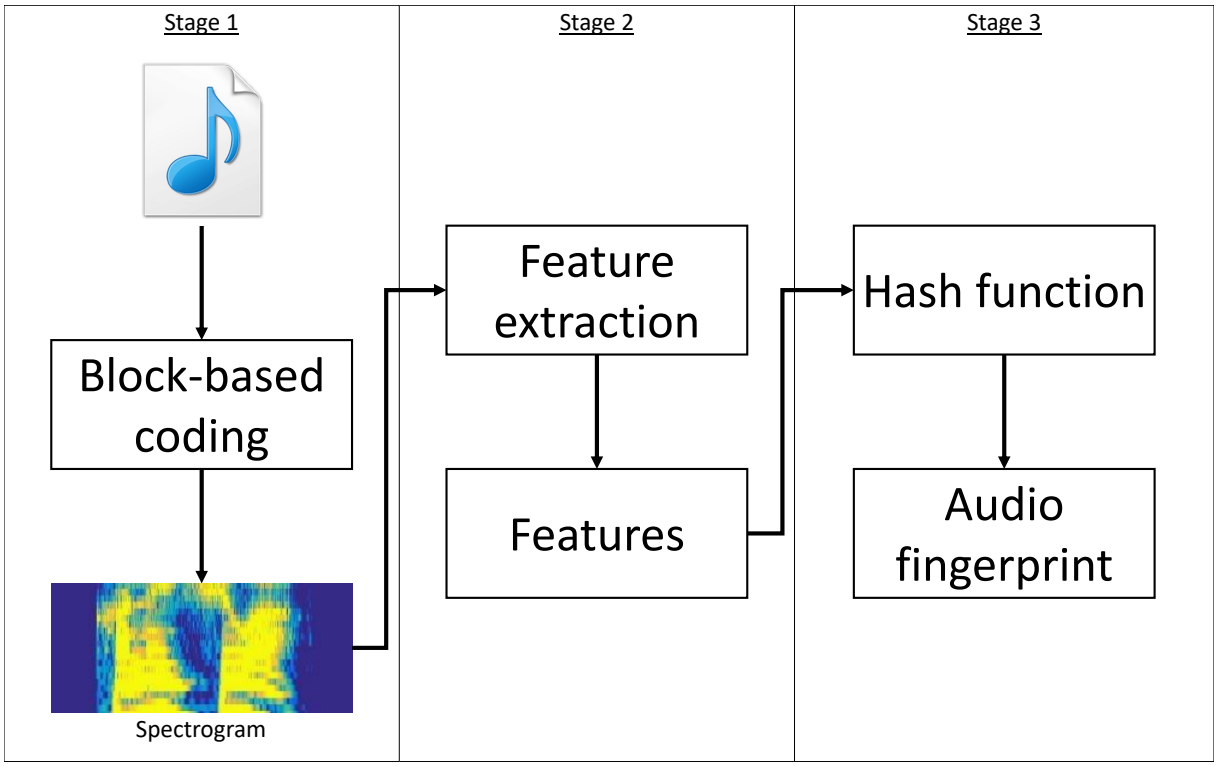

Figure 2.2: A common process of creating audio fingerprints.

2.2.2 Typical method used for creating audio fingerprints

In the attempts to derive the fingerprints of audio signals, many researches and different approaches have been conducted and proposed by researchers [12–14]. In spite of their ingenious ideas, their methods appear to consist of three stages as depicted in Fig. 2.2. In the first stage, an encoding technique is applied to transform raw audio waveforms into their representation models. In the second stage, a feature extraction module is used to analyze and extract unique features of the audio waveforms from their representation models. In the last stage, a hashing technique is used to pack the extracted features to create fingerprints for the input audio signals.

The main idea of stage one is converting a raw input waveform into a set of features that represents acoustic events and block-based coding is commonly used for this purpose. This method assumes that for a period of a few milliseconds, the audio signal is unchanged. For this reason, a windowing technique is usually applied to divide the signal into frames. Then linear transformation techniques, namely Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), and wavelet transformation, converts the frames into a set of acoustic features. The output of this stage is a representation model of the input signal, commonly known as a spectrogram. One important concern when using windowing technique is that there is a trade of between loss of information and computational complexity.

The main purpose of stage two is extracting features of the representation model that convey inherent information of the input audio signal. The extracted features of one signal should be distinguishable with features of other signals and resilient to attacks such as distortions and signal shiftings. For this purpose, a handful of techniques have been proposed for different rationales. Some researches utilized Linear Prediction Coefficients (LPCs), linear prediction cepstral coefficients (LPCCs), or Mel-frequency cepstral coefficients (MFCCs) to represent the spectral envelope. Some were inspired by the vibrations of the basilar membrane and used signs of energy to simulate its fluctuations. Other researchers used only maximum energies or peaks of the spectrograms with the justification that lower energies are unlikely to survive noise.

Creating fingerprints for audio signals from the extracted features is the main focus of stage three. The features obtained from stage two, which contain unique characteristics of the audio signals, can be regarded as fundamental fingerprints and the hash function of stage thee continues to increase the discrimination power of the fingerprints and to reduce the dimension of the fingerprints for searching efficiency. Templates of audio fingerprints vary depending on the hash function. They can be vectors or matrices of binary or decimal numbers. Some researchers compared values of the extracted features to create binary encoded fingerprints, others created their final fingerprints by applying pattern recognition methods to capture the relationships of the features [14–20].

2.2.3 Problems of audio fingerprints in speech

In case of music, the methods mentioned above achieved admirable results but the characteristics of speech have exposed their weaknesses. Block-based coding technique divides a speech signal into blocks having similar arbitrary widths and processes these blocks separately using Discrete Fourier Transform (DFT) or Discrete Cosine Transform (DCT). This technique is substantially sensitive to signal shifting and phoneme scaling; randomly blocking speech signals does not take into account the alignment of acoustic cues. Given a speech content, it is unlikely for a speaker to produce signals having the same length and

phonemes having the same duration. Figure 1.2 shows two speech signals that have the same content and are produced by the same speaker and their corresponding spectrograms. Perceptually, these two speech signals may sound the same but techniquely, the alignments and durations of their acoustic features are considerably different. Therefore, block-based coding will create different spectrograms for speech signals that have the same content and same speaker. One way to overcome the drawback of block-based coding is using filterbank based shift invariant coding but this technique greatly increase the dimension of speech signals because of its convolutional calculations. Data redundancy on the representation models makes it difficult to recognize the underlying structures of speech signals [9, 21].

The sources of problems do not confine only in the signal representation process, they also appear in the features extraction methods. To gather features for the production of speech fingerprints, several methods have been proposed; one used the signs of energies, another used the peaks of spectrogram. Although their methods achieved great experimental results, none of them take into account patterns of the features on the representation model. Given a pair speech signals that has the same content but different speakers and another pair that has the same speaker but different contents, a feature extraction method should realize the similarities and differences of the features on the representation model to improve the quality of speech fingerprints.

An experiment was carried out to evaluate the ability of block-based coding in representing speech signals. A Gammatone (GT) filterbank [22] was used in this experiment to process ten speech signals with the same content produced by two speakers. Although this is one of the most well-known techniques used in speech processing, testing the distortion between the original signals and the resynthesized signals could only achieve mean $\mu = 18.9$ dB and standard deviation $\sigma = 1.2$ dB in the Signal to Noise Ratio (SNR) (ranges of SNR are: below 25 dB is low, from 25 dB to 40 dB is high, and above 40 dB is excellent), and mean $\mu = 3.9$ and standard deviation $\sigma = 0.1$ in the perceptual evaluation of speech quality (PESQ)(PESQ scores are: 1 is bad, 2 is poor, 3 is fair, 4 is good, and 5 is excellent).

Based on the work by Ellis [5], another experiment was also conducted to investigate the effectiveness of spectrograms in the process of producing speech fingerprints. The results obtained from this experiment by using the previously described data proved that different fingerprints were generated for speech signals that had the same content and the same speaker despite the fact that this method worked very well with audio fingerprints. Section 2.3 goes into detail about advantages and disadvantages of various kinds of speech coding strategies.

2.3 Speech coding methods

2.3.1 Uniform filterbank

The time-frequency resolution of a signal representation is an important aspect in filterbank design. The shape of the filter, duration of translation, and filter function should be chosen adaptively to the input sounds (i.e., noises, transients, tones, and complex sounds) to increase the fidelity of acoustical cues as well as the signal structure of the resultant representation.

At the early stage of hearing research, it was believed that the cochlea was a frequency analyzer; therefore, orthonormal bases such as the well-known Fourier series and Krawtchouk-Tchebichef polynomials [23] were commonly used to expand audio signals

into a linear combination of pure tones as a way to mimic the cochlear [8]. Fourier Transform (FT) is sufficient to obtain the frequency components of a signal. It provides an insight into the prominent frequencies of a signal. It is calculated as follows:

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt, \tag{2.1}$$

where $f(t)$ and ω are a signal and frequency, respectively. With regard to periodic signals, FT is sufficient to obtain frequency components of a signal. However, it becomes troublesome with aperiodic signals. It provides only a frequency spectrum of the input signal; time localization of acoustical cues is unavailable. A wide variety of natural sounds such as speech is aperiodic and consists of different frequency components appearing at different time onsets. Therefore, using FT is not suitable for obtaining the complex structures of these signals. In the auditory system, sounds are converted by the auditory periphery into NAP. In NAP, not only are the places where neurons are fired important, the precise timings of the firing are also essential cues for the brain to understand the input waveforms [8]. Therefore, estimating the time and frequency localization of the acoustical cues of input signals is an important task to be considered.

One possible way to obtain the time localization of frequency components is using Short-time Fourier Transform (STFT). A short segment of a signal is taken, and it is assumed that this short-segmented signal is periodic, and FT can be applied to this signal. STFT can be represented as a windowed Fourier transform in $\mathbf{L}^2(\mathbb{R})$ as:

$$Sf(\tau, \omega) = \langle f, g_{\tau,\omega} \rangle = \int_{-\infty}^{+\infty} f(t)g(t-\tau)e^{-i\omega t}dt, \tag{2.2}$$

where $g(t-\tau)$ is a window function translated by τ. If the duration of the window function equals the duration of the input signal, Eq. (2.2) becomes Eq. (2.1), and signal decomposition is applied to the entire length of the signal. Thus, information about what frequencies appear at what time onset cannot be obtained. With STFT, however, the signal is segmented into short fragments, and spectrum analysis is applied to each segment. The frequency spectra of the segments are then concatenated. This way, the result of the STFT is a three-dimensional spectrogram of the signal. On the horizontal axis, a sequence of fragments can be seen as the time, on the vertical axis, the frequency, and on the other axis, the magnitudes of Fourier coefficients.

The practice of segmenting a signal into short fragments is called the windowing technique. Each window has its shape, such as a length and height. It is translated by a certain amount along a signal during the analysis process. The choice of the window shape and translation duration is related to the time-frequency resolution and the amount of data in the resulting spectrogram. The important point is to learn the time localization of the prominent frequency components composing a signal so that the structures of the signal can be emphasized in the corresponding spectrograms. For example, it is better to use a longer window to capture the oscillations in a lower frequency range and a shorter window to obtain transients in a higher frequency region. If the analysis window is translated with no-overlap, there might be a loss of information in the spectrogram, and if the window is translated with a high amount of overlap, important cues of the signal will be smeared in the spectrogram. As a result, the perceptual structures of speech signals are difficult to obtain. Although STFT provides a time-frequency representation of a signal, there is a shortcoming with it; STFT uses uniform window shapes. Therefore, if a complex signal contains both oscillations at low frequencies and transients at high frequencies, a fixed window can only be suitable for one of these cases.

2.3.2 Non-uniform filterbank

Finding suitable shapes of analysis windows to obtain a good time-frequency resolution is a challenging task in signal processing. A preferable choice is to use wide windows at low frequencies and narrow windows at high frequencies. This can be done by utilizing a scaling factor.

$$Wf(\tau, s) = \langle f, \psi_{\tau,s} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-\tau}{s} \right) dt. \tag{2.3}$$

Equation (2.3) shows the calculation of a continuous wavelet transformation. In the equation, there is a basis function, ψ, and this function is also called the mother wavelet. The shape of the analysis windows can be dilated by the scale factor s. By adjusting the scale factor, a family of wavelets with different shapes can be created. In addition, by applying this wavelet technique, a short window height can be used for a good frequency resolution at low frequencies and a short window length for a good time resolution at high frequencies.

On the basis of spectrograms, feature-emphasis techniques such as mel-frequency cepstral coefficients (MFCCs) [24–26] and linear prediction cepstral coefficients (LPCCs) [27, 28] can be applied to obtain significant acoustical feature vectors. Pattern-analysis techniques (e.g., support vector machine [29] for voice pathology detection, Gaussian mixture model [30] for speaker recognition, and deep neural networks [31] for emotion recognition) are then applied to the feature vectors.

Although a non-uniform filterbank (e.g., wavelet transform) provides a better time-frequency resolution in comparison with a uniform filterbank (e.g., STFT) and current speech-analysis techniques provide perfect signal reconstruction [32, 33]. The output provided by filterbank analysis in general contains highly redundant coding data. Thus, speech-signal patterns are smeared over spectrograms [9].

2.3.3 Auditory filterbank

Traditional wavelet filterbanks utilize a scale factor $s = 2^j$, where $0 \leq j \leq log_2(N)$, and N is the number of samples of an input signal [34]. The scaling factor used for controlling the time-frequency localization of wavelets is calculated without considering psychoacoustic findings. Thus, the center frequencies and the corresponding bandwidths of wavelet filters are not similar to those of auditory filters. Hence, the outputs of traditional wavelet filterbanks cannot be regarded as auditory representations.

Incorporating scientific discoveries into designing filters would provide a filterbank that is close to that of the auditory system. Psychoacoustic studies have calculated the ERB scale by estimating the ERB and the center frequency of an auditory filter by using human masking data [35]. Therefore, controlling the time and frequency localization of wavelets on the basis of the ERB scale would provide a representation that is more closely related to how sounds are represented in the auditory system.

Gammatone and gammachirp filterbanks are two variations of wavelet filterbanks. The center frequencies of these filterbanks are divided on the basis of the ERB scale, and the filter functions are gammatone and gammachirp functions that were designed on the basis of the characteristics of the basilar membrane. The outputs of these filterbanks are most similar to auditory representations in comparison with STFT and Gabor (GB) filterbanks.

Another commonly used approach in emphasizing significant features is auditory-inspired representations which focuses on important perceptual features. Inspired by the

amazing abilities of the human auditory system, different methods have been proposed to mimic the cochlear. For instance, neurograms [36–38] and cochleagrams [39–42] have been successfully applied to improve performance of various applications such as speech-emotion recognition, phoneme classification, and speech-intelligibility prediction. The advantage of these auditory spectrograms over conventional spectrograms is that psychoacoustic principles (e.g., using equivalent rectangular bandwidth scale and gammatone impulse response) are used to derive perceptual features from speech signals. Nevertheless, auditory spectrograms possess a similar drawback of conventional spectrograms, i.e., the abundant coding data obscures the perceptual structures of input signals in the representation [9].

2.3.4 Sparse representation

Although the wavelet transform provides a better time-frequency resolution in comparison with STFT, frame-based processing with overlapping windows generates high redundancy in coding data, causing the structures of input signals to be difficult to obtain. It is desirable to have an adaptive technique that provides a good time-frequency resolution for wavelet transforms, discards redundancy, and emphasizes the unique features of speech signals. The structures of speech signals can be emphasized by using sparse nonlinear representation because this kind of representation focuses on the highest energy of the signals with few coefficients. These few coefficients are said to be the "geometric" information of the signals [9]. Furthermore, research in the literature also provides evidence that, although with fewer coefficients, the number of approximation errors is lower than in the case of linear approximation.

The Nyquist sampling theorem specifies that to avoid losing information when capturing a signal, one must sample at least two times faster than the signal bandwidth. This is the most common method used to digitize sounds such as speech and music. Recent studies have discovered that a digitized signal can be further converted into and reconstructed from its sparse representation. The number of non-zero elements in this sparse representation is much less than the number of samples of the digitized signal itself. Thus, this sparse representation provides great benefits in signal compression, noise reduction, and pattern recognition. The literature has proof showing that the sparse representation is beneficial and favorable in a wide range of applications such as speech processing and computer vision.

Recent studies have emphasized the advantages of sparse representation over spectrogram representation as it increases the signal to noise ratio, it is shift invariant [43], and more importantly, the geometrical information of the speech signals is emphasized with the non-zero elements of the representation [9]. Given a discrete signal f and an orthonormal basis $\mathfrak{D} = \{g_m\}_{m\in\Gamma}$, an approximation of f is

$$f = \sum_{m\in\Gamma} \langle f, g_m \rangle \, g_m. \tag{2.4}$$

A sparse representation of f can be obtain by projecting the signal onto an orthonormal basis $\Lambda = \{g_m\}_{m\in\Lambda}$, where $\Lambda \subset \mathfrak{D}$. The orthogonal projection of f on the space V_Λ generated by the vectors in Λ is

$$f_\Lambda = \sum_{m\in\Lambda} \langle f, g_m \rangle \, g_m. \tag{2.5}$$

Thus, the resulting error is as follows:

$$\|f - f_\Lambda\|^2 = \sum_{m \notin \Lambda} |\langle f, g_m \rangle|^2 . \tag{2.6}$$

The approximation error is dependent upon the number of vectors in Λ; therefore, a threshold T can be used to control the trade-off between the sparseness of the sparse representation and the approximation error.

A current trend in audio and speech coding proposes using auditory sparse representation to focus on the perceptual structure of a signal [43]. One theory of efficient auditory coding hypothesizes that the auditory periphery produces an efficient spike code that conveys the maximum amount of information about an input signal [6]. According to current opinion on the sparse coding of sensory inputs [7] and theories of hearing [8], the auditory periphery emphasizes acoustical cues of a continuous speech waveform into neural activity patterns (NAPs) that are sent to the central nervous system, at which point, we are able to understand the speech waveform, e.g., linguistic and speaker individuality. Thus, mimicking a NAP would be beneficial for various speech-analysis techniques and applications, e.g., hearing aids in health care, speaker recognition in automation, and anti-spoofing in security.

Sparse representation, which has been becoming preferred over spectrogram representations, can be used to overcome the disadvantage of the redundancy of coding data. The orthogonal matching-pursuit (OMP) algorithm is commonly used for obtaining sparse representation of speech signals [34]. An advantage of sparse representation is that it provides high signal reconstruction quality using a low number of non-zero elements. A previous study [44] reported that the sparse representation provided good signal-reconstruction quality using only 5 to 10% of the dimension of the original signal under clean and noisy conditions. Subsequently, the underlying structures of speech signals can be emphasized by the non-zero elements and are said to be the "geometric" information of the signals [9].

Although the original OMP algorithm can be used to discard the redundancy of coding data, the obtained underlying structures cannot be regarded as perceptual structures because psychoacoustic principles are not incorporated into the algorithm. Previous studies [45, 46] used perceptual MP (PMP) algorithms to improve sinusoidal audio modeling. Although these algorithms provide a perceptual sparse representation, which increases the perceptual quality of the resynthesized signals, their drawback is that they operate on the Bark scale. Psychoacoustic research derived the equivalent rectangular bandwidth (ERB) scale as a function that relates the number of ERBs to the center frequencies of auditory filters by using a notched-noise method to better explain psychoacoustical data [35]. Therefore, using the ERB scale is more suitable than using the Bark scale in creating auditory sparse representation.

2.3.5 Optimal kernel

Determining optimal kernels, g_m, to project audio signals onto the space V_Λ such that the resultant representation is similar to that of the auditory system is an important issue. A theoretical study suggested that, given a mixture of sounds, a suitable shape and length of a kernel could improve the fidelity of a representation [6]. This study also reported that the optimal kernel for representing a combination of mammalian vocalizations and environmental sounds has a rapid rise and slow decay that are similar to the characteristics

of the envelopes of auditory revcor filters and gammatone/gammachirp filters. Research reported in [47] evaluated three kinds of kernels (i.e., Damped Sinusoid (DS), Gabor, and gammatone) with some speech signals from the TIMIT dataset and suggested that the gammatone kernel was the best choice in terms of signal reconstruction quality and atom rate. An extended version of the gammatone filter, known as the gammachirp, is described as an optimal auditory filter in comparison with the Gabor and gammatone filter [48]. The gammachirp can provide an excellent fit to 12 sets of notched-noise masking data [49]. Therefore, the gammachirp kernel is a good candidate to be considered for auditory representation. This paper aims to evaluate the gammachirp kernel along with damped sinusoid, Gabor, and gammatone kernels in the process of creating auditory representations.

Another drawback of the original OMP algorithm is that it uses Gabor basis to decompose speech signals. A previous study [47] evaluated three types of kernels—damped sinusoid (DS), Gabor (GB), and gammatone (GT)—with some speech signals from the TIMIT dataset and suggested that gammatone kernels were the best choice in terms of signal-reconstruction quality and atom rate. Psychoacoustic research has found that another important characteristic of the auditory periphery is that the impulse responses measured at the basilar membrane have a gamma-like temporal envelope and non-linear up-chirp frequency modulation [50]. The gammachirp (GC) provided an excellent fit to 12 sets of notched-noise masking data [49] and was described in a previous study [48] as an optimal auditory filter in comparison with the GB and GT filters. By using the GC kernel, the auditory representations would be more similar to that of the auditory periphery.

2.3.6 Auditory masking

Incorporating psychoacoustic principles into auditory representations is mostly about masking effects, i.e., frequency and temporal masking. The underlying idea is to obtain only audible kernels, and thus, the perceptual structures of the speech signals are more refined, and the auditory representation is more similar to that of the auditory periphery. Research reported in [21, 45, 46, 51] proposed temporal masking models to remove inaudible elements to refine perceptual patterns on sparse representations of speech signals. However, the drawback of this research is that the frequencies of the kernels and the masking surface are calculated on the basis of the Bark scale. Moore and Glasberg described in their work that the ERB scale is more closely related to how sounds are represented in the auditory system [35]. Research reported in [52] used the Gabor function to develop a time-frequency masking kernel. The underlying idea is to calculate a joint time-frequency masking surface instead of considering frequency masking and temporal masking separately. However, the drawback of this research is its use of the Gabor kernel, which is a symmetrical kernel and in contrast with the optimal kernel found in [6, 48, 49]. This paper aims to calculate the frequencies of the kernels and the masking surface on the basis of the ERB scale instead of the Bark scale.

The masking effect is another important phenomenon of the auditory periphery that is incorporated with the MP algorithm. The underlying idea is to obtain only audible kernels; thus, the perceptual structures of the speech signals are more refined, and the auditory representation is more similar to that of the auditory periphery. Previous studies [21, 51] proposed temporal masking models to remove inaudible elements to refine perceptual patterns on sparse representations of speech signals. However, the drawback

of these studies is that the frequencies of the kernels and masking surface are calculated on the basis of the Bark scale. Moore and Glasberg described that the ERB scale is more closely related to how sounds are represented in the auditory system [35]. A previous study [52] used the GB function to develop a time-frequency masking kernel. The underlying idea is to calculate a joint time-frequency masking surface instead of considering frequency masking and temporal masking separately. However, the drawback of this research is its use of the GB kernel, which is a symmetrical kernel and in contrast with the optimal kernel found in previous studies [6, 48, 49].

Chapter 3

Auditory Sparse Representation

3.1 Summary

In this chapter, an algorithm is constructed to reproduce the output of the auditory periphery that is the NAPs of the auditory nerves. The purpose of the algorithm is deriving auditory sparse representations of speech signals as a way to mimic the NAPs. Because speech fingerprints cannot simply be extracted as the case of fingerprints or facial patterns by using electrical scanners, they can only be calculated. To do so, a gammachirp auditory filterbank and psychoacoustic principles are employed to ensure the perceptual features on the representations are as similar to the NAPs as possible. Furthermore, an orthogonal matching pursuit algorithm is used to produce sparse representations because NAPs are sparse signals.

3.2 Proposed method

Sparse representations of speech signals can be calculated by using a Matching Pursuit (MP) algorithm. Figures 3.2 and 3.3 show block diagrams of Orthogonal Matching Pursuit (OMP) and Perceptual Matching Pursuit (PMP) algorithms, respectively. Generally, the algorithms calculate a sparse representation of an input signal in four steps. First, the algorithms project an input speech signal onto an overcomplete time-frequency dictionary. Then, in step two, the highest values of the orthogonal projection of the input signal on a kernel of the time-frequency dictionary are selected. In step three, the time-frequency localization and magnitude of the selected kernel are used to update the sparse representation, resynthesize the signal, and remove the selected kernel from the projection. In step four, a threshold is chosen as a stopping rule. Matching pursuit is an analysis by synthesis technique, meaning that the resynthesized signal and the input signal are compared at each iteration using a metric such as PEMO-Q, PESQ, SNR, or LSD. If the threshold or the maximum number of iterations is reached, the algorithms halt; otherwise, they loop back to step one.

The MP and PMP algorithms reported in this text use the ERB scale to calculate the time-frequency dictionary to account for the characteristics of the auditory system. Four kinds of kernels are evaluated to find the optimal kernel for decomposing speech signals. In addition, a masking model, which is calculated from the selected kernels, is utilized to remove masked elements in an orthogonal projection to account for masking effects of the auditory system. Thus, the output sparse representation can convey the perceptual structures of the input speech signals.

The calculation of the time-frequency dictionaries using the four kinds of kernel functions based on the ERB scale is described in section 3.3. The MP, PMP, and a masking model evaluated in this study are explained in section 3.4.

3.3 Time-frequency dictionary

A matching pursuit algorithm requires a time-frequency dictionary, which is an overcomplete set of kernels, to decompose input signals. The work reported in this paper utilized the ERB scale to calculate the center frequencies and bandwidths of kernels to account for the characteristics of the auditory system. While revising the Zwicker loudness model, Moore et al. [53] provided a formula that related the number of ERBs to the frequency as follows:

$$\begin{aligned} \text{Number of ERBs, E} &= 21.4 \log_{10}(4.37 f_c + 1) \\ \Rightarrow f_c &= \frac{1}{4.37}\left(10^{\frac{\text{E}}{21.4}} - 1\right), \end{aligned} \tag{3.1}$$

where f_c(kHz) is the center frequency of an auditory filter, and ERBs is the number of the equivalent rectangular bandwidth. Furthermore, the bandwidth of an auditory filter can be calculated as follows:

$$\text{ERB}(f_c) = 24.7(4.37 f_c + 1), \tag{3.2}$$

where ERB is the equivalent rectangular bandwidth of an auditory filter at the center frequency f_c. Equations (6.1) and (6.2) are utilized to calculate the center frequencies and bandwidths of four kinds of time-frequency kernels, i.e., Gabor, damped sinusoid, gammatone, and gammachirp kernels. Figure 3.1 shows the kernel models that are evaluated in this work.

3.3.1 Gabor kernel

Among the four kinds of kernels evaluated in this experiment, Gabor is the only symmetrical one. It is also a popular spectrum used in the sparse coding techniques. For instance, Gabor dictionaries were developed in two independent studies to decompose input signals [34,54]. A Gabor kernel can be generated by multiplying a Gaussian function and a sinusoidal wave. The real part of a Gabor filter can be obtained by using the following equation:

$$g_{GB}(t) = K e^{-\pi(\text{ERB}(f_c)t)^2} \cos(2\pi f_c t + \phi), \tag{3.3}$$

where K is a normalizing factor so that the kernels have unit norm, f_c is the center frequency of a kernel, and ϕ is the phase of the carrier. Figure 3.1(a) shows 3 of the 64 Gabor kernels that were evaluated in this work.

3.3.2 Damped sinusoid kernel

One of the kernel models evaluated in this experiment was damped sinusoid (DS) kernels. The symmetrical Gabor kernels and asymmetrical DS kernels were compared in a study reported in [55]. This study provides proof that DS kernels are more suitable than the symmetrical Gabor kernels in representing transients in music signals. DS kernels can be

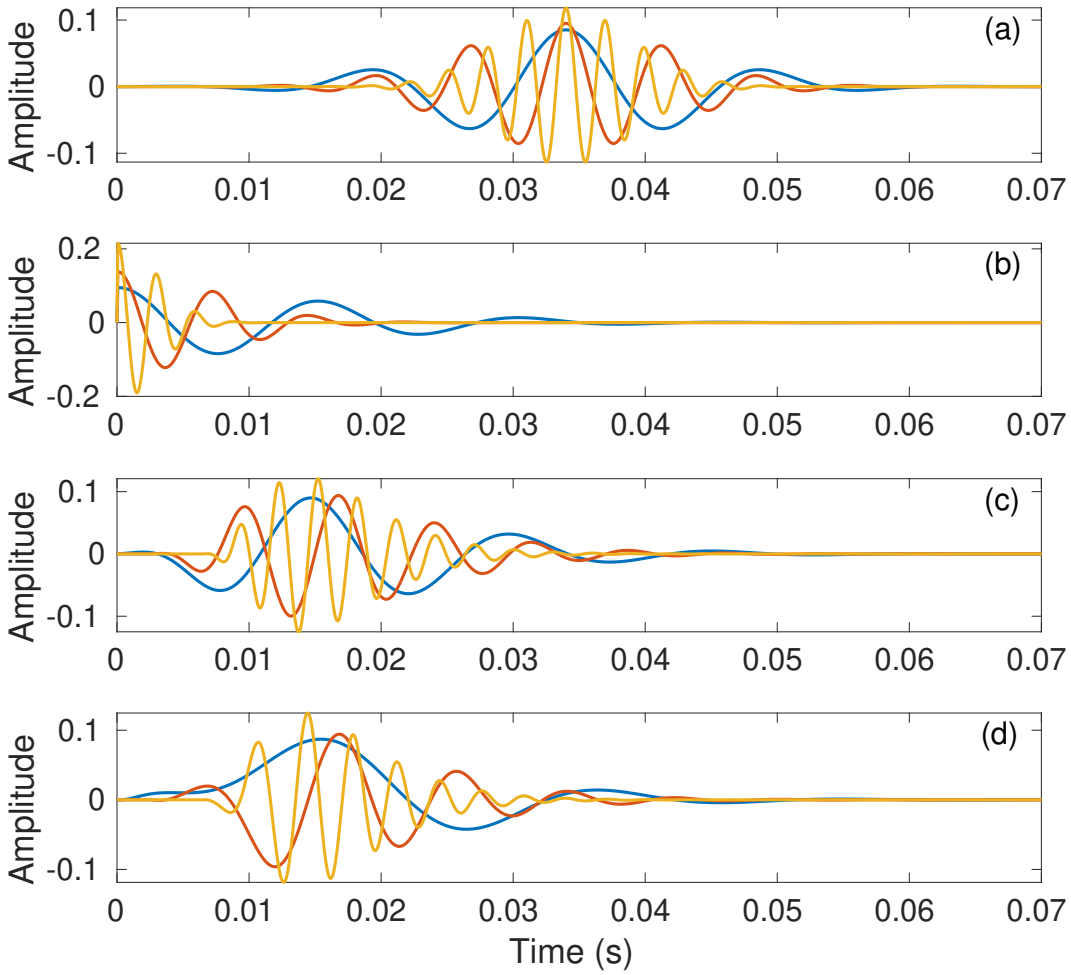

Figure 3.1: Four kinds of time-frequency dictionaries used in matching pursuit algorithm: (a) Gabor kernels, (b) damped sinusoid kernels, (c) gammatone kernels, and (d) gammachirp kernels.

generated by utilizing one-sided exponential windows. The real part of a DS kernel can be obtained by multiplying a cosine wave with an exponential decay function:

$$g_{DS}(t) = Ke^{-\pi(\mathrm{ERB}(f_c)t)^2}\cos(2\pi f_c t + \phi), \tag{3.4}$$

where K is a normalizing factor so that the kernels have unit norm, f_c is the center frequency of a kernel, and ϕ is the phase of the carrier. Figure 3.1(b) illustrates 3 of the 64 DS kernels that were evaluated in this work.

3.3.3 Gammatone kernel

One of the base spectra used in this experiment was gammatone kernels. Unlike Gabor and DS kernels, the shape of gammatone/gammachirp kernels is derived from physiological experiments. The fourth order of the gammatone function provides a temporal envelope that has a shape similar to that of the impulse response measured at the basilar membrane [50]. A gammatone kernel can be created by multiplying a gamma distribution and a sinusoidal carrier. The impulse response of the gammatone filter is given by:

$$g_{GT}(t) = at^{n-1}e^{-2\pi b\mathrm{ERB}(f_c)t}\cos(2\pi f_c t + \phi), \tag{3.5}$$

where $a, n = 4, b = 1.019\mathrm{ERB}(f_c), f_c$, and ϕ correspond to the amplitude, order of the filter, bandwidth of the filter, center frequency, and phase, respectively [56]. Figure 3.1(c) illustrates an example of three gammatone kernels.

3.3.4 Gammachirp kernel

The shape of the impulse response of the gammachirp kernel also has a gamma distribution; however, the gammachirp function was derived to be an optimum auditory filter. Research reported in [49] demonstrates that the gammachirp auditory filterbank has an excellent fit to human masking data. Therefore, the effectiveness of the gammachirp kernel in decomposing speech signals is evaluated in this work. The impulse response of the gammachirp filter is given by:

$$g_{GC}(t) = at^{n-1}e^{-2\pi b\mathrm{ERB}(f_c)t}\cos(2\pi f_c t + c\ln t + \phi). \tag{3.6}$$

The gammachirp function is an extended version of the gammatone function in that it has an additional chirp factor, $c\ln t$, used for controlling the asymmetry of its amplitude spectrum. Other parameters such as a, t, n, b, f, and ϕ are similar to those used in the gammatone function. Figure 3.1(d) illustrates three gammachirp kernels used in this experiment.

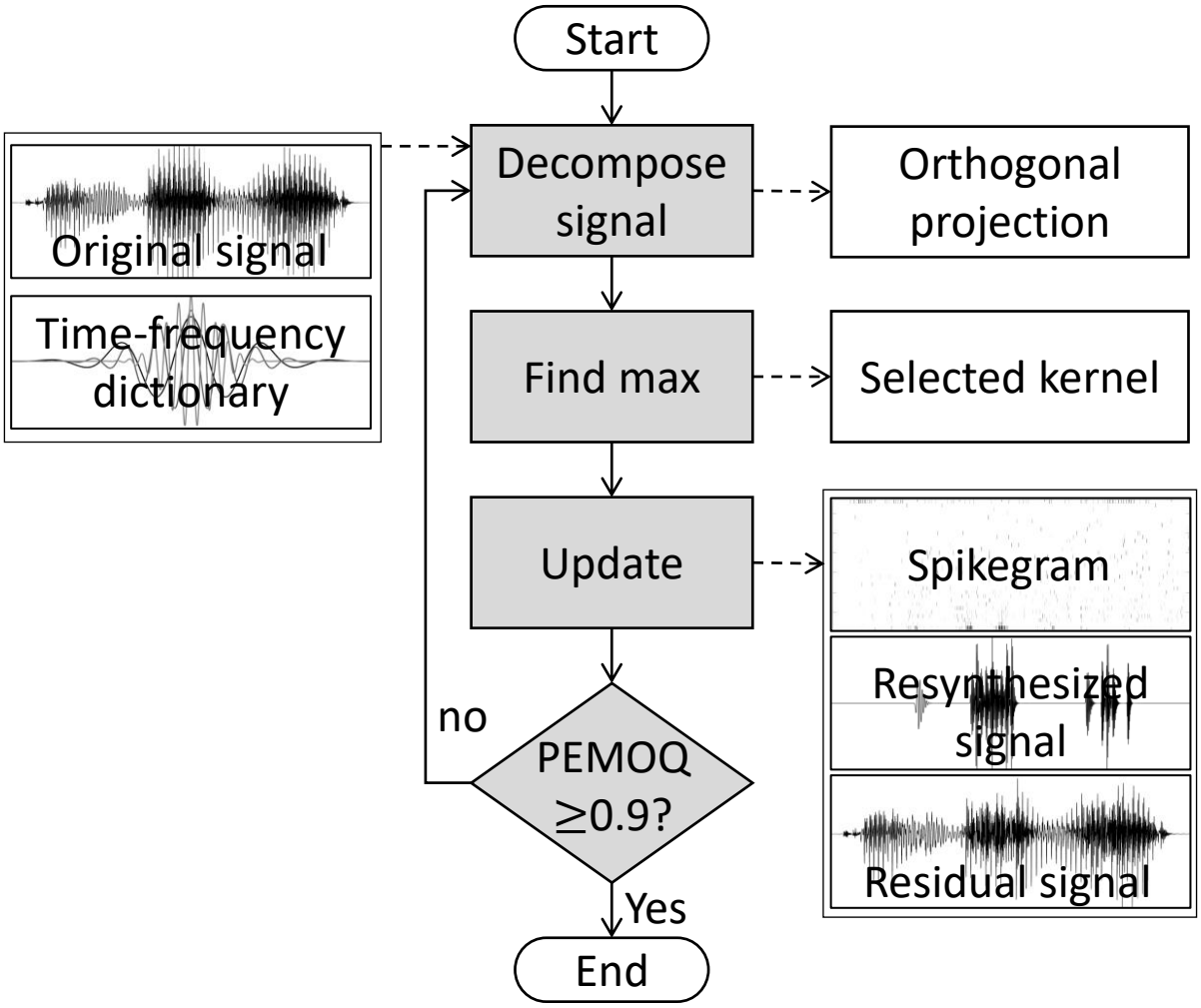

Figure 3.2: Processing pipeline of orthogonal matching pursuit algorithm evaluated in this work to derive spikegrams from speech signals.

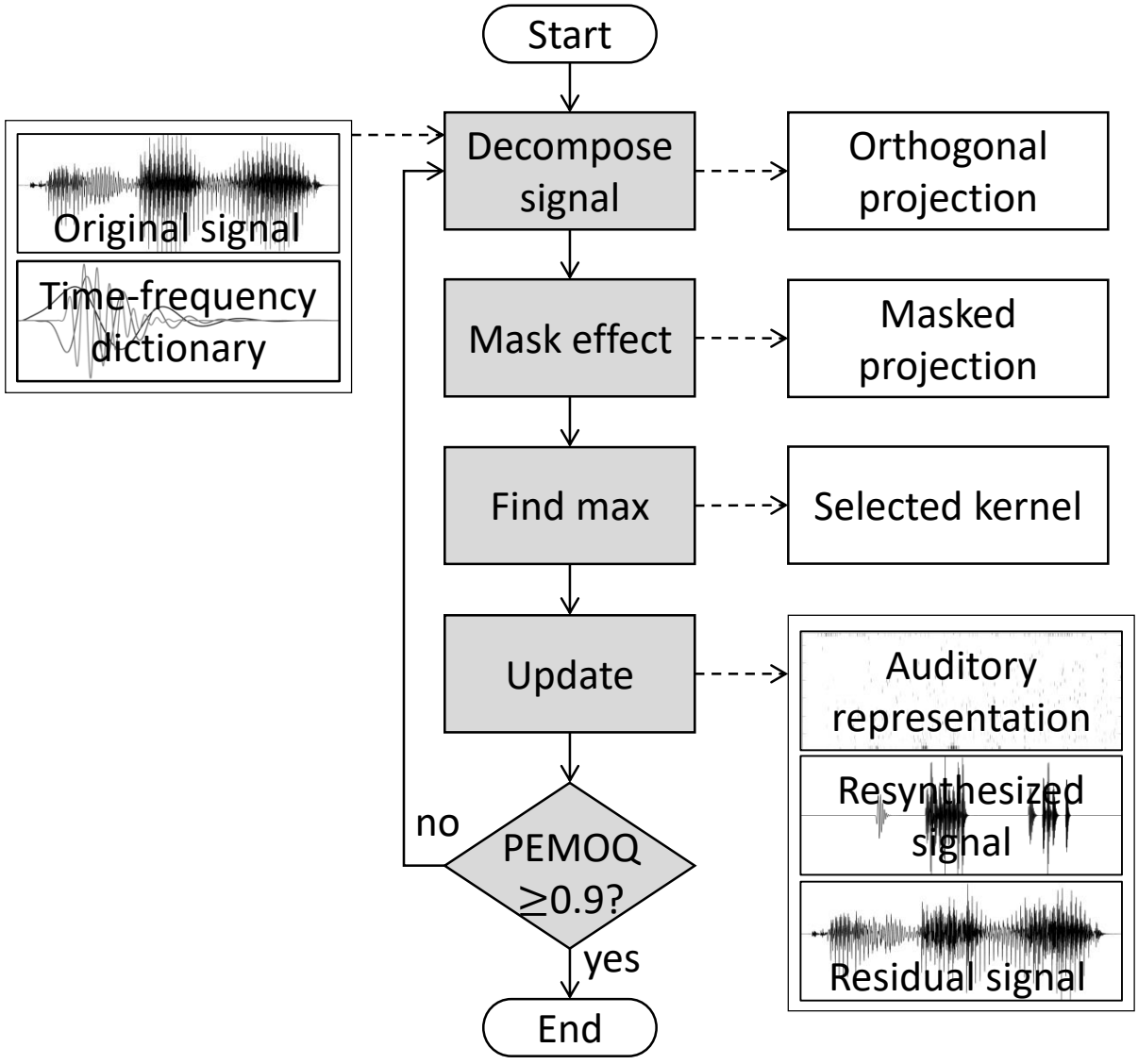

Figure 3.3: Processing pipeline of perceptual matching pursuit algorithm evaluated in this work to derive auditory representations from speech signals.

3.4 Matching pursuit algorithms

3.4.1 Orthogonal matching pursuit

The orthogonal matching pursuit algorithm was introduced by Mallat and Zhang in 1993 to decompose a signal into a linear combination of elementary waveforms [34]. The paper provides proof that the MP algorithm can decompose any function $f(t) \in L^2(R)$ into a linear expansion as follows:

$$f(t) = \sum_{n=0}^{+\infty} \langle R^n f(t), g_n(t) \rangle \, g_n(t), \tag{3.7}$$

where $g_n(t) \in \mathfrak{D}$ is a TF kernel of an overcomplete dictionary $\mathfrak{D}$, and the coefficient $\langle \cdot \rangle$ is the orthogonal projection of the residual signal $R^n f(t)$ on the kernel $g_n(t)$. Figure 3.2 depicts the decomposition process of the MP algorithm and the main calculation loop (i.e., decompose signal, find max, update, and halt) is depicted with the gray blocks and solid arrows. The MP algorithm starts with calculating the signal decomposition step (input: original signal, time-frequency dictionary; process: decompose signal; output: orthogonal projection of the original signal) as follows:

$$f(t) = \langle f(t), g_0(t) \rangle \, g_0(t) + R^0 f(t). \tag{3.8}$$

The output of this step is the orthogonal projection $\langle f(t), g_0(t) \rangle$ of the input signal $f(t)$ on a kernel $g_0(t) \in \mathfrak{D}$, and $R^0 f(t)$ is the residual signal. The energy reservation equivalent to Eq. (3.8) is as follows:

$$\|f(t)\|^2 - |\langle f(t), g_0(t) \rangle|^2 = \left\| R^0 f(t) \right\|^2. \tag{3.9}$$

The left side of Eq. (3.9) can be interpreted as the similarity between the original signal $f(t)$ and its resynthesized signal. Therefore, it can be seen that the lower the residual energy $\|R^0 f(t)\|^2$, the higher the quality of the resynthesized signal. Thus, in the second step (input: orthogonal projection, process: find max, output: selected kernel), the MP algorithm finds in the orthogonal projection space a kernel $g_0(t) \in \mathfrak{D}$ such that the inner product $|\langle f(t), g_0(t) \rangle|$ is maximum for the residual energy $\|R^0 f(t)\|^2$ to be minimum.

Information about the selected kernel (e.g., time, frequency localization, and magnitude) $g_0(t)$ is used in the update step to produce a spikegram, resynthesized signal, and residual signal. More specifically, the projection of $f(t)$ on the selected kernel $g_0(t)$ (i.e., $|\langle f(t), g_0(t) \rangle|$) becomes one spike on the spikegram, the resynthesized signal is updated as $\bar{f}_n(t) = \langle f(t), g_n(t) \rangle \, g_n(t) + \bar{f}_{n-1}(t)$, and the residual signal is updated as $R^n f(t) = R^{n-1} f(t) - \langle R^{n-1} f(t), g_{n-1}(t) \rangle \, g_{n-t}(t)$.

The next step (input: resynthesized signal, original signal; process: halt; output: yes or no) is applying stopping rules to halt the algorithm; otherwise, the algorithm may perform signal decomposition, find the max, and update the loop infinitely. The original MP algorithm that decomposes a function f in vector space introduces a precision factor ϵ such that:

$$\|f\| - \sum_{n=0}^{p-1} |\langle R^n f, g_n \rangle|^2 \leq \epsilon^2 \|f\|, \tag{3.10}$$

where p is the number of iterations. However, when the input signals are audio or speech, it is more meaningful to choose a more suitable precision factor (e.g., PEMO-Q, PESQ,

LSD, and SNR) to compare the distance between the original signal and the resynthesized signal. Therefore, at the third step of an iteration pth of the MP algorithm, the stopping rule uses a precision factor ϵ to compare the original signal and the resynthesized signal. If the stopping criteria is met, the algorithm halts, and a spikegram, a resynthesized signal of ϵ precision, and a residual signal can be obtained as output. Otherwise, the MP algorithm loops back to the first step (input: residual signal, time-frequency dictionary; process: decompose signal; output: orthogonal projection of the residual signal), and the signal decomposition takes the residual signal $R^p f(t)$ as input and projects it on the space created by the time-frequency kernels of the overcomplete dictionary. In the case when the quality of the resynthesized signals cannot satisfy the precision factor, a maximum number of iterations is chosen to ensure that the MP algorithm does not loop infinitely.

3.4.2 Masking model

Inaudible kernels have to be removed to obtain a better representation model for speech signals. For this reason, a masking model is necessary to separate noises from actual signals. The work reported in [21] employs a masking model that can remove unnecessary components effectively. In the present study, a similar model is applied to calculate masking patterns. The forward masking and backward masking patterns created by a selected kernel are calculated by multiplying the masking threshold with masking curves.

$$m_f = \theta \delta_f, \tag{3.11}$$

$$m_b = \theta \delta_b. \tag{3.12}$$

Equations (3.11) and (3.12) show the calculations for the forward masking and backward masking patterns of a selected kernel, where $m_f, m_b, \theta, \delta_f$, and δ_b are the forward masking pattern, backward masking pattern, masking threshold, forward masking curve, and backward masking curve, respectively.

Masking threshold

At an arbitrary iteration ith of the matching pursuit algorithm, the masking threshold caused by the selected kernel at this iteration is calculated by:

$$\theta(n_i, k) = 10 \log_{10} \left(\frac{a_i^2 p_k^2}{q_k} \right) - 4\Gamma(n_i, k) + 16, \tag{3.13}$$

$$q_k = \Upsilon_k + 10(\log_{10} 200 - \log_{10} d_k), \tag{3.14}$$

where n_i is the temporal position of the selected kernel in ERB k, a_i is the magnitude of the selected kernel, p_k is the peak value of the Fourier transform of the normalized kernel in ERB k, q_k is the elevated threshold of hearing in quiet for the same ERB and is calculated by Eq. (3.14), $\Gamma(n_i, k)$ is the tonality index for the ERB k at the temporal position n_i, Υ_k is the absolute threshold of hearing in ERB k, and d_k, calculated in milliseconds, is the duration of the selected kernel. Figure 3.4 shows an example of the masking threshold created by a selected kernel, $\theta \approx 21.4$ dB.

Forward masking curve

The decay of the forward masking curve is calculated as a logarithmic function of the forward masking duration:

$$\delta_f(k,n) = \alpha_k(\beta_k - \log_{10} n), \tag{3.15}$$

$$\alpha_k = \log_{10}\left(\frac{n_i + 0.1l_k + \rho_k f_s}{n_i + 0.1l_k + 1}\right), \tag{3.16}$$

$$\beta_k = \log_{10}(n_i + 0.1l_k + \rho_k f_s), \tag{3.17}$$

$$l_k = \text{round}(d_k f_s), \tag{3.18}$$

$$\rho_k = 100 \arctan(d_k), \tag{3.19}$$

where d_k in milliseconds and l_k in samples are the duration of the selected kernel, f_s is the sampling frequency, ρ_k in milliseconds is empirically defined as the effective duration of the selected kernel, and $\text{round}(n_i + 0.1l_k + 1) \leq n \leq \text{round}(n_i + 0.1l_k + \rho_k f_s)$. The right curve in Figure 3.4 shows an example of a forward masking curve created by a selected kernel.

Backward masking curve

The backward masking curve is also calculated as a logarithmic function of the backward masking duration:

$$\delta_b = \gamma_k(\log_{10} n - \eta_k), \tag{3.20}$$

$$\gamma_k = \log_{10}\left(\frac{n_i - 1}{n_i - d_b f_s}\right), \tag{3.21}$$

$$\eta_k = \log_{10}(n_i - d_b f_s), \tag{3.22}$$

where $d_b = 3$ (ms) is empirically selected as the effective duration of the backward masking of all kernels, and $n_i - d_b f_i \leq n \leq n_i - 1$. The left curve in Figure 3.4 shows an example of a backward masking curve created by a selected kernel. The gap between the backward masking and forward masking curves is the temporal position n_i of the selected kernel.

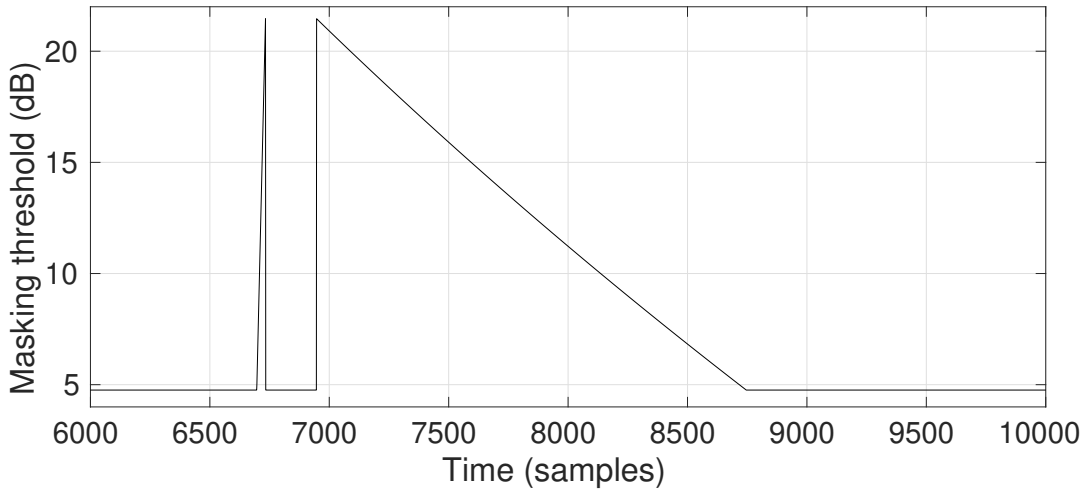

Figure 3.4: An example of masking patterns caused by selected kernel.

Algorithm 1: An orthogonal matching pursuit algorithm evaluated in this work to derive spikegrams from speech signals

Input: Original speech signal $f(t)$, Time-frequency dictionary $\mathfrak{D} = \{g_n(t)\}$
Output: Spikegram W, resynthesized speech signal $\bar{f}(t)$, residual signal $Rf(t)$
Initialization:
$\bar{f}(t) \leftarrow zeros(length(f(t)))$,
$Rf(t) \leftarrow f(t)$,
$C \leftarrow f(t) * \{g_n(t)\}$,
$W \leftarrow zeros(size(C))$,
$MAX\_PEMOQ \leftarrow 0.9$,
$maxi \leftarrow 5000$,
$i \leftarrow 1$

while $i \leq maxi$ **do**
  $(I_f, I_t) = find\_max(C)$
  $g_i(t) = \mathfrak{D}(I_f)$
  $W(I_f, I_t) = W(I_f, I_t) + C(I_f, I_t)$
  $\bar{f}(t) = \bar{f}(t) + C(I_f, I_t) \times g_i(t)$
  $Rf(t) = Rf(t) - C(I_f, I_t) \times g_i(t)$
  $S_{pemoq} = calculate\_PEMOQ(\bar{f}(t), f(t))$
  if $S_{pemoq} \geq MAX\_PEMOQ$ **then**
    break
  $C = Rf(t) * \{g_n(t)\}$
return $W, \bar{f}(t), Rf(t)$

Algorithm 2: A perceptual matching pursuit algorithm evaluated in this work to derive auditory representations from speech signals

Input: Original speech signal $f(t)$, Time-frequency dictionary $\mathfrak{D} = \{g_n(t)\}$
Output: Auditory representation W, resynthesized speech signal $\bar{f}(t)$, residual signal $Rf(t)$
Initialization:
$\bar{f}(t) \leftarrow zeros(length(f(t)))$,
$Rf(t) \leftarrow f(t)$,
$C \leftarrow f(t) * \{g_n(t)\}$,
$W \leftarrow zeros(size(C))$,
$M \leftarrow$ absolute threshold of hearing in quiet,
$C \leftarrow$ apply masking effect M,
$MAX\_PEMOQ \leftarrow 0.9$,
$maxi \leftarrow 5000$,
$i \leftarrow 1$

while $i \leq maxi$ **do**
  $(I_f, I_t) = find\_max(C)$
  $g_i(t) = \mathfrak{D}(I_f)$
  $W(I_f, I_t) = W(I_f, I_t) + C(I_f, I_t)$
  $\bar{f}(t) = \bar{f}(t) + C(I_f, I_t) \times g_i(t)$
  $Rf(t) = Rf(t) - C(I_f, I_t) \times g_i(t)$
  $S_{pemoq} = calculate\_PEMOQ(\bar{f}(t), f(t))$
  if $S_{pemoq} \geq MAX\_PEMOQ$ **then**
    break
  $C = Rf(t) * \{g_n(t)\}$
  $M = update\_masking\_pattern(g_i(t), I_f, I_t)$
  $C = apply\_masking\_effect(M)$
return $W, \bar{f}(t), Rf(t)$

3.4.3 Perceptual matching pursuit algorithm

The original matching pursuit algorithm is energy driven; it decomposes a function in the vector space by selecting the most suitable kernel at each iteration to minimize the residual energy. The resultant spikegrams describe the energy structures of input signals. Moreover, the algorithm does not take into account the masking effects happening in the human auditory system. As a result, many inaudible kernels are also selected during the coding process; thus, the resulting spikegrams contain many unnecessary spikes. For this reason, inaudible kernels should be removed to refine the structures of input signals and to achieve better coding efficiency.

Speech signals are related to perception; thus, it is reasonable to incorporate psychoacoustic principles with the matching pursuit algorithm to guide the selection of the most suitable kernels. The masking model described in section 3.4.2 is utilized to refine the perceptual structure by removing inaudible kernels. Along with using the ERB scale to derive time-frequency kernels and perceptual evaluation (e.g., PEMO-Q or PESQ) as a stopping rule, the masking model makes the perceptual matching pursuit algorithm become perception driven and can produce an auditory representation that conveys the perceptual structures of speech signals.

Figure 3.3 shows the processing steps of the perceptual matching pursuit algorithm. The difference between PMP and MP is that the masking effect step (input: orthogonal projection, selected kernel; process: mask effect; output: masked projection) is applied to the orthogonal projection before the find-max step. At the first iteration of the PMP, the masking patterns are set to the absolute threshold of hearing in quiet. Then, after the find-max step, the masking effect caused by the selected kernel is used to update the masking patterns using the masking model described in section 3.4.2. Then, this new masking surface is applied to the orthogonal projection of the next iteration and so on. The masking model also provides a stopping rule for the PMP. That is, after the masking effect is applied, if there is no non-zero element on the orthogonal projection, the algorithm will halt.

Algorithm 2 describes the implementation of the proposed method. The algorithm takes a speech signal $f(t)$ and a time-frequency dictionary $\mathfrak{D} = \{g_n(t)\}$ as inputs, where n is the number kernels. The outputs of the algorithm are the auditory representation W of the input speech signal, the resynthesized signal $\bar{f}(t)$, and the residual signal $Rf(t)$. Before the main calculation loop, the algorithm initializes the resynthesized signal $\bar{f}(t)$ to be a vector of zeros with the length of the input signal $f(t)$. The residual signal $Rf(t)$ is the input signal $f(t)$. C is the matrix of coefficients resulted by calculating the correlation between the input signal $f(t)$ and the TF dictionary $\{g_n(t)\}$. The auditory representation W is initialized as a matrix of zeros with the same size as the coefficient matrix C. The masking surface M is set to be the absolute threshold of hearing in quiet and then, this masking surface is applied to the coefficient matrix C to remove the elements that are below the threshold. The stopping criteria of the algorithm are the desired PEMO-Q score $MAX\_PEMOQ = 0.9$ and the maximum number of iterations $maxi = 5000$. In the main calculation loop, the algorithm starts to find the frequency channel I_f and temporal position I_t of the largest coefficient of C with the find-max step. The selected kernel of the ith iteration $g_i(t)$ can be found with the frequency channel index I_f. The auditory representation W is updated by adding the largest coefficient $C(I_f, I_t)$ to the existing W. The resynthesized signal $\bar{f}(t)$ is updated by adding the selected kernel $(C(I_f, I_t) \times g_i(t))$ to the existing $\bar{f}(t)$. The residual signal $Rf(t)$ is updated by subtracting the selected kernel $(C(I_f, I_t) \times g_i(t))$ from the existing $Rf(t)$. In the next step, the algorithm calculates

the current PEMO-Q score S_{pemoq} between the resynthesized signal $\bar{f}(t)$ and the input signal $f(t)$ and then compare the score with the desired PEMO-Q score. If the current score is larger than or equal to the desired PEMO-Q score, the algorithm exits the main calculation loop. Otherwise, the algorithm updates the coefficient matrix C by calculating the correlation between the residual signal $Rf(t)$ and the TF dictionary $\{g_n(t)\}$. The masking surface M is updated using the selected kernel $g_i(t)$ and the spatiotemporal indices (I_f, I_t) as described in section 3.4.2. Then, the updated masking surface is applied to the new coefficient matrix to remove inaudible elements. Then, the algorithm returns to the find-max step to find the next largest coefficient. Finally, when the desired PEMO-Q score is satisfied or the maximum number of iteration $maxi$ is reached, the algorithm returns the auditory representation W, the resynthesized signal $\bar{f}(t)$, and the residual signal $Rf(t)$; then halts.

Chapter 4

Perceptual Features of Auditory Sparse Representation

4.1 Summary

In this chapter, evaluations of the auditory sparse representations are conducted. The representations aim to mimic the NAPs; however, real NAPs of the auditory nerves are unavailable to compare to. Therefore, a speech analysis/synthesis experiment is conducted to evaluate the effectiveness of the representations in capturing significant perceptual features of speech signals.

4.2 Evaluation conditions

The effectiveness of an auditory representation can be evaluated in terms of three aspects: the higher the quality of the resynthesized speech signals, the better, the lower the number of non-zero elements, the better, and the ability to represent perceptual structures of speech signals. The first two aspects represent the trade-off between quality and coding data. Reducing the number of coefficients of a signal representation may lead to a reduction in the quality of the resynthesized signals while keeping an exceeding amount of coding data that may make a minor contribution to the quality of the resynthesized signals. An experiment was conducted to compare the signal reconstruction quality of spectrograms, spikegrams, and auditory representations.

4.3 Experiment setups

For the first experiment, parameters were prepared to create 11 sets of conditions to evaluate and compare: spectrograms created by a Gabor (GB-FB), a gammatone (GT-FB), and a gammachirp (GC-FB) filterbank, spikegrams created by an orthogonal matching pursuit algorithm with a damped sinusoid (MP-DS), a Gabor (MP-GB), a gammatone (MP-GT), and a gammachirp (MP-GC) dictionary, and auditory representations created by a perceptual matching pursuit algorithm with a damped sinusoid (PMP-DS), a Gabor (PMP-GB), a gammatone (PMP-GT), and a gammachirp (PMP-GC) dictionary. 630 speech samples spoken by 630 different speakers drawn from the TIMIT database [57] were used in the processes of creating the spectrograms, spikegrams, auditory representations, and resynthesized speech signals. Regarding the first evaluation aspect, PEMO-Q [58],

PESQ MOS score [59], SNR, and LSD were used to compare the distance between original speech signals and resynthesized speech signals. With regards to the second evaluation aspect, the sparseness or the amount of coding data of spectrograms, spikegrams, and auditory representations was evaluated by calculating the number of non-zero elements per second. Figure 4.1 illustrates an example of a speech signal in time domain and different kinds of its representations in time-frequency domain. Panel (a), (b), (c), and (d) are a speech signal in time domain, a spectrogram produced by a gammachirp filterbank, a spikegram by an MP-GC, and an auditory representation by a PMP-GC, respectively.

All the algorithms we used were composed in MATLAB 2020a environment including Signal Processing Toolbox version 8.4. Our experiments were conducted on a typical contemporary computer with an Intel Core i9-7900X CPU @ 3.30 GHz, 16 GB RAM, 500 GB solid-state drive, and Ubuntu 18.04.6 64-bit long-term support operating system.

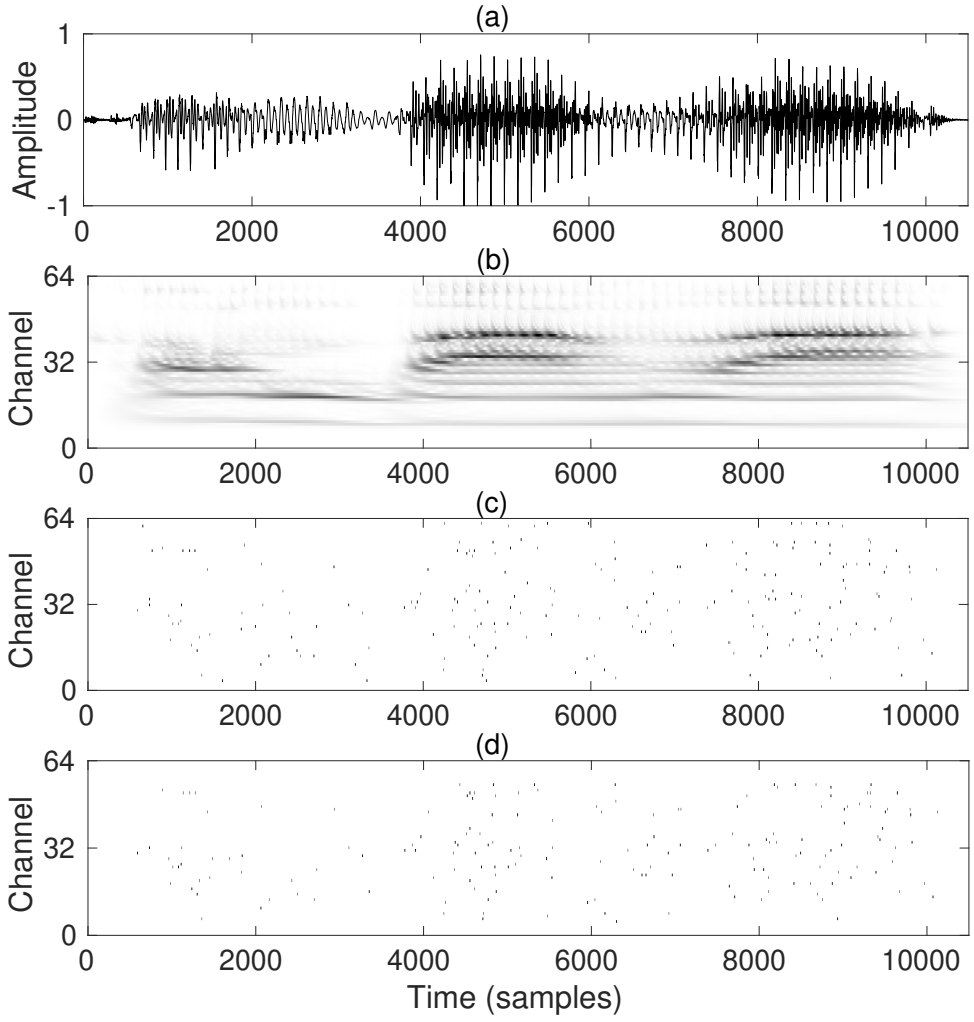

Figure 4.1: A speech signal and its representation models. Panel (a), (b), (c), and (d) are a speech signal in time domain, a spectrogram produced by a gammachirp filterbank, a spikegram by an MP-GC, and an auditory representation by a PMP-GC, respectively.

Table 4.1: Results of the speech analysis/synthesis experiment.

Spectrogram—filterbank (FB)				
		GB	GT	GC
PEMOQ		0.98 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
PESQ		4.38 ± 0.07	3.89 ± 0.13	4.16 ± 0.09
LSD		1.98 ± 0.18	1.28 ± 0.14	1.43 ± 0.16
SNR		6.38 ± 1.97	18.25 ± 1.92	21.39 ± 3.68
Rate		$m \times n$	$m \times n$	$m \times n$
Spikegram—orthogonal matching pursuit (MP)				
	DS	GB	GT	GC
PEMOQ	0.87 ± 0.04	0.89 ± 0.01	0.89 ± 0.01	0.89 ± 0.01
PESQ	2.93 ± 0.36	3.12 ± 0.37	3.20 ± 0.36	3.24 ± 0.31
LSD	102.34 ± 52.01	74.52 ± 45.72	77.15 ± 45.24	79.53 ± 45.78
SNR	17.91 ± 2.48	22.21 ± 3.26	21.92 ± 3.28	20.99 ± 3.31
Rate	1484	1206	1163	1151
Auditory representation—perceptual matching pursuit (PMP)				
	DS	GB	GT	GC
PEMOQ	0.80 ± 0.08	0.90 ± 0.00	0.89 ± 0.01	0.89 ± 0.02
PESQ	2.53 ± 0.27	3.17 ± 0.35	3.23 ± 0.31	3.27 ± 0.29
LSD	135.42 ± 57.71	75.64 ± 46.02	79.83 ± 46.97	81.60 ± 47.34
SNR	14.49 ± 1.38	22.27 ± 3.04	20.81 ± 2.51	19.73 ± 2.58
Rate	905	1238	1093	1066

4.4 Results and discussions

The evaluation results of the spectrograms, spikegrams, and auditory representations are shown in the top panel, the middle panel, and the bottom panel of Table 4.1, respectively. PEMO-Q, PESQ MOS, LSD, and SNR scores are in pairs of (mean score $\pm$ mean standard deviation). It can be seen that the signal representation methods achieved similar scores in PEMO-Q, PESQ MOS, and SNR. Arguably, the spikegrams and auditory representations provided slightly lower scores because a PEMO-Q score of larger-than-or-equal to 0.90 is used as one of the stopping rules in the MP and PMP algorithms. For this reason, the spikegrams and auditory representations provided similar evaluation scores with only less than 1,500 non-zero elements per second, while the spectrograms, though providing slightly higher scores, required entire $(m \times n)$ coefficients to represent speech signals, where m and n are the number of channels of the spectrogram and sampling frequency of the input signals, respectively. In terms of LSD, the spikegrams and auditory representations provided significantly lower scores in comparison with the spectrograms; this could be the result of the lower number of coding data. However, speech signals were used in this experiment, so it is more reasonable to compare the quality by using perceptual evaluations, e.g., PEMO-Q and PESQ MOS. At this point, it appears that sparse representations are preferable to spectrogram representations in representing speech signals.

In terms of the optimal kernel, the perceptual evaluation results in Table 4.1 show that the DS kernel had the lowest performance. More specifically, the MP-DS provided the lowest quality with the highest spike rate in comparison with the other kernels. In the case of PMP-DS, the algorithm could not reach the perceptual stopping rule (PEMO-Q ≥ 0.90). These results suggest that DS might not be an optimal kernel for speech signals. Among the spikegrams and auditory representations created by MP and PMP with GB, GT, and GC, it can be seen that the PEMO-Q scores were similar because PEMO-Q ≥ 0.90 was used as one of the stopping rules. However, the PESQ scores and rate of non-zero elements had opposite directions in the order of GB, GT, GC. More precisely, the PESQ score increased in the order of GB $\rightarrow$ GT $\rightarrow$ GC, and the rate of non-zero elements decreased in the order of GB $\leftarrow$ GT $\leftarrow$ GC. It can be observed that the gammachirp kernel provided a higher perceptual quality with a lower rate in comparison with the other kernels. With regards to the first and the second evaluation aspects, it appears that auditory representations created by PMP-GC provided the highest perceptual quality and the lowest number of non-zero elements per second.

In terms of masking effects, comparing spikegrams created by MP-GT and MP-GC with auditory representations created by PMP-GT and PMP-GC, it can be seen that these methods provided similar evaluation results; however, the PMP produced a lower number of non-zero elements. This means that the masking effects helped to remove a number of unnecessary elements in the auditory representation. The masking model described in section 3.4.2 was designed specifically for gammatone and gammachirp kernels; therefore, its performance is unstable with DS and GB kernels.

Chapter 5

Unique Patterns of Auditory Sparse Representation

5.1 Summary

In this chapter, unique patterns on auditory sparse representations are evaluated. In Chapter 4, it has been verified that the representations contain important perceptual features; however, the uniqueness of the patterns has to be evaluated. It means that if several speech signals are produced by the same speaker speaking the same linguistic content, the patterns on their representations should be similar. Therefore, a general landmark-based hashing technique is used in this chapter to evaluate the patterns. The NAPs should cary different kinds of patterns related to different information such as speaker individuality, linguistic content, emotions, etc. Specialized algorithms should be constructed for specific tasks to achieve optimal performance.

5.2 Landmark-based pattern analysis technique

Perceptual patterns on auditory representations are built by using a landmark-based analysis technique [20]. In this technique, each spike on an auditory representation is regarded as an anchor point or a landmark and paired with subsequent landmarks to become spike patterns. The pairing criteria used in [5] invloves two steps that are building a target zone and selecting matched landmarks.

Each landmark has two important information that are current temporal location (t_1), and current frequency location (f_1). A temporal interval Δt is selected to determine the width of the target zone and frequency interval Δf is selected to determine the height of the target zone. Thus, the target zone with an area of $(t_1 + \Delta t) \times (f_1 \pm \Delta f)$ is used to search for subsequent landmarks for the current landmark.

Among the matched landmarks in the target zone, a number of k landmarks that are nearest in time to the current landmark are chosen to form the perceptual patterns. This process is repeated until the end of the auditory representation. Figure 5.1 illustrates the process of this technique. The resulting perceptual patterns can be regarded as a graph with vertices are the spikes and edges are the pairs among the spikes.

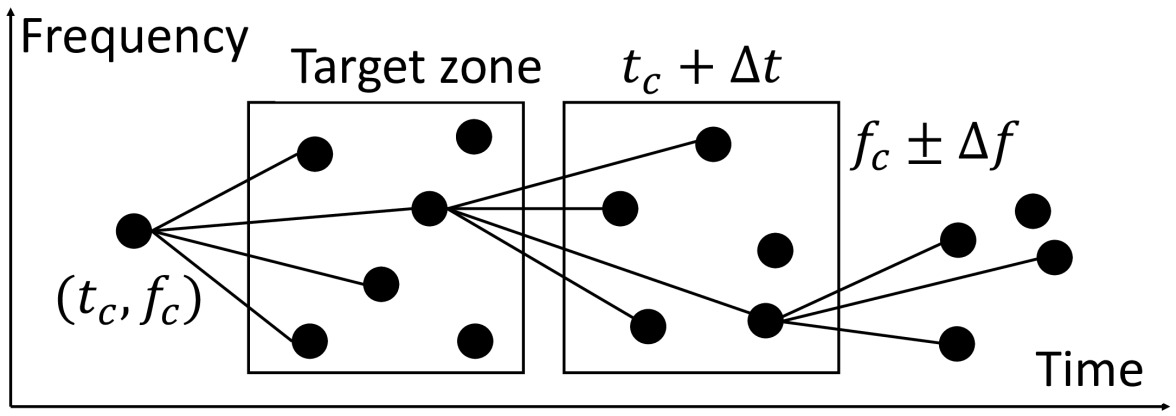

Figure 5.1: An example of landmark-based pattern analysis.

5.3 Hashing technique

Cosidering a landmark $L_1(t_1, f_1)$ and its matched subsequent landmark $L_2(t_1, f_2)$, information about this matched pair is kept as a vector as follow $[t_1 f_1 f_2 \Delta t]$, where $\Delta t = t_2 - t_1$. Together with other matched pairs, the entire graph (or entire perceptual patterns) is encoded into a matrix consisting of 4 columns that are start time column t_1, start frequency column f_1, end frequency column f_2, and delta time column Δt.

The hash table consists of 2 columns, the first column is the start time column t_1. The second column is calculated with the following formula uint32 $(F1 + \Delta f + \Delta t)$.

Eventually, an auditory representation is transformed into a graph representing its perceptual patterns by landmark-based analysis technique, and then this graph is converted into a hash sequence by using the hashing technique.

5.4 Evaluation conditions

The purpose of this section is to evaluate the effectiveness of auditory representations in conveying the perceptual structures of speech signals. It is argued that sparse representation is preferable to spectrogram representation because the structures of signals can be emphasized by the non-zero elements of sparse representation, which is thus beneficial to pattern recognition. Therefore, an experiment was conducted to compare the perceptual structures of auditory representations, spike patterns of spikegrams, and patterns of spectrograms.

5.5 Experiment setups

An experiment was conducted to evaluate and compare perceptual structures in auditory representations with spike patterns in spikegrams and patterns in spectrograms. Robust landmark-based audio fingerprinting (RLBAF) [5] was employed for this purpose. RLBAF is designed to obtain unique structures of music signals known as audio fingerprints. These unique structures are compared to identify short excerpts of music in a music fingerprint database. The algorithm of this application is believed to be the computational routines of a successful commercial service [20]. Therefore, this application is employed to evaluate perceptual structures of auditory representations.

The speech data used in this experiment was created by three female and three male speakers uttering three pieces of speech content: "ohayo gozaimasu," "konichiwa," and "kombanwa." Ten utterances by each speaker and each speech content (180 utterances in total) were used for creating a speech fingerprint database. Spectrogram representations, spikegram representations, and auditory representations of this speech database were created. Then, RLBAF was used to obtain the structures in these three kinds of representations, and, these structures were then hashed and kept as a speech fingerprint database. In the same manner, another 18 utterances were used for querying the database. Given an arbitrary query, for instance, "konichiwa" spoken by female number 2, let us see if the structures of this query can be used to recover ten other "konichiwa" utterances produced by the same speaker in the fingerprint database.

Table 5.1: Confusion matrix illustrates matching results produced by using GC auditory sparse representations as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13
Ain't That a Bitch 01	1												
Attitude Adjustment 02		1											
Crash 03			1										
Fallen Angels 04				1									
Falling in Love 05					1								
Full Circle 06						1							
Hole in My Soul 07							1						
Kiss Your Past Good Bye 08								1					
Nine Lives 09									1				
Pink 10										1			
Something's Gotta Give 11											1		
Taste of India 12												1	
The Farm 13													1

Table 5.2: Pattern matching results produced by using landmark-based pattern analysis.

#	Methods	Mean recall	Mean precision	Mean F1 score
1	STFT-FB	0.15 ± 0.21	0.42 ± 0.45	0.19 ± 0.22
2	GB-FB	0.57 ± 0.28	0.11 ± 0.20	0.11 ± 0.19
3	GT-FB	1.00 ± 0.00	0.10 ± 0.10	0.16 ± 0.14
4	GC-FB	1.00 ± 0.00	0.06 ± 0.07	0.11 ± 0.12
5	MP-GB	0.92 ± 0.13	0.57 ± 0.23	0.68 ± 0.17
6	MP-GT	0.94 ± 0.13	0.60 ± 0.21	0.70 ± 0.16
7	MP-DS	1.00 ± 0.00	0.61 ± 0.24	0.73 ± 0.19
8	MP-GC	1.00 ± 0.00	0.68 ± 0.22	0.78 ± 0.17
9	PMP-GB	0.91 ± 0.13	0.59 ± 0.22	0.69 ± 0.17
10	PMP-GT	0.93 ± 0.14	0.62 ± 0.21	0.71 ± 0.16
11	PMP-DS	1.00 ± 0.00	0.63 ± 0.18	0.76 ± 0.13
12	PMP-GC	1.00 ± 0.00	0.66 ± 0.21	0.77 ± 0.16

Table 5.3: Confusion matrix illustrates matching results produced by using STFT spectrograms as input for RLBAF application. Each row of leftmost column represents ten utterances produced by speaker uttering speech content. Top row represents 18 labels corresponding to 3 male, 3 female speakers, and 3 pieces of speech content.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	2																	
(Ohayo, Female2) 02		2																
(Ohayo, Female3) 03																		
(Ohayo, Male1) 04																		
(Ohayo, Male2) 05					2													
(Ohayo, Male3) 06																		
(Konichiwa, Female1) 07																		
(Konichiwa, Female2) 08								1										
(Konichiwa, Female3) 09																		
(Konichiwa, Male1) 10										4						2		
(Konichiwa, Male2) 11											2							
(Konichiwa, Male3) 12																		
(Kombanwa, Female1) 13																		
(Kombanwa, Female2) 14																		
(Kombanwa, Female3) 15							3								5			
(Kombanwa, Male1) 16		1														1		1
(Kombanwa, Male2) 17																		
(Kombanwa, Male3) 18																		2

Table 5.4: Confusion matrix illustrates matching results produced by using PMP-GC auditory representations as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	3						1						1	4	1			
(Ohayo, Female2) 02		6						2					1	1				
(Ohayo, Female3) 03			9												1			
(Ohayo, Male1) 04				5						2	2						1	
(Ohayo, Male2) 05					6										1		3	
(Ohayo, Male3) 06		1				7					1						1	
(Konichiwa, Female1) 07	1	1					6						2					
(Konichiwa, Female2) 08		1					3	5					1					
(Konichiwa, Female3) 09			3						5						2			
(Konichiwa, Male1) 10				2						4						4		
(Konichiwa, Male2) 11					1						4						5	
(Konichiwa, Male3) 12												8						2
(Kombanwa, Female1) 13	2												5	3				
(Kombanwa, Female2) 14		1												9				
(Kombanwa, Female3) 15									1						9			
(Kombanwa, Male1) 16																10		
(Kombanwa, Male2) 17																	10	
(Kombanwa, Male3) 18						1												9

5.6 Results and discussions

Before discussing the main results with speech signals, a small experiment was conducted to verify if the proposed method was effective with music. The RLBAF application described in the previous section had three parts that were STFT spectrogram as input, land-mark based pattern analysis, and hashing method. In this experiment, the STFT spectrogram was replaced by the proposed auditory sparse representation as input and the other parts were kept the same. The music database used in this experiment was the same with one used by the RLBAF including 13 music tracks. For each track, a 6-second segment was selected randomly and used as a query. The result of this experiment is shown as a confusion matrix on Table 5.1. It can be seen on the diagonal line of the confusion matrix that the recall and precision are 100%. It can be concluded that the proposed auditory sparse representation is as effective as the baseline system in identifying music.

Table 5.2 summarizes the matching results of the main experiment with speech signals. Rows 1 to 4, 5 to 8, and 9 to 12 of the table correspond to the results obtained by using spectrograms, spikegrams, and auditory representations as inputs for the RLBAF application, respectively. The results are illustrated in pairs of (mean score $\pm$ mean standard deviation). In general, the matching results produced by using spectrograms were significantly lower than those of the spikegrams and auditory representations. Table 5.3, 5.5, 5.6, 5.7 show the confusion matrices of matching results produced by using spectrograms as input for RLBAF application. The precisions of these cases were remarkably low; the highest case was only 11%, which means that most of the results were incorrect. Although the precision of STFT-FB was the highest, the recall was the lowest, only 15%. Table 5.3 shows a confusion matrix of this case. The leftmost column is the speech fingerprint database, and each row of this column represents ten speech fingerprints corresponding to ten speech utterances. The top row of the table represents 18 labels corresponding to the 6 speakers and the 3 pieces of speech content, and correct matches are shown on the diagonal cells of the table. Because the recall in this case was very low, the cells are almost empty. The highest result in this case is presented in row #15; the query used in this case was produced by female number 3 speaking "kombanwa." The recall was eight matches over ten matches in total; however, there are only five correct matches over ten in total as shown in column #15, and there are three incorrect matches shown in column #07, that is, the label corresponding to female 1 speaking "konichiwa." Although the recalls of GB-FB, GT-FB, and GC-FB were higher than those of STFT-FB, 100% with GT-FB and GC-FB, the precisions and the F1 scores were the lowest of all cases. It can be seen that using spectrograms as inputs for the RLBAF application, their patterns resulted in either low or highly incorrect detection.

The matching results obtained by using spikegrams and auditory representations as inputs for the RLBAF are summarized in rows 5 to 8 and rows 9 to 12 in Table 5.2, respectively. Table 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.4 show the confusion matrices of matching results produced by using spikegrams and auditory representations as input for RLBAF application. It can be seen that the sparse representations provided significantly higher precision and F1 scores in comparison with the spectrogram representations. At this point, it appears that sparse representations are also preferable to spectrogram representations in representing the structures of speech signals.

In terms of the optimal kernel, examining the matching results produced by using spikegrams and auditory representations, the results show a similar pattern. The recall

produced by the DS and GC kernels was the highest, by the GT kernel the second highest, and by the GB kernel the lowest. The precisions and the F1 scores were in an increasing order of GB $\rightarrow$ GT $\rightarrow$ DS $\rightarrow$ GC. It appears that the GC kernel provided the highest recall, precision, and F1 score. Table 5.4 shows a confusion matrix produced by using PMP-GC. Examining the diagonal cells of the table, it can be seen that the precision was above 50% in most of the cases, and the seven highest results are shown in rows 3, 12, 14, 15, 16, 17, and 18, where correct matches were at least eight out of ten in total. Taking a closer look at the case shown in row 12, the query used in this case was produced by male number 3 speaking "konichiwa." There were two incorrect matches shown in column #18, that is, the label corresponding to male number 3 speaking "kombanwa." This means that although the pieces of speech content were incorrect, the speaker was the same. This holds for all of the seven highest results listed above. With regards to the evaluation conditions, it appears that the spikegrams and auditory representations created by the GC kernels provided the highest matching results.

5.7 Conclusion

Inspired by the amazing performance of our auditory system in complicated listening tasks, such as speaker identification, speech recognition, and sound localization, in extremely noisy environments. The goal of this study is improving reliability of speech fingerprint used for authenticating speakers and linguistic contents by mimicking the auditory representation. There are three sub-objectives. First, auditory representations are used as the representation model for speech signals by using a matching pursuit algorithm and psychoacoustic principles. Second, perceptual structures on an auditory representation are obtained using a landmark-based technique. Third, a hashing method is used to combine perceptual features to create unique fingerprints.

Auditory representations are derived from speech signals as a way to mimic the neural activity patterns of the auditory nerves. The auditory representation is created by considering the time-frequency resolution, sparse representation, optimal kernel, and masking effects. It was evaluated in terms of three aspects: the higher the quality of the resynthesized speech signals, the better, the lower the number of non-zero elements, the better, and the ability to represent the perceptual structures of speech signals.

The auditory representation has an advantage over the traditional spectrogram representation in that the ERB scale is utilized to control the center frequencies and bandwidths of kernels. This strategy makes the time-frequency resolution of the auditory representation more similar to that of the NAP. Another advantage is that perceptual structures can be emphasized in the auditory representation because it is a sparse representation. Thus, these two advantages make features in auditory representations more similar to that of the NAP. The effectiveness of the auditory representation was evaluated in an analysis by synthesis experiment. The TIMIT dataset and evaluation techniques such as PEMO-Q, PESQ, LSD, and SNR were used to compare the original speech signals and the resynthesized speech signals. The results of this experiment show that although auditory representations require a significantly lower number of non-zero elements, similar perceptual scores can be achieved in comparison with spectrogram representations.

The auditory representation created by a PMP algorithm also has an advantage over the spikegram created by an MP algorithm in that masking effects are employed to remove unnecessary elements of the representation. This way, the perceptual structures in auditory representations are more refined and more similar to the NAP. The effectiveness

of the auditory representation and the spikegram were also compared in the analysis by synthesis experiment. The results showed that even with fewer non-zero elements, the auditory representation can have similar evaluation scores, especially in the case of gammatone and gammachirp kernels.

The perceptual structures of auditory representations and patterns of spectrograms were also compared by using RLBAF. The results of this experiment show that even with significantly fewer non-zero elements, the auditory representation can provide noticeably higher recalls, precision, and F1 scores. This means that perceptual structures contribute highly to the performance of the speech fingerprint algorithm.

Perceptual structures of the auditory representation and patterns of the spikegram were also compared by using RLBAF. The results showed that even with fewer non-zero elements, the perceptual structures of the auditory representation can provide similar recalls, precision, and F1 scores. In general, the unnecessary elements in the spikegram neither contribute to the quality of the resynthesized speech signals nor the performance of the speech fingerprint algorithm.

The auditory representation created by using GC kernels has an advantage over other kernels in that it is most similar to the impulse response measured at the basilar membrane. Thus, the frequency and the temporal masking effects produced by the GC kernel are more similar to auditory masking. Consequently, the perceptual structures of the GC auditory representation are most similar to the NAP. The perceptual structures of the GC auditory representation were compared with those of other kernels by using RLBAF. The results showed that the GC kernel can provide the highest recalls, precision, and F1 scores.

In theory, by examining the four psychoacoustic principles—the time-frequency resolution, sparse representation, optimal kernel, and masking effects; and in practice, by examining the three evaluation aspects, it appears that the auditory representation created by the perceptual matching pursuit algorithm with gammachirp kernel can provide the highest performance of the speech fingerprint algorithm.

Table 5.5: Confusion matrix illustrates the matching results produced by using GB–FB spectrograms as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	2	1	1			1	2	3	2	1	2			2			1	
(Ohayo, Female2) 02		5	1	1			3	1	2	1				2		1	1	
(Ohayo, Female3) 03	5	1	2	1	1	1	1	2	1	2					1			
(Ohayo, Male1) 04	2	1		3				1		4	2			3	1	1		
(Ohayo, Male2) 05	1	1		2		1	2	3	1		3	1			1	1	1	
(Ohayo, Male3) 06	1	1	1	1	1	3	2			2	1		1				2	2
(Konichiwa, Female1) 07	1	1	2		1		5	3						2	1	1		1
(Konichiwa, Female2) 08	1	2						3		3	2			3	1	2		1
(Konichiwa, Female3) 09		1	1	1			2	3	1	1	2	2		1	1	2		
(Konichiwa, Male1) 10		1			1		2	6		1	1			1	1	2	2	
(Konichiwa, Male2) 11	1	1		1		1	2	2	2	2	2			2		1	1	
(Konichiwa, Male3) 12	2			2			1	3	1	3	3			2	1			
(Kombanwa, Female1) 13	2	1		2			1	1		2			4	3		1	1	
(Kombanwa, Female2) 14	1	1		2	2	2		3		1	2		2			2		
(Kombanwa, Female3) 15	3	3	1				2	1			1	1	1	2		1	2	
(Kombanwa, Male1) 16	1			2			2			3	1	1	1	3		1	2	1
(Kombanwa, Male2) 17	2		1	1		1	1	2		1	2	1	1	1	1	1	2	
(Kombanwa, Male3) 18				2	1			3	1	1	2				2	2	1	3

Table 5.6: Confusion matrix illustrates the matching results produced by using GT–FB spectrograms as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	1					2	1			1	1			1	1		2	
(Ohayo, Female2) 02		1	1			2			1	1	1		1	1	1			
(Ohayo, Female3) 03			1	2				1	2	2	1				1			
(Ohayo, Male1) 04				4	1				1	1	1				1		1	
(Ohayo, Male2) 05	1		1			1		1	1	1			1	1			2	
(Ohayo, Male3) 06	1					1			1	3	1		1			1	1	
(Konichiwa, Female1) 07	2	3					1	1	1	1				1				
(Konichiwa, Female2) 08	1		3			2	1		2						1			
(Konichiwa, Female3) 09		2	1	1	1	1	1		1		1						1	
(Konichiwa, Male1) 10								1	1	1	1				1	2	1	2
(Konichiwa, Male2) 11				1			2				1		1		1	2	1	1
(Konichiwa, Male3) 12				1	2				1	1			1	2			1	1
(Kombanwa, Female1) 13	1	3					1	1					1	1		1	1	
(Kombanwa, Female2) 14				1			2		1				1			2	2	1
(Kombanwa, Female3) 15		1				2	2			2				1		1	1	
(Kombanwa, Male1) 16	1	1				2		2	1						1	1		1
(Kombanwa, Male2) 17		2		1			1	1		3				1			1	
(Kombanwa, Male3) 18						1		1			1				1	2	1	3

Table 5.7: Confusion matrix illustrates the matching results produced by using GC–FB spectrograms as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01		2	2		1		1		1	1	1		1					
(Ohayo, Female2) 02		2	1	1	2		1	2								1		
(Ohayo, Female3) 03					3		1	1		1	1	1				1		1
(Ohayo, Male1) 04			1		2			1			1			1		1	1	2
(Ohayo, Male2) 05		1	1		2					2	1					1	1	1
(Ohayo, Male3) 06	1		2					1	1	1	1			1		1	1	
(Konichiwa, Female1) 07	1				1		1		1		3	1				1		1
(Konichiwa, Female2) 08		3	1				1	1			1		1	1		1		
(Konichiwa, Female3) 09	2		1	1		1		1		1	1		1				1	
(Konichiwa, Male1) 10	1	1		1	1									3		2	1	
(Konichiwa, Male2) 11		1	1		2		1		1	1				1		1		1
(Konichiwa, Male3) 12		2		1	2			2			1							2
(Kombanwa, Female1) 13	1	1	1					2					1	2	1			1
(Kombanwa, Female2) 14		3			1	1		3			1						1	
(Kombanwa, Female3) 15			2		1	1	1		1	1	1				1		1	
(Kombanwa, Male1) 16			1		2	1		1			1			2		1	1	
(Kombanwa, Male2) 17			1		3		1		1		1					1	1	1
(Kombanwa, Male3) 18		1	2				1	1	1	1						2		1

Table 5.8: Confusion matrix illustrates the matching results produced by using MP–DS spikegrams as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	5						1		1					1		1		1
(Ohayo, Female2) 02		10																
(Ohayo, Female3) 03			9						1									
(Ohayo, Male1) 04				5						4		1						
(Ohayo, Male2) 05					7					1						1		1
(Ohayo, Male3) 06						9										1		
(Konichiwa, Female1) 07		1					6	1					1		1			
(Konichiwa, Female2) 08		1					1	6						2				
(Konichiwa, Female3) 09			1						4			3				1		1
(Konichiwa, Male1) 10				1						9								
(Konichiwa, Male2) 11											4	4						2
(Konichiwa, Male3) 12												6						4
(Kombanwa, Female1) 13	2						2	1					5					
(Kombanwa, Female2) 14	1	1					1	3						4				
(Kombanwa, Female3) 15															10			
(Kombanwa, Male1) 16				3												6	1	
(Kombanwa, Male2) 17				1						1	2					3	2	1
(Kombanwa, Male3) 18												2						8

Table 5.9: Confusion matrix illustrates the matching results produced by using MP–GB spikegrams as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	5			1				1						2	1			
(Ohayo, Female2) 02		6											1	3				
(Ohayo, Female3) 03			3						2	1	1			1	1		1	
(Ohayo, Male1) 04				5						2								
(Ohayo, Male2) 05					6			1				1			1		1	
(Ohayo, Male3) 06					2	4			1		1						1	1
(Konichiwa, Female1) 07							5	1						3				1
(Konichiwa, Female2) 08		3					4	2						1				
(Konichiwa, Female3) 09		3							4			1			1			
(Konichiwa, Male1) 10	1			2						2	1							
(Konichiwa, Male2) 11											3						4	
(Konichiwa, Male3) 12											1	6						3
(Kombanwa, Female1) 13	1							2					4	3				
(Kombanwa, Female2) 14	1	1											2	6				
(Kombanwa, Female3) 15													1		9			
(Kombanwa, Male1) 16																10		
(Kombanwa, Male2) 17											1						9	
(Kombanwa, Male3) 18																		7

Table 5.10: Confusion matrix illustrates the matching results produced by using MP–GT spikegrams as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	7							1						2				
(Ohayo, Female2) 02	1	7						1						1				
(Ohayo, Female3) 03		1	4		1				2				1		1			
(Ohayo, Male1) 04				7						1						2		
(Ohayo, Male2) 05					6						2						1	1
(Ohayo, Male3) 06		2				4					1	1			1		1	
(Konichiwa, Female1) 07		1					3	3					2	1				
(Konichiwa, Female2) 08		2					3	2						2				1
(Konichiwa, Female3) 09			3			1			4									
(Konichiwa, Male1) 10				3						4						3		
(Konichiwa, Male2) 11											5							
(Konichiwa, Male3) 12												7						3
(Kombanwa, Female1) 13							1	2					5		2			
(Kombanwa, Female2) 14							1	1					1	7				
(Kombanwa, Female3) 15															10			
(Kombanwa, Male1) 16										1						9		
(Kombanwa, Male2) 17					3												4	
(Kombanwa, Male3) 18						1						2						7

Table 5.11: Confusion matrix illustrates the matching results produced by using MP–GC spikegrams as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	3						1						1	4	1			
(Ohayo, Female2) 02		7						2						1				
(Ohayo, Female3) 03			9												1			
(Ohayo, Male1) 04				6						1	2					1		
(Ohayo, Male2) 05					7			1									2	
(Ohayo, Male3) 06						8					1						1	
(Konichiwa, Female1) 07		1					6	1					2					
(Konichiwa, Female2) 08		3					3	3					1					
(Konichiwa, Female3) 09			3						5						2			
(Konichiwa, Male1) 10				2						4						4		
(Konichiwa, Male2) 11					1						4						5	
(Konichiwa, Male3) 12												8						2
(Kombanwa, Female1) 13	1						2	1					6					
(Kombanwa, Female2) 14		1												9				
(Kombanwa, Female3) 15									1						9			
(Kombanwa, Male1) 16																10		
(Kombanwa, Male2) 17																	10	
(Kombanwa, Male3) 18						1												9

Table 5.12: Confusion matrix illustrates the matching results produced by using PMP–DS auditory representations as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	7	1					1						1					
(Ohayo, Female2) 02		8					2											
(Ohayo, Female3) 03			9						1									
(Ohayo, Male1) 04				6						4								
(Ohayo, Male2) 05				2	7						1							
(Ohayo, Male3) 06						9					1							
(Konichiwa, Female1) 07	1						6	2					1					
(Konichiwa, Female2) 08							3	5						2				
(Konichiwa, Female3) 09	2		1				1	1	4						1			
(Konichiwa, Male1) 10				2		1				6						1		
(Konichiwa, Male2) 11											6					1	1	2
(Konichiwa, Male3) 12												7						3
(Kombanwa, Female1) 13	1						2	1					5	1				
(Kombanwa, Female2) 14	2							3						5				
(Kombanwa, Female3) 15															10			
(Kombanwa, Male1) 16		1		2												7		
(Kombanwa, Male2) 17						1						1				4	3	1
(Kombanwa, Male3) 18												3						7

Table 5.13: Confusion matrix illustrates the matching results produced by using PMP–GB auditory representations as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	5												2	2	1			
(Ohayo, Female2) 02		6											1	3				
(Ohayo, Female3) 03			4						2		1			1	1		1	
(Ohayo, Male1) 04				5						2								
(Ohayo, Male2) 05					6			1				1					2	
(Ohayo, Male3) 06						6			1		1						1	1
(Konichiwa, Female1) 07							4	1						4				1
(Konichiwa, Female2) 08		1					5	2						2				
(Konichiwa, Female3) 09		3							4						1			
(Konichiwa, Male1) 10	1			2						2	1							
(Konichiwa, Male2) 11											3						4	
(Konichiwa, Male3) 12											1	6						3
(Kombanwa, Female1) 13								3					4	3				
(Kombanwa, Female2) 14	1	1											2	6				
(Kombanwa, Female3) 15													1		9			
(Kombanwa, Male1) 16																10		
(Kombanwa, Male2) 17											1						9	
(Kombanwa, Male3) 18																		7

Table 5.14: Confusion matrix illustrates the matching results produced by using PMP–GT auditory representations as input for RLBAF application.

#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
(Ohayo, Female1) 01	7							1						2				
(Ohayo, Female2) 02	1	7												2				
(Ohayo, Female3) 03			6						2				1		1			
(Ohayo, Male1) 04				6						1						3		
(Ohayo, Male2) 05	1				6		1										2	
(Ohayo, Male3) 06						7						1			1		1	
(Konichiwa, Female1) 07		1					2	4					2	1				
(Konichiwa, Female2) 08		2					3	3						2				
(Konichiwa, Female3) 09	1		3			1			4									
(Konichiwa, Male1) 10				3						3						3		
(Konichiwa, Male2) 11											5							
(Konichiwa, Male3) 12												7						3
(Kombanwa, Female1) 13							1	1					5	1	2			
(Kombanwa, Female2) 14							1	3						6				
(Kombanwa, Female3) 15															10			
(Kombanwa, Male1) 16										1						9		
(Kombanwa, Male2) 17					2												4	
(Kombanwa, Male3) 18						1						2						7

Chapter 6

Speech Fingerprints Identification Algorithms

6.1 Summary

In this chapter, a speech fingerprints identification algorithm is constructed to evaluate the uniqueness and usefulness of the proposed speech fingerprints. The present study assumes that speech fingerprints are contained in the NAPs of the auditory nerves and experimental results reported in Chapter 4 and 5 show that the auditory sparse representation conveys important perceptual features and contains distinguishable patterns. However, a vast majority of research articles in the literature considers speech fingerprints as binary or real hash sequences. The advantages of the hash sequences are their effectiveness because of the high identification accuracy and their efficiency because of the high indexing speed when dealing with large-scale datasets. Therefore in this chapter, a deep hashing algorithm is constructed to convert the proposed speech fingerprints into hash sequences to employ the aforementioned advantages.

Deep hashing algorithms used for speaker identification and retrieval aim to produce discriminative hash codes for a set of speech signals. The basis of this task is highly related to speaker individuality. However, existing deep speaker hashing algorithms were constructed without considering speaker individuality. Previous studies have demonstrated the importance of speaker individuality in speech analysis and synthesis applications. Furthermore, recent studies have demonstrated the advantages of sparse representations of speech signals over the traditional spectrograms. Therefore, a method is proposed to hash the significant acoustical features related to speaker individuality by using auditory sparse representations. In speaker identification and speaker retrieval experiments with the VoxCeleb2 dataset, 64-bit hash codes—produced by the proposed method—achieved 99.91% in top-1 accuracy and 97.55% in MAP@100, which are highly competitive with other state-of-the-art methods.

The rest of this chapter is organized as follows. Section 6.4.1 describes an algorithm, which incorporates a gammachirp auditory filterbank and an orthogonal matching pursuit, used to produce auditory sparse representations. The representations can be regarded as speech fingerprints. Section 6.4.2 describes a deep hashing method that is equivalent to a speech fingerprint identification algorithm. Section 6.5 describes experiments and evaluations. Section 6.6 reports the experimental results and discusses the uniqueness and effectiveness of the proposed speech fingerprints.

6.2 Introduction

Speaker identification and retrieval are useful for many applications such as automatic access control for various services and automatic detection of speakers in complex scenes. As speech data grows gigantically, speaker identification and retrieval tasks with high accuracy and reasonable speed have become problematic. Deep hashing for speaker identification and retrieval has become very common recently because of its high accuracy and low computational cost when dealing with large-scale datasets [60,61]. Such a system often consists of a pre-processing module and a deep learning to hash algorithm. The effectiveness of a deep hashing method depends on both pre-processing and hashing techniques.

The pre-processing module is responsible for converting raw speech signals into feature vectors, which serve as inputs for a deep hashing algorithm. Short-time Fourier transform spectrograms are often used as the feature vectors [62]. Features related to speaker individuality, i-vector/x-vector, are commonly used in the existing methods [63,64]. Some methods [65,66] use feature enhancement techniques to increase how discriminative hash codes are. However, current pre-processing techniques did not consider perceptual features and speaker individuality inherent in speech signals carefully.

The work in [67] reported that speaker individuality is related to the F0 contour and that speaker individuality can be controlled by using the Fujisaki F0 model. Also, the work reported in [68] regards the speaker individuality of a speaker as a distribution of $\log \mathrm{F0}$. The mean and variance of a $\log \mathrm{F0}$ distribution can be linear scaled to convert speaker individuality. Furthermore, common vocoder systems [69–72] produce F0 contours by searching within the range of 40 to 800 Hz. This means that the range of frequencies from 40 to 800 Hz is very important for speaker individuality.

Several studies [47, 52] in speech-coding methods pointed out that the redundant coding data of spectrogram representations obscured the underlying structures of speech signals. Sparse representations produced by orthogonal matching pursuit (OMP) algorithms were proposed to overcome this issue by focusing on significant features. Furthermore, gammatone/gammachirp auditory filterbanks were used with OMP algorithms to produce auditory sparse representations, which contained perceptual structures of speech signals, to mimic the neural activity patterns (NAPs) of the auditory periphery [21]. Experimental results reported in [21,73] showed that the auditory sparse representations were more beneficial to speech-coding and pattern-analysis applications than spectrogram representations. Therefore, utilizing such auditory sparse representation would increase the effectiveness of a deep hashing algorithm in speaker identification and retrieval tasks.

Speaker identification and retrieval are specialized tasks in the domain of speech; therefore, it is more meaningful to use features that are related to speaker individuality. In this study, we propose a method that maps significant acoustical features related to speaker individuality into hash sequences in the Hamming space. The contributions of our study are listed as follows.

- We focus on the frequency range between 40 and 800 Hz because it is important for speaker individuality.
- We utilize the advantages of an auditory sparse representation using a deep hashing technique. The equivalent rectangular bandwidth (ERB) scale and a gammachirp (GC) function are used to design a GC auditory filterbank; together with an OMP algorithm, speech signals are converted into auditory sparse representations as a

way to mimic NAPs.

- An extensive experiment with the VoxCeleb2 dataset is conducted to evaluate the effectiveness of the proposed method on speaker identification and retrieval tasks.

6.3 Related Works

Deep hashing methods used for speaker identification and retrieval in the literature are largely separated into two categories: classifier-based and feature-based methods.

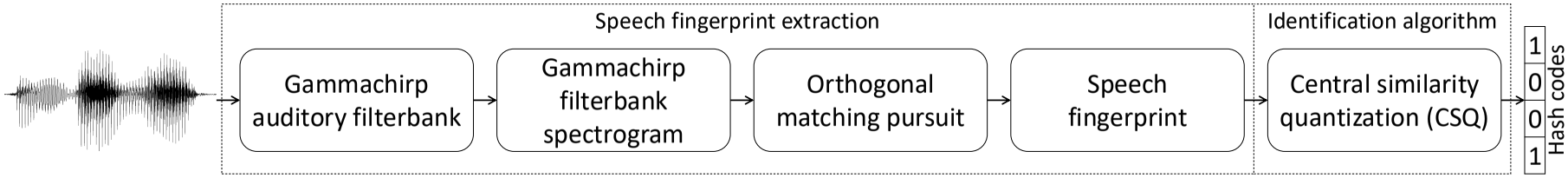

Figure 6.1: Processing pipeline of the speech fingerprint identification system.

6.3.1 Classifier-based Techniques

In this category, the main focus is constructing powerful non-linear classifiers that can tolerate the variations of noisy inputs. Different architectures [74, 75] and loss functions [76–78] have been proposed to learn the distance between data points of a distribution. These techniques concentrate on designing mapping functions, dealing with optimization problems, and increasing quantization qualities. Traditional input features are often used in these methods such as the short-time Fourier transform spectrograms and MFCCs.

Applications in this category pay less attention to the importance of input features inherent in speech signals. Therefore, utilizing the inputs that contain highly distinguishable features would increase the effectiveness of the classifiers.

6.3.2 Feature-based Techniques

In this category, speech signals are usually treated with pre-processing techniques such as voice activity detection and noise reduction to mitigate the interference of noisy datasets. Feature enhancement techniques such as those using log domain features and Mel-frequency cepstral coefficients (MFCCs) focus on important acoustical features [65, 66]. On the basis of low-level acoustical features, data-driven techniques such as i-vector [79], d-vector [80], and x-vectors [81] are also used to increase the inter-class distance and to reduce the intra-class distance. Based on i-vector, several deep hashing techniques were proposed and achieved high performance in speaker identification and retrieval tasks [63,64]. However, these techniques mainly focus on the magnitude spectrum of the subband frequencies of speech signals. Thus, the hash codes are representatives of the energy distribution of the speech signals.

Psychoacoustical and experimental studies have found that speech signals convey latent patterns unique to each speaker [67, 68]. Therefore, it is useful to construct a technique that hashes the inherent speaker individuality. This way, the resultant hash sequences can be regarded as the representatives of speaker identities.

6.4 Proposed Method

Our proposed method consists of two main parts: auditory sparse representation and deep hashing algorithms. Figure 6.1 shows the processing steps of the proposed method. The auditory sparse representation algorithm extracts important acoustical features related to speaker individuality to provide input features. The deep hashing algorithm maps the input features to the hash sequences in the Hamming space.

6.4.1 Auditory sparse representation algorithm

Auditory sparse representation was studied extensively in [73]. In this study, a GC auditory filterbank is used for producing significant acoustical features related to speaker individuality and an OMP algorithm is used for producing sparse representations of speech signals. These two components convert speech signals into sparse codes that are similar to NAPs of the auditory system. The dashed-line rectangle on the left of Fig. 6.1 illustrates the processing steps of this algorithm.

Gammachirp auditory filterbank

The GC auditory filterbank used in this study has three important properties that provide highly accurate information about speaker individuality. The first property is using a frequency range from 40 Hz to 800 Hz because this region conveys information about fundamental frequency F0 [67, 68]. Therefore, acoustical features related to speaker individuality should be derived from this region.

The second property is using the ERB scale to calculate the center frequencies and bandwidths of auditory filters to account for the frequency selectivity characteristic of the auditory system. Moore and Glasberg [53] constructed a formula that describes the relationship between the number of ERBs and frequency as follows:

$$\begin{aligned} \text{Number of ERBs, E} &= 21.4 \log_{10}(4.37 f_c + 1) \\ \Rightarrow f_c &= \frac{1}{4.37}\left(10^{\frac{\text{E}}{21.4}} - 1\right), \end{aligned} \tag{6.1}$$

where f_c(kHz) is the center frequency of an auditory filter, and E is the ERB number on the ERB scale. Furthermore, the formula used to calculate bandwidths of the auditory filters are as follows:

$$\text{ERB}(f_c) = 24.7(4.37 f_c + 1), \tag{6.2}$$

where ERB$(\cdot)$ takes a center frequency f_c and gives a corresponding filter bandwidth. The center frequencies f_c and bandwidths of the GC auditory filterbank are calculated by using Eq. (6.1) and (6.2).

The third property is using a GC function to analyze speech signals. There are several functions that can be used as filter functions such as the exponential, damped sinusoid, Gabor, and gammatone. However, a psychoacoustic study found that the impulse response measured at the basilar membrane has a gamma distribution and an up chirp [50]. Human masking data was also used to derive an auditory filter that simulate this phenomenon [49]. Experimental results in [73] showed that the GC function provided highest perceptual scores. Therefore, the GC function should provide optimal features for analyzing speaker individuality. The GC function is as follows:

$$\gamma(t) = a t^{n-1} e^{-2\pi b \text{ERB}(f_c) t} \cos(2\pi f_c t + c \ln t + \phi), \tag{6.3}$$

where $a, n = 4, b = 1.019, t, c$ ln$(\cdot)$, and ϕ correspond to the amplitude, order of the filter, coefficient of the filter bandwidth, time, chirp factor, and phase, respectively [56]. Panel (a) and (b) of Fig. 6.2 show a three-second speech signal from the VoxCeleb2 dataset and its corresponding GC filterbank spectrogram, respectively.

Orthogonal matching-pursuit algorithm

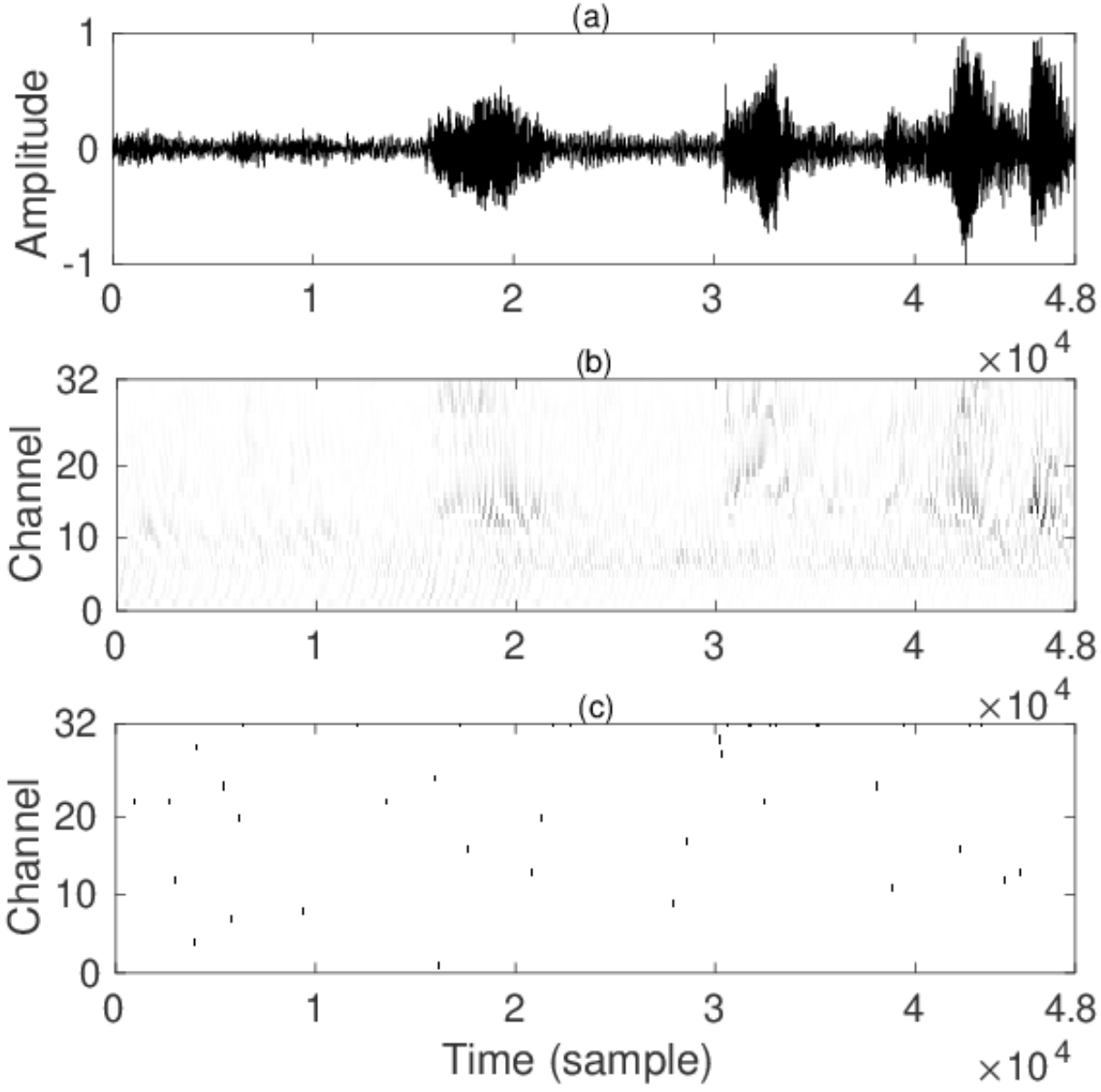

Figure 6.2: Sample output of the auditory sparse representation algorithm. (a) is a speech signal in the VoxCeleb2 dataset, (b) is its GC filterbank spectrogram, and (c) is the corresponding auditory sparse representation.

Algorithm 3: An orthogonal matching pursuit algorithm used in this paper to produce auditory sparse representations of speech signals.

Input: Speech signal $s(t)$, gammachirp filterbank $\Gamma = \{\gamma_n(t)\}$
Output: Auditory sparse representation C
Initialization:
$r(t) \leftarrow s(t)$,
$S \leftarrow s(t) * \Gamma$
$C \leftarrow zeros(size(S))$,
$N \leftarrow 6000$,
$i \leftarrow 1$

while $i \leq N$ **do**
    $\alpha, I_f, I_t = max(S)$
    $C(I_f, I_t) = C(I_f, I_t) + \alpha$
    $\gamma_i(t) = \Gamma(I_f,)$
    $r(t) = r(t) - \alpha \times \gamma_i(t)$
    $S = r(t) * \Gamma$
return C

A sparse representation of a speech signal is a representation that contains a very low number of non-zero elements that are said to be the "geometric" information of the speech signals [9]. Previous studies used OMP algorithms to emphasize the underlying structures of speech signals on the sparse representations [21, 73]. Experimental results of these studies showed that sparse representations outperformed spectrogram representations in perceptual evaluations and pattern analysis applications. Although GC auditory spectrograms convey acoustical features related to speaker individuality, we further refine the features by utilizing the advantages of sparse representations.

Processing steps of the OMP are briefly described in Algorithm 3. In general, the OMP requires a speech signal $s(t)$ and the GC auditory filterbank $\Gamma = \{\gamma_n(t)\}$ as inputs, where n is the number of GC filters; and produces an auditory sparse representation C of the input speech signal.

In the initialization steps before the main calculation loop, the algorithm set a residual signal $r(t)$ to be the same as the input speech signal $s(t)$. Then, it calculates S as the correlation matrix between the input speech signal $s(t)$ and all of the GC filters $\gamma_n(t)$ of the filterbank Γ. Then, it pre-allocates the auditory sparse representation C to be a matrix of zeros with the same dimensions as the correlation matrix S. Finally, the OMP sets the maximum number of iterations N and the counter i to be 6000 and 1, respectively.

The main calculation loop, which has six steps in total, is where the auditory sparse representation C is produced. The steps are as follows:

1. The first step $i \leq N$ is the stopping criterium of the algorithm. Preferably, perceptual scores such as PEMO-Q [58] or PESQ [59] should be used as the stopping criterium to control the trade-off between perceptual quality and number of coefficients. However, performing a perceptual evaluation in each iteration would increase computational complexity of the algorithm significantly, especially when working with large-scale datasets. Therefore, we empirically set $N = 6000$ to obtain 2000 coefficients per second.

2. Next, the algorithm searches on the correlation matrix S for the largest coefficient α, as well as the frequency channel index I_f, and the time offset I_t of α.

3. In the third step, the coefficient at the coordinate (I_f, I_t) of auditory sparse representation C is increased by α.

4. In the fourth step, the GC filter of the ith iteration $\gamma_i(t)$ is extracted from the GC filterbank Γ using frequency channel index I_f.

5. In the fifth step, the energy produced by the coefficient α and the GC filter $\gamma_i(t)$ is removed from the residual signal $r(t)$.

6. In the last step, the correlation matrix S is replaced by the correlations between the residual signal $r(t)$ and the GC filters of the GC filterbank Γ. Then, the algorithm goes back to the first step.

When the maximum number of iterations is reached, the algorithm halts and returns the auditory sparse representation C of the input speech signal. Figure 6.2 (c) shows an example of the auditory sparse representation.

6.4.2 Deep hashing algorithm

Central similarity quantization (CSQ) [82] is a deep learning to hash algorithm, which converts an image into a binary hash sequence. The hash sequences are seen as data points in the Hamming space, and the Hamming distance is used to measure the similarity among them. First, the algorithm calculates a set of predefined hash centers by drawing samples from a Hadamard matrix or a Bernoulli distribution. Then, a deep learning algorithm is trained to produce hash sequences that approach a hash center based on the Hamming distance. As a result, similar images are mapped into a cluster of hash sequences around a pre-defined hash center. Experimental results show that CSQ achieves outstanding performance on the ImageNet [83], MS COCO [84], and NUS_WIDE [85] datasets.

We have used the auditory sparse representation algorithm to derive the inherent speaker individuality of each speaker. Now, we use CSQ to map similar speaker individualities into the same binary hash cluster in the Hamming space. Then, we can perform the speaker identification and retrieval by calculating the Hamming distance between the hash sequences.

Optimization problem of CSQ

The key to the intricacies and effectiveness of CSQ is its loss function to solve a central similarity optimization problem. The optimization problem is presented as follows:

$$\min_{\Theta} L_T = L_C + \lambda_1 L_Q, \tag{6.4}$$

where Θ and λ_1 are parameters of the deep hash function and a hyper-parameter, respectively. L_C is designed based on the well-known maximum a posterior (MAP) estimation and L_Q is introduced as a quantization loss to ensure that the generated hash codes converge to their corresponding hash centers.

Let $\mathcal{X} = \{x_i\}_{i=1}^{N}$ be the training set consisting of N samples, $\mathcal{C} = \{c_1, ..., c_q\}$ be a set of hash centers drawn from a Bernoulli distribution. Because each data points x_i is associated with a hash center, we have the following semantic hash centers of N elements $\mathcal{C}^{'} = \{c_1^{'}, ..., c_N^{'}\}$, where $c_i^{'}$ is the hash center of a data point x_i. Assume that $\mathcal{H} = \{h_i\}_{i=1}^{N}$ is a set of hash codes corresponding to $\mathcal{X}$. $\mathcal{H}$ can be obtained by maximizing the following likelihood probability:

$$\log\ P(\mathcal{H}|\mathcal{C}^{'}) \propto \log\ P(\mathcal{C}^{'}|\mathcal{H})P(\mathcal{H}) = \sum_{i=1}^{N} \log\ P(c_i^{'}|h_i)P(h_i), \tag{6.5}$$

where $P(\mathcal{H})$ is a prior distribution of the hash codes, $P(\mathcal{C}^{'}|\mathcal{H})$ is the posterior probability or the likelihood function, and $P(c_i^{'}|h_i)$ is the conditional probability of the hash center $c_i^{'}$ given the hash code h_i. A Gibbs distribution is used to model $P(\mathcal{C}^{'}|\mathcal{H})$ as follows:

$$P(c_i^{'}|h_i) = \frac{1}{\alpha e^{\beta D_H(c_i^{'}, h_i)}}, \tag{6.6}$$

where α and β are constants, and $D_H(c_i^{'}, h_i)$ is the Hamming distance between a hash code h_i of a data point x_i and its corresponding hash center $c_i^{'}$. It can be seen that the conditional probability $P(c_i^{'}|h_i)$ is inversely proportional to the Hamming distance $D_H(c_i^{'}, h_i)$. Therefore, by maximizing $P(c_i^{'}|h_i)$, the distance $D_H(c_i^{'}, h_i)$ can be minimized, and thus, the hash code h_i of a data point x_i is encouraged to approach its hash center

$c_i^{'}$. Binary cross entropy (BCE) is used to measure the Hamming distance between a hash code and its center,

$$D_H(c_i^{'}, h_i) = \text{BCE}(c_i^{'}, h_i) = -\frac{1}{K}\sum_{k=1}^{K} c_{i,k}^{'} \log\, h_{i,k} + (1 - c_{i,k}^{'}) \log\,(1 - h_{i,k}). \tag{6.7}$$

Substituting Eq. 6.7 to Eq. 6.6 and taking logarithm on both sides, we obtain the following direct proportion: $\log\, P(c_i^{'}|h_i) \propto -\frac{1}{K}\sum_{k=1}^{K} c_{i,k}^{'} \log\, h_{i,k} + (1 - c_{i,k}^{'}) \log\,(1 - h_{i,k})$. Substituting $\log\, P(c_i^{'}|h_i)$ into Eq. 6.5, we arrive at the following central similarity loss function:

$$L_C = \frac{1}{K}\sum_{i=1}^{N}\sum_{k=1}^{K} c_{i,k}^{'} \log\, h_{i,k} + (1 - c_{i,k}^{'}) \log\,(1 - h_{i,k}). \tag{6.8}$$

The second part of the composite loss function, L_Q, is introduced to ensure that the generated hash codes h_i converge to their corresponding hash centers c_i. The bi-modal Laplacian prior for quantization is used for this purpose: $L_Q = \sum_{i \neq j}^{N}(|||2h_i - \mathbf{1}| - \mathbf{1}||_1)$, where $\mathbf{1} \in \mathbb{R}^K$ is the vector of ones. However, this function is a non-smooth function or non-differentiable. This problem can be relaxed by using a smooth surrogate of the absolute function $|x| \approx log\, \cosh\, x$. Consequently, L_Q can be rewritten as

$$L_Q = \sum_{i=1}^{N}\sum_{k=1}^{K} \log\, \cosh\,(|2h_{i,k} - 1| - 1). \tag{6.9}$$

Substituting Eq. 6.8 and 6.9 into 6.4, the composite loss function L_T can be used to optimize the parameters of Θ.

CSQ framework

Figure 6.3 shows the framework of the central similarity quantization. The framework has 4 important parts:

1. At first, the input to the CSQ, $\{(x_i, x_j, c_i, c_j)\}$, is a pair of auditory sparse representations (x_i, x_j) and their corresponding hash centers c_i, c_j, with $i \neq j$. More specifically, each auditory sparse representation, x_i, is a 32×300 matrix extracted from a speech signal by the extraction method described in Section 6.4.1. Each hash center c_i is a binary sequence of length K.

2. Generally, the next part is a convolutional neural network (CNN) used for feature learning. In the present study, four convolutional layers from ResNet50 [86] are used for this part; however, other CNN frameworks such as AlexNet [87] can also be used for the same purpose.

3. The third part is a hash layer—consisting of three fully connected layers and a ReLU activation function—that is used to convert high dimensional features (outputs of the CNN framework) into hash codes corresponding to the input auditory sparse representation.

4. The last part is the optimization strategy of CSQ as described in Section 6.4.2. The composite loss function L_T is used to encourage the generated hash codes to approach their corresponding hash centers.

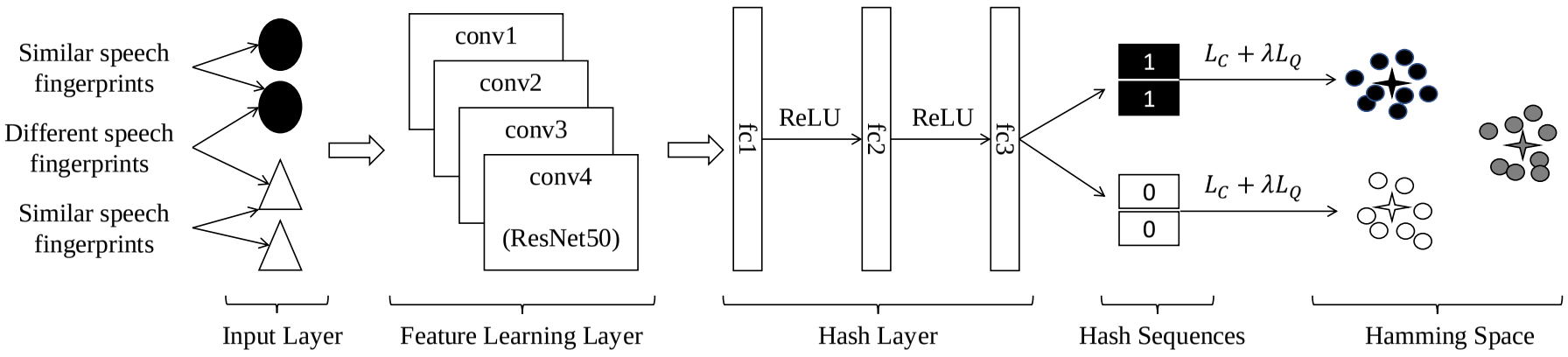

Figure 6.3: Architecture of central similarity quantization.

Table 6.1: Splits of VoxCeleb2 dataset used in speaker identification and retrieval experiments.

	No. Speakers	No. Utterances
Training set	3,641	848,957
Validation set (10 utt./speaker)	3,641	36,410
Test set (20 utt./speaker)	3,641	72,820
Total	3,641	958,187

Table 6.2: Evaluation results of proposed method and other state-of-the-art methods in MAP (%) and Top-1 (%). Real and binary are real-valued and binary-valued hash codes.

Method	Code length	MAP	Top-1
i-vector (binary) [64]	150	—	79.23
x-vector (binary) [64]	150	—	76.74
RSS (binary) [64]	150	—	74.65
i-vector (real) [62]	150	27.70	93.81
AM-Softmax (real) [62]	512	95.82	98.65
DAMH (binary) [62]	256	94.55	98.19
SEM (real) [65]	2048	97.70	98.00
SEM (binary) [65]	32	95.60	96.30
Proposed method	**64**	**97.55**	**99.91**

6.5 Experiment and Evaluation

6.5.1 Dataset and pre-processing

We used the VoxCeleb2 dataset [88] to evaluate the effectiveness of our proposed method. More specifically, only speakers whose number of utterances are at least 100 were selected; thus, the total number of speakers and utterances that were used in our experiments were 3,641 and 958,187, respectively.

For each utterance, we cropped a three-second segment randomly. Then, we used the algorithm as described in Section 6.4.1 to convert all the utterances into auditory sparse representations with a dimension of 32×48,000. Then, we used a max pooling technique to reduce the dimensionality of the feature vectors. More specifically, we used a $32 \times 1 \times 160$ (10 milliseconds duration) frame with no overlapping to produce input feature vectors with a dimension of 32×300.

6.5.2 Experiment setups and evaluation metrics

We obtained the hash centers by sampling the Bernoulli distribution. More specifically, we generated 3,641 64-bit binary hash centers that corresponded to the number of speakers in the evaluation dataset. Furthermore, only a set of hash centers, where the minimum distance between two hash centers and the mean distance of all the hash centers were at least 20 and 32, respectively, was used.

Table 6.1 shows the splits of data used in our experiments. For each speaker, we randomly selected 10 utterances for validation—36,410 utterances in total—and 20 utterances for testing—72,820 utterances in total. The remaining 848,957 utterances of the whole dataset were used for the training process. We used the same splits in both speaker identification and retrieval experiments. We used top-1 accuracy for the speaker identification task and mean average precision at 100 (MAP@100) for the speaker retrieval task because the lowest number of utterances per speaker was 100.

We trained the deep hashing algorithm for 50 epochs. After every 10 epochs, we evaluated the performance of the trained deep hashing algorithm with the validation set using MAP@100 and saved the model as a checkpoint. Then, we used the model that produced the highest MAP@100 on the validation set for calculating MAP@100 and top-1 accuracy on the test set.

6.6 Results and Discussion

Researchers researching deep hashing have evaluated their methods using different types and sizes of datasets. Therefore, we compared our results with two recent benchmarks on VoxCeleb datasets. More specifically, we compared our results with those produced by DAMH [62], which was the state-of-the-art method in 2019, and Random Speaker-variability Subspace (RSS) [64]. We also compared our results with SEM [65], which was the state-of-the-art method in 2021. Table 6.2 shows the results of our experiments.

Regarding the work reported in [64], i-vector and x-vector were used as speaker embeddings and binary hash codes were produced by two methods. The first method used Local Sensitive Hashing (LSH) [**?**], and this method was used as the baseline for comparison. The second hashing method introduced RSS to avoid redundant and potentially overlapping projections. The similarity between CSQ and RSS is that the distribution of the generated

hashes is a part of the overall algorithms. Therefore, the intra-class and inter-class of the binary hash clusters of the speakers are controlled by explicit techniques rather than random projection such as using LSH. Consequently, this is one advantage of our proposed method over the classical hashing method, using LSH on top of i-vector/x-vector. The second advantage of our proposed method over RSS is that we used only 64-bit hash length while RSS used 150 bits. It is evident that the shorter the length, the shorter the indexing time, and the lower the storage size. Another advantage of our proposed method is that the speaker identification results are noticeably higher than those reported in [64], 20.68% and 25.26% higher than i-vector and RSS, respectively.

With respect to [62], i-vector was also used as a baseline for comparison in speaker identification and retrieval tasks. It can be seen that our proposed method outperformed this baseline. The method that produced the top performance in [62] was called AM-Softmax; however, this method generated 512 real-valued hash codes that greatly increased indexing time and storage cost. Moreover, its MAP and top-1 accuracy were 1.73% and 1.26% lower than those of our proposed method, respectively. A hashing technique was applied to AM-Softmax to produced binary hash codes and called DAMH. Regarding DAMH, its 256-bit binary hash codes produced 94.55% in MAP and 98.19% in top-1 accuracy. The advantages of our proposed method over DAMH are threefold. First, it can be seen that the MAP and top-1 accuracy produced by our method were 1.72% and 3.00% higher than those of the 2019 state-of-the-art. Second, a higher performance can be achieved with a shorter code length, that is, 64-bit instead of the 256-bit of DAMH. Finally, we used a lower number of utterances in the training set to train our deep hashing method, which was 848,957 utterances instead of the 903,572 utterances in DAMH. In summary, our proposed method was more effective than DAMH because it required less training data yet performed better with a shorter code length.

With respect to SEM with a 2048-bit real-valued code length, the MAP by our method was almost the same—only 0.15% lower—but the top-1 accuracy was 1.91% higher. The disadvantage of SEM (real) is that it used 2048-bit real-valued hashes, which greatly increased the storage size and the identification and retrieval time. Furthermore, only MAP@10 was calculated; the calculated MAP may not hold if the evaluation range is increased to 100. A dimension reduction technique and the *sign* function were applied to SEM (real) to obtain SEM (binary) to deal with the storage size and indexing time problems at the cost of its performance. More specifically, both the MAP and top-1 accuracy were lower than those of our method by 1.95% and 3.61%, respectively. We acknowledge that our method operated with a smaller dataset in comparison with that of SEM. However, a longer hash code length can be used to increase the coding capability and a farther hash center distance can be applied to increase the inter-class discrimination.

6.7 Conclusion

Deep speaker hashing techniques in the literature have tried to produce discriminative hash codes that can be used for large-scale speaker identification and retrieval. Results from psychoacoustical studies have shown that the inherent speaker individualities in speech signals are unique to each speaker. Although existing deep speaker hashing methods can achieve high accuracy, they do not take into account the importance of speaker individuality. In this study, we proposed a method that maps significant acoustical features related to speaker individuality to the Hamming space. More specifically, we used the equivalent rectangular bandwidth (ERB) scale and a gammachirp (GC) function

to design an auditory filterbank; together with an orthogonal matching pursuit (OMP) algorithm, we converted speech signals into auditory sparse representations as a way to mimic neural activity patterns (NAPs). Furthermore, we focus on the frequency range from 40 to 800 Hz because it is important for speaker individuality analysis. Then, we used the CSQ to map auditory sparse representations into hash sequences in the Hamming space. Experimental results on the VoxCeleb2 dataset showed that our proposed method is highly competitive with other state-of-the-art methods as it achieved 99.91% accuracy in a speaker identification task and 97.55% MAP@100 in a speaker retrieval task with only 64-bit binary hash sequences.

Chapter 7

Conclusion

7.1 Summary

In cyber physical systems, speech is an essential part of various kinds of applications such as controlling automated systems, communications, and digital properties, etc. Therefore, speech security is important to ensure that speech is safe and convenient tool to use. A solution to the problem is using speech fingerprints.

The first goal of this study is constructing an algorithm to approximate the NAPs of the auditory nerves. This algorithm is important because speech fingerprints are inherent in the NAPs. A speech analysis/synthesis experiment was conducted to evaluate the performance of this algorithm. Results suggest that auditory sparse representations convey significant perceptual features of speech signals. At this point, it can be concluded that the auditory sparse representations are similar to the NAPs.

The second goal is to verify the uniqueness of the patterns on the auditory sparse representations. The NAPs cary different patterns specifically to different speakers, linguistic contents, emotions, etc. The auditory sparse representations are proposed to mimic the NAPs; thus, they should contain unique patterns depending on different kinds of information of speech signals. A general landmark-based pattern analysis technique was used for this purpose. Experimental results show that the proposed auditory sparse representations contain highly distinguishable patterns. At this point, it can be concluded that the proposed method provides an effective representation model for speech signals that is potentially useful for different kinds of speech analysis applications.

The third goal is constructing a speech fingerprint identification algorithm. A gammachirp auditory filterbank and an orthogonal matching pursuit algorithm were used to extract speech fingerprints from speech signals. Also, a deep hashing technique (CSQ) was used as the identification algorithm. Experimental results show that the proposed speech fingeprints achieve high performance in distinguishing speakers and very effective in a large dataset.

7.2 Remaining issues

A masking model was used in the proposed method to removed inaudible kernels and the PMP-GC provided the highest results in both experiments. However, it was unclear if the masking model had over or under estimated the masking patterns.

With regards to the second experiment, the RLBAF is a very basic pattern analysis technique and it was originally designed to work with spectrograms. Thus, the RLBAF

might not be able to obtain the potential of the auditory representation.

7.3 Future work

The nonlinearity characteristics of the auditory periphery are still a mystery to science. Thus, in the future, when we have a better understanding about the current characteristics and discover more physiological and psychoacoustical phenomenons, extensions can be made to obtain auditory representations that are more similar to the NAPs.

Deep learning has become the state-of-the-art pattern analysis recently; therefore, a deep learning algorithm can be design specifically to exploit the advantages of the auditory representations.

Appendix A

Supplementary Material

A.1 Creating speech database for the speech fingerprint matching experiment

The speech fingerprint matching experiment has to main steps. The first step is creating a speech fingerprint database for references and the second step is querying the database. We recorded the speech signals for this experiment in a sound proof room. Six people including three male and three female (from 23 to 30 years of age) volunteered for the recording process. The speech contents we used were three Japanese words: /Ohayo-gozaimasu/, /Konichiwa/, and /Kombanwa/. Each volunteer uttered each word 20 times. Then, we used a free software to extract 11 utterances for each speaker and each word (speaker, word) randomly, 198 speech signals in total. We used MATLAB to down-sample the speech signals from 44,100 Hz to 16,000 Hz. We used 10 pairs (speaker, word) for creating the speech fingerprint database and 1 pair (speaker, word) for querying the database.

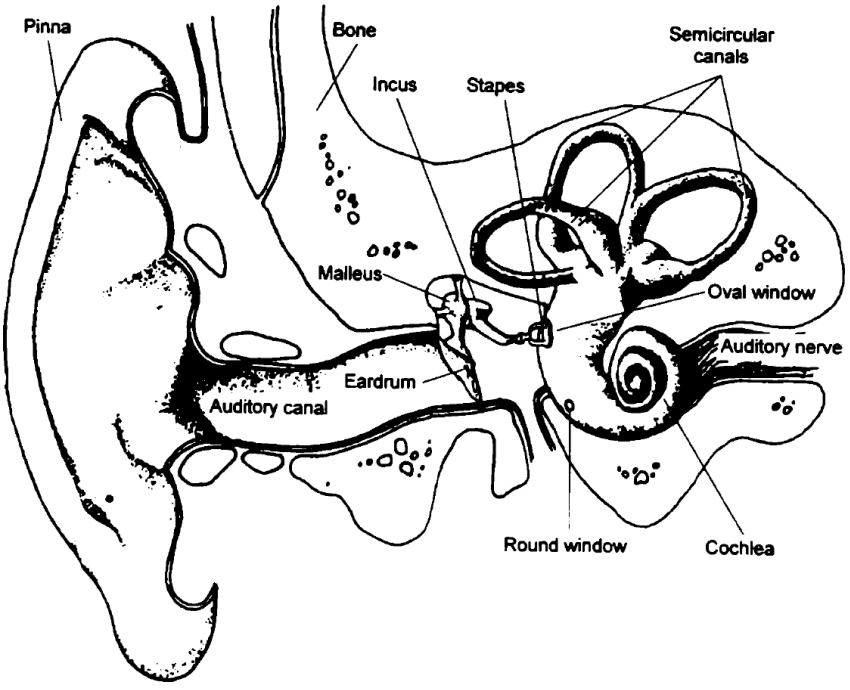

Figure A.1: Anatomy of the human auditory periphery. The figure was captured from [1].

A.2 Neural Activity Patterns

If someone asked: "Could you please show me your ears?" We would probably be showing the two round and springy flaps on either sides of our head. That would normally correct in the common sense. However, those are just pinnae, the ears are much more complicated, and we have not fully understood how they work yet. Generally, our ears comprise of two parts, i.e., the auditory periphery, and the auditory cortex. Figure A.1 illustrates an anatomy of the human auditory periphery. The auditory periphery consists of three parts, i.e., the outer ear, middle ear, and inner ear. The sounds we hear first arrive at the pinna of the outer ear, travel through the auditory canal, and beat the ear drum. The ear drum and the ossicles of the middle ear transfer the waves of air pressure into mechanical movements of the stapes, which hammer the oval window of the cochlear. The spiral snail-like shape of the cochlear is filled with almost incompressible fluids. The pressure of the fluids caused by the hammering of the stapes generates travelling waves on the basilar membrane. These travelling waves stimulate the inner hair cells and outer hair cells of the inner ear. Stimulated inner hair cells discharge a flux of neurotransmitter into the auditory nerve causing bioelectrical spikes. Thus far, incoming sounds have been converted into bioelectrical spikes at the auditory nerve. The Neural Activity Patterns (NAPs) of the spikes carry the information about the sound to the auditory cortex.

Bibliography

[1] B. Moore, "An introduction to the psychology of hearing: Sixth edition," *Brill, Leiden, The Netherlands*, 2013. [Online]. Available: https://brill.com/view/title/24210

[2] Y. Lit, S. Kim, and E. Sy, "A survey on amazon alexa attack surfaces," *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, pp. 1–7, 2021.

[3] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230820300474

[4] P. Cano, E. Batle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," *2002 IEEE Workshop on Multimedia Signal Processing.*, pp. 169–173, 2002.

[5] D. Ellis, "Robust landmark-based audio fingerprinting," https://labrosa.ee.columbia.edu/ dpwe/resources/matlab/fingerprint/, 2006.

[6] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.

[7] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[8] R. F. Lyon, "Theories of hearing," *Human and Machine Hearing: Extracting Meaning from Sound*, pp. 23–32, 2017.

[9] S. Mallat, "Sparse Representations," *A Wavelet Tour of Signal Processing*, pp. 1–31, 2009.

[10] D. Milano, "Content control: Digital watermarking and fingerprinting," https://www.digimarc.com/docs/default-source/technology-resources/white-papers/rhozet_wp_fingerprinting_watermarking.pdf.

[11] S. E. Siwek, "The true cost of sound recording piracy to the u.s. economy," August 2007, https://www.riaa.com/wp-content/uploads/2015/09/20120515_SoundRecordingPiracy.pdf.

[12] X. Liu and C.-C. Bao, "Audio bandwidth extension based on temporal smoothing cepstral coefficients," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, no. 1, pp. 1687–4722, 2014.

[13] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for song detection in broadcast audio," 2002. [Online]. Available: files/publications/aes2002-pcano.pdf

[14] N. Chen and W.-G. Wan, "Speech hashing algorithm based on short-time stability," *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II*, pp. 426–434, 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-04277-5_43

[15] N. Chen and W. Wan, "Robust speech hash function," *ETRI Journal*, vol. 32, no. 2, pp. 345–347, 2010.

[16] J. Haitsma, A. Kalker, and J. Oostveen, "Robust audio hashing for content identification," *International Workshop on Content-Based Multimedia Indesing (CBMI'01), Brescia, Italy, September 19-21, 2001*, 2001.

[17] H. Özer, B. Sankur, and N. Memon, "Robust audio hashing for audio identification," *2004 12th European Signal Processing Conference*, pp. 2091–2094, 2004.

[18] A. B. Salem, S.-A. Selouani, H. Hamam, and J. Caelen, "A highly robust audio hashing system using auditory-based front-end processing," *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1413–1416, 2009.

[19] Y. Jiao, L. Ji, and X. Niu, "Robust speech hashing for content authentication," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 818–821, 2009.

[20] A. L.-C. Wang, "An industrial-strength audio search algorithm," *Proc. 2003 ISMIR International Symposium on Music Information Retrieval*, October 2003. [Online]. Available: https://www.ee.columbia.edu/ dpwe/papers/Wang03-shazam.pdf

[21] R. Pichevar, H. Najaf-Zadeh, L. Thibault, and H. Lahdili, "Auditory-inspired sparse representation of audio signals," *Speech Communication*, vol. 53, no. 5, pp. 643–657, 2011, perceptual and Statistical Audition.

[22] M. Unoki and M. Akagi, "A method of signal extraction from noisy signal based on auditory scene analysis," *Speech Communication*, vol. 27, no. 3, pp. 261–279, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639398000776

[23] B. M. Mahmmod, A. R. Ramli, T. Baker, F. Al-Obeidat, S. H. Abdulhussain, and W. A. Jassim, "Speech enhancement algorithm based on super-gaussian modeling and orthogonal polynomials," *IEEE Access*, vol. 7, pp. 103 485–103 504, 2019.

[24] L. Fan, "Audio example recognition and retrieval based on geometric incremental learning support vector machine system," *IEEE Access*, vol. 8, pp. 78 630–78 638, 2020.

[25] Q. Li, Y. Yang, T. Lan, H. Zhu, Q. Wei, F. Qiao, X. Liu, and H. Yang, "Msp-mfcc: Energy-efficient mfcc feature extraction method with mixed-signal processing architecture for wearable speech recognition applications," *IEEE Access*, vol. 8, pp. 48 720–48 730, 2020.

[26] R. H. Aljuhani, A. Alshutayri, and S. Alahdal, "Arabic speech emotion recognition from saudi dialect corpus," *IEEE Access*, vol. 9, pp. 127 081–127 085, 2021.

[27] A. Dey, S. Chattopadhyay, P. K. Singh, A. Ahmadian, M. Ferrara, and R. Sarkar, "A hybrid meta-heuristic feature selection method using golden ratio and equilibrium optimization algorithms for speech emotion recognition," *IEEE Access*, vol. 8, pp. 200 953–200 970, 2020.

[28] T. Zhang, Y. Wu, Y. Shao, M. Shi, Y. Geng, and G. Liu, "A pathological multi-vowels recognition algorithm based on lsp feature," *IEEE Access*, vol. 7, pp. 58 866–58 875, 2019.

[29] M. K. Reddy and P. Alku, "A comparison of cepstral features in the detection of pathological voices by varying the input and filterbank of the cepstrum computation," *IEEE Access*, vol. 9, pp. 135 953–135 963, 2021.

[30] T.-H. Tsai, P.-C. Hao, and C.-L. Wang, "Self-defined text-dependent wake-up-words speaker recognition system," *IEEE Access*, vol. 9, pp. 138 668–138 676, 2021.

[31] M. B. Er, "A novel approach for classification of speech emotions based on deep and acoustic features," *IEEE Access*, vol. 8, pp. 221 640–221 653, 2020.

[32] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[33] T. Necciari, P. Balazs, N. Holighaus, and P. L. Søndergaard, "The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 498–502, 2013.

[34] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[35] B. R. Glasberg and B. C. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Research*, vol. 47, no. 1, pp. 103–138, 1990.

[36] Y. Fu and X. Yuan, "Composite feature extraction for speech emotion recognition," *2020 IEEE 23rd International Conference on Computational Science and Engineering (CSE)*, pp. 72–77, 2020.

[37] M. S. Alam, M. S. A. Zilany, W. A. Jassim, and M. Y. Ahmad, "Phoneme classification using the auditory neurogram," *IEEE Access*, vol. 5, pp. 633–642, 2017.

[38] N. Mamun, W. A. Jassim, and M. S. A. Zilany, "Prediction of speech intelligibility using a neurogram orthogonal polynomial measure (nopm)," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 760–773, 2015.

[39] A. Lauraitis, R. Maskeliūnas, R. Damaševičius, and T. Krilavičius, "Detection of speech impairments using cepstrum, auditory spectrogram and wavelet time scattering domain features," *IEEE Access*, vol. 8, pp. 96 162–96 172, 2020.

[40] F. Adeeba and S. Hussain, "Native language identification in very short utterances using bidirectional long short-term memory network," *IEEE Access*, vol. 7, pp. 17 098–17 110, 2019.

[41] F. Li and M. Akagi, "Unsupervised singing voice separation using gammatone auditory filterbank and constraint robust principal component analysis," *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1924–1928, 2018.

[42] Z. Peng, Z. Zhu, M. Unoki, J. Dang, and M. Akagi, "Speech emotion recognition using multichannel parallel convolutional recurrent neural networks based on gammatone auditory filterbank," *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1750–1755, 2017.

[43] R. Pichevar, H. Najaf-Zadeh, L. Thibault, and H. Lahdili, "New trends in biologically-inspired audio coding," in *Signal Processing*, sebastian miron ed. InTech, 2010. [Online]. Available: http://www.intechopen.com/books/signal-processing/new-trends-in-biologically-inspired-audio-coding

[44] M. Edalatian, A. A. Soitani, and N. Faraji, "Sparse representation of human auditory system," *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 302–306, 2016.

[45] P. Vera-Candeas, N. Ruiz-Reyes, and F. López-Ferreras, "Bark scale-based perceptual matching pursuit for improving sinusoidal audio modeling," *Digital Signal Processing*, vol. 19, no. 2, pp. 229–240, 2009.

[46] B. Bouchhima, R. Amara, and M. Turki Hadj-Alouane, "Perceptual orthogonal matching pursuit for speech sparse modelling," *Electronics Letters*, vol. 53, no. 21, pp. 1431–1433, 2017.

[47] K. Daoudi and N. Vinuesa, "An analysis of psychoacoustically-inspired matching pursuit decompositions of speech signals," *International Conference on Natural Language, Signal and Speech Processing*, 2017.

[48] I. Toshio, "An optimal auditory filter," *Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Accoustics*, pp. 198–201, 1995.

[49] T. Irino and R. D. Patterson, "A time-domain, level-dependent auditory filter: The gammachirp," *The Journal of the Acoustical Society of America*, vol. 101, no. 1, pp. 412–419, 1997.

[50] E. de Boer and A. L. Nuttall, "The mechanical waveform of the basilar membrane. I. Frequency modulations ("glides") in impulse responses and cross-correlation functions," *The Journal of the Acoustical Society of America*, vol. 101, no. 6, pp. 3583–3592, 1997.

[51] H. Lahdili, H. Najaf-Zadeh, R. Pichevar, and L. Thibault, "Perceptual matching pursuit for audio coding," in *Audio Engineering Society Convention 124*, May 2008. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=14613

[52] G. Chardon, T. Necciari, and P. Balazs, "Perceptual matching pursuit with Gabor dictionaries and time-frequency masking," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3102–3106, 2014.

[53] B. Moore and B. Glasberg, "A revision of Zwicker's loudness model," *Acta Acustica united with Acustica*, vol. 82, pp. 335–345, 03 1996.

[54] S. Qian and D. Chen, "Signal representation using adaptive normalized Gaussian functions," *Signal Processing*, vol. 36, no. 1, pp. 1–11, 1994.

[55] M. Goodwin and M. Vetterli, "Matching pursuit and atomic signal models based on recursive filter banks," *IEEE Transactions on Signal Processing*, vol. 47, no. 7, pp. 1890–1902, 1999.

[56] R. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," *Auditory Physiology and Perception*, pp. 429–446, 1992.

[57] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," 1993.

[58] R. Huber and B. Kollmeier, "PEMO-Q–A new method for objective audio quality assessment using a model of auditory perception," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1902–1911, 2006.

[59] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, pp. 749–752, 2001.

[60] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—a survey," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2016.

[61] J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 769–790, 2018.

[62] L. Fan, Q.-Y. Jiang, Y.-Q. Yu, and W.-J. Li, "Deep hashing for speaker identification and retrieval," *Proc. Interspeech 2019*, pp. 2908–2912, 2019.

[63] L. Schmidt, M. Sharifi, and I. L. Moreno, "Large-scale speaker identification," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1650–1654, 2014.

[64] S. Shon, Y. Lee, and T. Kim, "Large-scale speaker retrieval on random speaker variability subspace," *Proc. Interspeech 2019*, pp. 2963–2967, 2019.

[65] C. Chen, D. Jiang, J. Peng, R. Lian, Y. Li, C. Zhang, L. Chen, and L. Fan, "Scalable identity–oriented speech retrieval," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[66] Y.-Q. Yu, S. Zheng, H. Suo, Y. Lei, and W.-J. Li, "Cam: Context–aware masking for robust speaker verification," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6703–6707, 2021.

[67] M. Akagi and T. Ienaga, "Speaker individuality in fundamental frequency contours and its control," *Journal of the Acoustical Society of Japan (E)*, vol. 18, no. 2, pp. 73–80, 1997.

[68] T. V. Ho and M. Akagi, "Cross-lingual voice conversion with controllable speaker individuality using variational autoencoder and star generative adversarial network," *IEEE Access*, vol. 9, pp. 47 503–47 515, 2021.

[69] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006.

[70] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008. [Online]. Available: https://doi.org/10.1121/1.2951592

[71] M. Mauch and S. Dixon, "Pyin: A fundamental frequency estimator using probabilistic threshold distributions," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 659–663, 2014.

[72] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[73] D. K. Tran and M. Unoki, "Matching pursuit and sparse coding for auditory representation," *IEEE Access*, vol. 9, pp. 167 084–167 095, 2021.

[74] T. Zhu, X. Qin, and M. Li, "Binary Neural Network for Speaker Verification," *Proc. Interspeech 2021*, pp. 86–90, 2021.

[75] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119, 2016.

[76] S. Novoselov, V. Shchemelinin, A. Shulipa, A. Kozlov, and I. Kremnev, "Triplet Loss Based Cosine Similarity Metric Learning for Text-independent Speaker Recognition," *Proc. Interspeech 2018*, pp. 2242–2246, 2018.

[77] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System," *Proc. Interspeech 2019*, pp. 361–365, 2019.

[78] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In Defence of Metric Learning for Speaker Recognition," *Proc. Interspeech 2020*, pp. 2977–2981, 2020.

[79] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[80] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, 2014.

[81] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[82] L. Yuan, T. Wang, X. Zhang, F. E. Tay, Z. Jie, W. Liu, and J. Feng, "Central similarity quantization for efficient image and video retrieval," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3080–3089, 2020.

[83] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[84] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," *Computer Vision – ECCV 2014*, pp. 740–755, 2014.

[85] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-WIDE: A Real-World Web Image Database from National University of Singapore," *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, July 8-10, 2009.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[87] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[88] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *INTERSPEECH*, 2018.

Publications

Journal

[1] D. K. Tran and M. Unoki, "Matching Pursuit and Sparse Coding for Auditory Representation," in IEEE Access, vol. 9, pp. 167084-167095, 2021, doi: 10.1109/ACCESS.2021.3135011.

Lecture note

[2] Dung Kim Tran and Masashi Unoki, "Study on speech representation based on spikegram for speech fingerprints," Springer Lecture Notes Smart Innovation, Systems and Technologies Volume 82: Advances in Intelligent Information Hiding and Multimedia Signal Processing, J. S. Pan et al. (eds.), pp. 153-160, 2018.

International conference

[3] Dung Kim Tran, Masato Akagi, and Masashi Unoki, "Deep Hashing for Speaker Identification and Retrieval Based on Auditory Sparse Representation," APSIPA2022 (accepted).

[4] Dung Kim Tran and Masashi Unoki, "Study on speech representation based on spikegram for speech fingerprints," Proc. IIHMSP2017, Shimane, Aug. 2017 (CDROM). 10.1007/978-3-319-63859-1_20.

Domestic conference

[5] Dung Kim Tran, Nguyen Huy Quoc, Masashi Unoki, "Study on perceptual matching pursuit algorithm to create speech representation for speech fingerprint," 日本音響学会 2019 年度春季研究発表会講演論文, 3 5 16, 電気通信大学, March 2019.

[6] Dung Kim Tran, Nguyen Huy Quoc, Masashi Unoki, "Study on speech representation for speech fingerprint using perceptual matching pursuit algorithm," IEICE Technical Repor t, EMM2018 59, pp. 71 76, 別府国際コンベンションセンター, Sept. 2018.

[7] Dung Kim Tran and Masashi Unoki, "Investigation of spikegram-based signal representation for speech fingerprints," IEICE Technical Report, EMM2017-35, pp. 241-246, 内田洋行東京本社ショールーム, July. 2017.