UNIVERSITY OF CYPRUS

Department of Electrical and Computer Engineering

Noise-Robust Classification using Rank Order Kernels

Alexandros Kyriakides

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the University of Cyprus

June, 2012

APPROVAL PAGE

Alexandros Kyriakides

Noise-Robust Classification using Rank Order Kernels

The present Doctorate Dissertation was submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering, and was approved on 29 June, 2012 by the members of the Examination Committee.

Committee Chair _____
Marios Polycarpou, Ph.D.

Research Supervisor _____
Constantinos Pitris, Ph.D, M.D.

Committee Member _____
Andreas Spanias, Ph.D.

Committee Member _____
Constantinos S. Pattichis, Ph.D.

Committee Member _____
Julius Georgiou, Ph.D.

Abstract

The need to process and classify signals is encountered in many applications. Signals are abundant in nature and can arise from numerous sources. In many cases however, signals also contain high levels of noise. This poses a unique challenge when processing the signals in order to obtain useful information needed for classification.

In this thesis, we show that by using an appropriate representation transformation of the signal and by kernel-based feature-extraction methods, we can mitigate the effect of noise. We describe a biologically-inspired classification system which can classify various types of noisy signals, without the need to perform extensive pre-processing on the signal. We introduce the concept of rank order kernels which employ rank order coding. Rank order coding is a type of temporal coding which has been proposed as a possible explanation of how neurons encode information. We formulate an image distance metric based on rank order kernels and use it for classification.

We focus on the problem of Automatic Speech Recognition (ASR) in order to demonstrate the capability of our classification system. The accurate recognition of speech is a vital element in human-computer interfaces. One of the main obstacles to building robust ASR systems is the problem of noise. With our methodology, we transform speech signals to two-dimensional time-frequency image representations and classify them using the rank order kernel distance metric.

In our attempt to create a noise-robust speech recognition system we found that it was also necessary to develop an endpoint detection system which was also robust to noise. This thesis therefore also presents an endpoint detection system which uses a spectrogram representation of speech and variance kernels in order to separate speech from non-speech. Our endpoint detection system is used as a pre-processing step to our speech recognition system.

Our endpoint detection algorithm and rank order kernel method can also be

applied to other types of signals. We show how the endpoint detection algorithm is used to detect the endpoints of micro-Doppler signatures in ultrasound signals, and how the rank order kernels can be used to classify Raman spectra obtained from bacterial samples. The classification system we develop in this thesis can be used on any type of signal by first converting the signal to an appropriate two-dimensional image representation and then performing classification using the rank order kernel distance metric.

Περίληψη

Η ανάγκη για την επεξεργασία και ταξινόμηση των σημάτων, συναντάται σε πολλές εφαρμογές. Τα σήματα είναι άφθονα στη φύση και μπορούν να προκύψουν από πολλές πηγές. Σε πολλές περιπτώσεις όμως, τα σήματα περιέχουν, επίσης, υψηλά επίπεδα θορύβου. Αυτό αποτελεί μια μοναδική πρόκληση κατά την επεξεργασία των σημάτων προκειμένου κάποιος να λάβει χρήσιμες πληροφορίες που απαιτούνται για την ταξινόμηση.

Στην παρούσα εργασία, δείχνουμε ότι με τη χρήση ενός κατάλληλου μετασχηματισμού στην αναπαράσταση του σήματος και των kernels, μπορούμε να ελαχιστοποιήσουμε την επίδραση του θορύβου. Περιγράφουμε ένα βιολογικά εμπνευσμένο σύστημα ταξινόμησης το οποίο μπορεί να χαρακτηρίσει διάφορους τύπους σημάτων, χωρίς την ανάγκη για εκτεταμένη προ-επεξεργασία στο σήμα. Έχουμε εισαγάγει την έννοια των rank order kernels που χρησιμοποιούν κωδικοποίηση σειράς κατάταξης (rank order coding). Η κωδικοποίηση σειράς κατάταξης είναι ένα είδος κωδικοποίησης που έχει προταθεί ως μια πιθανή εξήγηση για το πώς οι νευρώνες κωδικοποιούν πληροφορίες. Έχουμε διαμορφώσει ένα μέτρο με βάση τα rank order kernels και το χρησιμοποιούμε για ταξινόμηση.

Εστιάζουμε την προσοχή μας στο πρόβλημα της αυτόματης αναγνώρισης ομιλίας, προκειμένου να αποδείξουμε την ικανότητα του συστήματος ταξινόμησης. Η αναγνώριση της ομιλίας είναι ένα σημαντικό στοιχείο στην επικοινωνία του ανθρώπου με τον υπολογιστή. Ένα από τα κύρια εμπόδια είναι το πρόβλημα του θορύβου. Με τη μεθοδολογία μας, έχουμε μετατρέψει τα σήματα ομιλίας σε δύο διαστάσεις. Δημιουργούμε αναπαραστάσεις χρόνου-συχνότητας και τις ταξινομούμε με τα rank order kernels.

Στην προσπάθειά μας να δημιουργήσουμε το σύστημα αναγνώρισης ομιλίας βρήκαμε ότι ήταν επίσης αναγκαίο να αναπτυχθεί ένα σύστημα endpoint detection. Στην παρούσα εργασία παρουσιάζεται ως εκ τούτου, επίσης, ένα σύστημα endpoint detection το οποίο χρησιμοποιεί φασματογράφημα της φωνής και variance kernels προκειμένου να διαχωρίσει την ομιλία από την μη-ομιλία. Το σύστημα endpoint detection χρησιμοποιείται ως προ-επεξεργασία για την ομιλία στο σύστημα αναγνώρισης.

Ο αλγόριθμος endpoint detection και το rank order kernel μπορεί επίσης να εφαρμοστεί και σε άλλα είδη σημάτων. Έχουμε δείξει πώς ο αλγόριθμος endpoint detection μπορεί να χρησιμοποιηθεί για υπερηχητικά σήματα, και τα rank order kernels μπορούν να χρησιμοποιηθούν για την ταξινόμηση φασμάτων Raman.

Acknowledgments

Completing a PhD thesis is a highly demanding and challenging task. It is predominantly an individualistic venture. This accomplishment however, would have not been possible without the assistance, advice, and support of others. I would therefore like to thank all those who contributed to the successful fulfillment of this thesis.

First and foremost I would like to thank my thesis supervisor Costas Pitris. I regard myself fortunate to have found a supervisor with whom I could work on a field of research which was of interest to me. I am very grateful for his help and understanding. I would also like to thank Julius Georgiou who provided the motivational spark for the research direction of this thesis. I feel privileged to have worked with Andreas Spanias. His expertise in speech and signal processing proved essential to the success of my endeavors.

It was important for me to have a pleasant, comfortable, and inspiring work environment. The KIOS Research Center provided this environment for me. I would therefore like to thank all the KIOS faculty, researchers, and support staff. Of special mention are the Director of the KIOS Research Center, Marios Polycarpou, and our two wonderful Administrative Assistants, Skevi Chrysanthou and Despina Petrou. My fellow researchers at the KIOS Research Center are too many to list all of them by name, but I would like to thank each and every one of them for contributing to the ambiance of the KIOS Center. I am particularly grateful to Yiannis Tofis, Christos Laoudias, Demetris Stavrou, Giorgos Milis, Demetris Eliades, Constantinos Hadjistassou, Antonis Hadjiantonis, Evgenia Bousi, and Theofanis Lambrou who each in their own way contributed to enhancing my experience at work.

My friends Sophocles Ioannides, Demetris Soteropoulos, and Ioannis Demetriades were extremely supportive during the difficult times, and for this I will be

eternally indebted. Above all I would like to thank my parents and my sister. My parents have been standing by me all my life and everything I have achieved, I owe it to them.

Software

This thesis was written entirely with the `emacs` text editor and the LaTeX typesetting system on a machine running the GNU/Linux operating system. I would like to express my utmost respect and admiration to all the people who develop and support free and open source software. Thank you for making the world a better place.

Publications

Book Chapters

1. E. Kastanos, A. Kyriakides, K. Hadjigeorgiou, and C. Pitris, "Identification and Antibiotic Sensitivity of UTI Pathogens Using Raman Spectroscopy," *Urinary Tract Infections*, Dr. Peter Tenke (Ed.), InTech, 2011. ISBN: 978-953-307-757-4.

Published journal articles

1. A. Kyriakides, E. Kastanos, K. Hadjigeorgiou, and C. Pitris, "Classification of Raman spectra using the correlation kernel," *J. of Raman Spectroscopy*, 2010. DOI 10.1002/jrs.2809.

2. E. Kastanos, A. Kyriakides, K. Hadjigeorgiou, and C. Pitris, "A Novel Method for Urinary Tract Infection Diagnosis and Antibiogram Using Raman Spectroscopy," *J. of Raman Spectroscopy*, 2010. DOI 10.1002/jrs.2540.

3. E. Kastanos, A. Kyriakides, K. Hadjigeorgiou, and C. Pitris, "A Novel Method for Bacterial UTI Diagnosis Using Raman Spectroscopy," *Int. J. of Spectroscopy*, 2012. DOI:10.1155/2012/195317

Published conference proceedings

1. C. Pitris, A. Kyriakides, and P. Ioannides, "Decomposition and Analysis of Unresolvable Optical Coherence Tomography Signals," *ICO Topical Meeting on Optoinformatics/Information Photonics 2006*, St Petersburg, Russia, September 4-7, 2006.

2. E. Kastanos, A. Kyriakides, K. Hadjigeorgiou, and C. Pitris, "Urinary tract infection diagnosis and response to antibiotics using Raman spectroscopy," *Proc. SPIE 7169, 71690I (2009) , Photonics West*, San Jose, California USA, January 24-29, 2009.

3. E. Kastanos, A. Kyriakides, K. Hadjigeorgiou, and C. Pitris, "UTI diagnosis and antibiogram using Raman spectroscopy," *Proc. SPIE 7368, 73680U (2009)*, European Conference on Biomedical Optics, Munich, Germany, June 14-19, 2009.

4. A. Kyriakides, K. Hadjigeorgiou, E. Kastanos, and C. Pitris, "Classification of Raman Spectra using Support Vector Machines," *IEEE 9th International Conference on Information Technology and Applications in Biomedicine,* November 5-7, 2009, Larnaca, Cyprus, DOI: 10.1109/ITAB.2009.5394428.

5. K. Hadjigeorgiou, E. Kastanos, A. Kyriakides, and C. Pitris, "Raman Spectroscopy for UTI Diagnosis and Antibiogram," *IEEE 9th International Conference on Information Technology and Applications in Biomedicine,* November 5-7, 2009, Larnaca, Cyprus, DOI: 10.1109/ITAB.2009.5394425

6. E. Kastanos, K. Hadjigeorgiou, A. Kyriakides, and C. Pitris, "Surface enhanced Raman spectroscopy for urinary tract infection diagnosis and antibiogram," *Proc. SPIE 7560, 75600A (2010),* Photonics West, San Francisco, California USA, January 23-28, 2009

7. K. Hadjigeorgiou, E. Kastanos, A. Kyriakides, and C. Pitris, "Surface Enhanced Raman Spectroscopy for Urinary Tract Infection diagnosis and antibiogram," *Nanotheranostics: Fabrication and Safety Concerns, International Conference,* April 27-30, 2010, Ayia Napa, Cyprus.

8. E. Kastanos, K. Hadjigeorgiou, A. Kyriakides, and C. Pitris, "Classification of bacterial samples as negative or positive for a UTI and antibiogram using surface enhanced Raman spectroscopy," *Proc. SPIE 7911, 791107 (2011). Photonics West*, San Francisco, California USA, January 22-27, 2011

9. A. Kyriakides, E. Kastanos, K. Hadjigeorgiou, and C. Pitris, "The correlation kernel and support vector machines for the classification of Raman spectra," *Proc. SPIE 7890, 789047*, 2011.

10. A. Kyriakides, E. Kastanos, K. Hadjigeorgiou, and C. Pitris, "Raman spectra classification with support vector machines and a correlation kernel," *Proc. SPIE 8087, 808706 (2011), European Conference on Biomedical Optics,* Munich, Germany, May 22-26, 2011.

11. A. Kyriakides, C. Pitris, A. Fink, and A. Spanias, "Isolated word endpoint detection using time-frequency variance kernels," *Forty Fifth Asilomar Conference on Signals, Systems and Computers, 2011*, 6-9 Nov. 2011, DOI: 10.1109/ACSSC.2011.6190170.

Journal articles to be published

1. A. Kyriakides, A. Spanias, and C. Pitris, "Noise-Robust Endpoint Detection using Time-Frequency Variance Kernels."

2. A. Kyriakides, J. Georgiou, A. Spanias, and C. Pitris, "Noise-Robust Speech Recognition using Rank Order Kernels ."

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Intelligent Systems

A system, be it human or machine, exhibits intelligence if it is able to learn from past experience and make predictions about the future. Learning is the process of building models by using information. We humans learn because our brain is essentially a machine that builds models of the world based on the information gathered from our experiences. Using these models we are able to construct theories and make predictions about the future, and thus adapt to our environment. Understanding speech for example, is a process of interpreting sound information and making predictions on the different words that the sounds represent.

The human brain has the amazing ability to extract relevant patterns from the input it receives from our senses, which allows it to build appropriate models and make predictions. Even though these pattern recognition models exist in our mind, we cannot consciously describe them. A young child for example, can easily recognize a dog when it sees one. Based on the information available to the brain from the visual system, the prediction is made that the object is a dog. However, nobody can describe explicitly and in detail what features and models the brain uses to make the prediction.

The task of artificial intelligence researchers is to develop intelligent systems. According to Hawkins [40], one of the key ingredients needed to develop an intelligent system is the use of *invariant representations*. An invariant representation is a way of portraying information about an object so that even if the information changes slightly, the object is still recognizable for what it is. For example, humans can rec-

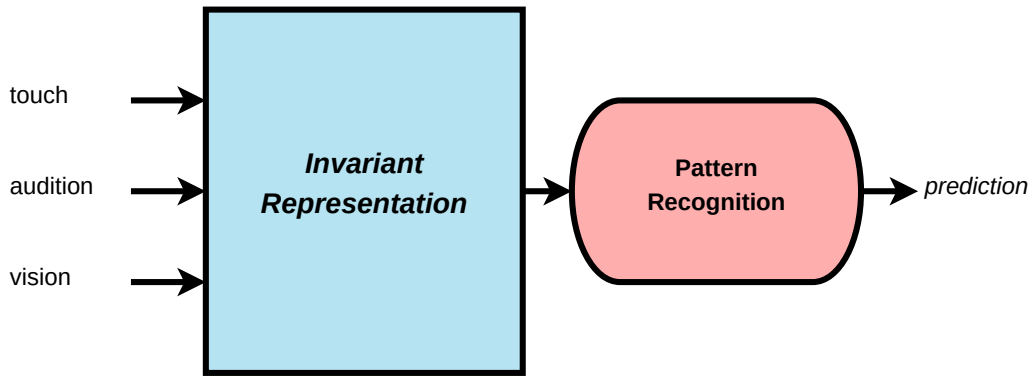

Figure 1.1: A schematic showing how a classification system can make predictions from multiple modalities by first transforming input signals into an invariant representation and then performing pattern recognition.

ognize speech without being greatly affected by changes in pronunciation and voice tone. Another argument made by Hawkins [40], is that at a high level the brain processes all information with a common algorithm, irrespective of the modality (e.g. touch, acoustic, visual). In this thesis, we have used these two comprehensive proposals of invariant representation and a common processing algorithm as inspiration in order to create a noise-robust classification system. This system is able to make predictions using input from various modalities by first converting the input signal into an invariant representation and then applying a pattern matching algorithm. Figure 1.1 illustrates this concept.

1.2 Thesis statement

In this thesis we describe a classification system which can classify various types of noisy signals, without the need to perform extensive pre-processing on the signal. We do not focus on intricate filtering or advanced noise-reduction techniques. Instead, we focus on finding appropriate representations for signals, and in combination with a noise-robust feature-extraction mechanism, we show how noisy signals can be classified with high accuracy even when the noise levels are high. We demonstrate how this classification system is well suited for noise-robust automatic speech recognition.

1.3 Automatic Speech Recognition

Automatic Speech Recognition (ASR) systems have numerous commercial applications. The use of voice to interface with devices, such as mobile phones, is becoming increasingly common. The future of human-machine interaction lies with voice control. Electric appliances in the house, home automation systems, home robots, and in-car systems will all be controlled with the user's voice. Voice control is not yet widespread however, because of the poor performance of current ASR systems.

The field of ASR has advanced significantly in the past few decades. Current state-of-the-art ASR systems perform extremely well when the vocabulary of words is limited, when there is no background noise, and finally, when the voice and pronunciation of the human speaker is not too different from the voice and pronunciation of the speakers used to train the system. When the variability of the speech signal increases, due to factors such as background noise, the performance of ASR systems drops significantly. Humans, on the other hand, have an amazing ability to process speech. Humans can understand continuous streams of speech input consisting of a large vocabulary of words. This ability is un-hindered even when a high level of background noise is present. All the more impressive is the fact that in the case of multiple speakers talking at the same time, a human listener can separate out and focus on the speech of only one single speaker of interest.

Researchers have turned to biology in an attempt to find ideas which can help to improve the speech recognition performance of ASR systems so that these systems can eventually reach, and maybe even surpass, the speech recognition performance of humans. For example, Mel Frequency Cepstral Coefficients (MFCCs) are a now the standard method for feature creation in ASR. The frequency distribution and bandwidth of MFCCs attempt to imitate those of the cochlea in the human ear. The general model of current state-of-the-art ASR systems however, bears no resemblance to human neurobiology. Current state-of-the-art ASR systems use a frame-based model and employ Hidden Markov Models (HMMs). The HMM statistical model is in disagreement with biological models, and although improvements are continuously being made, this model seems to have reached its limitations. The performance of such statistical approaches depends on accurate estimates of probability distributions during the supervised training phase. For this reason, the latest significant improvements in ASR were a result of the availability

of vast speech databases and extensive computational power which has allowed for improved statistical estimates of the characteristics of speech. Nonetheless, these incremental improvements are both inefficient and unsatisfactory. The only way to drastically improve ASR performance is by finding a more appropriate speech recognition model. Consequently, several researchers have indicated the need for new, biologically-inspired, ASR models [50, 89, 95, 99].

1.4 Biologically-inspired representations

Although we do not have a perfectly clear and detailed view of how the human brain works, we can still draw inspiration from the results of research studies performed on the brain. Such studies, for example, have established the fact that the brain's cerebral cortex decomposes visual images into features of oriented edges. Of great interest for the purposes of this thesis is a publication by Christopher deCharms et al. [16] which shows evidence that the brain also decomposes sound into visual-like features. From their experiments with primates, they found that the primary auditory cortex shows complex patterns of sound-feature selectivity. Certain neurons are more sensitive to "edges" in frequency-time, and to transitions in frequency or intensity. They observed that stimuli designed for a particular neuron's preferred feature pattern can drive that neuron with higher sustained firing rates than have typically been recorded with simple stimuli. "This suggests that the cortex decomposes an auditory scene into component parts using a feature-processing system reminiscent of that used for the cortical decomposition of visual images." [16]

In the current thesis we therefore draw inspiration from the human visual system which has primitive features for detecting patterns in images in order to detect patterns in speech. We use a two-dimensional time-frequency representation of sound together with another biologically-inspired idea, that of Rank Order Coding (ROC). ROC is a temporal coding technique which has been hypothesized as a possible description of how neurons code information. The information is distributed through a large population of neurons and is represented by the relative timing of the neuronal spikes. ROC has the advantages that it is easier to implement, it is less subject to changes in intensity of the stimulus, and the information is available as soon as the first spike arrives. Recent experimental studies on the auditory system of cats and somatosensory system of humans show that ROC might be responsible for coding

sensory information with only one spike per neuron [120]. ROC has been primarily used for image processing applications [34,113], but it has also been used for speech recognition. Rouat et al. compared a simple speech recognition prototype which uses ROC with a conventional HMM system. They found that the ROC system did surprisingly well compared to the HMM system [95].

Our work combines the idea of a two-dimensional image representation of signals, with the idea of rank order coding, in order to formulate the mechanism of *Rank order kernels*. Rank order kernels operating on a two-dimensional image are a form of invariant representation, which is robust to noise. We demonstrate this robustness to noise by focusing on speech recognition experiments. The bio-inspired aspects of our algorithm are the transformation of sound to a two-dimensional image representation and the use of rank order coding.

1.5 Outline

This thesis presents a noise-robust approach to classification using rank order kernels. We have chosen to focus on the problem of Automatic Speech Recognition (ASR) because it is an important field of research and because the performance of ASR systems is greatly influenced by noise. Our speech recognition experiments are performed using a speech corpus which we have collected ourselves, and a set of twenty publicly-available noise files. The isolated-word speech recognition system we have developed in the current thesis is based on image similarity metrics which require accurate endpoint detection of speech. For this reason, we also developed a biologically-inspired noise-robust endpoint detection system which we use as a pre-processing step.

The following is an outline of this thesis. **Chapter 2** gives a detailed description of the speech corpus we have created, and a description of the noise files used in our experiments. **Chapter 3** describes the Endpoint Detection System we have developed. **Chapter 4** is the core contribution of this thesis which describes the formulation of Rank Order Kernels. The result is a novel image similarity metric. **Chapter 5** applies Rank Order Kernels to the problem of Automatic Speech Recognition and shows that our method is robust to noise. **Chapter 6** shows how the same Endpoint Detection System and Rank Order Kernel approach can be used for two other applications. **Chapter 7** summarizes our work, talks about possible future research related to our

work, and states the main contributions of this thesis.

Chapter 2

Speech Data and Noise Data Used

In this chapter we give a detailed description of the data that we used for this thesis. We created our own corpus of spoken isolated words which we used for our experiments. In order to test the performance under noisy conditions we acquired a set of 20 noise types which were publicly available.

2.1 Isolated Word Speech Corpus

We decided to create our own speech corpus because we did not find a freely-available corpus suitable for our experiments. We needed a large set of isolated-word recordings from a wide range of speakers. For this reason we recruited 15 male speakers and 15 female speakers. We chose 100 words which the speakers were asked to utter into the microphone. It was required that each word was uttered 10 times, but not sequentially. In total therefore, we have 1000 recordings for each speaker. Each recording had a duration of 2 seconds. The speaker initiated and completed the utterance of the word within those 2 seconds. We created a Graphical User Interface (GUI) which randomly presented one word at a time to the human speaker. Figure 2.1 shows how this GUI looks. The data was recorded as part of an undergraduate thesis [26]. Table 2.1 shows the 100 words we used in our recordings. Table 2.2 shows the ages of the 30 speakers.

The recording environment was a small room, the walls of which were covered with egg cartons to reduce reverberation. Figure 2.2 shows a photograph of the computer used to perform the recordings in the room. The speakers were asked to speak at a close distance to the microphone in order to increase the signal-to-noise

Table 2.1: Our speech corpus of isolated word recordings consists of these 100 words. Each word was uttered 10 times by each of 15 male and 15 female speakers.

1	add	26	four	51	no	76	silent
2	AM	27	fourteen	52	north	77	six
3	answer	28	front	53	now	78	sixteen
4	backward	29	go	54	off	79	sixty
5	bat	30	goodbye	55	OK	80	slower
6	buy	31	hello	56	on	81	sound
7	bye	32	high	57	one	82	south
8	cancel	33	higher	58	open	83	start
9	close	34	house	59	pause	84	stop
10	computer	35	hundred	60	play	85	switch
11	curtains	36	keyboard	61	PM	86	telephone
12	door	37	left	62	power	87	ten
13	down	38	lights	63	preserve	88	thirteen
14	east	39	low	64	pull	89	thirty
15	eight	40	lower	65	push	90	three
16	eighteen	41	mat	66	rear	91	TV
17	eighty	42	menu	67	remove	92	twelve
18	eleven	43	microphone	68	repeat	93	twenty
19	faster	44	monitor	69	reserve	94	two
20	fifteen	45	mouse	70	reverse	95	type
21	fifty	46	music	71	right	96	up
22	five	47	mute	72	screen	97	west
23	flamingo	48	nine	73	seven	98	window
24	forty	49	nineteen	74	seventeen	99	yes
25	forward	50	ninety	75	seventy	100	zero

Table 2.2: The ages of the 15 male and 15 female speakers who were recruited for our recordings. The recordings were taken during the year 2009, all under the same recording conditions and with the same recording equipment.

Speaker	**Age**	**Speaker**	**Age**
male01	14	female01	21
male02	17	female02	22
male03	24	female03	22
male04	20	female04	22
male05	23	female05	22
male06	23	female06	21
male07	22	female07	22
male08	22	female08	21
male09	19	female09	22
male10	24	female10	23
male11	19	female11	30
male12	22	female12	22
male13	40	female13	21
male14	31	female14	24
male15	23	female15	23

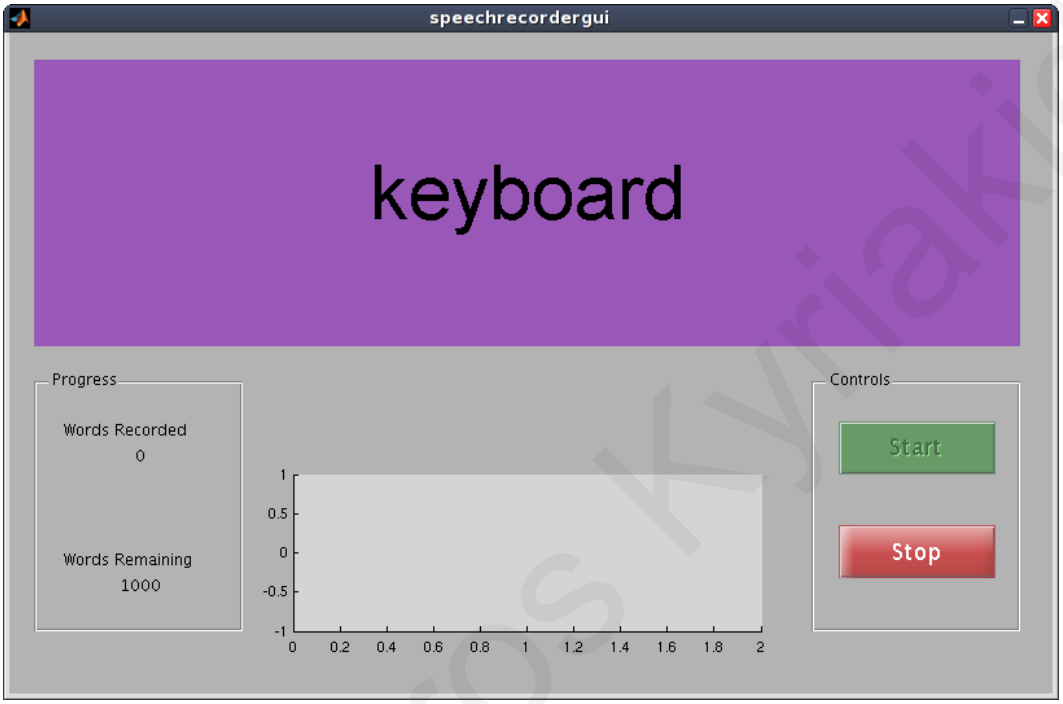

Figure 2.1: A screenshot showing the Graphical User Interface used for recording the words of our speech corpus. In this example, the speaker is asked to utter the word "keyboard". The recording duration is 2 seconds, after which, the waveform of the recording is presented on the axis in the lower center. The controls on the right can be used to start and stop the recording process, while the numbers on the left side indicate the progress.

Figure 2.2: A photograph of the recording room showing the computer and microphone used for the recordings. The walls are covered with egg cartons to reduce reverberation.

ratio (SNR) of the recording. The speakers were also asked to not speak directly on to the microphone in order to minimize the number of recordings with breathing sounds and air puffs. In general, we instructed the speakers to not introduce any extra unwanted noise. Even so, we found that a large number of our recordings contained sound artifacts such as:

- breathing noises
- microphone clicks
- tongue/mouth sound when opening the mouth before talking
- chair sounds
- keyboard/mouse sounds

It is common for recordings to contain artifacts, as is mentioned in several publications [1,53,109]. Among the artifacts mentioned in these publications are hisses, clicks, coughs, gulps, tongue clicks, lip-smacks, breaths, and microphone clicks. In this thesis we term the recordings which do not contain artifacts as "clean" recordings, and those which do contain artifacts as "non-clean" recordings. We use both clean and non-clean recordings in our experiments.

For some of our experiments, it was necessary to manually label some of the recordings into "clean" and "non-clean" as well as to manually select the endpoints of the spoken word in the recording. This was done by a human expert who manually labeled the endpoints by visual inspection of the waveform, and by repeated listening of the segmented waveform until the segmentation was satisfactory. This is similar to the procedure carried out by Acero [3], but without using the spectrogram.

2.2 Noise types

We used a total of twenty noise types. Fifteen of them were obtained from the NOISEX-92 [108, 121] database[1]. The other five are publicly-available noise types

[1]These were downloaded from `http://spib.rice.edu/spib/select_noise.html` where they were available at the time of this writing.

found online[2]. The noise files were used as added noise to our own speech recordings. For this reason, only 2 seconds of sound was required from each noise file. From the noise files we downloaded, we decided to discard the first second of the recording and to use the next two seconds of the file as the added noise.

Each noise type is described in the following paragraphs, with the help of figures. For each noise type, the figures show the waveform, the power spectral density estimate using Burg's method, and the spectrogram representation of the noise. Autoregressive modeling can identify the frequencies for each noise type which have high energy. Burg's method is a way to estimate the autoregressive parameters. Several methods are available to estimate an autoregressive model. The Yule-Walker method is one such other method. Later in this thesis we will use the Levinson-Durbin algorithm to solve the normal equations that arise from the least-squares formulation, in order to create the spectrograms of spoken words. The various estimation methods generally lead to similar results. The Yule-Walker method and least-squares method estimate the autoregressive parameters directly. Burg's method first estimates the reflection coefficients and then the parameter estimates are determined using the Levinson-Durbin algorithm.

Air conditioner

This type of noise is a recording from an air conditioner. It produces a low energy noise with almost no variations. Most of the energy is concentrated below 2kHz. This type of noise is useful for conducting experiments because it represents the type of noise which is common in a quiet office environment. A graphical analysis can be seen in Figure 2.3.

Speech babble

This noise is a recording from a canteen with 100 people speaking. There are several conversations taking place. The conversations are mainly in the background, without any legible speech in the foreground. This type of noise is useful to test if a system is susceptible to an environment where there are many people talking in the background. A graphical analysis can be seen in Figure 2.4.

[2]These were downloaded from `http://www.partnersinrhyme.com/pir/PIRsfx.shtml` where they were available at the time of this writing.

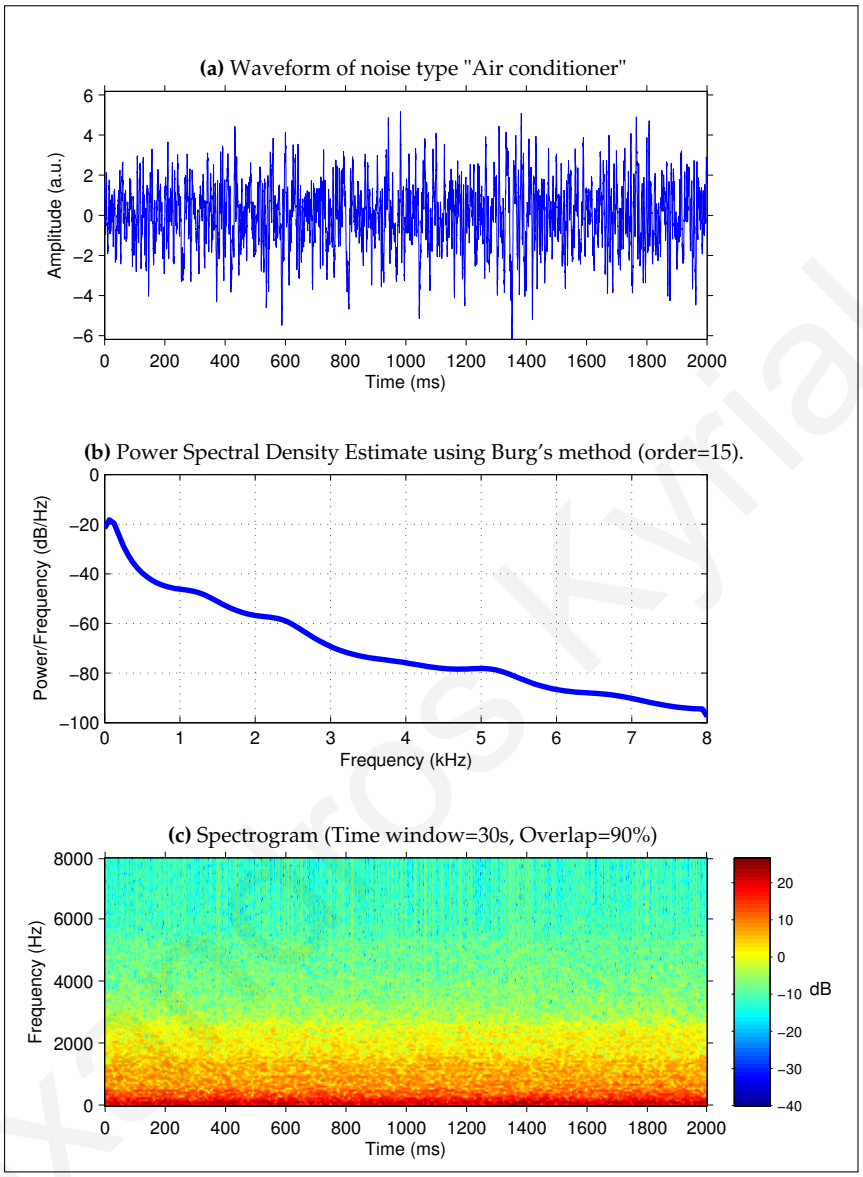

Figure 2.3: Analysis of the noise type "Air conditioner". Most of the energy is below 2kHz.

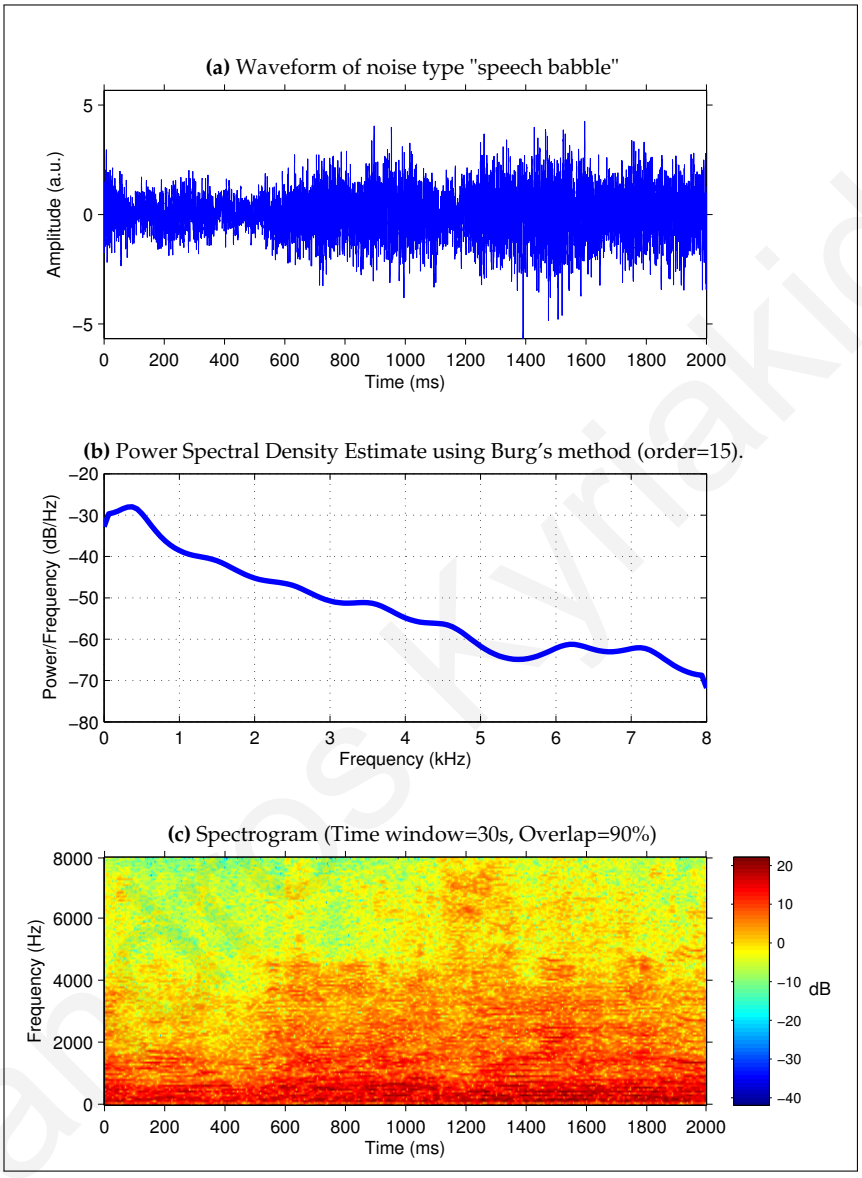

Figure 2.4: Analysis of the noise type "speech babble". It shows the characteristics of speech which has most of its energy below 4kHz, but also includes significant information above 4kHz. The "banded" structure can also be seen which is a characteristic of vowel sounds produced by a fundamental frequency and harmonics.

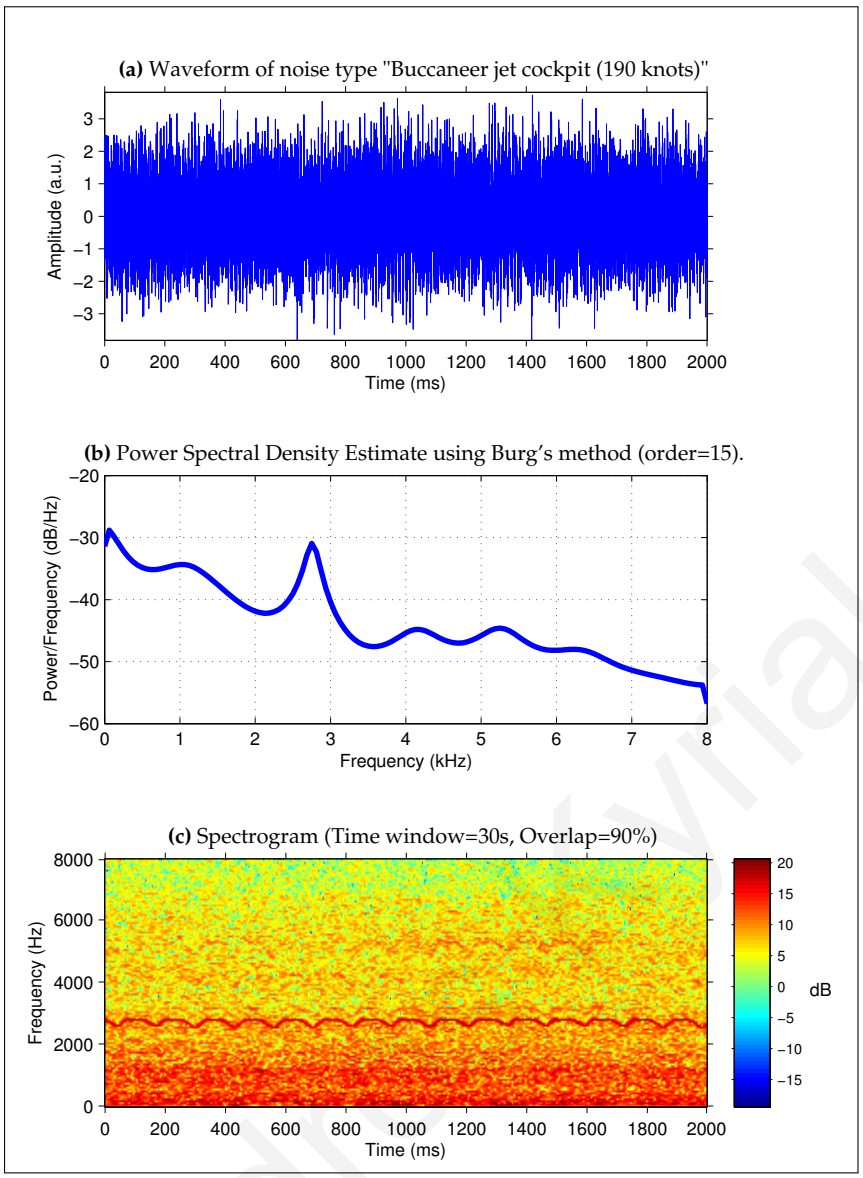

Figure 2.5: Analysis of the noise type "Buccaneer jet cockpit at 190 knots". The noise covers all frequencies and there is a high energy region just below 3kHz.

Buccaneer jet cockpit at 190 knots

This noise type is a recording from the cockpit of a Buccaneer jet traveling at 190 knots. The jet was moving at an altitude of 1000 feet, with airbrakes out. There is a characteristic sound with a frequency just below 3kHz which is varying with time. A graphical analysis can be seen in Figure 2.5.

Buccaneer jet cockpit at 450 knots

This noise type is a recording from the cockpit of a Buccaneer jet traveling at 450 knots. The jet was moving at an altitude of 300 feet. A graphical analysis can be

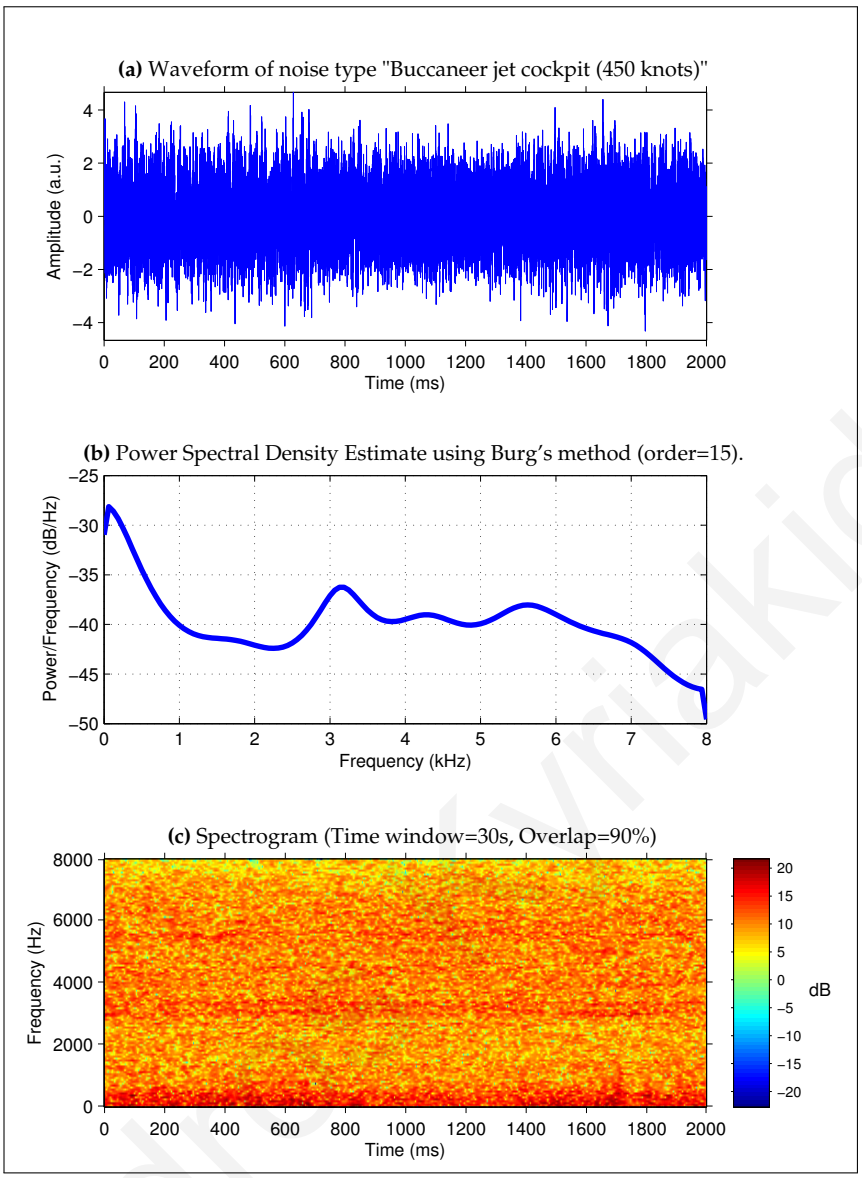

Figure 2.6: Analysis of the noise type "Buccaneer jet cockpit at 450 knots". The noise covers all frequencies.

seen in Figure 2.6.

Conference Room

This noise was recorded from an empty conference room. There is a uniform low frequency sound. This noise is again useful for testing applications which will operate in a quiet office environment. A graphical analysis can be seen in Figure 2.7.

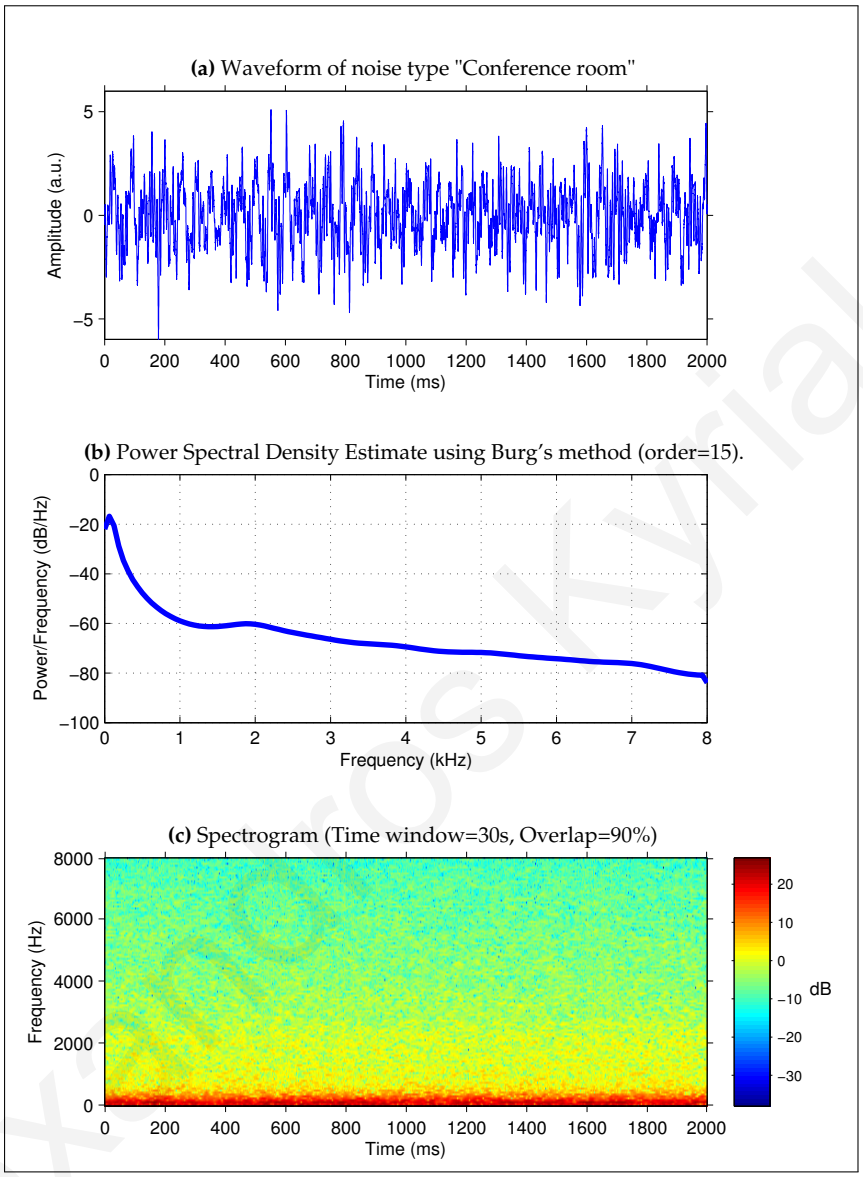

Figure 2.7: Analysis of the noise type "Conference Room". Almost all the energy is concentrated at low frequencies.

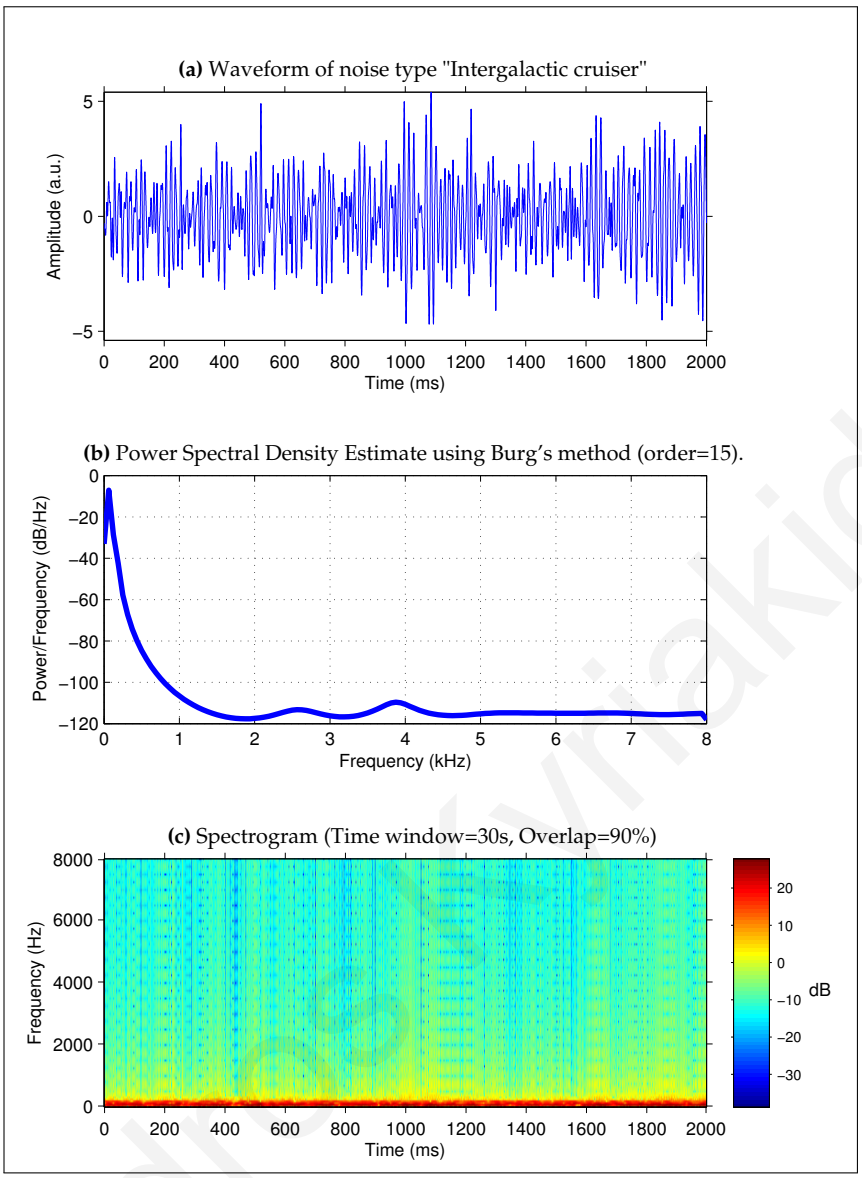

Figure 2.8: Analysis of the noise type "Intergalactic Cruiser". Almost all the energy is concentrated at very low frequencies.

Intergalactic Cruiser

This is a simulated noise of an intergalactic space cruiser. It is a low-frequency background noise. From the noise types we have used, this noise type is the one with most of its energy in the very low frequencies. A graphical analysis can be seen in Figure 2.8.

Destroyer Engine Room

This noise was recorded from the engine room of a destroyer type warship. Most of the energy is below 3kHz. A graphical analysis can be seen in Figure 2.9.

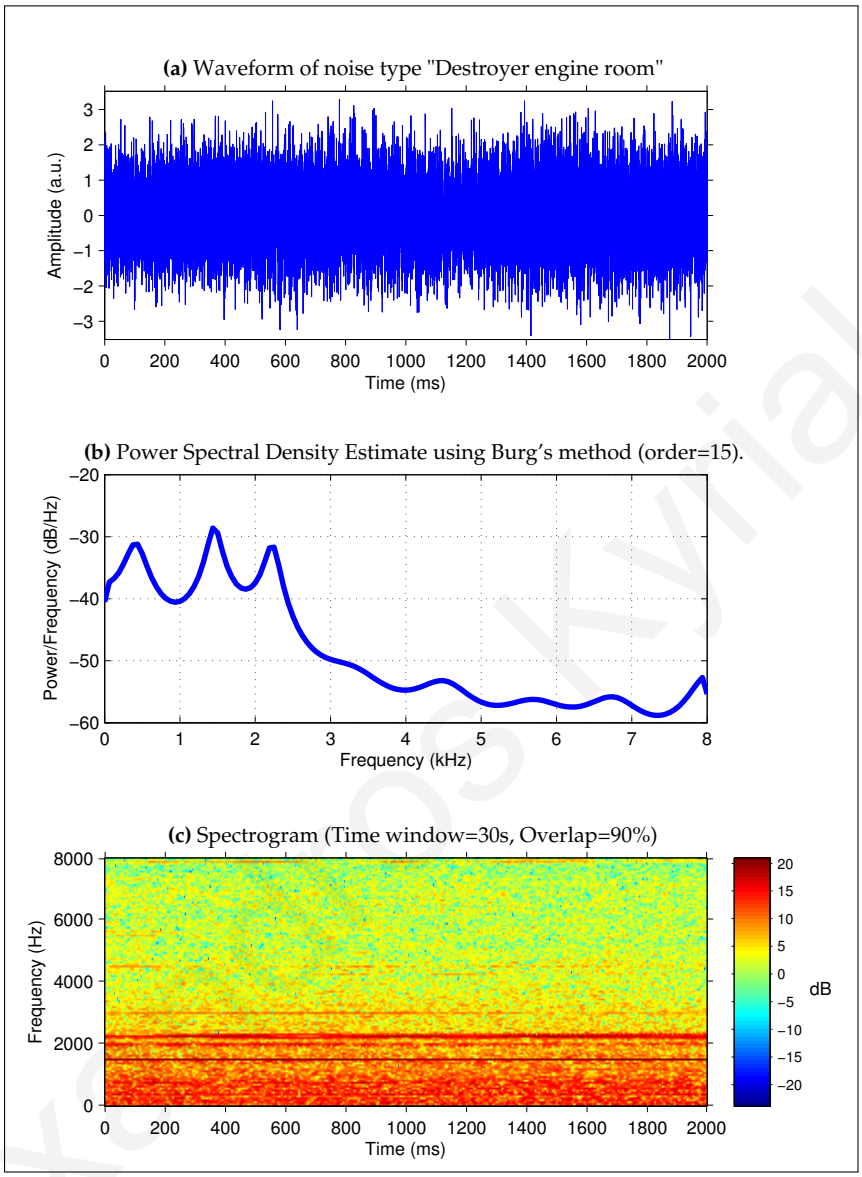

Figure 2.9: Analysis of the noise type "Destroyer Engine Room". The noise covers all frequencies but most of its energy is below 3kHz. There are characteristic frequency peaks around 2kHz.

Destroyer Operations Room

This noise was recorded from the operations room of a destroyer type warship. During the recording there were people talking in the operations room. Therefore, the recording also includes some background speech. A graphical analysis can be seen in Figure 2.10.

F16 cockpit

This noise was recorded at the co-pilot's seat in a two-seat F-16, traveling at a speed of 500 knots, and an altitude of 300-600 feet. There are two frequency regions with high energies. A graphical analysis can be seen in Figure 2.11.

Factory floor (1)

This noise was recorded in a factory near plate-cutting and electrical welding equipment. Most of the energy is in the low frequencies, but there are also high frequencies at certain points in time. A graphical analysis can be seen in Figure 2.12.

Factory floor (2)

This noise was recorded in a car production hall. A graphical analysis can be seen in Figure 2.13.

HF radio channel

This is a recording of noise in an HF radio channel after demodulation. A graphical analysis can be seen in Figure 2.14.

Jet airliner cabin

This is the noise one would hear while sitting in the passenger cabin of a commercial airliner. It is mainly low energy noise caused by the jet engines. A graphical analysis can be seen in Figure 2.15.

Leopard military vehicle

A recording from a Leopard military vehicle moving at a speed of 70 km/h. The engine makes high energy low-frequency noise. A graphical analysis can be seen in

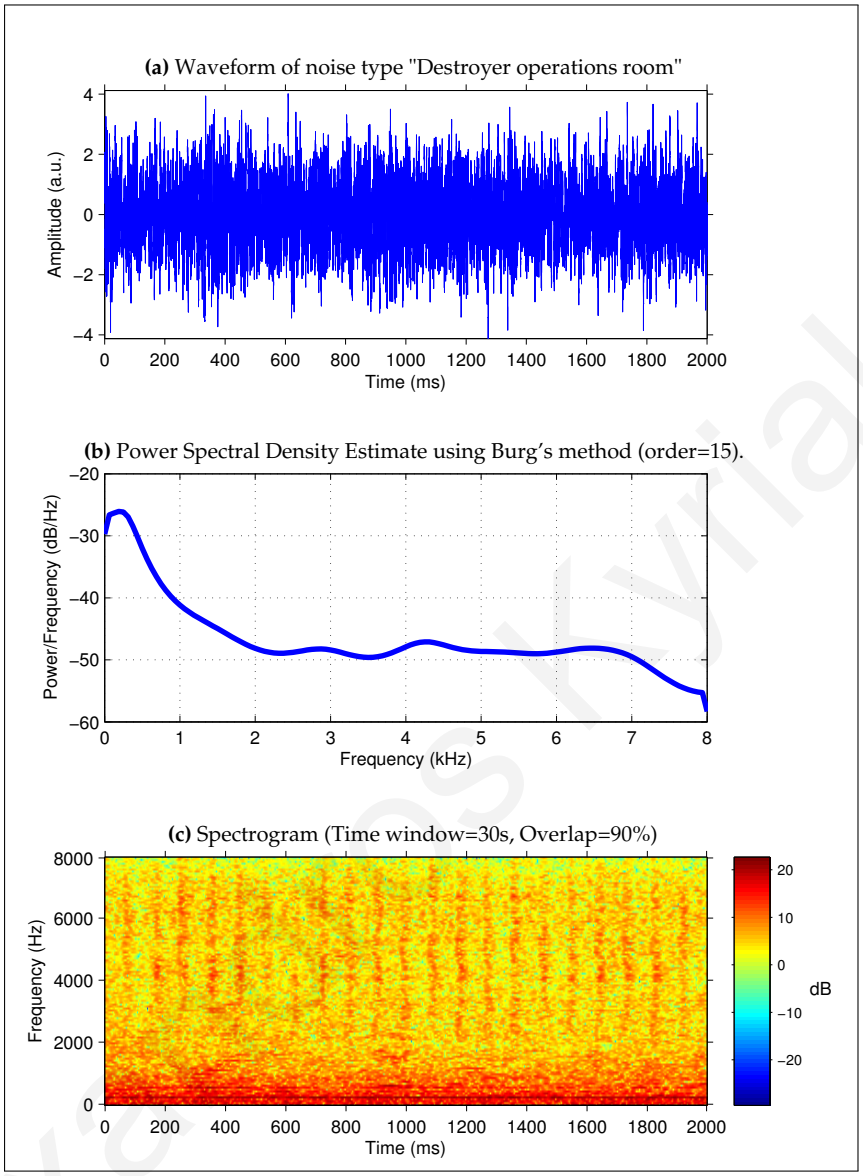

Figure 2.10: Analysis of the noise type "Destroyer Operations Room". Most of the energy is concentrated in the low frequencies. This noise type includes both the sound of the ship and of the people talking in the operations room. The characteristic patterns of speech can be seen in the spectrogram.

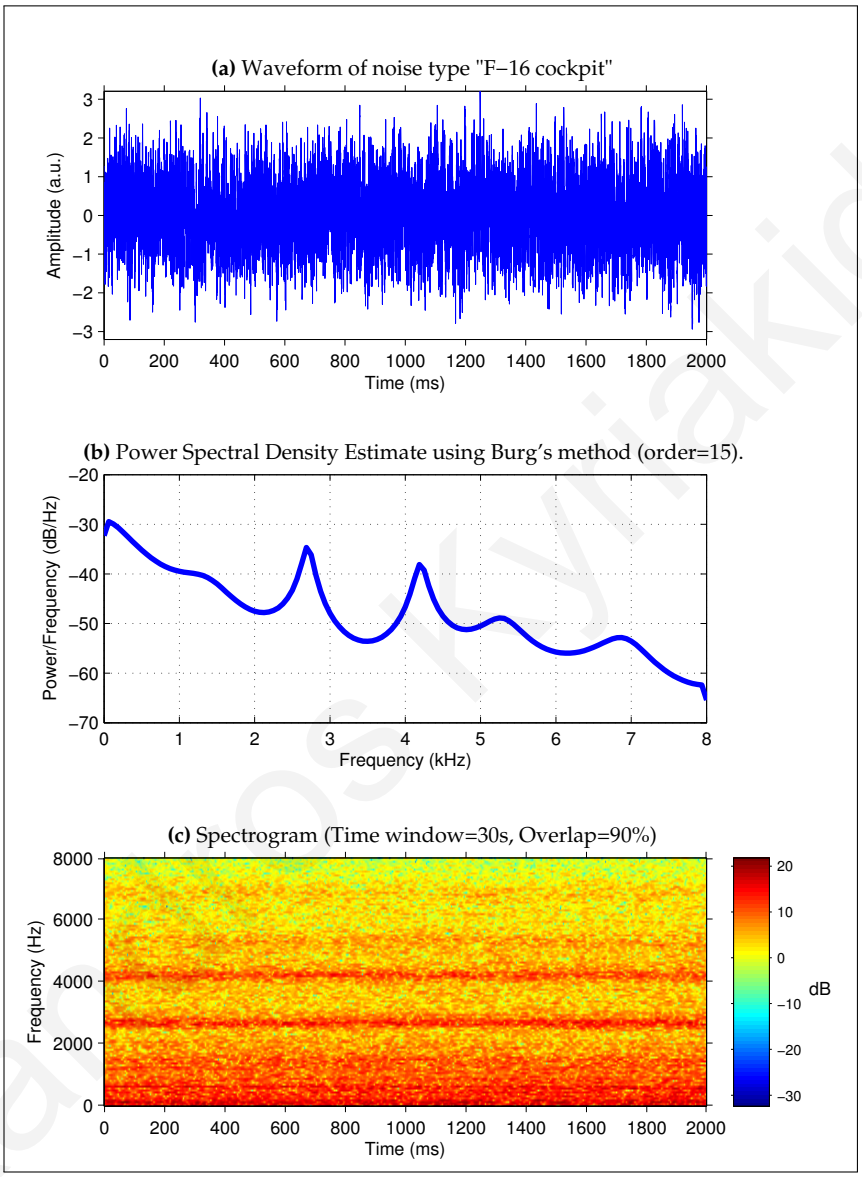

Figure 2.11: Analysis of the noise type "F16 cockpit". This noise type covers all frequencies but there are two frequency regions, around 3kHz and 4kHz, with very high energy.

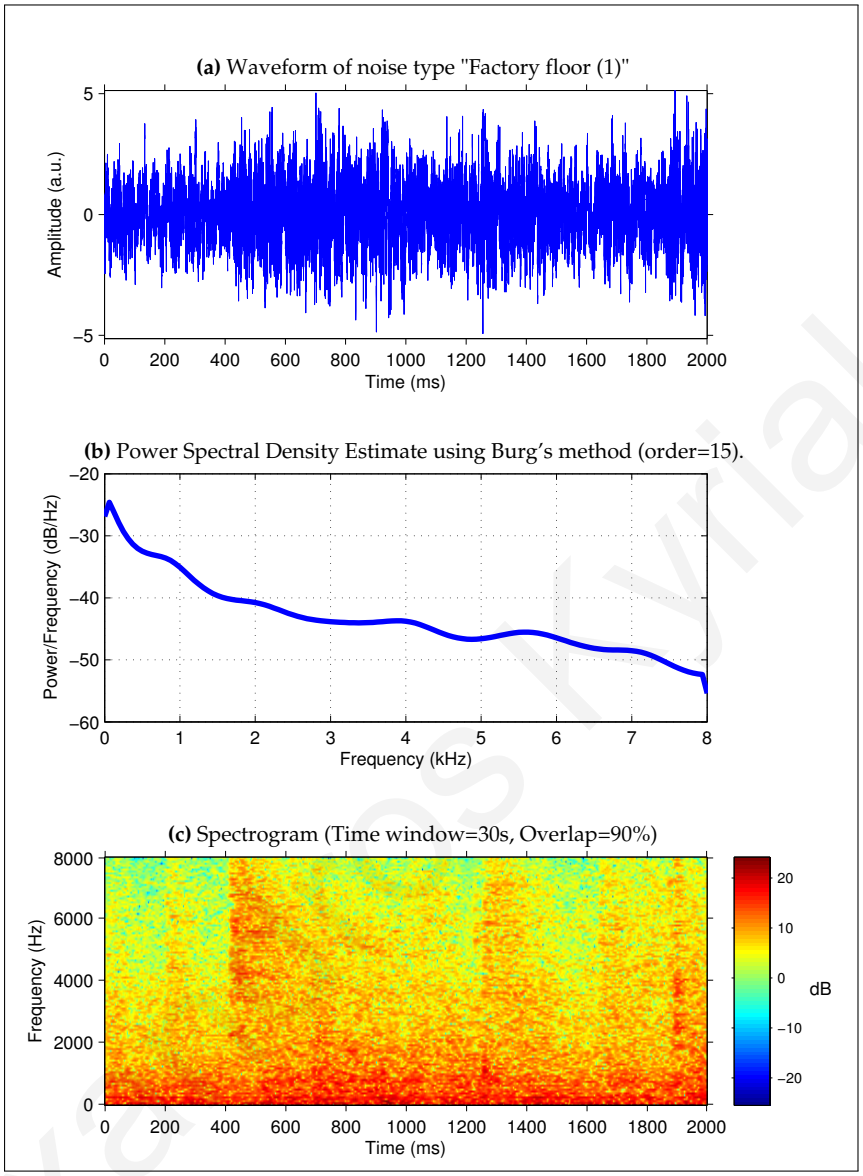

Figure 2.12: Analysis of the noise type "Factory floor (1)". Most of the energy is in the low frequencies, but there are also high frequencies. It can be seen from the spectrogram that at certain points in time there are sounds which span the whole frequency range. These are sounds caused by the machinery in the factory.

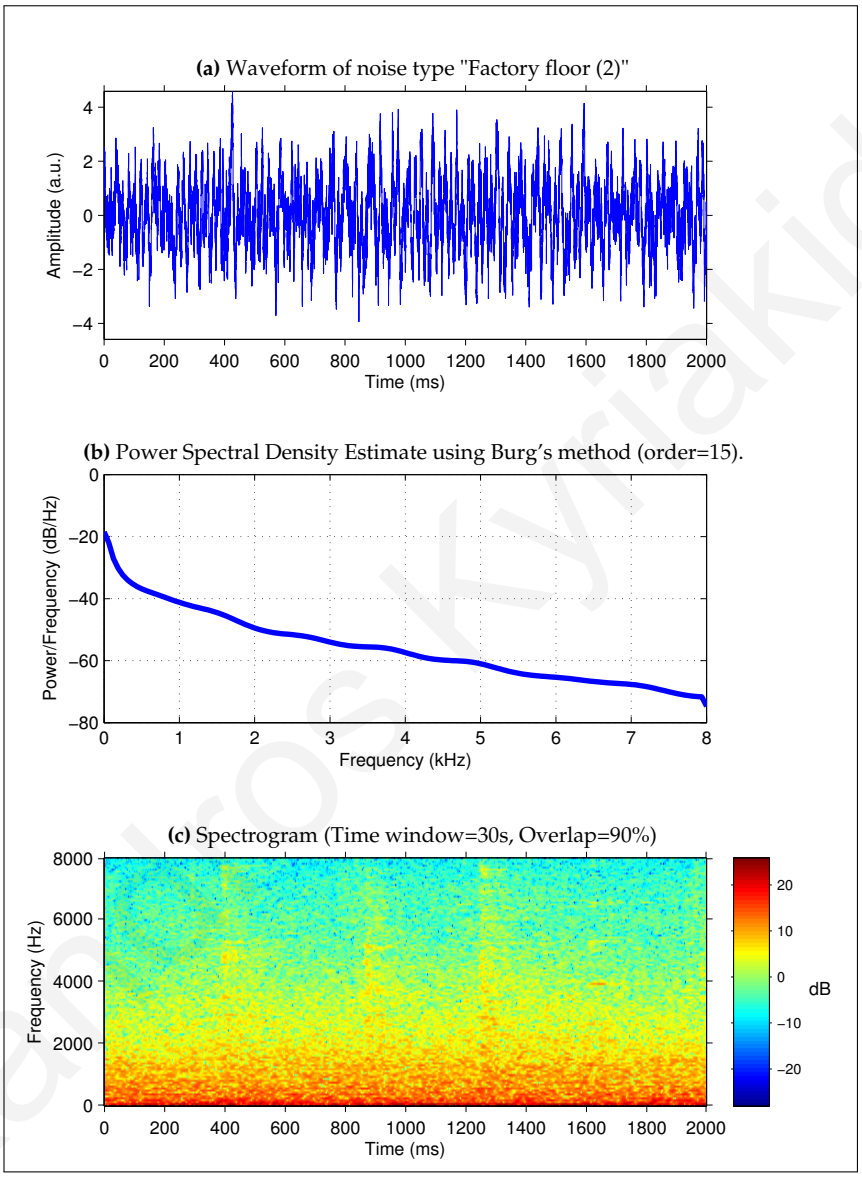

Figure 2.13: Analysis of the noise type "Factory floor (2)". Most of the noise is in the low frequencies. There are some regions with higher frequency sound as well.

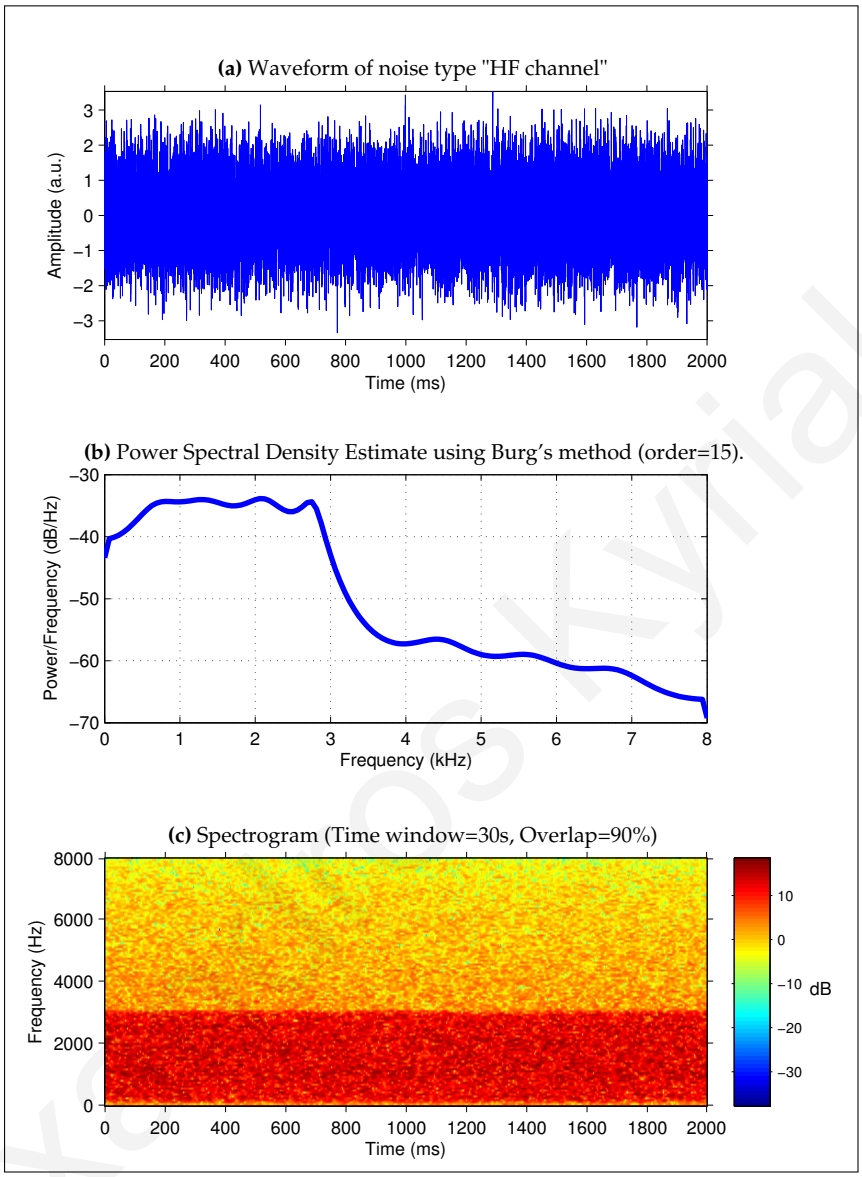

Figure 2.14: Analysis of the noise type "HF radio channel". This is a uniformly distributed type of noise, like white noise, but with the energy concentrated below 3kHz.

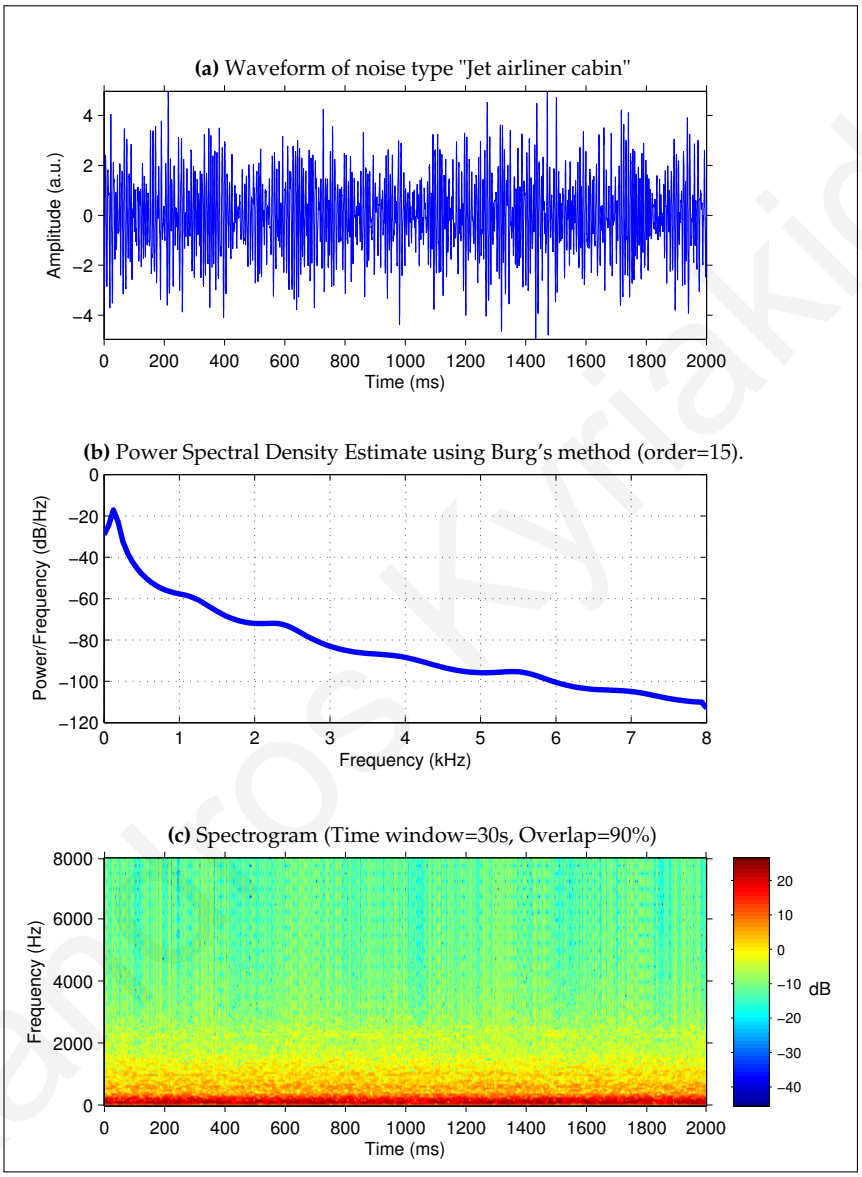

Figure 2.15: Analysis of the noise type "Jet airliner cabin". Most of the noise is low frequency noise.

Figure 2.16.

M109 military tank

This noise was recorded from an M109 military tank which was moving at a speed of 30 km/h. There are some higher frequency sounds in addition to the low frequency sounds of the engine. A graphical analysis can be seen in Figure 2.17.

Machine Gun

This is a recording of a .50 caliber gun fired repeatedly with short pauses in between. In the 2-second recording which we used, the noise starts off with one short burst of firing, followed by a pause, and then another short burst of firing. A graphical analysis can be seen in Figure 2.18.

Car interior

This recording was made inside a Volvo 340 while driving at 120 km/h, in 4th gear, on an asphalt road, in rainy conditions. This is a very useful recording for testing speech recognition applications because such applications are usually required to operate in the car while driving. A graphical analysis can be seen in Figure 2.19.

Street traffic

This is a recording made on a street with cars passing by. A graphical analysis can be seen in Figure 2.20.

Pink

Pink noise has the characteristic that the power spectral density of the noise is inversely proportional to the frequency. It is called pink noise because visible light with the same frequency characteristics appears pink to the human eye. A graphical analysis can be seen in Figure 2.21.

White

White noise has a flat power spectral density. The energy in one frequency band is equal to the energy in any other frequency band. It is called white noise because

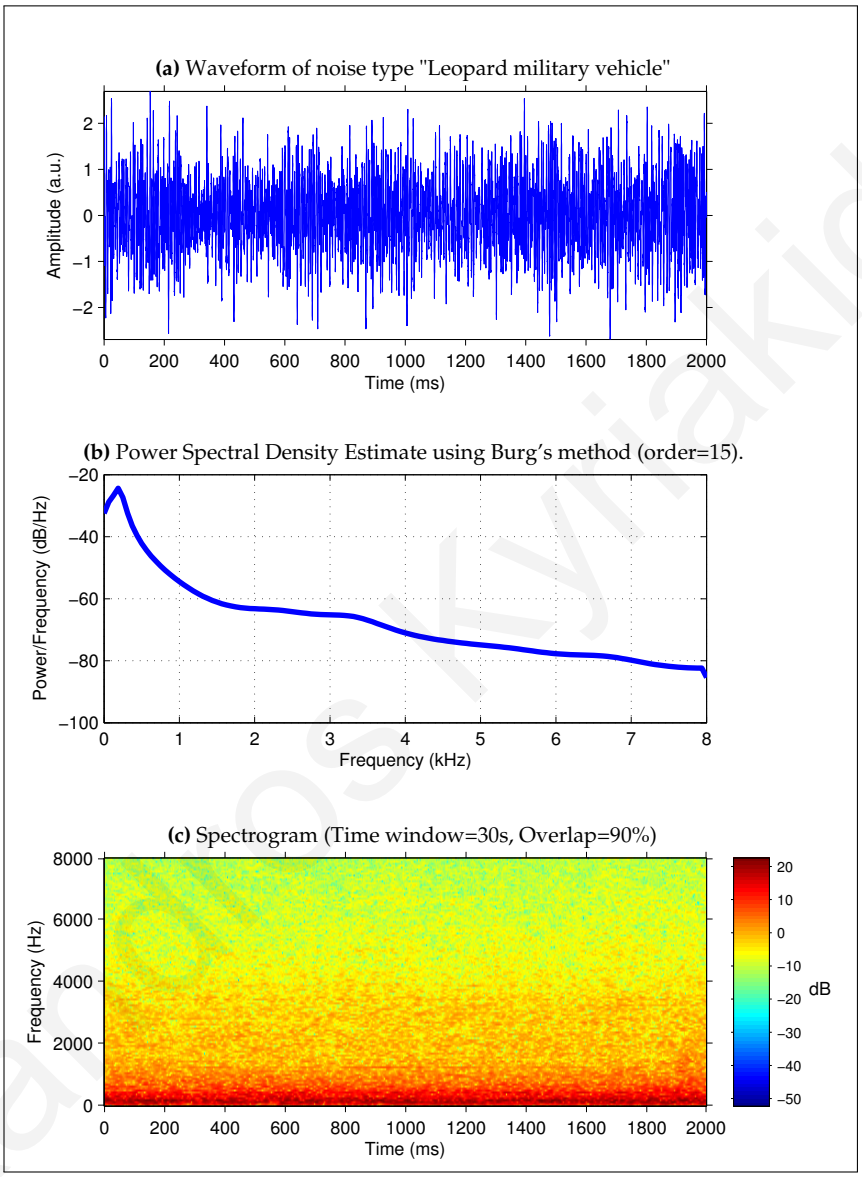

Figure 2.16: Analysis of the noise type "Leopard military vehicle". Most of the noise is in the low frequencies. Some variations can be seen in the very low frequencies. These are caused by changes in the amount of power applied to the vehicle's engines.

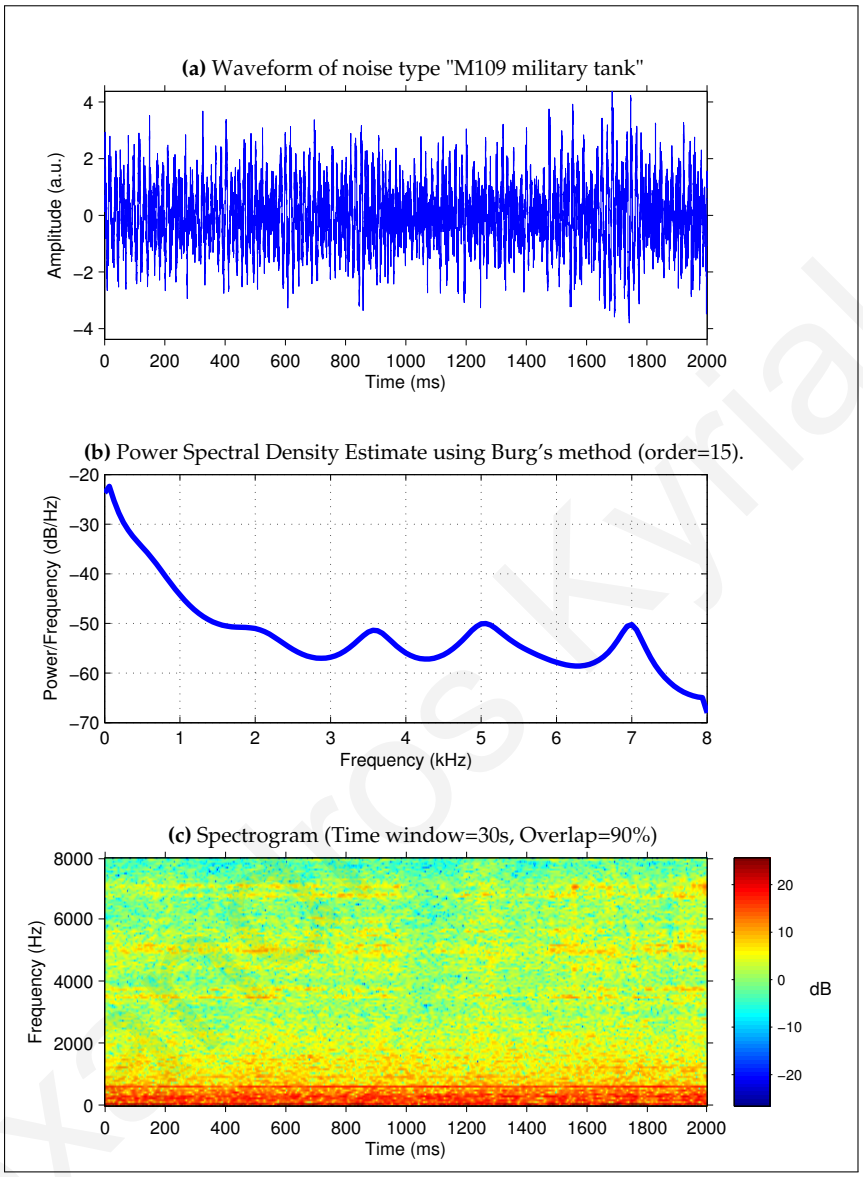

Figure 2.17: Analysis of the noise type "M109 military tank". Most of the energy is in the low frequencies, but there are also some characteristic higher frequencies.

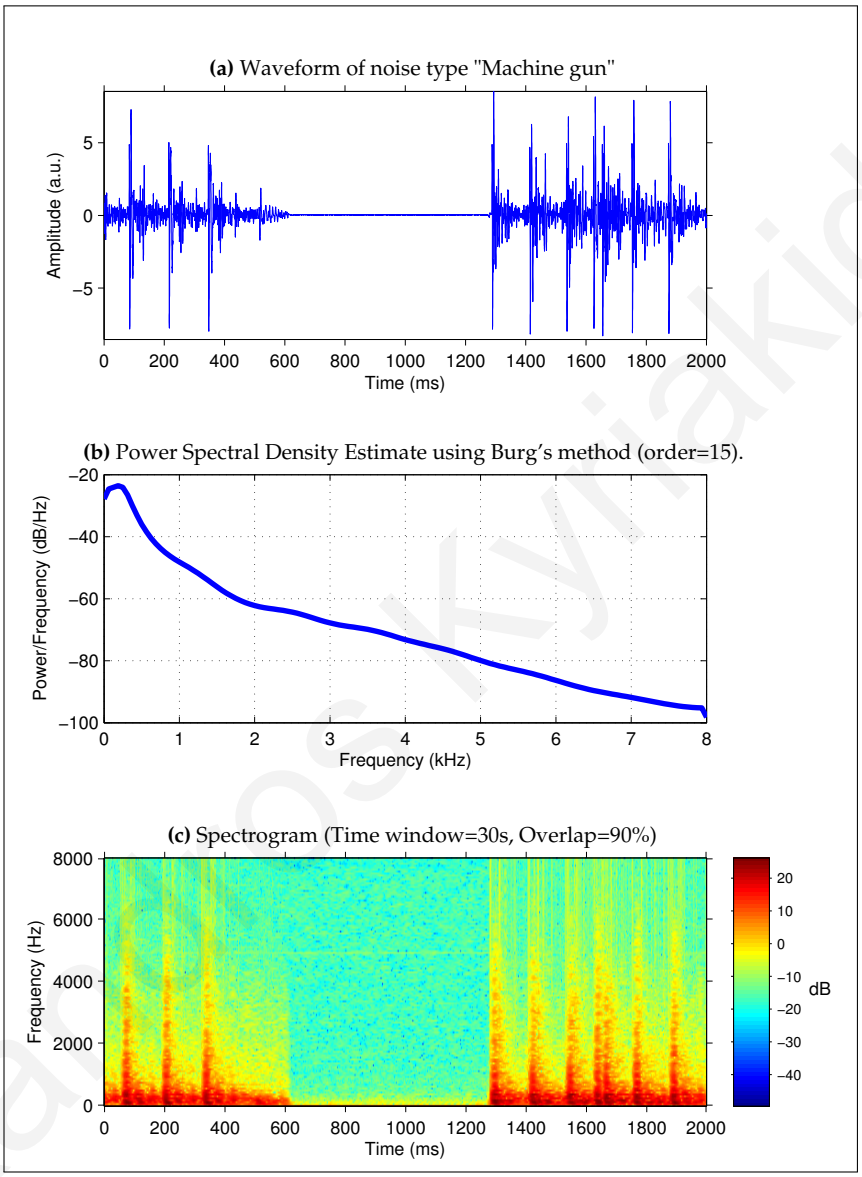

Figure 2.18: Analysis of the noise type "Machine Gun". The machine gun is firing two short bursts, with a pause in between. The sound from the bullet shots show a very characteristic pattern.

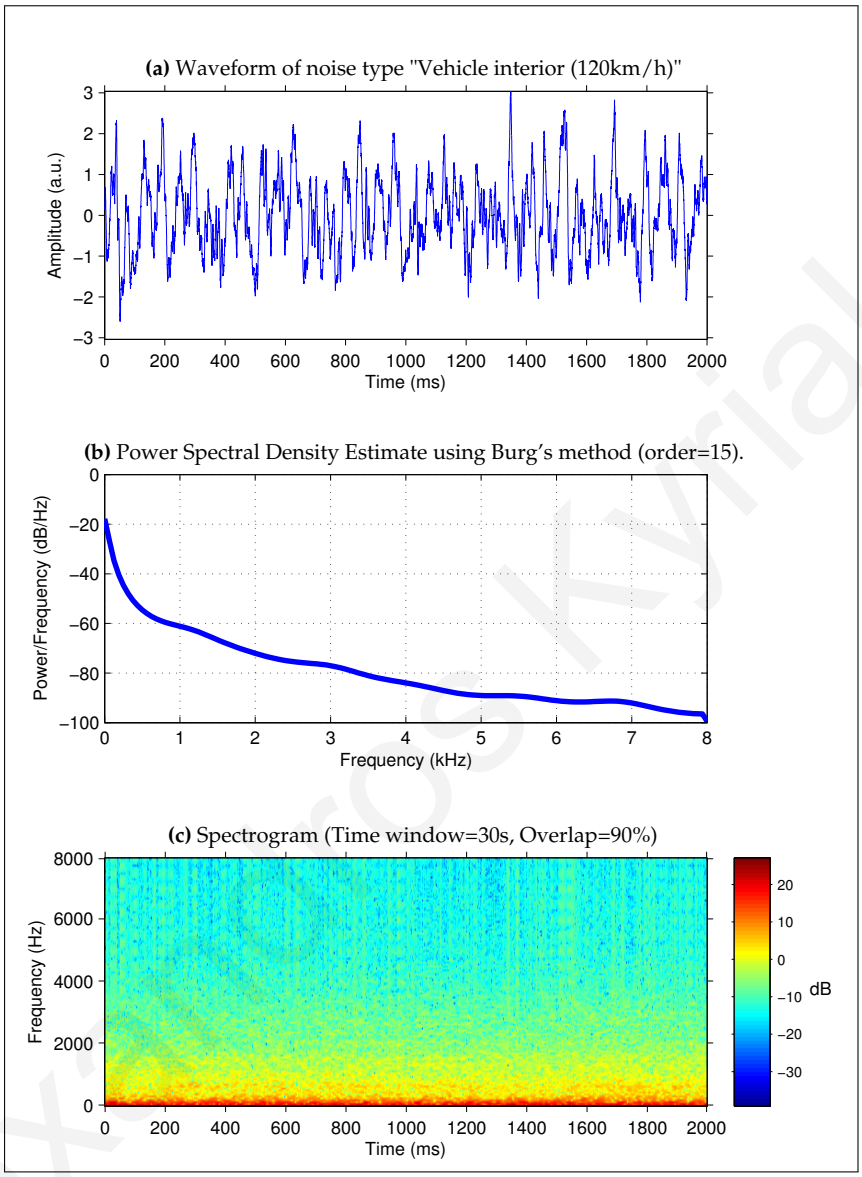

Figure 2.19: Analysis of the noise type "Vehicle interior (Volvo car at 120 km/h)". Most of the energy is in the low frequencies.

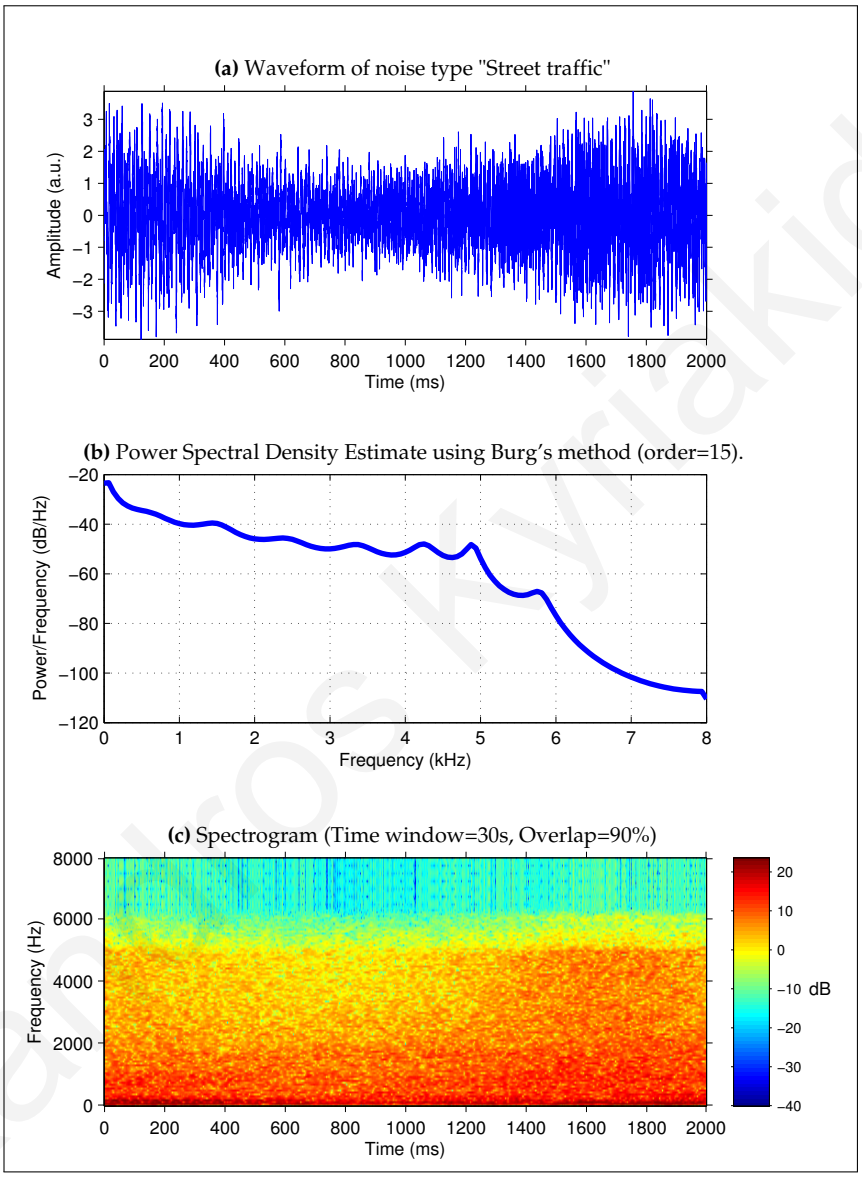

Figure 2.20: Analysis of the noise type "Street traffic". Most of the energy is below 6kHz.

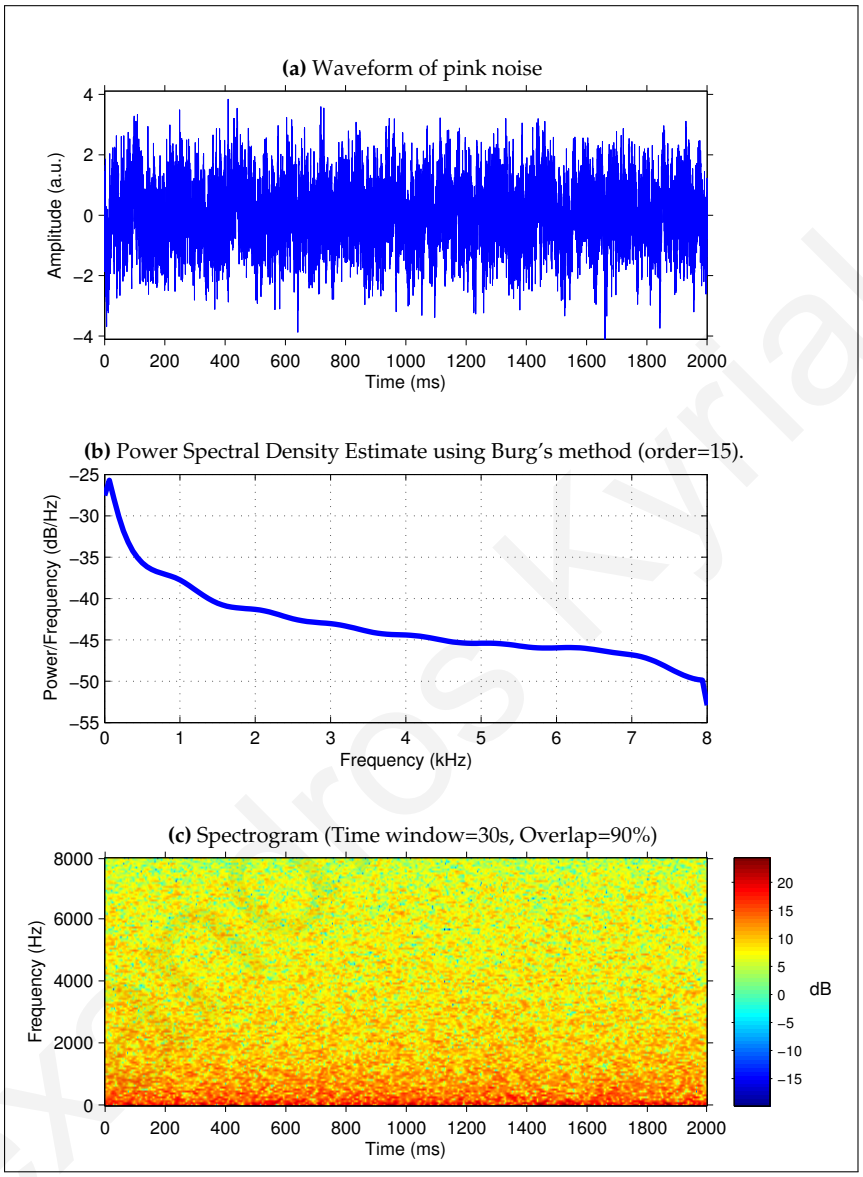

Figure 2.21: Analysis of pink noise.

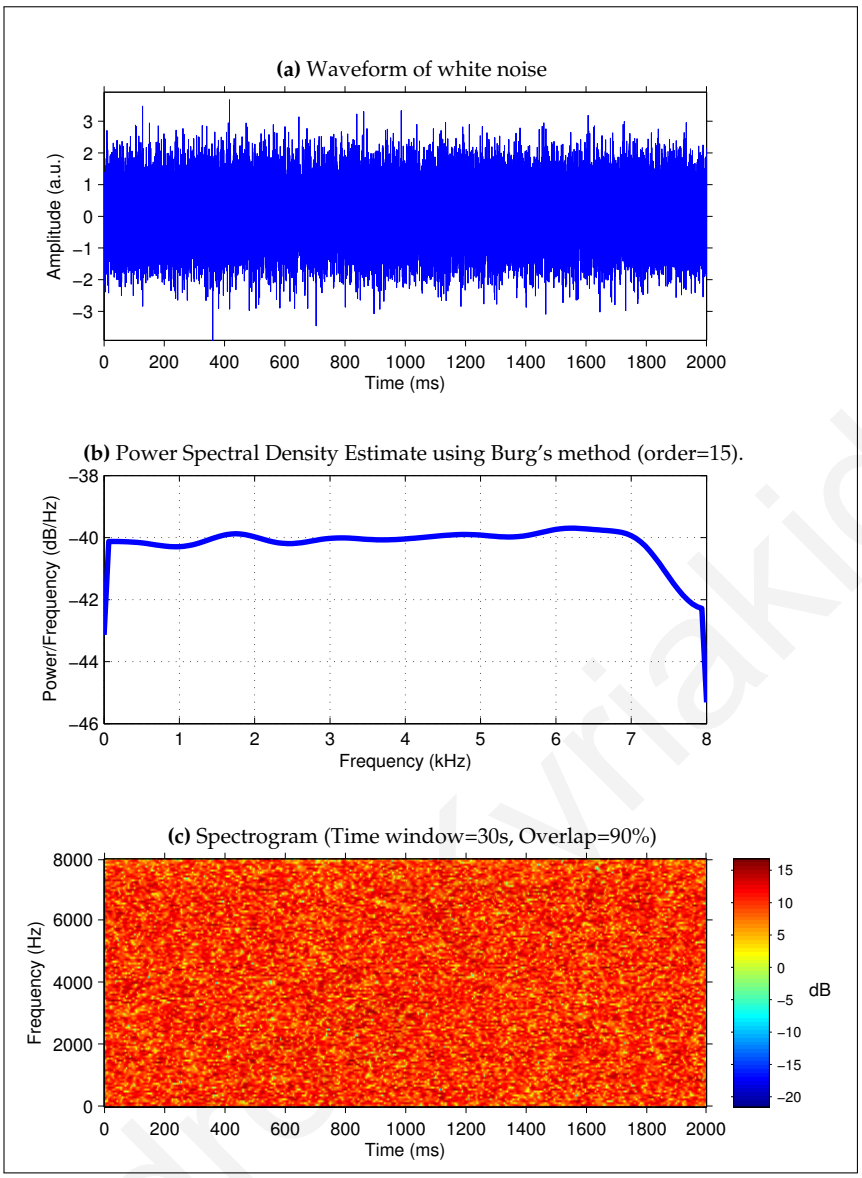

Figure 2.22: Analysis of white noise.

visible light with the same frequency characteristics appears white to the human eye. A graphical analysis can be seen in Figure 2.22.

Chapter 3

Endpoint Detection

3.1 Motivation

The accuracy and robustness of a speech recognition system can be greatly increased by first separating the regions of the input sound signal into speech and non-speech regions. This is especially important in the case of an isolated word recognition system [80]. In the case of isolated word recognition, it is assumed that the input sound signal consists of a single word. Only a certain region of this signal is the actual spoken word. Before the start of the word, and after the end of the word, there are non-speech regions which consist of silence and background noise. Endpoint detection is the process of finding the *start point* and *end point* of a word in a signal, and thus separating the speech region from the non-speech regions. Once endpoint detection is performed, only the speech segment is used as input to the isolated word speech recognition algorithm.

Assuming that endpoint detection is performed accurately, this segmentation can be important for two reasons. The first reason is that the speech recognition algorithm will not need to process non-speech regions. This makes the recognition process faster by reducing computation, and also more accurate. The second reason is that the words can be normalized in terms of time duration. It is evident that when a word is spoken, the time duration of the word is not always the same, even for the same exact word. The duration of the word can change based on the human speaker uttering the word and it can also change based on the situation under which the word is spoken. This variability in time duration can greatly influence speech recognition systems which are based on pattern matching, which

is the typical method used by isolated-word recognition systems [3]. One such pattern-matching speech recognition system is the one presented in this thesis. For our speech recognition system it is important that the duration of each word is normalized. A time-warping operation can normalize the word to a specific time duration. This normalization can only be achieved however, by first performing accurate endpoint detection.

3.2 Background

Detecting the endpoints of a spoken word is not a trivial task, except in the cases of extremely high signal-to-noise ratio [85]. It is a problem that has been studied for several decades [56]. It is one of the most fundamental, important, and difficult problems encountered in speech processing [17]. The large number of publications which address this subject is simple evidence of the high level of difficulty that this problem presents. To a newcomer in the field this may seem strange, because humans have no trouble in distinguishing when a spoken word starts and stops, even in the presence of a significant amount of noise. It still eludes us however how to make a machine that can perform equally as well as humans in distinguishing speech from non-speech. Robust endpoint detection is an unsolved problem in speech processing [92] and it is an important area of speech processing research because it affects numerous applications [130]. Such applications include robust speech recognition, discontinuous transmission, real-time speech transmission on the Internet, and combined noise reduction and echo cancellation schemes in the context of telephony [90]. In a real-world evaluation of a discourse system which used an endpoint detector and an isolated-word speech recognition system, it was found that more than half of the recognition errors were due to errors of the endpoint detection system [46].

Endpoint detection algorithms perform well under conditions of stationary noise and high signal-to-noise ratios (above 30dB). When the signal-to-noise ratio (SNR) is high, the energy of all the speech sounds is greater than the energy of the background noise and so a simple energy threshold is sufficient in determining the regions of speech [85]. However, once noise and other non-speech events, called "artifacts", are introduced then the performance of these algorithms drops significantly. Acoustic background noise and sound artifacts, such as breathing noises before or after the

word which can be wrongly classified as speech, are a problem in real-world sound recordings. It is therefore of paramount importance that the issues of background noise and artifacts are addressed if an endpoint detection algorithm will be used in real-world situations. In such situations, noise can be introduced by the speaker, the recording environment, and by the transmission system [53].

The required characteristics of an ideal endpoint detection system are [96]: reliability, robustness, accuracy, adaptation, simplicity, real-time processing, and no a-priory knowledge of the noise. The most difficult of these to achieve is robustness to noise [47].

3.2.1 Voiced and Unvoiced speech

For certain words it is very difficult to distinguish the endpoints. This is because some phonemes, such as weak fricatives ("f", "th", "h"), weak plosive bursts ("p", "t", "k"), and final nasals [85], have very low energy. This makes it difficult to differentiate the spoken phoneme from background noise, especially considering that such low-energy phonemes appear at the beginning or end of a word. Speech can be divided into **voiced** and **unvoiced** speech. [1] A voiced sound is one which is produced by the larynx. During the production of a voiced sound the vocal chords are vibrating. All vowel sounds are voiced sounds. Voiced sounds also include some nasals (e.g. "m", "n"), certain plosives (e.g. "b", "g"), and voiced fricatives (e.g. "v", "z"). Unvoiced sounds do not use the larynx. The vocal chords do not vibrate when producing unvoiced sounds. These unvoiced sounds include the sibilants (e.g "s", "z", "sh"), plosives (e.g. "t", "k", "p"), and unvoiced fricatives (e.g. "f", "th"). One of the most difficult problems in speech analysis is to correctly classify speech into "voiced speech", "unvoiced speech", and "silence" [83].

3.2.2 Voice Activity Detection

The problem of endpoint detection is closely related to the problem of Voice Activity Detection (VAD), which is also known as Speech Activity Detection (SAD). A VAD system attempts to label regions of a sound signal as either "speech" or "non-speech".

[1] A particularly enlightening demonstration of different phonetic sounds in American English, complete with audio and video, can be found at `http://www.uiowa.edu/~acadtech/phonetics/english/frameset.html`

The sound signal is broken up into small time frames (for example 10ms segments) and each frame is then classified as being a frame with speech or a frame without speech. A VAD system can be used as an endpoint detection system by either applying simple rules, or by using more complex algorithms [73]. A simple rule can be used in the case of a sound recording which consists of only one isolated spoken word. In this case, a simple rule would be to mark the first frame for which the VAD detects speech as the start of the word , and to mark the last frame for which the VAD detects speech as the end of the word. VAD systems are especially important in telecommunication applications where efficient coding of transmitted speech can be achieved by applying silence compression during the non-speech segments of the signal [9]. The benefits of this approach are apparent when one notes that in a phone-based communication about 60% of the time the transmitted signal contains just silence [90].

3.2.3 Summary of methods

Many methods have been tried in order to improve the accuracy and performance of endpoint detection and voice activity detection. The long list of methods used includes: energy thresholds [100, 127, 129], log energy [4], zero crossing rate [8, 48, 85], pitch detection [14], spectral analysis [66], least-square periodicity measures [115], cepstral analysis [39], Linear Predictive Coding coefficients [83], formant shape [43], smoothed likelihood ratio [15], noise robust features and decision rules [56, 66, 107, 129], hybrid detection [53], fusion [110], specialized order statistics filters [92], rank-order statistics [17], autocorrelation functions [135], Mel-scale filter banks [132], spectral entropy [44, 64, 103, 130], noise suppression [131], non-linear likelihood-based projections derived from a Baysian classifier [88], time-frequency features [47], long-term spectral envelope (LTSE) [91], Poincaré recurrence metric [38], bispectrum [126], Hidden Markov Models [3], dynamic programming [57], Support Vector Machines [1], multilayer neural networks [79], Higher Order Statistics [36, 71], a multiple observation likelihood ratio test [93], third-order spectra [70], and change-point detection [57]. The large number of methods tried by researchers over the past few decades is testament to the difficulty of the problem of endpoint detection. Although many approaches have been proposed, the problem of accurate noise-robust endpoint detection still remains unsolved. Some algorithms perform

better than others, but they all have their shortcomings when the noise levels are high. The earliest approaches using classical methods could only perform well at low noise levels. The more recent approaches which employ time-frequency features are better at handling noise.

Classic methods

Classic endpoint detection methods and voice activity detection (VAD) methods use signal energy and zero crossing rate [48, 85]. The zero crossing rate (ZCR) is useful for detecting unvoiced speech [23] because unvoiced speech has low energy but high ZCR. When using signal energy and ZCR, the sound signal is divided into small time windows, which are typically around 10ms. For each time window, the energy of the signal is calculated. If the energy surpasses a preset threshold, then the time window is classified as "speech" based on the assumption that speech regions have a higher energy than the background noise. To accommodate varying noise levels, the preset threshold can be recalculated for each analysis window if needed. The ZCR uses a similar approach. The ZCR of noise is assumed to be significantly larger than that of speech. A preset threshold can therefore be used based on the ZCR. Both these assumptions for energy level and ZCR fail at low SNRs. Figure 3.1 taken from [104] shows the energy and ZCR for the spoken digits "zero six" under non-noisy conditions. It can be seen from the figure that the ZCR is important for detecting the "s" sound at the beginning of the word "six" which appears between the 1200ms mark and 1400ms mark.

Classifier-based vs. Rule-based methods

Although there are many types of endpoint detection methods, they can be divided into two broad categories [44, 88]. One category is rule-based. In their great majority these methods employ thresholds for making decisions. These methods extract features from the sound signal and compare them to a threshold. The thresholds can either be fixed, or they can be adaptive and change based on the input signal. It is important to note that although the threshold-based methods are not explicitly trained on training data, the thresholds are still defined empirically after experimenting with speech data. When using rule-based methods, a specific rule needs to be created for each feature. So every time a new feature is introduced, a new rule has to also

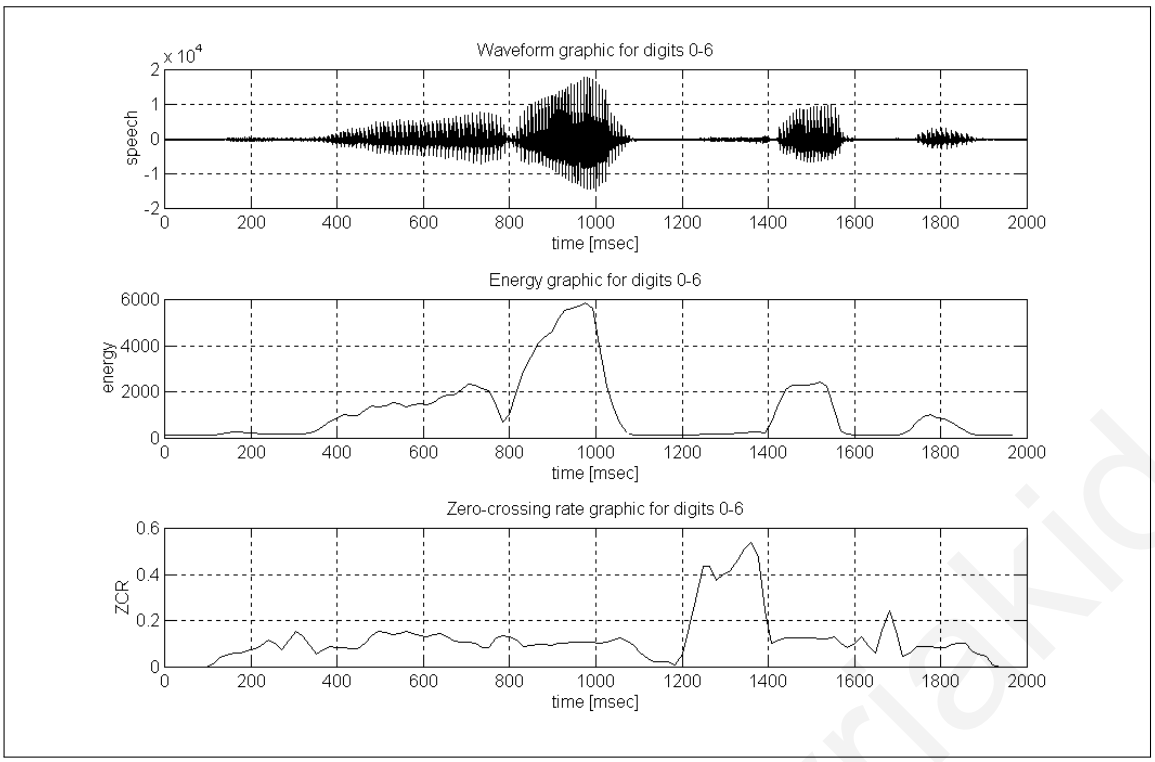

Figure 3.1: Waveform, energy and ZCR graphics for the words "zero-six". (Figure taken from [104])

be defined. This makes it difficult to have a large number of features when using rule-based methods. The other category is classifier-based which rely on pattern-matching. These methods also extract features, but in this case the features are used in order to estimate model parameters. Training data is used to train a model to recognize patterns. One disadvantage of classifier-based models, including models used for speech recognition as well as those used for for endpoint detection, is that there could be a mismatch between training conditions and testing conditions. If the models are trained on noiseless data, and then tested on data with noise, and if the features are not robust to noise, then the accuracy of the model drops significantly. Such models are therefore not suitable for noisy environments [90]. Another concern when using classifier-based models is the relatively large amounts of data needed for training when a large number of features is used [88].

Explicit vs. Implicit methods

Another important way to differentiate between types of endpoint detection methods is to separate them into explicit and implicit methods [53]. In the explicit methods, the

endpoints are determined independently, without the need of a speech recognition system. Therefore the endpoints are chosen explicitly, before any speech recognition system is used. In contrast, the implicit methods use a speech recognition system to determine the endpoints. As stated in [53], for implicit methods:

> All (reasonable) combinations of beginning points and ending points are used and the best output from the pattern similarity stage and decision rule (lowest distance) is used to implicitly define the word endpoint as well as the recognized word.

The literature suggests that implicit methods perform better than explicit methods [3]. Furthermore, hybrid methods have also been developed which use aspects from both explicit and implicit methods. It has been shown that these hybrid methods provide the best endpoint detection performance [57]. One way to overcome the poor accuracy of endpoint detection when performing speech recognition is to eliminate explicit endpoint detection altogether [128]. Systems which use Hidden Markov Models or Dynamic Time Warping, do not require explicit endpoints. They only require that the speech to be recognized is completely contained within the input signal.

3.2.4 Comparison of methods

The broad categories of *classifier-based vs. rule-based* methods and *explicit vs. implicit* methods are useful for describing endpoint detection methods. To compare the methods even further, more characteristics of endpoint detection methods can be considered. Table 3.1 attempts to list these characteristics in a concise way, and also provides references to publications associated with some of these characteristics.

3.2.5 Time-frequency features

In order to improve the performance of endpoint detection systems under noisy conditions researchers have attempted to use a wide variety of features. Some of the most successful endpoint detection systems use time-frequency features [47, 57, 88, 132].

Table 3.1: Comparison of the characteristics of endpoint detection methods, with references to examples.

Training required [1]	No training required [17]
Use of thresholds [85]	No use of thresholds [57]
Frame-by-frame decisions [8]	Requires whole word before making decision [47]
Requires calculation of statistics before start [103]	Does not require calculations before start [85]
Computationally intensive [17]	Low computational cost [8]
Estimation of background noise [57, 66]	No estimation of background noise
Pitch and frequency information used [47, 88, 132]	No frequency information used [53, 85]
Noise-reducing filters used [80]	No filtering [85]
Robust to noise [38, 66]	Not robust to noise, or only robust to one type of noise [17, 85]
Uses longer-term information [91]	Uses shorter-term information [8]
Isolated word endpoint detection [57]	Continuous speech endpoint detection [4]
Real-time processing [56, 88]	Batch-mode processing [88]
Assumes an initial recording interval (e.g. 100ms [85], 200ms [66]) of no speech being present	Does not make any assumptions about where the speech starts

Time-frequency parameter

Junqua et al. [47] created a parameter they called the "time-frequency parameter" (TF) which is based on the energy in the frequency band 250-3500 Hz. This frequency band corresponds to the vowel portions of speech and is therefore useful for distinguishing between speech and noise. With this simple approach they obtained better results than previous methods which did not use frequency-based features. A disadvantage of their method is that it needs to empirically determine the thresholds and has ambiguous rules which are not easily determined by a human [132]. Also, as we show later, better results can be obtained by using higher frequencies so that non-vowel sounds can also be captured.

Adaptive Time-frequency parameter

To improve on the TF parameter method, Wu et al. [132] created the "adaptive time-frequency parameter" (ATF) which uses multi-band spectrum analysis instead of a single band. Their method is termed "adaptive" because it adaptively chooses the proper frequency bands to use. They employ 20 frequency bands based on the mel-scale frequency bank. An important observation in their experiments is that when noise is added to a speech signal, some frequency bands are corrupted more severely by the noise than others. For performing good endpoint detection it is therefore important to ignore the frequency bands which have little word signal information and to keep the useful frequency bands which contain more word signal information. This is easily illustrated by an example. Figure 3.2 shows the smooth and normalized frequency energies of a clean speech signal, for 20 frequency bands and 100 time frames. Figure 3.3 shows the same speech signal with added white noise at 10dB SNR. Comparing sub-figures (b) in these two figures it can be seen that the fifth frequency band (top graph) retains word signal information after the addition of white noise. Conversely, the eighteenth frequency band (bottom graph) loses almost all word signal information when white noise is added. In this example therefore, the fifth frequency band is useful in determining the endpoints of the word whereas the eighteenth frequency band should be ignored. An additional observation from this work is that as the energy of the background noise increases, the number of useful frequency bands, which can be used for endpoint detection, decreases.

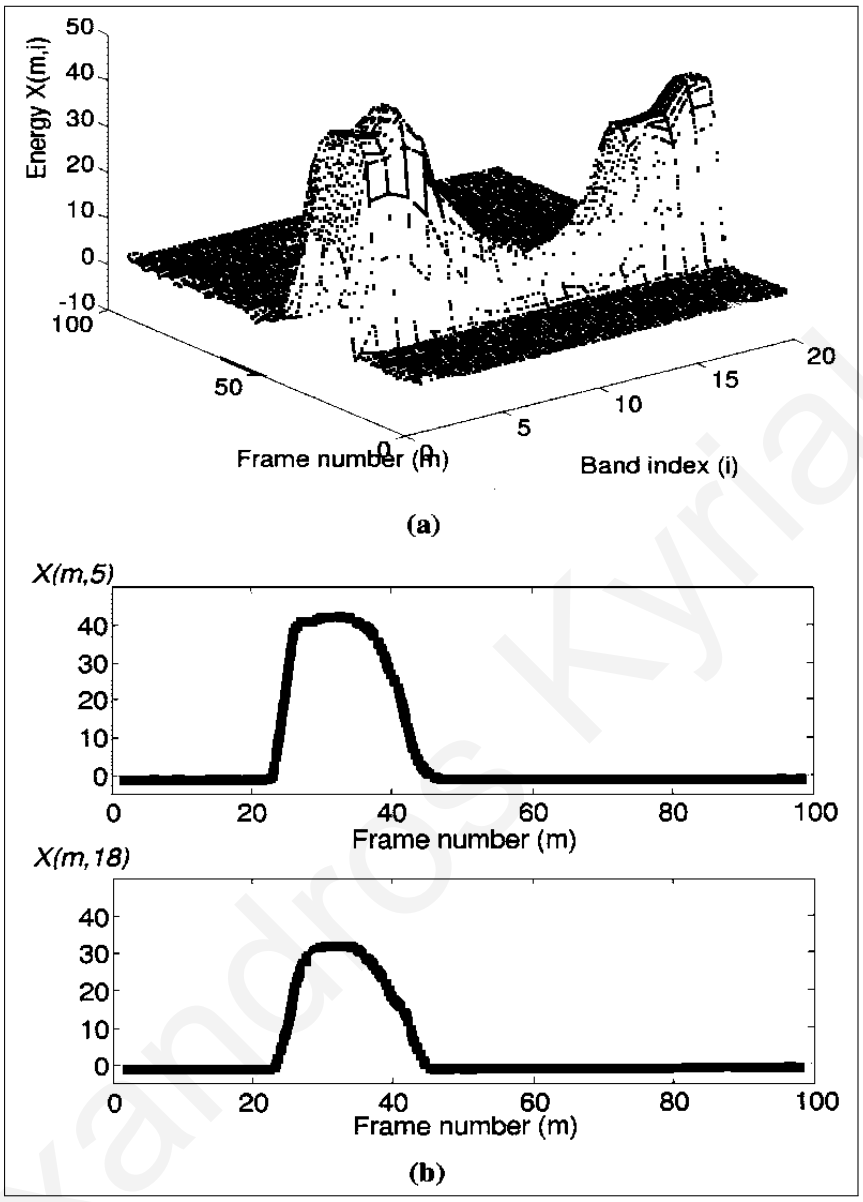

Figure 3.2: Multi-band spectrum analysis of a clean speech signal with length of 100 time frames. **(a)** Smoothed and normalized frequency energies, $X(m,i)$, on 20 frequency bands. **(b)** Smoothed and normalized frequency energies, $X(m,5)$ and $X(m,18)$, on the fifth and eighteenth frequency bands. (Figure taken from [132])

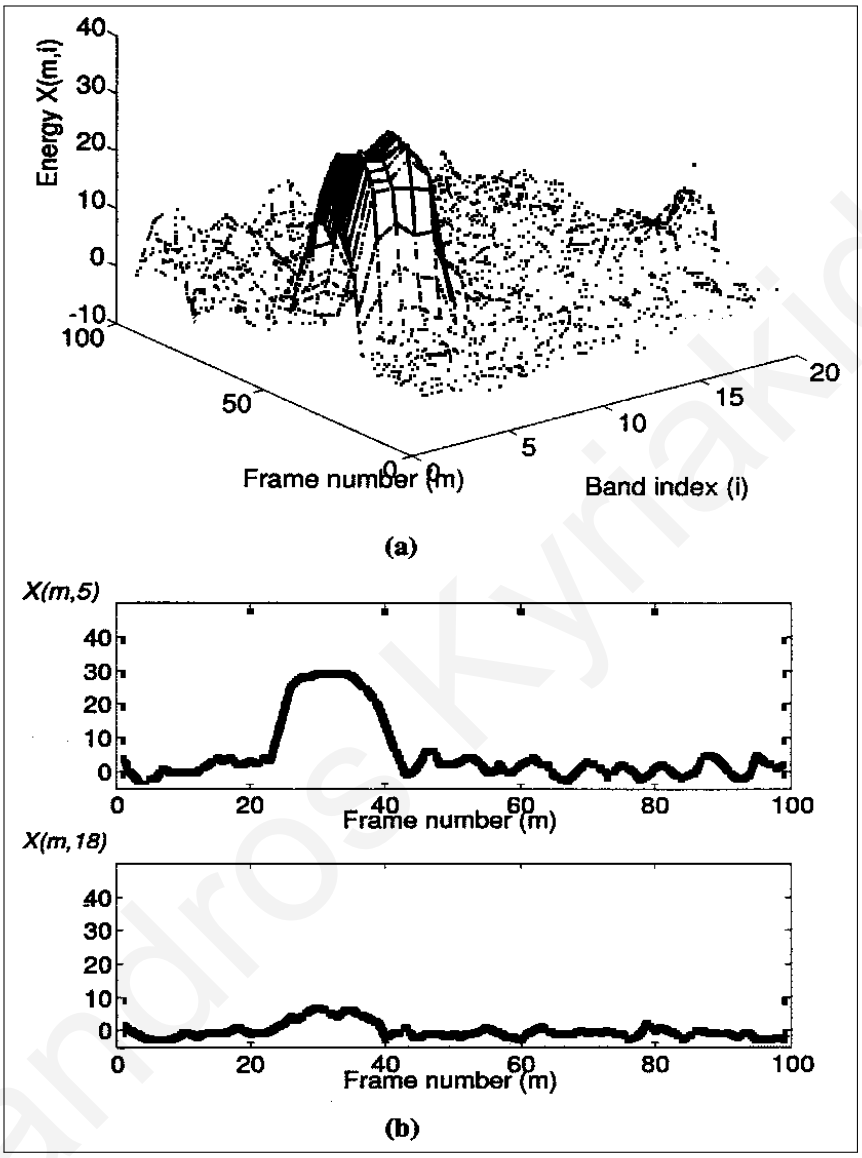

Figure 3.3: Multi-band spectrum analysis of the speech signal in Figure 3.2 with additive white noise of 10dB. **(a)** Smoothed and normalized frequency energies, $X(m,i)$, on 20 frequency bands. **(b)** Smoothed and normalized frequency energies, $X(m,5)$ and $X(m,18)$, on the fifth and eighteenth frequency bands. (Figure taken from [132])

Spectral representations

Features used by speech recognition systems are typically based on spectral representations which can be derived from the short-term Fourier transform of the signal [88]. It is therefore reasonable to use such features for endpoint detection systems as well. The short-time Fourier transform is the most commonly-used tool for spectral analysis. In speech processing, another commonly-used tool is Linear Predictive Coding (LPC) which can be used to capture the spectral envelope of a speech signal. An interesting extension to using spectral representations for endpoint detection is the use of spectral entropy [103, 130].

Linear Predictive Coding

Linear predictive coding (LPC) is defined as a digital method for encoding an analog signal in which a particular value is predicted by a linear function of the past values of the signal [12]. A linear predictive filter calculates the value of the next sample by a linear combination of the previous samples. LPC can be used to model speech as an autoregressive process. The LPC model consists of a number of coefficients which are the weights of the linear combination of previous samples. The *order* of the LPC model defines the number of previous samples to use, and therefore also defines the number of coefficients. The coefficients can be calculated using an optimization algorithm which minimizes, in the least-squares sense, the error between the actual signal samples and predicted signal samples. A higher order model, with more coefficients, uses a larger number of previous samples for making predictions and is therefore better at capturing rapidly-changing characteristics of the signal. A lower order model on the other hand, can only capture characteristics which do not change as rapidly. In our work we use the Levinson-Durbin algorithm to find the coefficients.

Speech production in humans can be modeled as air being pushed from the lungs (the source) and through the vocal tract (a filter) to generate speech. This is called the *source-filter model* for sound production. The source-filter model is the model that is used in linear predictive coding. It is based on the idea of separating the source from the filter in the production of sound [12]. LPC attempts to find a set of parameters to model the vocal tract during the production of speech.

Endpoint detection systems can segment a speech signal into time frames and

perform LPC analysis on each frame. The LPC coefficients obtained for each frame can then be used by endpoint detection systems as features [83]. Furthermore, the frequency response of the linear predictive filter obtained after LPC analysis can itself be used to create features. The order of the autoregressive model which is chosen is very important. In speech processing, it is common to use an order in the range from 8 to 14. In endpoint detection systems, it is common to use 8th-order [83, 128] and 10th-order [71] LPC analysis. Rabiner [83] used a distance function which included the 8 LPC coefficients as features, in order to discriminate each 15ms time frame of the signal into one of three classes: "silence", "unvoiced speech", and "speech". The distance was calculated on the training data by applying a nearest-neighbor approach. The authors claim that an advantage of this technique with LPC coefficients is that all spectral information of the signal is used. Although LPC analysis has been used successfully in endpoint detection systems, there are also some authors who claim that the linear prediction model is not well suited for endpoint detection systems [57]. One claim is that LPC is quite successful for modeling vowels, but not particularly suitable for modeling nasal sounds and fricatives [132].

Spectral entropy

A very interesting approach which has been used successfully in endpoint detection systems is one which calculates entropy [51] in the time-frequency domain. This is referred to as spectral entropy [103]. First, the probability density function (pdf) of the spectrum of each time frame of the signal is estimated. This can be achieved by using the spectrogram of the sound signal. In [103], the spectrogram is derived using the Fast Fourier Transform (FFT). The pdf can then be calculated by using the frequency components of each frame. Following this, the spectral entropy is measured based on the pdf. It was found that spectral entropy has a higher value in the segments of the signal which contain speech than in segments without speech. This remained true even in the presence of different types of noise at low SNRs. A specific enhancement to the algorithm was made which places an upper and lower bound on the pdf. The lower bound effectively removes noise which has almost constant power spectral density values over all frequencies, like white noise. The upper bound eliminates noise which is concentrated on specific frequency bands,

such as a single-frequency sine wave.

Detecting patterns in the spectrogram

A recent publication on endpoint detection describes an algorithm which detects speech based on the banded structure of the spectrogram of speech [130]. It attempts to detect patterns in the spectrogram which represent speech. Figure 3.4 shows the waveform and spectrogram of a mixed signal consisting of vehicle noise, multi-talker babble noise, factory noise, speech, and white noise. The banded structure only appears in the case of speech. Spectral entropy is a measure which attempts to capture this banded structure in order to detect the parts of the spectrogram which contain speech. Wu and Wang [130] claim that the spectral entropy alone cannot adequately capture this banded structure, and so they devised the adaptive band-partitioning spectral entropy (ABSE) parameter. This parameter separates the spectrogram into 32 frequency bands and adaptively discards frequency bands which are corrupted by noise. This is the same idea as the adaptive time-frequency (ATF) parameter proposed by Wu et al. [132], described previously. Additionally, the multi-band analysis can also enhance the banded nature of the speech spectrogram. The most important advantage of the ABSE parameter is that it is robust to noise. The banded structure in the speech spectrogram is itself robust to additive noise. This is illustrated in Figure 3.5 where the waveforms and spectrograms of clean speech and noisy speech are compared. Four kinds of noise (vehicle noise, factory noise, white noise, and multi-talker babble noise) are added to the clean speech at 0dB. The banded structure of the speech can still be seen in all the spectrograms of the noisy speech. In some cases (e.g. for white noise) the banded structure degrades more than in other cases (e.g. for vehicle noise), but it is still present. Looking at this example, one can speculate that an algorithm that captures the patterns of speech in a spectrogram can lead to a noise-robust endpoint detection system.

3.3 Endpoint Detection System

In this section we describe a novel endpoint detection system for isolated words. The first step converts the sound signal into a time-frequency representation called a spectrogram. The spectrogram is generated using a low-order autoregressive

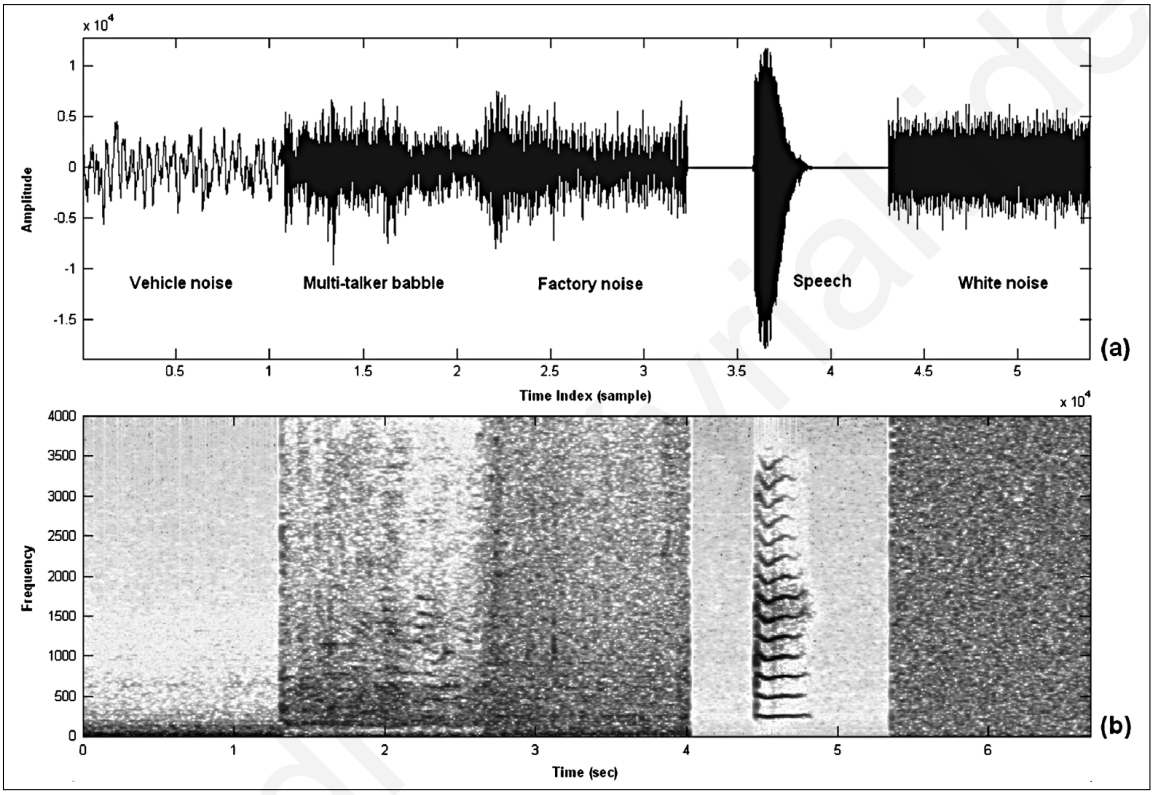

Figure 3.4: The banded structure is a characteristic which only appears in the spectrogram for speech. **(a)** Mixed signal waveform of vehicle noise, multi-talker babble noise, factory noise, speech, and white noise. **(b)** The spectrogram of the corresponding signal obtained by using the short-time Fourier Transform. (Figure taken from [130])

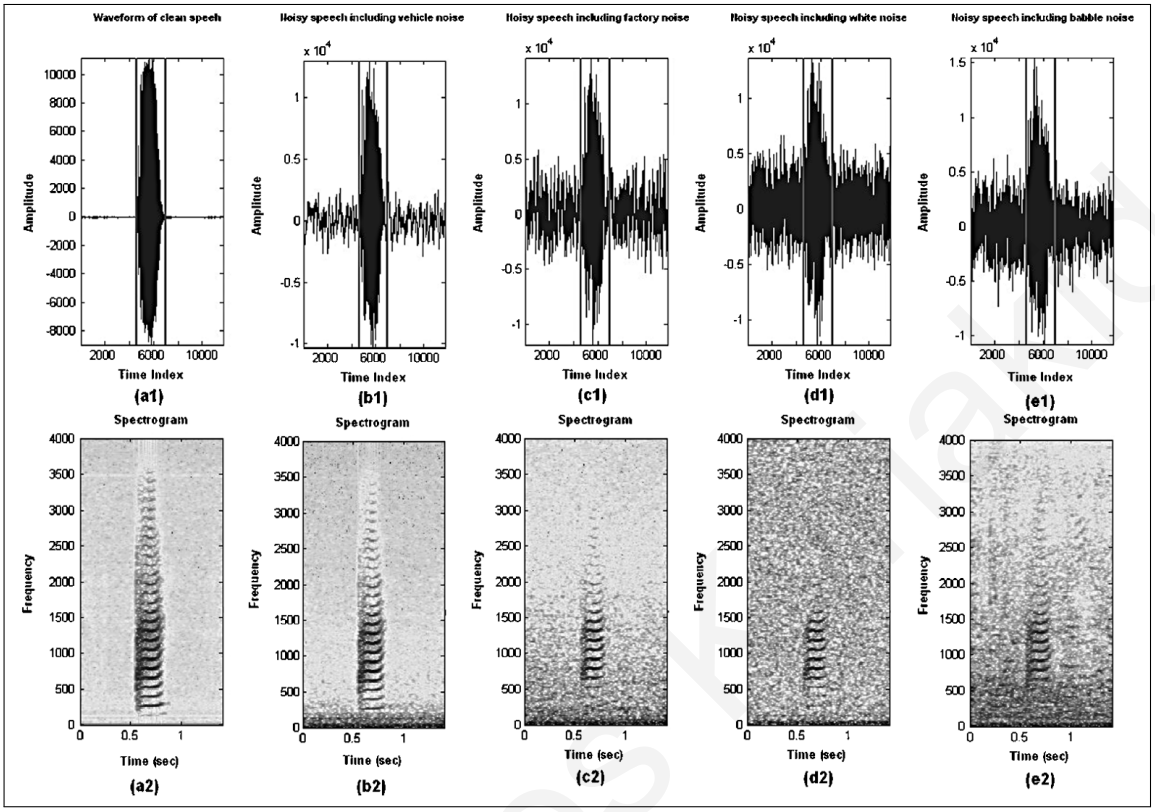

Figure 3.5: The banded structure in the speech spectrogram is robust to noise. The spectrogram on the left shows the banded structure present in clean speech. The rest of the spectrograms show that this banded structure remains (although degraded to some degree) even when various types of noise are added at an SNR of 0dB. **(a1)** Waveform of clean speech. **(b1)** Waveform of speech with vehicle noise. **(c1)** Waveform of speech with factory noise. **(d1)** Waveform of speech with white noise. **(e1)** Waveform of speech with babble noise. **(a2)** Spectrogram of clean speech. **(b2)** Spectrogram of speech with vehicle noise. **(c2)** Spectrogram of speech with factory noise. **(d2)** Spectrogram of speech with white noise. **(e2)** Spectrogram of speech with babble noise. (Figure taken from [130])

model. Subsequently, by using a variance kernel to process the spectrogram, we are able to detect regions in the spectrogram which have high local variance. These high-variance regions correspond to speech. A threshold, which is automatically calculated for each word, separates the high-variance regions from the low-variance regions in order to separate speech from non-speech. Our experiments show that this method is robust to noise even at low SNRs. We call our endpoint detection system the *Variance Kernel method*.

The time-frequency representation of sound is analogous to the representation used by the human auditory system. The cochlea in the human ear acts as a filter bank which separates the incoming sound into different frequency bands [7]. The magnitude of the excitation in each frequency band changes with time, depending on the sound. The brain then processes this input by using both time and frequency information. The spectrogram is a representation which includes both time and frequency information.

3.3.1 Overview of methodology

When a sound signal containing speech is converted to a spectrogram, the regions containing speech show some distinguishing patterns. This was illustrated in Section 3.2.5. Our methodology attempts to capture these two-dimensional patterns. We treat the spectrogram as an image and perform a transformation on the image using a two-dimensional image filter. The image filter calculates the standard deviation of the pixel values of each 5×5 square area of the image. The resulting values, after the filter is applied, are standard deviation values. Nevertheless, we choose to call the resulting image the "variance image.". As explained in a later section, using standard deviation values instead of variance values aids in the calculation of the automatic threshold.

The spectrogram image has high pixel intensity values for time-frequency locations with high energy. The variance image has high pixel values for regions of the spectrogram which have high variance. Figure 3.6 shows an example of these transformations. In the example shown in the figure, the spectrogram was created using a 256-point Short Time Fourier Transform (STFT). The sampling frequency of the input signal shown in Figure 3.6 (a) is 8kHz. For the STFT, a window size of length 30ms was used, with a 90% overlap between windows. The energy (E) was then converted

to decibels to obtain the image pixel values (I) of the spectrogram ($I = 10 \log_{10} E$) in Figure 3.6 (b). This example uses the STFT with relatively short window sizes and large overlap in order to create the spectrogram. This creates a spectrogram which is easier for visualization by a human. This spectrogram however, is not well suited for detecting the regions of speech using a 5×5 standard deviation filter (or variance kernel). This is evident in the variance image shown in Figure 3.6 (c), which shows the standard deviation values after the filter is applied to the spectrogram. The high-variance regions in this variance image do not adequately capture the regions of speech.

In our endpoint detection algorithm, we calculate the spectrogram using a fourth order LPC filter, and we use longer time windows with less overlap. We have found that this type of spectrogram is appropriate for detecting speech regions using a variance kernel.

3.3.2 Description of Algorithm

In this section we describe our endpoint detection algorithm, the Variance Kernel method, in detail. The algorithm consists of the following steps:

1. The input signal is first down-sampled to 16kHz if necessary, and then passed through a high-pass filter.

2. A spectrogram is created using a 4th order LPC analysis filter. Frequencies below 200Hz in the spectrogram are removed. The pixel values are converted to decibels.

3. A variance image is created by applying a 5×5 variance kernel on the spectrogram.

4. If the highest value in the variance image does not exceed a preset global threshold, then the algorithm terminates here with the decision that no speech is present in the signal. Otherwise, the algorithm decides that speech is present in the signal and therefore continues to the next steps in order to set the endpoints.

5. The variance image is converted to a gray-scale image and a threshold is automatically calculated using Otsu's method on the gray-scale image.

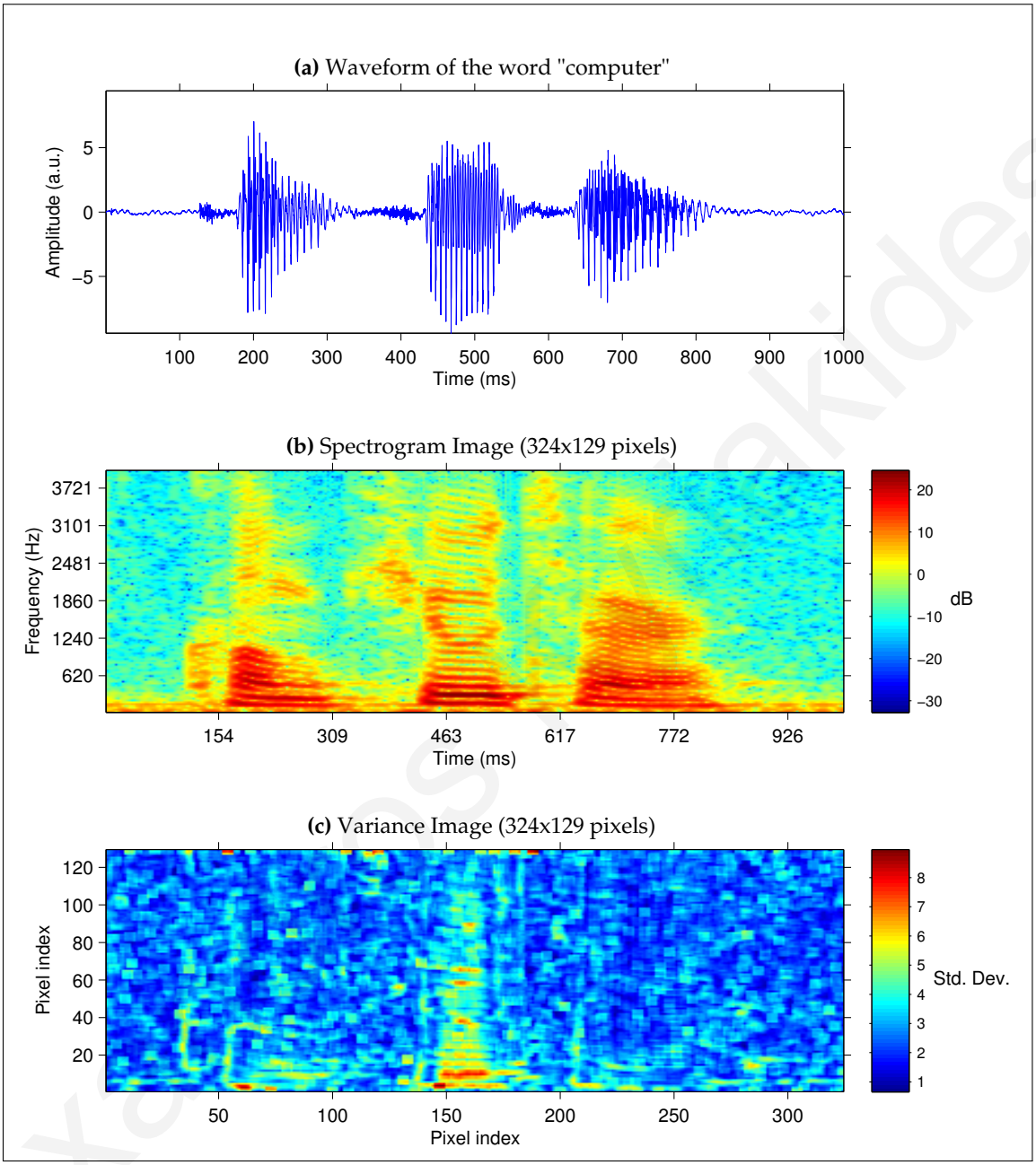

Figure 3.6: Transformation of a sound signal, first to a spectrogram, and then to a variance image. **(a)** The waveform of the word "computer" sampled at 8kHz. **(b)** The spectrogram calculated using the STFT with a window size of 30ms and 90% overlap between windows. **(c)** The variance image, calculated using a 5×5 standard deviation image filter.

6. The gray-scale image is converted to a binary image consisting of black and white pixels, by using the automatic threshold calculated in the previous step.

7. Small isolated components in the binary image of area less than 25 pixels are removed from the binary image. Narrow horizontal lines in the binary image of width less than 10 pixels are also removed from the binary image.

8. The decision is made to place the first endpoint (beginning of word) at the leftmost white pixel in the binary image, and the second endpoint (end of word) at the rightmost white pixel in the binary image.

The flowchart in Figure 3.7 describes the endpoint detection algorithm. The next few sections explain each step in detail.

Sampling Frequency

Most endpoint detection systems use a sampling frequency of 8kHz for the input signal. When performing frequency analysis therefore, only frequencies below the Nyquist frequency of 4kHz can be captured. In our experiments however, we have found that certain phonemes contain energies above 4kHz. For example, the sibilants (such as "s" and "z") have a large proportion of their energy above 4kHz. This is nicely illustrated in Figure 3.8 which shows the spectrogram of the word "six". The black dashed horizontal line marks the 4kHz frequency position. The high-energy region at the center of the waveform is the vowel sound "i". Most of the vowel's energy is concentrated below 4kHz. Conversely, it can be clearly seen that the fricative "s" sounds at the beginning and end of the word "six" have most of their energy above 4kHz. It is therefore useful to have frequency information above 4kHz when performing endpoint detection in order to capture such phonemes. Voiced speech has most of its energy collected in the low frequencies, whereas most energy of unvoiced speech is found in the higher frequencies [86]. For this reason, we have decided to use a sampling frequency of 16kHz for our endpoint detection system so that frequency information up to 8kHz is available. Nevertheless, our endpoint detection algorithm still performs well if the lower sampling rate of 8kHz is used. At this lower sampling frequency however, there is a slight drop in performance.

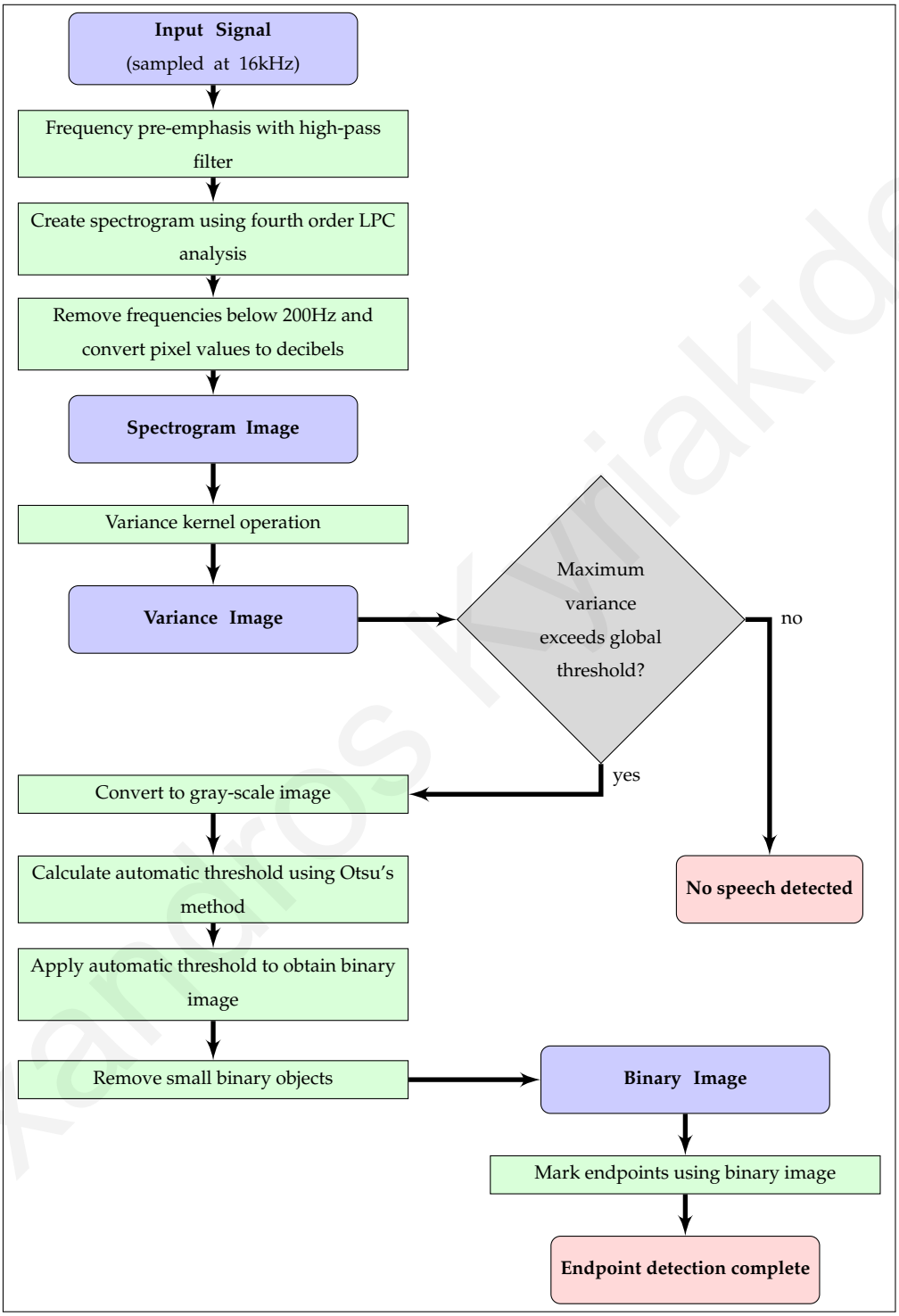

Figure 3.7: A flowchart describing the endpoint detection system.

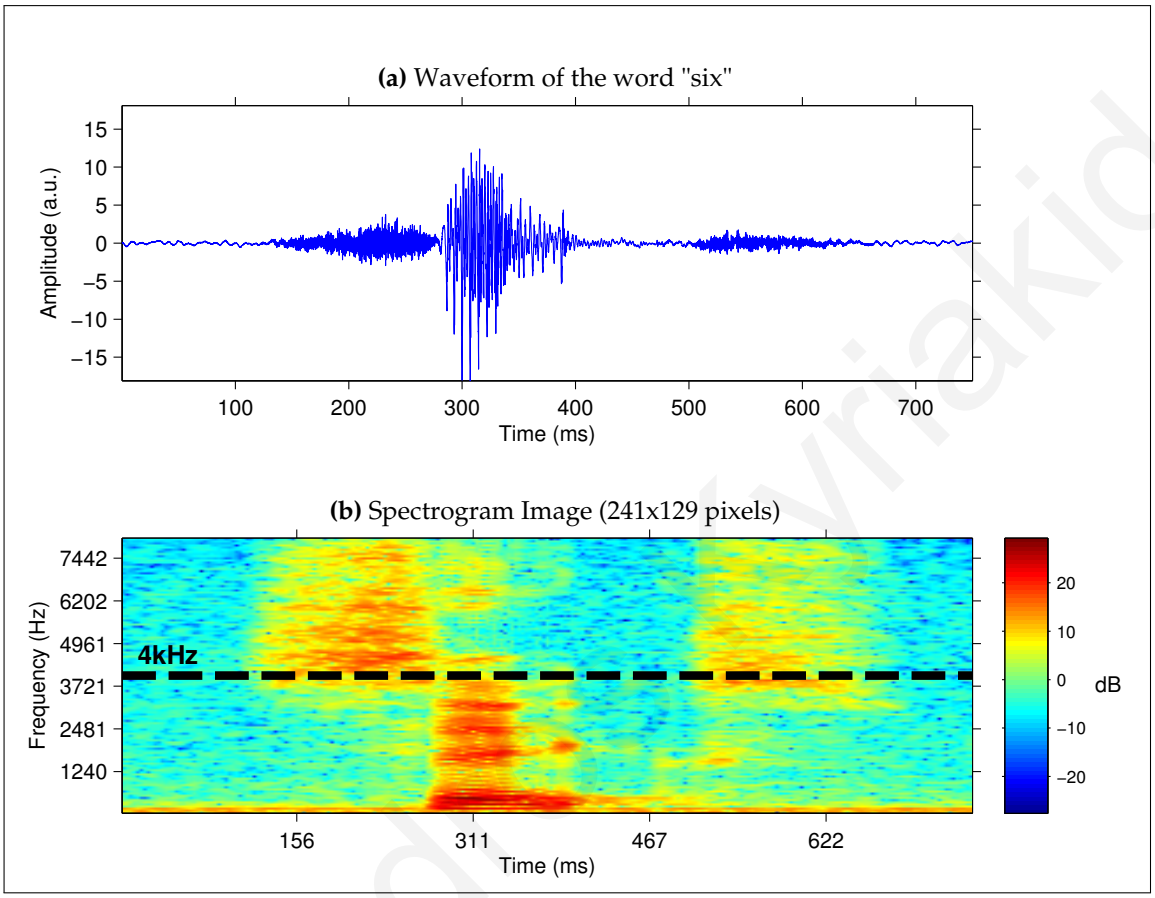

Figure 3.8: The spectrogram of the word "six". It reveals that the consonant sounds of the word have a large proportion of their energy above the frequency of 4kHz. **(a)** The waveform of the word "six" sampled at 16kHz. **(b)** The spectrogram calculated using the STFT with 30ms time window and 90% overlap. The black dashed horizontal line marks the 4kHz frequency. It is evident that there are two regions above the 4kHz line which have high energy, at the beginning and end of the word.

Frequency pre-emphasis

It is common for speech processing systems to use a frequency pre-emphasis filter as an initial step. This high-pass filter restrains the low frequencies of the signal and increases the accuracy of the system. The filter also helps to emphasize the high frequencies present in fricatives [3]. The idea of frequency pre-emphasis is related to the physiology of human hearing. The human ear is insensitive to low frequency signals, but is good at capturing the higher frequency properties of speech [80]. In our endpoint detection system we used a digital filter with transfer function $H(z) = 1 - 0.9375z^{-1}$.

Spectrogram

In order for our method to be robust to noise we used a time-frequency representation which is not commonly used in the literature. The most common way of creating a spectrogram is by using the short-time Fourier Transform. After extensive experimentation however, we have found that for the purpose of endpoint detection, spectrogram estimation using a fourth order Autoregressive (AR) model produces superior results. This AR model utilizes a Linear Predictive Coding (LPC) filter to obtain the parameters which are then used to create the spectrogram. The spectrogram can then be used in conjunction with a variance kernel to greatly improve the noise robustness of the endpoint detector. Although typically a 10th or 14th order AR model is used for the vocal tract, the advantage of using a lower order AR model is that it captures the essential endpoint and energy characteristics of speech. This is consistent with studies in voicing detection in early LPC systems.

Our decision to use a 4th order LPC model was made after experimentation. We tried LPC models of various orders, ranging from 2 to 50, and we obtained the best results with a fourth order model. The use of fourth order LPC models can be found in several publications. They have been used for the cancellation of side-tone oscillations in mobile phones [98] and for fixed-point implementations in speech coding [13]. Spectral moments (mean, variance, skewness, and kurtosis) based either on LPC or DFT spectra have been used with some success to classify acoustic transients, voiceless speech, and voiced speech [78]. In specific applications using the spectral moments approach, good results were obtained with fourth order LPC models [78]. An important algorithm which uses a 4th order LPC model is a pitch

detector known by the name of "Simple Inverse Filtering Tracking" (SIFT) [65, 81]. The authors state that the success of the SIFT method is strongly dependent upon a proper choice of the number of filter coefficients, which is defined by the order of the LPC model. Although the use of fourth order LPC models is not new, our idea to perform isolated word endpoint detection using a spectrogram generated by fourth order LPC analysis is a novel one.

Another distinguishing feature of our algorithm is that it uses relatively long window sizes for creating the spectrogram. We use a window size of 100ms with a 50% overlap, multiplied by a Hamming window. The input signal is therefore segmented into time frames of length 100ms. Each time frame corresponds to one column of the spectrogram image. To calculate the pixel values of each column of the spectrogram image, a 4th-order LPC analysis is performed on each time frame so that a fourth order digital filter is obtained. The amplitudes of the frequency response of this filter at different frequencies constitute the pixel values of the spectrogram. We chose to divide the frequency response into 129 different frequencies, on a linear scale. Consequently, each column of the spectrogram has 129 pixels.

Once the spectrogram is obtained, two operations are carried out on the image. The first operation removes the rows of the spectrogram which correspond to frequencies below 200Hz. This is done to remove any DC, low frequency hum, or very low frequency noise [83]. Such low frequency noise can be seen at the bottom part of the spectrogram images in Figure 3.6 (b) and Figure 3.8 (b). After this first operation, the number of rows in the spectrogram is reduced from 129 to 126. The second operation transforms the image so that the new pixel values are ten times the log magnitude of the original pixel values: $I_{new} = 10 \log_{10} I_{orig}$. This step, from input waveform to spectrogram, can be seen in Figure 3.9. Sub-figure (a) shows the input waveform. Sub-figure (b) shows the spectrogram image obtained after the operations described above.

Variance Image

The variance image is calculated by processing the spectrogram using a 5×5 standard deviation image filter. This image filter takes as input each of the 5×5 square areas in the spectrogram. The output value of the filter is the standard deviation of the 25 pixel values covered by each square area. The location of the center pixel of each

5×5 area in the spectrogram defines the location of the pixel in the variance image which will acquire the output value of each filter operation. All possible 5×5 squares in the spectrogram are processed, with maximum possible overlap. For the pixels on the borders of the spectrogram image, symmetric padding is used. That is, the values of padding pixels are a mirror reflection of the border pixels. In this way, the spectrogram image and the variance image have exactly the same size.

The standard deviation image filter can also be called a kernel because it operates on a fixed-size two-dimensional pixel area. Although the actual operation calculates the standard deviation, we choose to call it a variance kernel, because it sounds more intuitive. It captures local regions of the spectrogram in which the pixel values *vary* a great deal.

The final choice for the size of the kernel was made after experimentation with several kernel sizes. Both square and rectangular kernels were tried, with either dimension ranging from 3 to 31 pixels. Only odd numbers were used for each dimension so that the center pixel of the kernel could be easily defined. In our experiments we found that the 5×5 kernel worked the best. However, we cannot claim that this size is the optimal for all situations. We have not carried out extensive testing as far as the kernel size is concerned. For this specific methodology it was found that smaller size kernels work better than larger ones.

The kernel covers 5 pixels in the horizontal direction, which represents time, and 5 pixels in the vertical direction, which represents frequency. In this specific case therefore, based on the parameters used to calculate the spectrogram, each 5×5 kernel covers a time region of 250ms and a frequency region of 310Hz. The benefits of using of a relatively long time region for separating speech from non-speech has also been reported by others [91].

The variance image shown in Figure 3.9 (c) was obtained after processing the spectrogram as described above. The pixel values displayed are standard deviation values (σ). The regions of the spectrogram which contain speech have higher variance than the non-speech regions.

Decision on the presence of speech

Before calculating the endpoints, our endpoint detection system makes a decision on whether speech is present in the complete input sound signal or not. There are two

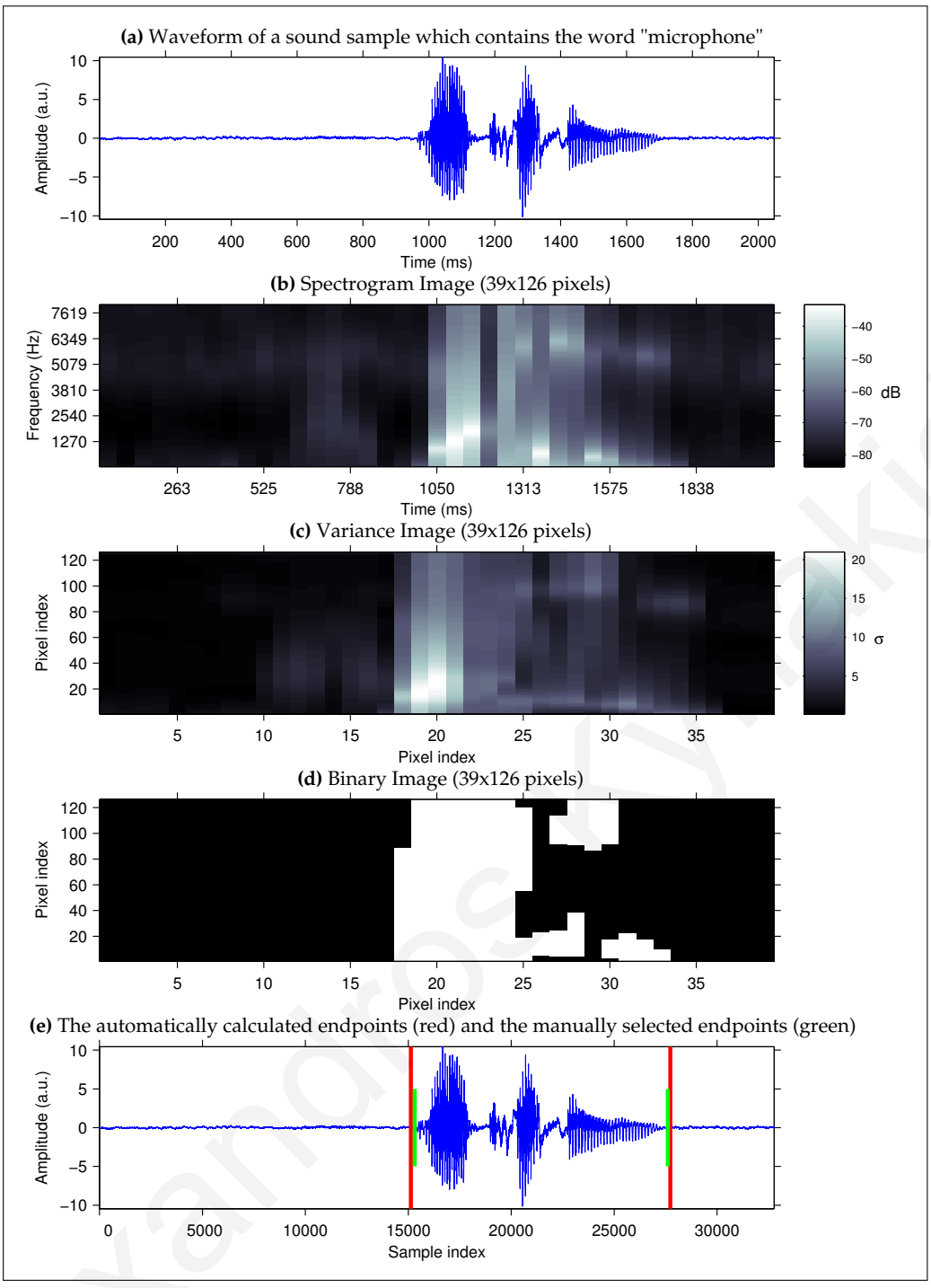

Figure 3.9: The step-by-step procedure of our endpoint detection system. **(a)** The input sound signal sampled at 16kHz which contains the word "microphone". **(b)** The spectrogram image calculated using fourth order LPC analysis. **(c)** The variance image calculated using a 5×5 standard deviation kernel. **(d)** The binary image obtained after automatic thresholding using Otsu's method. The endpoints of the word are the leftmost white pixel and the rightmost white pixel in this image. **(e)** The endpoints of the word calculated using our endpoint detection system are marked with the red lines. The green lines are for reference and they indicate the correct endpoints which were manually selected by a human.

cases where this is useful. The first case is when the input sound does not actually contain any speech. The second case is when the input does contain speech, but at the same time has a high level of noise. High levels of noise can mask the speech making it more difficult to discern from the waveform, spectrogram, and variance image. For each input instance therefore, it is appropriate to make a decision on whether there is speech present or not. As we are dealing with isolated word endpoint detection, an input instance is one complete sound recording. In our experiments for example, each input instance consisted of 2 seconds of sound.

If the algorithm decides that there is no speech present in an input that does actually contain speech, then this is considered to be a "miss". In some cases, when the noise levels are high, it is more desirable to have a "miss" instead of detecting speech and marking it with endpoints which are very far from being correct. That is, in some applications it is sometimes better to have a "miss" than to be "wrong".

The decision on the presence of speech is based on the maximum standard deviation value in the variance image and a pre-set global threshold. If the maximum value in the variance image is above the global threshold, then the decision is made that speech is present in the input signal. If the maximum value in the variance image is equal to or below the global threshold, then the decision is made that no speech is present in the input signal. We call it a *global* threshold because it is pre-set to a specific value and this same value applies to all input signals. This is in contrast to the automatic threshold described in the next section which is automatically calculated separately for each input signal, and therefore it changes from one input instance to the next.

The global threshold can be tweaked in order to balance between the number of wrong endpoint decisions and the number of missed words. A higher global threshold decreases the endpoint detection errors, but increases the miss rate. A lower global threshold decreases the miss rate, but increases the number of endpoint errors. A lower global threshold however, can also result in more correct endpoint detections for some words which would have otherwise been considered a "miss" if the global threshold was set higher. The maximum pixel value in the generated spectrograms typically differs from the minimum pixel value by approximately 50dB. Through experimentation we have found that a value of $\sigma = 10$ is a good global threshold in order to have a satisfactory balance between the miss rate and the error rate. A lower global threshold can be used if the miss rate needs to be

reduced. Figure 3.11 shows an example of a noisy input signal being processed by our endpoint detection system. The signal contains added white noise at an SNR of 0dB. The calculated endpoints shown in sub-figure (e) are considered correct because they are very close to the manually selected endpoints. Looking at the variance image however in sub-figure (c), one can see that the maximum pixel value is $\sigma = 6.11$. If the global threshold was set to $\sigma = 10$, then this instance would have been classified as a "miss" because the decision would have been that no speech is present in this input signal, and therefore no endpoints would have been calculated. In contrast, if a global threshold of $\sigma = 5$ was used, then the endpoints of this word would have been found correctly.

Automatic Threshold

The variance image is converted to a binary image using an automatically calculated threshold. In order to calculate the threshold, the values in the variance image are linearly scaled so that they fall in the range from 0 to 1. This is achieved by finding the minimum and maximum values in the variance image, σ_{min} and σ_{max}, respectively. The value of σ_{min} is subtracted from each pixel of the variance image, and the result is divided by $\sigma_{range} = \sigma_{max} - \sigma_{min}$. The threshold is then calculated on this gray-scale image using Otsu's method [76].

Otsu's method finds a threshold that maximizes the inter-class variance. One class is formed by the values below the threshold and the other class by the values above the threshold. Maximizing the inter-class variance is the same as minimizing the intra-class variance. Intuitively, this method finds a threshold which separates the pixel values into two groups which are as "compact" as possible, and as far apart from each other as possible. We call this the *automatic threshold* to distinguish it from the global threshold which was described in the previous section. The automatic threshold is used to transform the gray-scale image to a binary image. All pixels which fall below the threshold are converted to "black" pixels (value=0), and pixels which fall above the threshold are converted to "white" pixels (value=1). An example of such a binary image is seen in Figure 3.9 (d).

The values in the variance image that we use are standard deviation values. If we had used variance values in the variance image, converted them to gray-scale, and carried out Otsu's method, the result would change. A different binary image

would be obtained because the automatic threshold would change. We have found that using standard deviation values instead of variance values yields better results.

The following equations describe the steps to go from the spectrogram image with symmetric padding ($\tilde{S}$), to the variance image (V) , to the binary image (B). T is the automatic threshold calculated by Otsu's method.

$$V(p,q) = \sqrt{\frac{1}{25}\sum_{i=p-2}^{p+2}\sum_{j=q-2}^{q+2}(\tilde{S}(i,j)-\mu)^2} \tag{3.1}$$

$$\mu = \frac{1}{25}\sum_{i=p-2}^{p+2}\sum_{j=q-2}^{q+2}\tilde{S}(i,j) \tag{3.2}$$

$$B(p,q) = \begin{cases} 1, & \text{if } V(p,q) > T \\ 0, & \text{if } V(p,q) < T \end{cases} \tag{3.3}$$

Removal of binary objects

Some input signals contain sound artifacts which can cause the endpoint detection system to make mistakes. These artifacts manifest themselves in the binary image as binary objects. A binary object is a collection of connected white pixels. The connectivity can be defined based on some sort of neighborhood. In order to remove unwanted binary objects we perform two operations.

The first operation removes horizontal lines which have a height less than 10 pixels. This can be useful for removing any isolated constant-frequency noises, such as a sine wave. A sine wave with a single frequency will show up as a horizontal line in the spectrogram, variance image, and binary image. In the binary image, it will have a height less than 10 pixels, and so it will be removed. The binary object connectivity in this case is defined by the neighborhood in the horizontal direction only.

The second operation removes binary objects using a two-dimensional eight-connected neighborhood. It removes all binary objects which have an area less than 25 pixels.

Both these operations are demonstrated in the example shown in Figure 3.10. The input signal contains some sound artifacts before and after the spoken word "eight". The spectrogram image and variance image are not shown in the figure. The binary image obtained immediately after the automatic thresholding of the variance image

is shown in sub-figure (b). There are three artifacts marked as A1, A2, and A3 in the image. By performing the binary object operations described above, these three artifacts are removed. The two artifacts marked A1 and A2 have a pixel height less than 10 pixels. Therefore they are removed by the first binary object removal operation. The artifact marked A3 has a height greater than 10 pixels, but its total area is less than 25 pixels. It is therefore removed by the second binary object removal operation. Sub-figure (c) shows the binary image after the binary objects have been removed. It is worth noting that the binary image shown previously in Figure 3.9 (d) was obtained after the removal of the binary objects.

Establishing the endpoints

The endpoints of the word are determined directly from the binary image. The column pixel index (c_1) of the leftmost white pixel in the binary image determines the first endpoint, which is the start of the word. The column pixel index (c_2) of the rightmost white pixel in the binary image determines the second endpoint, which is the end of the word. If the digitally sampled waveform of the input signal has a length of N time samples, and the binary image has a width of C pixels, then the sample index of the first endpoint is $n_1 = \lceil N(c_1/C) \rceil$, and the sample index of the second endpoint is $n_2 = \lfloor N(c_2/C) \rfloor$. It is important to note that each pixel in the spectrogram represents 50ms of time in the horizontal direction. This is also true in the binary image. Therefore, the endpoint detection algorithm has an endpoint decision resolution of 50ms. In Figure 3.9 (d), the binary image is shown, and in sub-figure (e) the endpoints calculated as described are marked with red lines. The smaller green lines are for reference and they show the correct endpoints which were manually selected. The automatically calculated endpoints are very close to the manually selected endpoints. In the example in Figure 3.10, it is evident that the binary object removal operations, described before, significantly improved the detection of the endpoints of the word. In sub-figure (d), the dashed vertical red lines indicate where the endpoints would have been placed if the binary objects were not removed. The solid red vertical lines show where the endpoints are placed after the binary objects are removed. They are much closer to the correct manually-selected endpoints which are marked with green vertical lines.

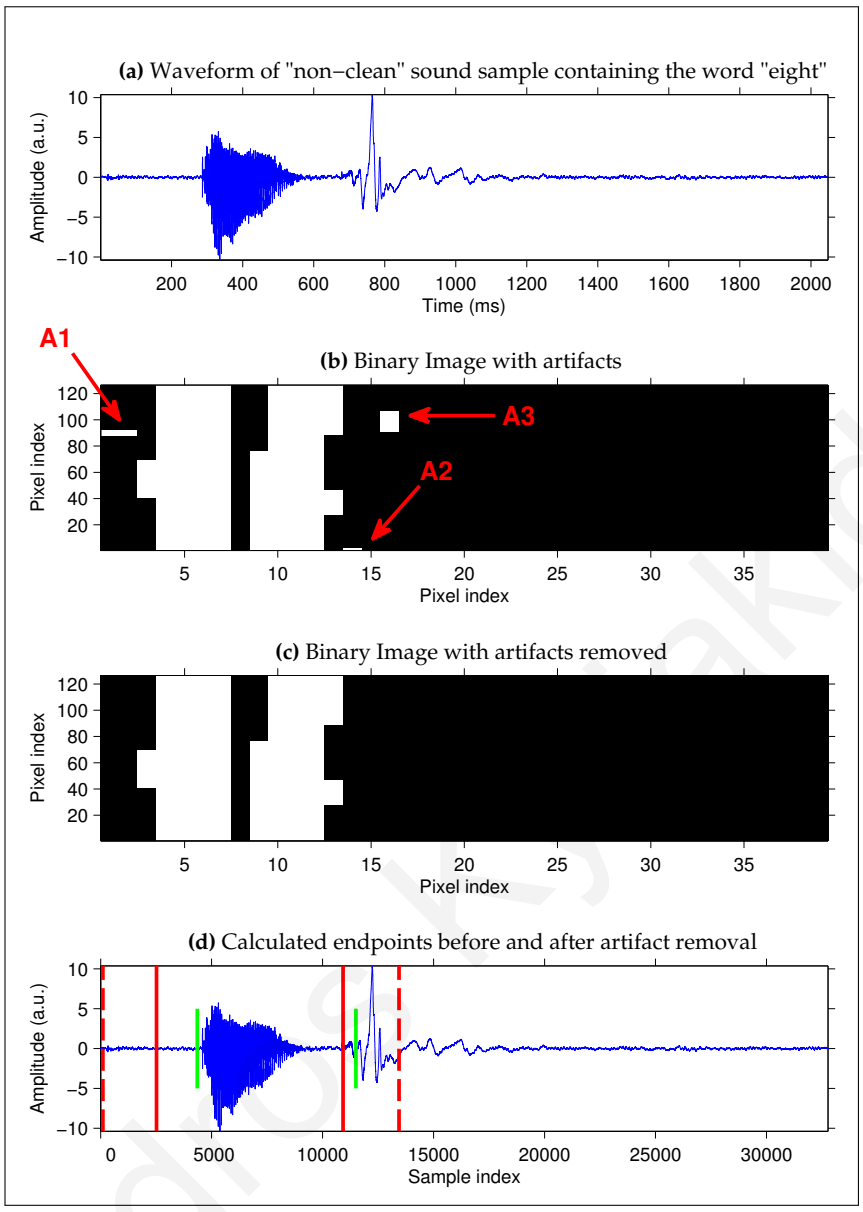

Figure 3.10: An example of how the removal of binary objects improves endpoint detection. **(a)** The waveform of a sound sample which contains the word "eight". This sound sample is non-clean. It contains "artifacts" before and after the word. **(b)** The binary image obtained after automatic thresholding of the spectrogram variance image (not shown). This binary image is before the removal of binary objects. The image shows three artifacts: A1, A2, and A3. **(c)** The binary image after removal of the binary objects. Artifacts A1 and A2 were removed by the rule which removes binary objects which have a pixel height less than 10 pixels. Artifact A3 was removed by the rule which removes isolated binary objects of pixel area less than 25 pixels. **(d)** The waveform showing the endpoints which would have been calculated if binary object removal was not performed (dashed red lines), and the actual endpoints calculated after binary object removal (solid red lines). The green lines show the manually selected endpoints, for reference.

3.3.3 Characteristics of the algorithm

The isolated-word endpoint detection system we have developed can be characterized as an explicit rule-based system. It is explicit because it calculates the endpoints of the word without the need to use a speech recognition system to verify the endpoints. It is rule-based because there is no classification model involved and no training is needed. The system uses a set of rules, based on thresholds, to determine the position of the endpoints. Two thresholds are used. One threshold is the global threshold which is pre-set by the user. It is not difficult to set this threshold to a value which is appropriate for the application at hand. The value of the global threshold can be selected empirically in order to achieve the desired balance between miss rate and error rate. The other threshold is the automatic threshold which adapts itself to each input instance without any user intervention.

Referring to the list of characteristics in Table 3.1, we can compare our endpoint detection system, which we call the Variance Kernel method, to those implemented by others. An important feature of our endpoint detection implementation is that it requires the whole recording to be available in order to make a decision. It does not make a decision on a frame-by-frame basis. The spectrogram is calculated on a frame-by-frame basis, but the automatic threshold operation requires the whole recording. The use of frequency information is a particularly important characteristic of the system. This information is made available by the spectrogram representation. Also, by using a relatively wide kernel, the algorithm captures longer-term information in the sound, which has proved to be advantageous.

There is no explicit calculation of the statistics or nature of the background noise. The algorithm can proceed in the same way irrespective of the type of noise. This allows it to perform well under many types of noise conditions. Also, it does not require an initial period of non-speech "silence" at the beginning of the recording.

The algorithm in its current implementation works in batch mode and not in real-time. It would be possible however, to make adjustments so that it works in real-time.

Computational cost

There are four operations in this endpoint detection system which define the computational complexity of the overall system. They are the calculation of the spec-

trogram, the application of the variance kernel, the calculation of the automatic threshold, and the operation which removes the binary objects. The main computational cost is due to the spectrogram calculation. The spectrogram calculation constitutes about 87% of the total computational cost. The variance kernel transformation takes about 3%, the automatic threshold calculation about 2%, and the binary object removal operation about 8%. The performance of the system, in terms of time, can therefore be improved if the time to perform the spectrogram calculation is reduced. The spectrogram calculation can be performed in a parallel fashion. The above measurements were made using a spectrogram calculation which was performed sequentially. In a parallel implementation, each column of the spectrogram can be independently calculated. A separate parallel process can be instantiated to compute each column. This can lead to a speed increase of a factor approximately equal to the number of columns in the spectrogram. In our examples, the spectrograms have 39 columns. The spectrogram calculation is greatly dependent on LPC analysis. We therefore believe that by using optimized LPC analysis algorithms, an even greater improvement can be made to the speed of the overall system.

Robustness to noise

The most significant advantage of this endpoint detection system is its robustness to noise. The results presented in the following sections support this claim. Additionally, a few illustrative examples can emphasize this point. In Figure 3.11 an example is presented where a sound sample containing the word "microphone" is processed by the endpoint detection system. The recording is corrupted with additive white noise at an SNR of 0dB. The high level of noise can be seen in the waveform in sub-figure (a). The frequency pre-emphasis step, accentuates the high frequencies. Consequently, and because white noise contains energy in all the frequencies, the pixels in the upper part of the spectrogram image have higher values than the ones in the lower part of the image in sub-figure (b). The variance kernel manages to select the region which contains speech because the variance in the non-speech regions is relatively low. Even if the pixels in the upper part of the spectrogram have high energies, the image representation has low variance due to the uniformity of the noise. This allows the variance kernel to disregard the noise. It is important to note however that the highest pixel value in the variance image in sub-figure (c) is

σ = 6.11. If the pre-set global threshold was chosen to be above this value, then for this particular recording, the endpoint algorithm would decide that no speech was present. If the global threshold was set to a value below 6.11 however, the endpoints selected by the system would be very close to the correct ones selected manually by a human. This is seen in sub-figure (e) where the waveform is shown without the added noise so that it is easier to visualize the endpoints. The red lines indicate the endpoints detected by the system. The green lines indicate the correct endpoints. The word is only slightly clipped at the beginning by about 10ms, and at the end by about 40ms.

A word which poses particular difficulty for endpoint detection systems is the word "six". Significant endpoint detection errors are most likely seen for this word because it begins and ends with high frequency, low energy, noise-like sounds [128]. The word "six" starts and ends with the fricative sound "s". This makes it difficult to discern the ends of the word in the presence of noise. Wilpon and Rabiner [128] looked at various examples of the word "six" and found that in general their endpoint detection system did not detect the ending fricative of the word and that only a small portion of the beginning fricative was detected.

There is criticism in the literature that LPC coefficients are not suitable for modeling fricative sounds [132] and that LPC does not work well in adverse environments [130] . Nevertheless, our approach of using LPC analysis to create a spectrogram and subsequent processing by a variance kernel has proved to be a suitable approach for detecting fricative sounds, even under noisy conditions. To demonstrate this, we present examples using two sound recordings, each containing the word "six". White noise is added to one recording at an SNR of 10dB and to the other recording at an SNR of 0dB. Figure 3.12 and Figure 3.13 reveal the effectiveness of the algorithm in these two cases. Sub-figure (e) in both figures shows with red lines the endpoints calculated by the algorithm. The green lines indicate the position of the manually selected endpoints. In both cases, the automatically-detected endpoints are very close to the manually-selected ones, showing that the endpoints of the word "six" were found correctly even when a high level of noise is present. In the case where the SNR is 0dB, the energy level of the noise is so high that if one just looks at the waveform representation of the input signal (sub-figure (a)), only the high-energy vowel sound in the middle of the word can be distinguished from the noise.

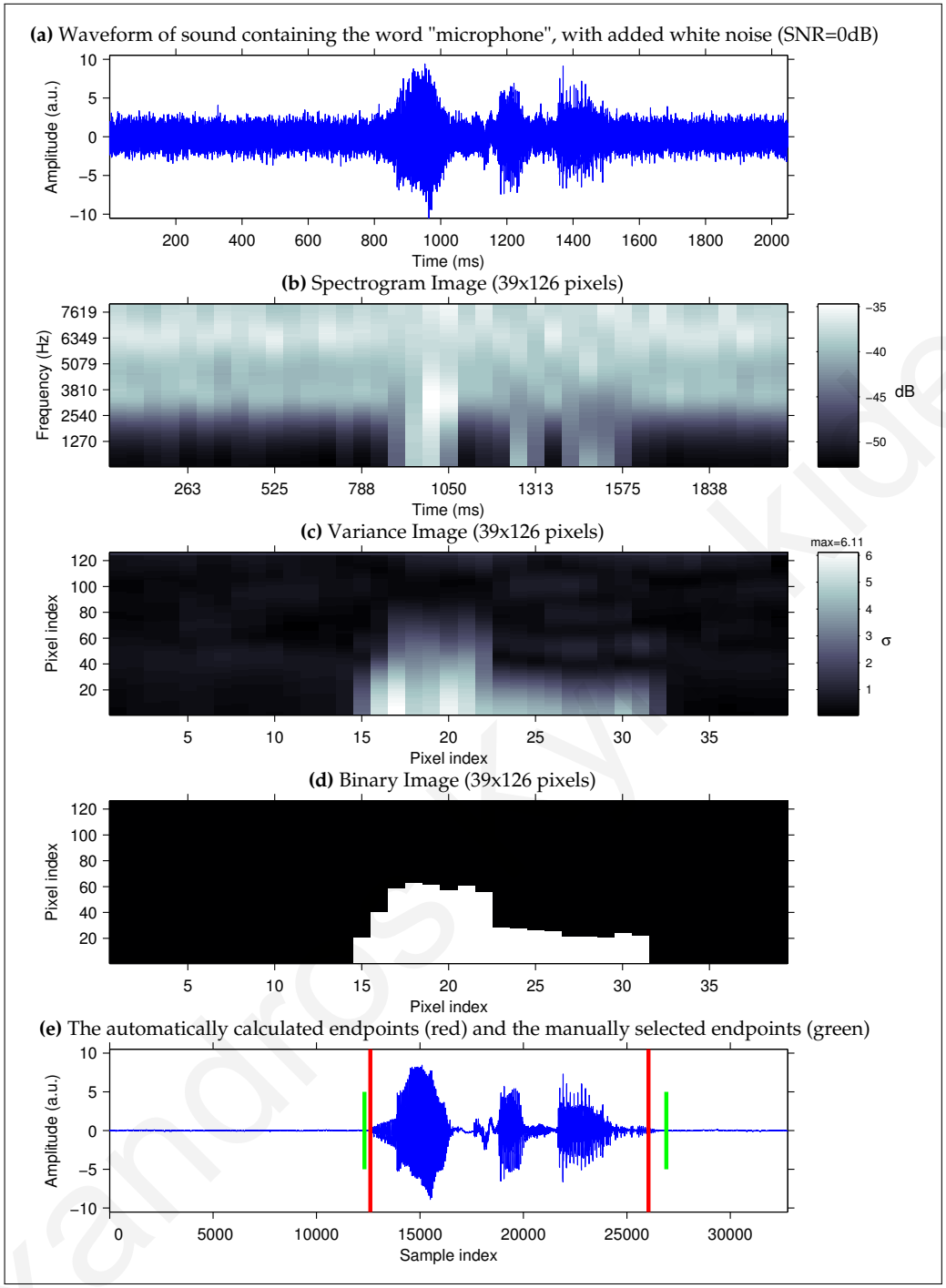

Figure 3.11: The same procedure as the one shown in Figure 3.9, but this time with a different sound sample. In this example, the sound is corrupted with added white noise at an SNR of 0dB. The spectrogram in sub-figure **(b)** has more energy in the high frequencies because of the high-pass filter which was applied before the spectrogram was generated. The variance image in sub-figure **(c)** shows that the highest pixel value in the image is $\sigma = 6.11$. Consequently, if the global threshold was set above this value, then the word in this sound sample would not be detected. In sub-figure **(e)** the manually selected endpoints are shown in green for reference.

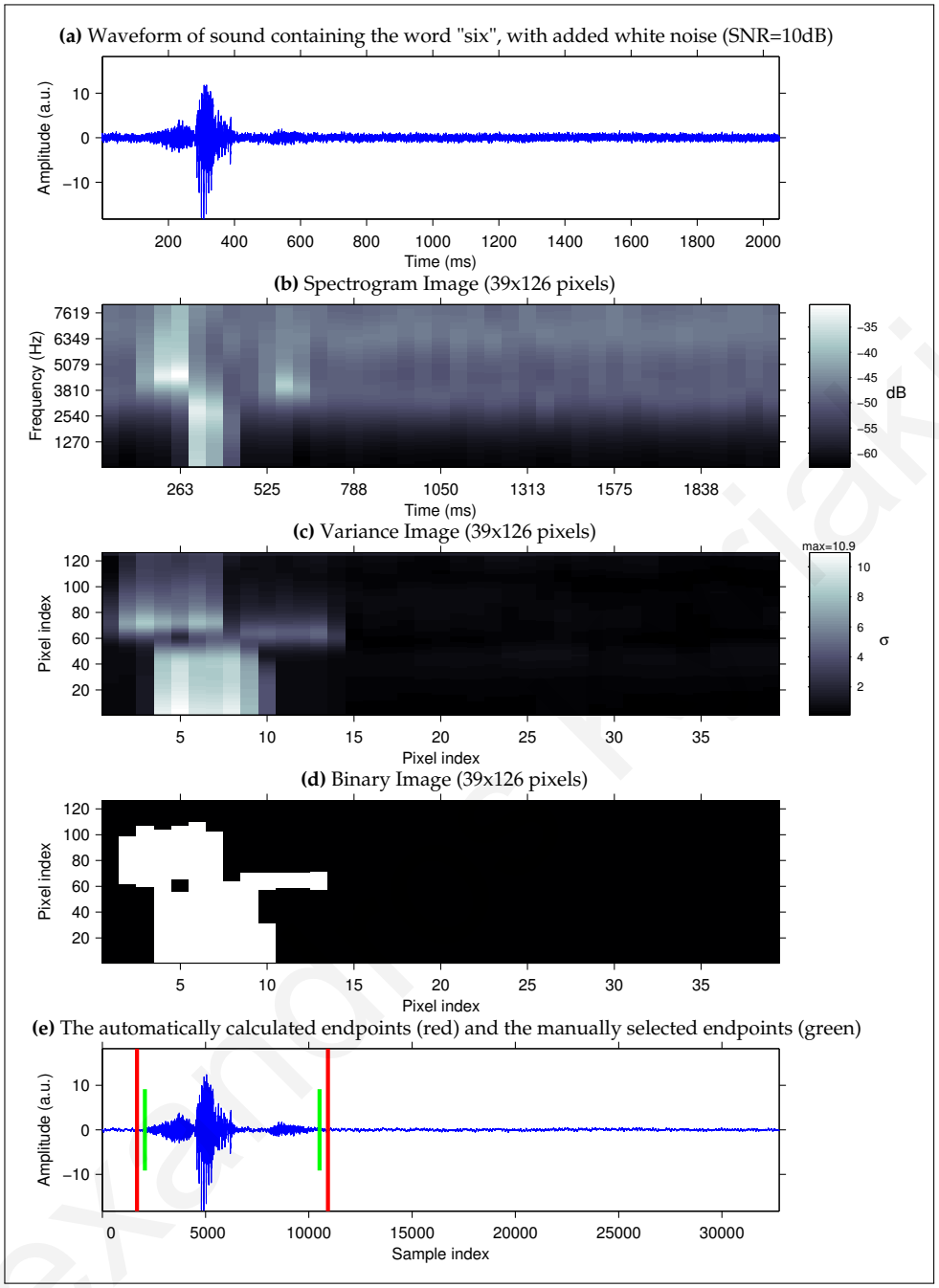

Figure 3.12: A sound sample containing the spoken word “six” corrupted with white noise with an SNR of 10dB. The endpoint detection system adequately captures the fricative sounds at the beginning and end of the word. As a result it detects the endpoints of the word correctly.

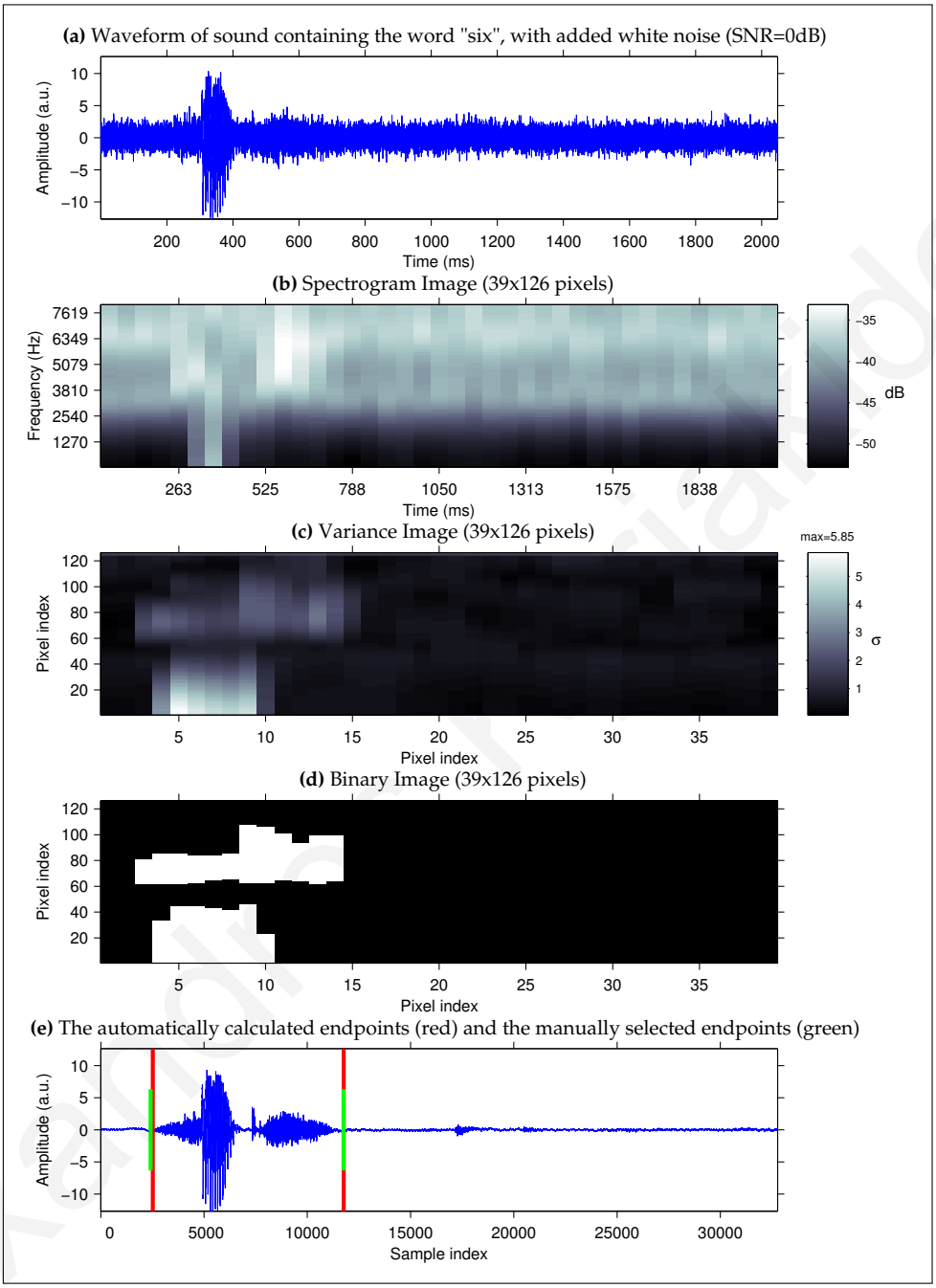

Figure 3.13: A sound sample containing the spoken word “six” corrupted with white noise with an SNR of 0dB. The endpoint detection system adequately captures the fricative sounds at the beginning and end of the word. As a result it detects the endpoints of the word correctly.

Robustness to amplitude scaling

In real-world applications, different input signals can have different intensities. Some endpoint detection systems which use energy-based thresholds are sensitive to changes in the overall intensity of the input signal. If an input signal has a higher overall energy, then more of the signal is classified as "speech". This is because more of the signal will be above the energy threshold. This can lead to false detections. A greater number of non-speech frames will be wrongly classified as speech frames.

When testing endpoint detection systems, one can simply scale the amplitude of the input signal by a certain factor in order to see how the system behaves in such situations. With some endpoint detection systems, scaling the amplitude *up* by a certain factor causes most of the input signal to be classified as speech. Correspondingly, scaling the amplitude *down* by a certain factor causes most of the input signal to be classified as non-speech.

In batch systems, which process one input signal at a time, the problem of having input signals with inconsistent intensities can be solved by normalization. The energy of the input signal or the maximum amplitude of the signal can be normalized to a certain value before the signal is presented to the endpoint detection system. It is still an issue however, to select the normalization parameters which are most appropriate for the endpoint detection system at hand.

One advantage of the endpoint detection system we have presented, is that it is unaffected by amplitude scaling. When the amplitude of the input signal is scaled up, the pixel values of the spectrogram image increase because more energy is present in the signal. When the amplitude is scaled down, the values of the spectrogram decrease. The variance image however, remains the same in both cases.

In a sound signal, if the amplitude is scaled by a factor of C then the energy is multiplied by a factor of C^2. Spectral energy is also multiplied by C^2. After LPC analysis therefore, the values obtained for the two-dimensional time-frequency representation are multiplied by C^2 because the pixel values represent spectral energy. These pixel intensity values (I) are then converted to decibels by taking the log to the base 10 and multiplying by 10. The log transformation means that a multiplicative change becomes an additive change. So the multiplication of the energy by C^2 means that a constant value is *added* to each pixel value in the final spectrogram: $10\log_{10}(C^2 I) = 10\log_{10}(C^2) + 10\log_{10}(I)$. The variance of a set of values does not

change if the same constant is *added* to all the values. And so the variance of each pixel area in the final spectrogram does not change when the amplitude of the input signal is multiplied by a constant factor.

In the figures we have presented until now, which show the endpoint detection system in action, we have arbitrarily chosen to normalize the input signal so that its RMS value is 1. One such figure is Figure 3.9. If the amplitude of the same input signal in that figure is scaled up or down by a certain factor, the result of the endpoint detection decision will not change. This can be seen in Figure 3.14 where the amplitude is scaled up by a factor of 100, and in Figure 3.15 where the amplitude is scaled down by a factor of 100.

3.4 Evaluation of Endpoint Detection

An endpoint detection system can be evaluated by running tests, using several inputs, in order to measure performance. There are two reasonable ways of evaluating the performance of an endpoint detector [128].

3.4.1 Evaluation by using a speech recognition system

The first way to evaluate an endpoint detection system is by initially performing endpoint detection and subsequently measuring the recognition accuracy of a speech recognition system. The detected endpoints are used to decide which part of the sound signal is passed as input to the speech recognition system. The assumption is that the speech recognition accuracy will be higher when the endpoint detection accuracy is higher. The accuracy of the speech recognition system can therefore be used as an indication of the performance of the endpoint detection system.

It can be argued that the reason for performing endpoint detection is so that it can ultimately improve speech recognition accuracy. With this reasoning, the evaluation of an endpoint detection system using speech recognition accuracy is more relevant. It should be noted however, that in some cases, even gross errors in endpoint locations do not necessarily lead to speech recognition errors [128]. Some speech recognition systems can still perform well even when the endpoints are not correct. Even if a small segment of the word is passed to such a speech recognizer, it can still recognize the word correctly. Furthermore, it could be the case

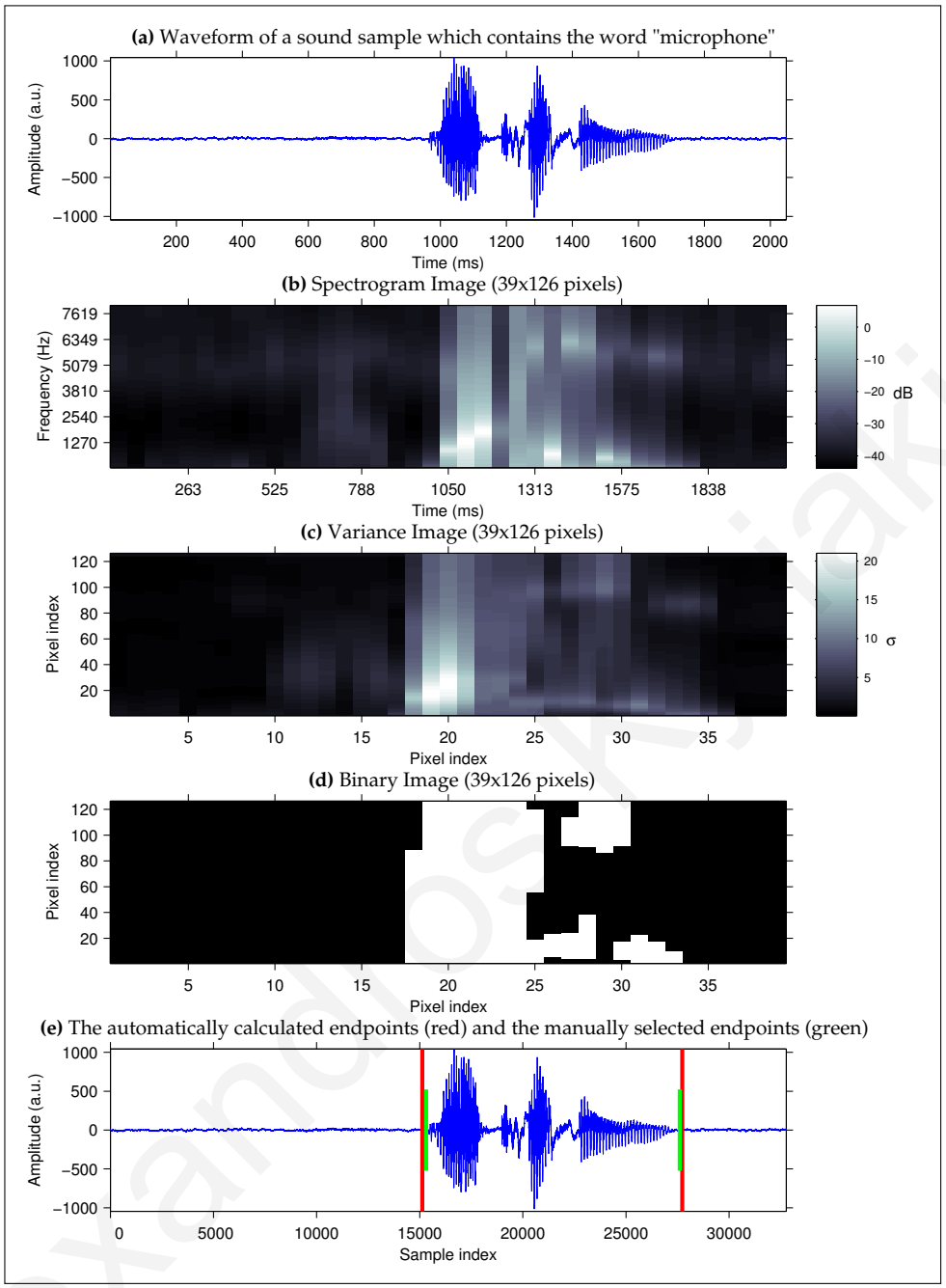

Figure 3.14: The endpoint detection system is robust to amplitude scaling. In this example, the input signal is the same as the one in Figure 3.9, but with the amplitude *scaled up* by a factor of 100. The values in the spectrogram image have increased, but the values in the variance image have remained the same. Therefore, the endpoint detection decision does not change.

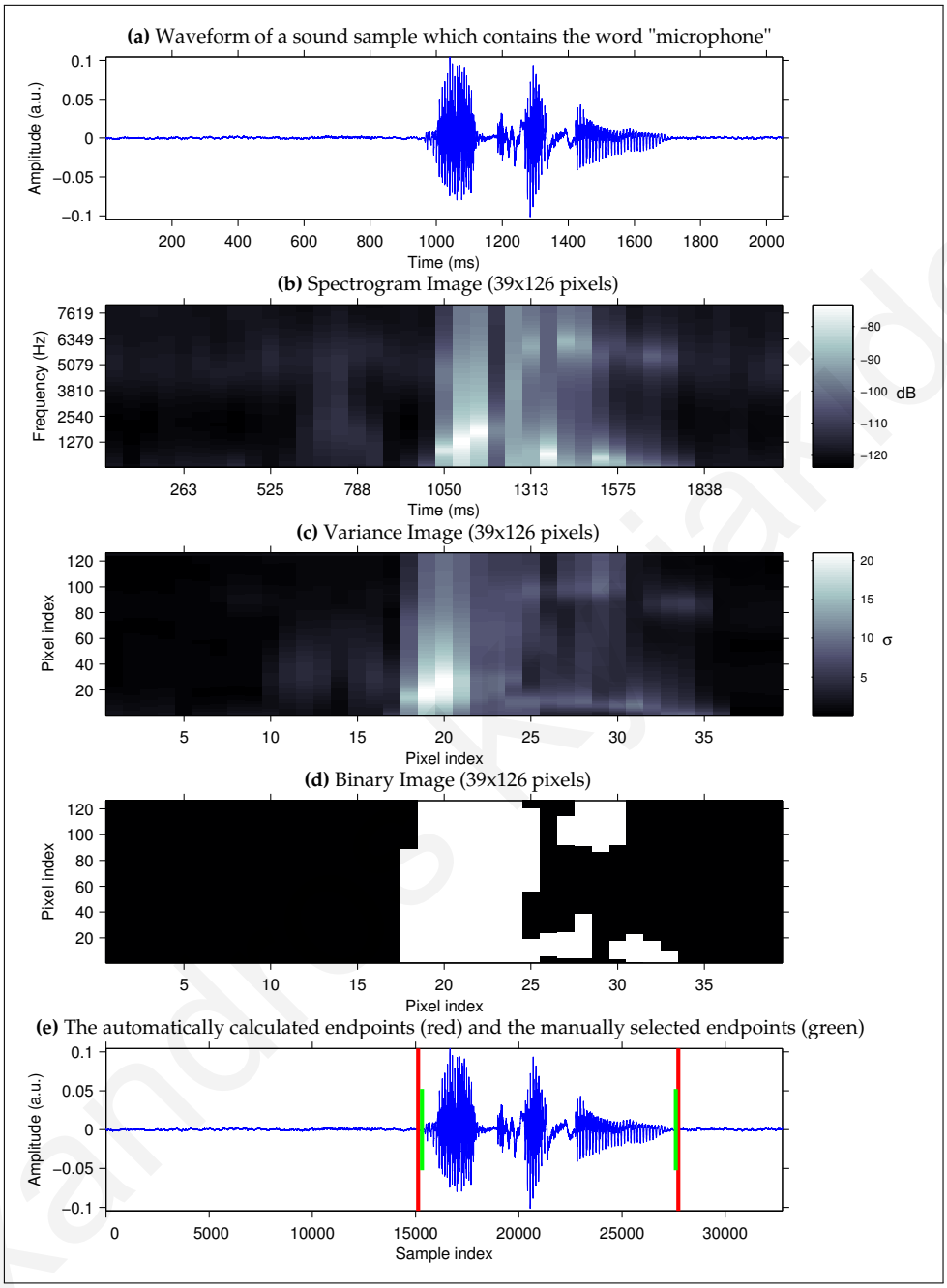

Figure 3.15: The endpoint detection system is robust to amplitude scaling. In this example, the input signal is the same as the one in Figure 3.9, but with the amplitude *scaled down* by a factor of 100. The values in the spectrogram image have decreased, but the values in the variance image have remained the same. Therefore, the endpoint detection decision does not change.

that the speech recognition system was trained on just a small segment of the word instead of the whole word. This could happen if the same endpoint detection system was used in the training phase. In this case, the speech recognition system would perform better if just a small segment of the word is presented to it instead of the whole word, because it was trained on just that small segment. By using a speech recognition system to evaluate an endpoint detection system one cannot guarantee that the endpoint detection system is being evaluated in an independent way and one cannot claim that the endpoints found by the endpoint detection system are the ones that would be accepted by a human as being the true endpoints of the word. For this reason, we did not use this way to evaluate our endpoint detection system.

3.4.2 Evaluation by comparing to pre-selected endpoints

The second way of evaluation requires that the "correct" endpoints of the words in the input data are already known. The endpoints calculated by the endpoint detection system can then be compared to these correct endpoints. The correct endpoints can be manually determined by a human. This is the approach we have taken in our evaluation tests.

Manually marking the endpoints of words is a cumbersome procedure which takes a long time. For this reason, instead of using a human to manually mark the endpoints, some researchers choose to use some standard endpoint detection algorithm to automatically mark the endpoints which they then consider as being "correct". For example, in one particular publication [66], the standard G.729 algorithm was used for automatic marking instead of manually marking the words. The authors made the assumption that the output of the G.729 is "correct" and they evaluated their own speech pause detection system based on this assumption. As our experiments show, this assumption is a good one when there is no noise present in the input. In the absence of noise, the G.729 algorithm has a high speech pause hit rate (non-speech hit rate) and a low error rate. The G.729 algorithm is discussed in more detail in Section 3.5.2. It is one of the algorithms with which we compared the performance of our own endpoint detection system.

Evaluation of Endpoint Detection Systems

Manual endpoints might not always result in the best performance of a speech recognition system [127]. Even so, comparing to manual endpoints provides a good objective criterion which is independent of the speech recognition system, and is therefore good for comparison [3]. When comparing the automatically calculated endpoints to the manual endpoints, some measures need to be defined. When dealing with endpoint detection systems, the following are useful:

Correct rate The percentage of words for which the endpoints were found correctly. The endpoints are considered correct if they fall within a certain pre-defined time distance from the manual endpoints.

Wrong rate The percentage of words for which the detected endpoints were wrong. The endpoints are considered to be wrong if they fall outside the pre-defined time distance from the manual endpoints.

Miss rate The percentage of words for which no endpoints were detected. A spoken word is present in the input sound, but the endpoint detection system does not detect any speech.

Mean and Standard Deviation of differences: The average difference (μ) and standard deviation (σ) in milliseconds between the automatic endpoints and the manual endpoints [3].

Distribution of differences: Further to calculating μ and σ, histogram plots of the differences in endpoint locations can be used to visualize the distribution of the differences [128].

The above measures are used for endpoint detection systems. They make direct comparisons between the automatic endpoints and the manual endpoints in order to find the time differences between the two. For our evaluation tests we decided to use the first three measures: Correct rate, Wrong rate, and Miss rate. These measures use a pre-defined time distance which determines if the endpoints are correct or not. Gu et al. [38] for example, decided to use a time distance of 75ms for the beginning endpoint and 100ms for the ending endpoint. So if the automatic beginning endpoint was less than 75ms from the manual beginning endpoint, and the automatic ending endpoint was less than 100ms from the manual ending endpoint, then the endpoint

detection for that word was considered to be correct. Otherwise, it was considered to be wrong. In our evaluation, we decided to use a different definition for correctness. When the calculated endpoints are different than the manual endpoints, the error can be in one of two directions. The error can either be in the direction towards the word, in which case some of the word is "clipped", or the error can be in the direction away from the word, in which case some extra non-speech "noise" is added to the word. We believe that it is more important that the calculated endpoints do not clip the word, and that it is less important if some extra non-speech noise is added to the ends of the word. For this reason, we chose to use the following criterion to define correctness:

> A word is considered to have been correctly endpointed if each of the automatically calculated endpoints differs from the corresponding manually selected endpoint by no more than 50ms in the direction towards the word, and by no more than 150ms in the direction away from the word. Equivalently, the endpoints of a word are considered to be wrong if either endpoint clips more than 50ms of the word or if either endpoint adds more than 150ms of non-speech noise.

Evaluation of Voice Activity Detection systems

Voice Activity Detection (VAD) systems do not explicitly define endpoints. Instead, the input sound signal is divided into short time frames and a classification to "speech" or "non-speech" is made for each frame. When evaluating VADs it is therefore more convenient to use measures based on short time pulses, instead of comparing word endpoints. It is a binary classification exercise. Each short time pulse belongs either to the class "speech" or "non-speech". The following measures are useful when evaluating VADs:

Rejection rate: The percentage of speech pulses that are not detected by the endpointer [3]. A high rejection rate in the VAD could cause a speech recognizer to reject or delete a word.

False Alarm Rate of speech (FAR0): The percentage of speech pulses that are wrongly classified as non-speech [91]. This is the same as the Rejection rate above.

False Alarm Rate of non-speech (FAR1): The percentage of non-speech pulses that are wrongly classified as speech.

Non-Speech Hit Rate (HR0): The percentage of non-speech pulses that are correctly classified as speech [91]. This can also be called the speech pause hit rate [66].

Speech Hit Rate (HR1): The percentage of speech pulses that are correctly classified as speech [91].

Error Rate (ER): The percentage of all speech pulses that are wrongly classified.

The measures above should be used in combination so as to give a complete and accurate picture of the performance of a VAD. Stating only one measure would give a partial and inaccurate view. For example, a trivial implementation of a VAD which classifies all pulses as speech will have a speech hit rate of HR1=100%. So it would be important to also state the non-speech hit rate, which in this case would be HR0=0%. It should be noted that HR0 and HR1 provide the same information as FAR0 and FAR1, because FAR0=100%-HR1, and FAR1=100%-HR0. In order to have a complete picture of the performance of the VAD, it would be necessary to also include the Error Rate (ER) in the performance measurements. Some publications [66, 91, 93] which state only the speech hit rate (HR1) and the non-speech hit rate (HR0), without stating the error rate (ER), may be misleading.

The following pair of examples demonstrates this point. The examples show that even if HR0 and HR1 stay the same, the ER can change.

Example1: The total number of sound pulses is 100. The total number of speech pulses is 25, and 5 of these are correctly classified as speech. Therefore HR1=5/25=**20%**. The total number of non-speech pulses is 75, and 45 of these are correctly classified as non-speech. Therefore HR0=45/75=**60%**. A total of 20+30=50 pulses were classified incorrectly. The error rate is therefore ER=50/100=**50%**.

Example2: The total number of sound pulses is 100. The total number of speech pulses is 50, and 10 of these are correctly classified as speech. Therefore HR1=10/50=**20%**. The total number of non-speech pulses is 50, and 30 of these are correctly classified as non-speech. Therefore HR0=30/50=**60%**. In this case, a total of 40+20=60 pulses were classified incorrectly. The error rate is therefore ER=60/100=**60%**.

So the interpretation of HR0 and HR1 is highly dependent on the ratio of speech to non-speech pulses in the test set data. This ratio is not the same for all test sets. In our test set for example, 33% of pulses are speech pulses and 67% are non-speech pulses. If this ratio is not specified in the results together with HR0 and HR1, then the results could be misleading. For this reason, the error rate should always be stated in the results in order to provide a complete basis for comparing different VADs. In our results we state the non-speech hit rate (HR0), speech hit rate (HR1), and the error rate (ER).

3.5 Experiments

In order to test the performance of our endpoint detection system, which we call the Variance Kernel method, we have run several experiments. In the experiments we used speech data from our own speech corpus (see Chapter 2). In addition to our own endpoint detection system, we have also tested two publicly-available algorithms for comparison purposes. We used the VAD algorithm defined in the ITU G.729 standard [8] and the endpoint detector of the Sphinx-4 system [124]. In order to demonstrate the robustness of our system under noisy conditions, we digitally added 20 types of noise to the sound recordings with signal-to-noise ratios (SNRs) ranging from 20dB to -5dB.

3.5.1 Data

Our speech database consists of voice recordings from 15 male and 15 female speakers. For our endpoint detection experiments we used a test set with recordings of the following 15 words: “six”, “computer”, “telephone”, “one”, “eight”, “forward”, “microphone”, “power”, “OK”, “switch”, “pause”, “keyboard”, “silent”, “push”, and “north”. These words were chosen specifically in order to include some cases which are especially difficult for endpoint detection systems. Each word was spoken once by each speaker during a sound recording of 2 seconds. Our test set therefore contains a total of 450 recordings.

For evaluation purposes, each recording instance was manually labeled by a human expert. Using acoustic input and a graphical presentation of the waveform of the signal, the expert marked the two endpoints of the word in each recording.

Additionally, the expert also decided if each recording instance was either "clean" or "non-clean". The "clean" instances were the ones which did not contain any sound artifacts before the start or after the end of the spoken word. The "non-clean" instances were the ones which did contain sound artifacts before and/or after the spoken word. Such sound artifacts included tongue clicks, air puffs, and breathing noises. From the 450 recordings, it was determined that 185 of them were "clean" and 265 of them were "non-clean". The fact that almost 59% of the recordings were "non-clean" shows that, in real-life applications, it is important to consider the behavior of systems using recordings which contain artifacts, because such recordings are highly likely to occur. It is valuable to note that from the combined total of 900 seconds of recordings in our test set, approximately 295 seconds (33%) were speech, and 605 seconds (67%) were not speech.

Fifteen noise types were obtained from the NOISEX-92 [108, 121] database. In addition, five publicly-available noise types were used in our experiments. The following 20 types of noise were used:

1. Air conditioner
2. Speech babble (100 people speaking in a canteen)
3. Buccaneer jet cockpit at 190 knots
4. Buccaneer jet cockpit at 450 knots
5. Conference Room
6. Intergalactic Cruiser
7. Destroyer Engine Room
8. Destroyer Operations Room
9. F16 cockpit
10. Factory floor (1)
11. Factory floor (2)
12. HF radio channel
13. Jet airliner cabin
14. Leopard military vehicle
15. M109 military tank
16. Machine Gun
17. Vehicle interior (Volvo car at 120 km/h)
18. Street traffic
19. Pink
20. White

3.5.2 The G.729 algorithm

The G.729 algorithm is a speech coding algorithm standardized by the International Telecommunication Union (ITU) in 1996. It is described in Recommendation G.729

of the Telecommunication Standardization Sector (ITU-T). Annex B of this Recommendation defines a voice activity detector (VAD) and a comfort noise generator for use with G.729 [8]. The VAD was developed for fixed telephony and multimedia communications in order to reduce the transmission rate during silence periods of speech. It makes a voice activity decision every 10ms. A set of difference parameters is used for making a decision. The parameters are based on the full-band energy, the low-band energy (0-1kHz), the zero-crossing rate, and spectral distortion. The algorithm is adaptive to background noise. It keeps track of running averages of the background noise characteristics. The output of the VAD is either 0 or 1, indicating the presence or absence of voice activity, respectively.

It is common for researchers to compare their own algorithms with the G.729 VAD [66, 91]. Comparing a novel algorithm with this standard algorithm makes it also comparable with other algorithms. It is important to note however, that the G.729 algorithm was designed to be used in environments with low levels of noise.

The G.729 source code is freely available for download from ITU's web site [2]. The G.729 coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering. It also requires the input to be sampled with an 8kHz sampling rate, and coded with 16-bit linear PCM. For the filtering we used the G.712 algorithm[3], as indicated in the specification document of G.729. We found that this filtering improved the results by reducing the non-speech detection errors.

3.5.3 The Sphinx-4 speech recognizer

Sphinx-4 is a state-of-the-art speech recognition system [124]. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT). Sphinx-4 includes the `SpeechClassifier` java class which implements a simple VAD algorithm designed

[2] We obtained the source code from `http://www.itu.int/rec/T-REC-G.729-200701-I/en`. We compiled the code found in the `Soft/g729AnnexB/c_codeB/` directory and used the resulting `coder` binary executable.

[3] For the G.712 implementation we used the Filtering and Noise Adding Tool (FaNT) available for download at `http://dnt.kr.hs-niederrhein.de/download.html`.

by Bent K. Schmidt-Nielsen[4]. The algorithm classifies each audio frame as speech or not. For each fame, the average signal level and the background noise level are updated. If the average signal level is greater than the background noise level by a certain threshold value, then the current audio is marked as speech. Otherwise, it is marked as non-speech. The threshold value is configurable.

The wrapper java class `SpeechMarker` inserts markers into the sound input stream. The markers are identified as `SpeechStartSignal` and `SpeechEndSignal`. These two markers indicate the start and end of a period of speech. In a recording containing an isolated word, these two markers are therefore the endpoints of the word.

3.5.4 Experimental Procedure

The goal of our experiments was to test the endpoint detection performance of the Variance Kernel method and to compare it to the performance of G.729 and Sphinx-4 on isolated words. The performance measures are generated by comparing each algorithm's automatically calculated endpoints to the manually selected endpoints which were marked by a human.

Endpoint detection vs. Voice Activity Detection

As already mentioned in Section 3.4.2, there is a difference in the way performance can be measured depending on whether a system is an endpoint detection system or a voice activity detection system (VAD). In an endpoint detection system, the output is a set of endpoints which mark the start and end of a word. In a VAD, the output is a list of labels, one label for each short time frame (typically 10ms), indicating whether each frame is "speech" or "non-speech". Our Variance Kernel method and the Sphinx-4 implementation we used, both output a set of endpoints. The G.729 algorithm is a VAD, and therefore it outputs a list of voice activity labels. We decided to evaluate all three systems both as endpoint detection systems and as voice activity detection systems. For this reason, we needed to translate the endpoint detection outputs to voice activity detection outputs, and vice versa.

Translating endpoint detection outputs to voice activity detection outputs is straightforward. Considering that we are dealing with isolated words, we have

[4]Bent K. Schmidt-Nielsen works at MERL Research (`http://www.merl.com/people/?user=bent`)

two endpoints: the beginning and the end of the word. To translate the endpoints into voice activity detection outputs we need to label all the short time frames in each recording. We do this by labeling all time frames before the beginning endpoint of the word as non-speech, and we also label all the time frames after the ending endpoint of the word as non-speech. The time frames between the two endpoints are the only ones labeled as speech.

To translate VAD output to endpoints is a more involved task. The straightforward way would be to mark the first time frame which is labeled as speech to be the first endpoint, and the last time frame which is labeled as speech to be the second endpoint of the word. In some cases this can lead to significant errors however. For example, an individual time frame of 10ms might be labeled as speech, while all time frames before and after it are all labeled as non-speech. There is no word which has a duration of 10ms or less. Also, the opposite can occur. There could be isolated time frames which are labeled as non-speech while they are surrounded by speech frames. A rule needs to be in place to decide what to do in such cases.

The endpoint detection system in Sphinx-4 is actually based on an underlying VAD with a set of rules. This set of rules translates the VAD decisions into endpoints. In the default configuration, Sphinx-4 makes a VAD decision every 10ms, just like the G.729 algorithm. The Sphinx-4 rules are based on four configurable parameters. These parameters are the following:

Start Speech Time: This defines the minimum amount of time in speech to be considered as utterance start. Therefore, in order to mark the start of the word, a continuous sequence of time frames labeled as speech of time length at least equal to this parameter is needed.

End Silence Time: This defines the amount of time in silence (non-speech) to be considered as utterance end. Therefore, in order to mark the end of the word, a continuous sequence of time frames labeled as non-speech of time length at least equal to this parameter is needed.

Speech Leader: This defines the amount of time before speech start to be included as speech data. So when the start of a word is detected, based on the Start Speech Time parameter above, the endpoint is placed this amount of time before the first time frame which was labeled as speech.

Speech Trailer: This defines the amount of time after speech ends to be included as speech data. So when the end of a word is detected, based on the End Silence Time parameter above, the endpoint is placed this amount of time after the last time frame which was labeled as speech.

The above four parameters are used to translate VAD decisions into endpoints. The default values in Sphinx-4 are 200ms for the Start Speech Time and 500ms for the End Silence Time. For the Speech Leader and Speech Trailer parameters, the defaults for Sphinx-4 are 50ms. The Sphinx-4 VAD algorithm tends to clip the start and end of the words, and therefore these speech leader and trailer values are used.

In our experiments we decided to keep the default value of 200ms for the Start Speech Time. We also set the End Silence Time value to 200ms. Because our input sound recordings were only 2000ms in length, we deemed that the default value of 500ms for the End Silence Time was too long. For Sphinx-4 we kept the default value of 50m for the speech leader and trailer.

To translate the G.729 VAD decisions to endpoints, we implemented an algorithm which utilizes the same four parameters as those of Sphinx-4. For the G.729 endpointer we again used a value of 200ms for both the Start Speech Time and End Silence Time. Unlike Sphinx-4, the G.729 algorithm is more sensitive to speech and does not clip words. We therefore decided to use a value of 0ms for the speech leader and trailer. That is, the endpoints were marked exactly on the frames which were labeled as speech, without adding any extra signal before or after.

Configuration of parameters

The three algorithms under test (Variance Kernel, G.729, and Sphinx-4) change in the way they operate based on the values of certain parameters. Some of these parameters are built-in, while others are user-configurable. The built-in parameters are usually set by the authors of the algorithm to values which are found to work the best in most situations. The user-configurable parameters can be set by the user of the algorithm in order to achieve the desirable performance. The user-configurable parameters tend to influence the algorithm's sensitivity to noise. When a system is more sensitive to noise, it is more likely to detect speech where there is no speech. The less sensitive it is, the more likely it is to miss speech activity. We will talk about each of the algorithms in turn.

For the Variance Kernel method, there are many built-in parameters which can greatly affect the performance of the algorithm. Such built-in parameters include the length and overlap percentage of the time windows used to create the spectrogram, the order of the LPC analysis, the size of the variance kernel, the pixel threshold values used to remove binary objects, and the sampling frequency of the input signal. The values of these parameters are specified in Section 3.3.2. There is only one user-configurable parameter in the Variance Kernel algorithm. This parameter is the global threshold parameter described in Section 3.3.2. A higher global threshold makes the algorithm less sensitive. With a higher global threshold the miss rate increases. With a lower global threshold however, the number of wrong endpoint detections will increase. In our experiments we used a global threshold value of $\sigma = 10$. We found that this provides a good balance between the Miss rate and the Wrong rate.

For the G.729 VAD algorithm, there are many built-in parameters which are mainly related to thresholds. In Annex B of Recommendation G.729, the values of approximately 44 such parameters (or constants) are specified[5]. The executable program for the G.729 coder does not allow any user-configurable parameters. We have found however, that for short recordings with substantial levels of noise, the G.729 VAD has the tendency to classify the initial input frames as speech frames, even when there is no speech present. This results in a large error for the position of the endpoint at the beginning of the spoken word. In order to overcome this problem, we used a pre-padded input signal as input to the G.729 algorithm. The padding consisted of a signal with a duration of 2 seconds. This padding signal was created by taking the first 100ms of the original signal and concatenating 20 copies of it. As a result, during the experiments, a signal of 4 seconds duration is passed to the G.729 algorithm. Consequently, when evaluating the VAD results and calculating the endpoints, the first 2 seconds which are the padding are simply ignored.

For Sphinx-4, there are no built-in parameters per se, because all the parameters can be configured by the user using the Sphinx-4 API. Most of the parameters have default values which work well under most circumstances. The most important of these parameters in terms of configuration purposes is the threshold parameter.

[5]The documentation which specifies the constants can be downloaded from `http://www.itu.int/rec/T-REC-G.729-200701-I/en`

From the Sphinx-4 documentation[6]:

> If the current signal level is greater than the background level by this threshold, then the current signal is marked as speech. Therefore, a lower threshold will make the endpointer more sensitive, that is, mark more audio as speech. A higher threshold will make the endpointer less sensitive, that is, mark less audio as speech.

The default threshold value is 10. We tried threshold values ranging fro 7 to 13, and found that the default value of 10 gave the most satisfactory performance on our test data. Another parameter which can make a difference is the length of the each time frame. Again, we found that the default value of 10ms works well with our data. We therefore retained all the default values for the Sphinx-4 endpointer in our experiments, except for the End Silence Time parameter (mentioned above) which was changed from the default value of 500ms to 200ms.

Adding Noise

Our endpoint detection experiments were carried out using both the original sound recordings without any added noise, and also under various noise levels using various noise types. We added noise at levels ranging from 20dB SNR to -5dB SNR. The noise types are listed in Section 3.5.1.

The speech recordings in our database each have a duration of two seconds. Each of the noise sample files we acquired, have a longer time duration. For this reason it was necessary to clip each noise file so that it had the same duration as each of the speech recordings. We decided to use the same two seconds of noise from each of the noise files for all the experiments so that the results are repeatable. We chose to skip the first second of each noise file and to use the next two seconds.

In our tests, the noise was artificially added to the original speech recordings during the experiments. The original speech recordings were recorded under conditions without any background noise. Although artificially adding noise is an adequate way for evaluation purposes, it is not completely realistic. The reason is that when humans speak under noisy conditions, they tend to change their voice. When humans are aware that there is background noise, they change their voice so

[6]Taken from the javadoc at `http://cmusphinx.sourceforge.net/sphinx4/javadoc/edu/cmu/sphinx/frontend/endpoint/SpeechClassifier.html`

that it is easier for others to understand them. This is an involuntary tendency and it includes changes such as increase in sound intensity, increase in pitch, increase in vowel duration, and a shift in formant center frequencies. This is called the Lombard effect and some researchers have tested their algorithms under such Lombard conditions [47]. In order to do this, the speech recordings have to be performed in such a way that the noise can also be heard by the speaker during the recording. We did not do this, and so our experiments do not take into account the Lombard effect.

Running the tests

The tests were run in batch mode by passing each of the 450 sound recordings as input to each of the three algorithms using each of the 20 noise types, under each of seven noise level conditions. We therefore obtained 63000 outputs for each algorithm, and saved them in a large output file for subsequent analysis. For our Variance Kernel method and for the Sphinx-4 method we saved the calculated endpoints. For the G.729 method we saved both the endpoints, based on the rules we created, and the original VAD decisions.

3.6 Results

Our results can be used to analyze the performance of the three algorithms under test: the G.729 algorithm, the Sphinx-4 endpoint detector, and our Variance Kernel method. The results are based on a total of 450 manually endpointed recordings which we used in our experiments. From these, 185 of them do not contain any sound artifacts ("clean" recordings) and 265 of them do contain sound artifacts ("non-clean" recordings). The automatically calculated endpoints are evaluated by comparing them to manually selected endpoints which are considered to be the ideal endpoints for each word. Acero et al. [3] also presented endpoint detection results by using manually endpointed recordings. In their publication they present results based on 347 recordings.

There are two different ways to view the results of the three algorithms under test. They can either be seen as Endpoint Detection results or as Voice Activity Detection results. These two ways of evaluation were described in detail in Section 3.4.2.

Endpoint Detection makes a direct comparison between the automatic and man-

ual endpoints in order to find the time difference between them. We used the definition of correctness stated on page 80. A recording is considered to have been correctly endpointed if less than 50ms of the word is clipped on either end, and if less than 150ms of extra signal is added on either end of the word. For Endpoint Detection we present our results in the form of bar charts and tables.

Voice Activity Detection on the the other hand can be thought of as a classification exercise. The sound recordings are segmented into short time frames (typically 10ms in duration). The manual endpoints can be used to label the true class of each time frame as either speech or non-speech. Each time frame is then classified as speech or non-speech by the voice activity detector so that the hit rate and error rate can be calculated. For Voice Activity Detection we present our results in the form of line charts and tables.

In order to demonstrate the robustness to noise of our method, we used various types of added noise in our experiments. The complete set of Endpoint Detection results and Voice Activity Detection results for all noise types can be found in Appendix A. We also performed significance tests on the endpoint detection results using Fisher's exact test [29]. The significance tests show that for high levels of noise, when the SNR is less than 20dB, the performance of our variance kernel method is significantly different and better than the performance of both the G.729 algorithm and Sphinx4. The tests were carried out on the endpoint detection results using all the noise types combined, for clean and non-clean recordings separately. The p-values of the significance tests are shown in Appendix C. We now focus on five noise types which serve to show the strengths and weaknesses of the three algorithms.

3.6.1 Babble noise

Babble noise presents a particularly difficult challenge to the G.729 algorithm. Even when a low level of babble noise is added, the performance of the G.729 endpointer drops significantly. The reason for this is that the babble noise is classified as speech, even when the noise level is low. The Sphinx-4 endpoint algorithm however, performs well. Our Variance Kernel method also performs well, and in fact it performs just a bit better than Sphinx-4 for endpoint detection. As a voice activity detector, the Sphinx-4 algorithm is better at correctly classifying non-speech frames

than the other two methods when babble noise is added. For this reason, it also has the lowest error rate when used as a voice activity detector. The comparison in terms of endpoint detection performance can be seen in Figure A.2. The numerical values of the evaluation measures are available in Table A.2. The voice activity detection performance results can be seen in Figures A.24 and A.25. Table A.23 provides the numerical values of the performance measures.

3.6.2 Factory floor noise

In the collection of noise types that we have tried, there are two files with factory floor noise. The first of these contains sounds which create high local variance regions when converted to the spectrogram representation of our Variance Kernel method. This causes the accuracy of the Variance Kernel method to drop. Nonetheless, the Variance Kernel method still performs slightly better than the other two methods we tried with this type of noise. The comparison can be seen in Figure A.8. The numerical values of the evaluation measures are available in Table A.8. The voice activity detection performance results can be seen in Figures A.36 and A.37. Table A.29 provides the numerical values of the performance measures.

3.6.3 Machine gun noise

From all the noises tried in our experiments, machine gun noise poses the most challenging problem to the three endpoint detection methods. The irregular high-energy bursts of the machine gun sound are mistaken as speech by the endpoint detection algorithms when the noise level is high. This causes the three algorithms to have a high number of wrongly endpointed instances. Even so, the Variance Kernel method still outperforms the other two methods. This difference in performance between the Variance Kernel method and the other two methods is especially evident when the SNR is around 15db to 10dB. The comparison can be seen in Figure A.13. The numerical values of the evaluation measures are available in Table A.13. The voice activity detection performance results can be seen in Figures A.46 and A.47. Table A.34 provides the numerical values of the performance measures.

3.6.4 Car interior noise

Most of the energy of the car interior noise is in the low frequencies. Our Variance Kernel method performs outstandingly well in such cases because it ignores all frequencies below 200Hz. It is almost unaffected by the added noise, even at SNRs as low as -5dB. The G.729 algorithm also performs very well with this type of noise. It not affected significantly when the SNR is above 5dB. The Sphinx-4 endpoint algorithm however, does not perform well when the car interior noise is added. As the level of the noise increases, it wrongly classifies non-speech regions as speech. The comparison of the endpoint detection performance can be seen in Figure A.15. The numerical values of the evaluation measures are available in Table A.15. The voice activity detection performance results can be seen in Figures A.50 and A.51. Table A.36 provides the numerical values of the performance measures.

3.6.5 White noise

White noise uniformly corrupts all the frequencies of the speech signal with random noise. This type of noise provides a good approximation for various noises encountered in the environment, such as electrical noise. It is therefore useful to test the performance of endpoint detection methods with added white noise. In the case of the Variance Kernel method, the white noise has the effect of "masking" the high variance speech regions. The energy of white noise is distributed almost uniformly along all the frequency bands. Despite this drawback, the Variance Kernel method still outperforms the other two methods in the presence of added white noise. It is interesting to note that in the case of "non-clean" recordings, the Variance Kernel method performs better when a small level of white noise is added (with 20dB SNR) rather than when no noise is added. This is because some relatively low-energy sound artifacts (e.g. microphone clicks) are masked by the white noise and are therefore not mistaken as speech by the Variance Kernel. The G.729 voice activity detection algorithm has a higher speech hit rate than the other two algorithms at low SNRs. It can correctly classify some speech frames, when the other two algorithms wrongly classify the frames as speech. When the G.729 is used for endpoint detection however this does not result in higher accuracy. The reason is that under high levels of white noise, the G.729 algorithm misclassifies unvoiced speech frames as silence [8]. The endpoints resulting from the G.729 algorithm are therefore the

endpoints of the voiced region of the spoken word, which are not necessarily the endpoints of the whole word. This greatly decreases the accuracy of the G.729 algorithm as an endpoint detector in the presence of white noise. The comparison of the three methods can be seen in Figure A.16. The numerical values of the evaluation measures are available in Table A.16. The voice activity detection performance results can be seen in Figures A.52 and A.53. Table A.37 provides the numerical values of the performance measures.

3.7 Discussion

We have developed a method for performing endpoint detection which is based on a time-frequency image representation of sound. This method gives high accuracy even under the presence of high levels of background noise. We have compared our algorithm to the standard G.729 voice activity detector. It is well known that G.729 has a high error rate [9] under noisy conditions. Nevertheless, it provides a good standard for comparison because other researchers also use it as a benchmark [66, 90]. In addition, we have obtained results from a publicly-available open source endpoint detection algorithm which is part of the Sphinx-4 speech recognition system. The three methods were evaluated both as endpoint detectors and as voice activity detectors under noisy conditions using twenty different types of noise. The performance of our algorithm is shown to be comparable, and in many cases better, than G.729 and Sphinx-4.

The G.729 algorithm aims to keep the misclassification of unvoiced and silence frames to a minimum [8]. This is confirmed by our results which show that as the SNR decreases, the non-speech hit rate (HR0) for this algorithm remains at high levels, while the speech hit rate (HR1) decreases significantly. This can be seen in Figure A.23. Marzinzik and Kollmeier [66] have also used some of the same added noise that we have used in our experiments. Their results for G.729 agree with ours: the G.729 algorithm performs well with added vehicle noise, but with added babble noise its performance is extremely poor.

Our Variance Kernel algorithm uses two thresholds. One is automatically calculated using Otsu's method, and the other threshold is manually selected. It is very common to use thresholds in voice activity and endpoint detection systems. The values of the thresholds can have a strong impact on the performance of the

algorithm. Reliable threshold determination under noisy conditions remains an unsolved problem [57]. It is therefore important that our algorithm calculates one of the thresholds automatically because this makes the threshold adaptive and unbiased. The automatically-calculated threshold determines the endpoints. The additional manually-selected threshold is used as a parameter to control the tradeoff between the number of false detections and wrongly endpointed recordings. False detections can occur when a recording does not contain any speech.

Our endpoint detection system was designed in order to be used as a pre-processing step to a speech recognition system. Speech recognition performance is dependent on extracting complete speech segments from the recording, and not so much on accurate frame-level classification of speech and non-speech [57]. For this reason, it was important to test if our algorithm can find the endpoints of the words in the recordings, rather than being able to have a high frame-level accuracy. Evaluating an algorithm based on the accuracy of the calculated endpoints provides a better criterion for the subsequent success of the speech recognition system. On the other hand, a voice activity detection evaluation is based on the frame-level classification performance. Therefore, evaluation based on voice activity detection does not give the best indication to whether the speech activity detector will be a good pre-processing step for speech recognition. For example, the G.729 algorithm was designed to perform well as a voice activity detector, but it does not perform well as an endpoint detector because under noisy conditions it classifies many speech frames as non-speech. Our Variance Kernel method however, is better at detecting the endpoints of a word under noisy conditions. This can be clearly seen in Figure A.1.

Under noisy conditions the detected endpoints tend to capture only the high energy regions. Therefore, the endpoints move further in, missing the start and end of the word, which ultimately results in the clipping of the word. This is evident from the G.729 results where the VAD measures are good under noise but the endpoint detection results are very poor. Under high levels of noise, the G.729 algorithm only detects the high energy regions of the word. Our Variance Kernel method however, is better at detecting the low energy regions of the word under the same levels of noise. For speech recognition systems it is important that the endpoints of the word are detected as accurately as possible, because as stated by Lamel et al. [53]:

> Providing a great deal of latitude in the specification of the endpoint location tends to degrade the recognition performance severely. Hence, accurate location of endpoints is a strong requirement for a practical recognition system.

The use of both time and frequency parameters in endpoint detection systems has been shown to work well and lead to superior results [132]. In the method presented by Wu and Lin [132] however, the adaptive algorithm usually ignores high-frequency bands because these bands are the ones which are primarily corrupted with noise. Ignoring such high frequency bands could lead to problems when attempting to capture the endpoints of words like "six" which begins and ends with high-frequency, low energy, noise-like sounds. Our algorithm uses time-frequency information with the use of a spectrogram and can adequately detect speech at high frequencies, even when noise is present. Our method utilizes characteristics of the spectrogram which are robust to noise. This approach of using noise-robust spectral features was also successful in the spectral entropy approach presented by Wu and Wang [130]. Our algorithm is able to accurately endpoint spoken words even below 0dB SNR which is something that most other algorithms fail to achieve. Some authors even state that their algorithms are not supposed to work well below 0dB SNR.

The current state of our endpoint detection algorithm is ideal for voice-controlled equipment which operates under conditions with high background noise. In real-life recording environments, sound artifacts will always be present. Our recordings provide evidence of this. About 60% of our recordings were "non-clean" because they contained sound artifacts before or after the spoken word. As shown in our results, sound artifacts in the recording greatly affect the performance of voice activity and endpoint detection systems. When evaluating such systems, it is therefore important to use both "clean" and "non-clean" recordings as we have done in our experiments.

Chapter 4

Rank Order Kernels

4.1 Overview

In this chapter we will define Rank Order Kernels. The idea of Rank Order Kernels as described in this thesis is a novel one. It has been inspired by the following:

- Image basis functions
- Image kernel functions
- Rank Order Coding

Rank Order Kernels operate on two-dimensional images. They are used as a feature extraction procedure in order to compare images, for the purpose of classification. Rank Order Kernels are robust to noise. This is attributed to their Rank Order Coding aspect. Rank Order Kernels lead to a noise-robust distance metric between images.

4.2 Motivation

When comparing two images, the human brain is able to extract relevant information from the two images allowing it to determine how similar one image is to the other. It is able to extract significant features from the images and recognize important patterns. Although the notion of "patterns" in an image might be clear for a human to understand, these patterns cannot be explicitly defined in an exact mathematical way. It is difficult to say exactly which patterns are the ones that allow us humans to distinguish one image from another.

(a) Image from a natural scene

(b) Exactly the same pixels of the image on the left, shuffled and placed at random locations

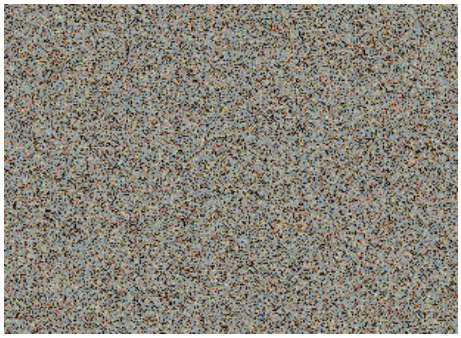

Figure 4.1: On the left is a natural image. The image on the right was produced by randomly shuffling the pixels of the image on the left. The image on the left is recognizable by a human because it contains certain primitive elements, such as edges. The image on the right has lost this information and therefore, it is not recognizable by a human. Image similarity metrics which use features based on primitive elements, such as the rank order kernel metric developed in this thesis, require this information.

Figure 4.1 shows two images. The image on the left is a natural image. The image on the right is obtained by just shuffling the pixels of the image on the left. Concerning the two images, Granai [37] states:

> The difference between the two images is clear to everybody, notwithstanding this it is not easy to define. In order to formalize such a distinction, it might be helpful to observe that an image presents peculiar elements such as wedges, textures, and smooth parts that are usually absent in random pixel combinations. Therefore, the problem of image representation must deal with these components or "primitives". Note that this topic is strongly related to the characteristics of the human visual system.

Following this reasoning, our task is to find components in images which will allow us to recognize patterns in order to create an image similarity metric for comparing images. We are dealing with two-dimensional images and so these components will themselves be two-dimensional in nature. A desirable characteristic of such components would be that they are robust to noise.

4.3 Background

Previous work regarding the search for image components can be loosely classified into two approaches:

1. Basis functions
2. Kernel functions

4.3.1 Image basis functions

Basis functions can be thought of as *synthesis functions* [74]. That is, any image can be modeled in terms of a linear superposition of basis functions. When modeling a two-dimensional gray-scale image, these basis functions are two-dimensional matrices, and so the basis functions themselves can also be visualized as gray-scale images. The basis functions are image components which can be used to describe a larger image.

The challenge is to find a suitable set of such basis functions. In the literature, the methods used to find these basis functions are optimization techniques which aim to minimize the reconstruction error and at the same time to maximize the sparsity of the representation [74, 105, 106]. The reconstruction error arises because the basis functions will not fully reconstruct the larger image. There will be a difference between the reconstructed image and the original image.

When the basis functions are used for discriminating between images, the optimization procedure for finding the basis functions can also optimize for discrimination [63]. The complete set of basis functions found by the optimization procedure can be called the *dictionary of elements*. This dictionary can be used to reconstruct larger images. For a given image, only a small number of elements from the dictionary is used to reconstruct the image, thus leading to a sparse representation.

Researchers have worked on finding basis functions for both visual images [37, 63, 74] and for spectrograms [49, 105, 106]. In this thesis, we are giving special attention to spectrogram images because we will use them for performing speech recognition.

Basis functions from visual images

Olshausen and Field [74] describe a procedure which aims to create an efficient coding of visual images using basis functions. They propose a sparse coding from

an over-complete set of basis functions. The training procedure used data derived from ten images of natural scenes. The resulting basis functions can be seen in Figure 4.2. It is interesting that these basis functions which were derived using a purely mathematical data-driven model resemble those found in the primary visual cortex of our brain.

One of the drawbacks of the procedure used to find basis functions is that it is computationally expensive. This is due to the optimization process which needs to be performed. When working with intensity images, each basis function is a gray-scale image. Therefore, each pixel can have any real value in the range from 0.0 to 1.0. This means that the number of possible basis functions is infinite. Nonetheless, a desirable characteristic of the basis functions is that they are learned from the images themselves. Learning takes place by using a set of training images. Small sections (patches) of the training images are used in order to find the basis functions. The size of the patches used is predefined and hence so is the size of the basis functions. Olshausen and Field [74] used ten images from natural scenes for training. The images were 512×512 pixels in size. From these ten images, image patches of size 12×12 pixels were selected at random. These small patches were then used for training in order to obtain the basis functions shown in Figure 4.2.

Basis functions from spectrograms

Smit and Barnard [105,106] used the process proposed by Olshausen and Field [74] on spectrogram images instead of visual images. The goal was to find a set of basis functions which could be used to reconstruct spectrogram images. For the training set they used the single digits ("oh", "zero", "one",... , "nine") from the TIDIGITS database [55]. They used 16 frequency channels for the spectrogram representation of the sounds. Frequencies below the threshold of human hearing were removed. The result is shown in Figure 4.3.

These basis functions are quite different from the basis functions obtained from visual images. It is intriguing to see that some of the basis functions capture the banded structure, which is a characteristic found in voice spectrograms. A subset of these basis functions can be used to reconstruct any spectrogram of the spoken digits. An example is shown in Figure 4.4. The example demonstrates how only four of the 30 basis functions can be used to reconstruct the spectrogram of the word "six".

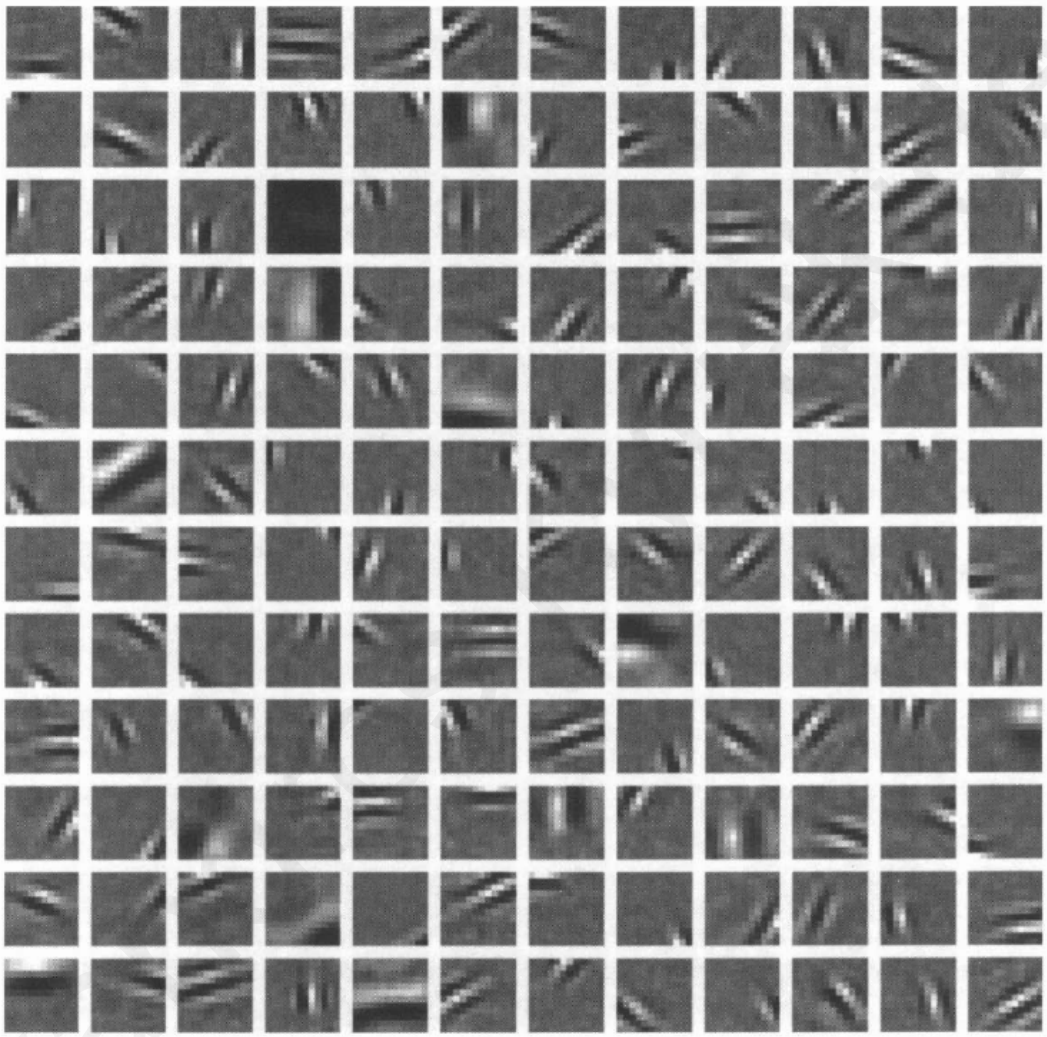

Figure 4.2: Basis functions for visual images. This is the set of 144 basis functions learned by a sparse coding algorithm using visual images as input. All have been normalized to fill the gray scale. (Figure taken from [74]).

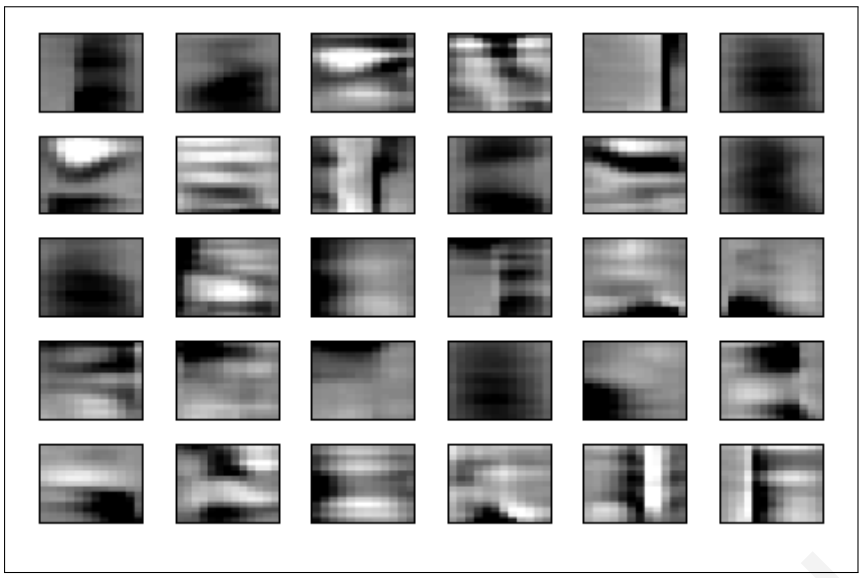

Figure 4.3: Basis functions obtained from spectrograms of spoken digits. The y-axis gives the frequency. The x-axis gives the time. The basis functions use 16 frequency channels and span a time length of 250ms. (Figure taken from [105]).

These four basis functions and their time location in the reconstructed spectrogram image characterize the word "six". The basis functions and their time location can therefore be used as features in order to recognize the word.

4.3.2 Kernel functions

When referring to images, a kernel function is some type of function operating over a rectangular area of pixels of the input image. The size of the kernel defines the height and width of the area of the pixels that the kernel function will operate on. The pixel values in the rectangular area covered by the kernel are the input to the kernel function, and the output of the kernel function is a single value. The kernel function can be used to transform an input image to an output image by "sliding" the kernel function over all locations of the input image. At each location, the output of the kernel function defines the pixel value of the output image at that location. This is shown graphically in Figure 4.5.

A convolution kernel is a very common type of kernel function. It consists of multiplying each input pixel by a coefficient (defined by the kernel itself), followed by a summation, to calculate the output value. The coefficients are arranged in a two-dimensional matrix which fully defines the convolution kernel (also called a

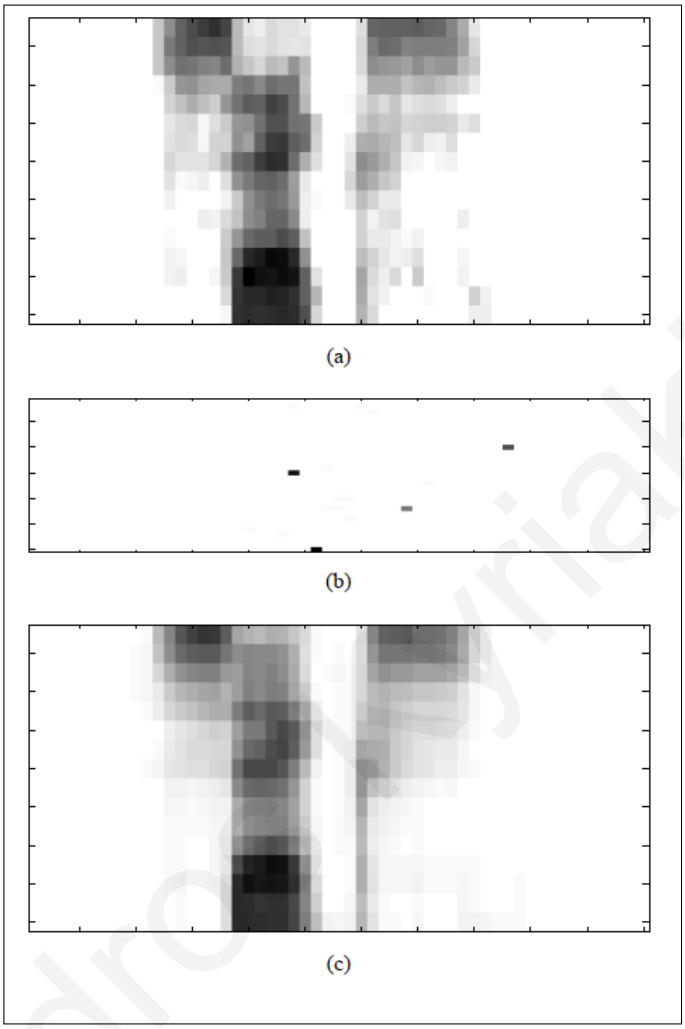

Figure 4.4: Showing how the basis functions from Figure 4.3 can be used to reconstruct and thus recognize the spoken word "six". (a) A spectrogram of the word "six". (b) The sparse code for the spectrogram using the basis functions in Figure 4.3. Only four of the basis functions (1, 9, 16, 21) are used to reconstruct the spectrogram. The x-axis shows the time location where each basis function is used. The y-axis indicates the numeric identifier of the basis function. Basis function 16 is used first, then basis function 1, followed by basis function 9, and then basis function 21. (c) The spectrogram after it has been reconstructed from the sparse code. (Figure taken from [105]).

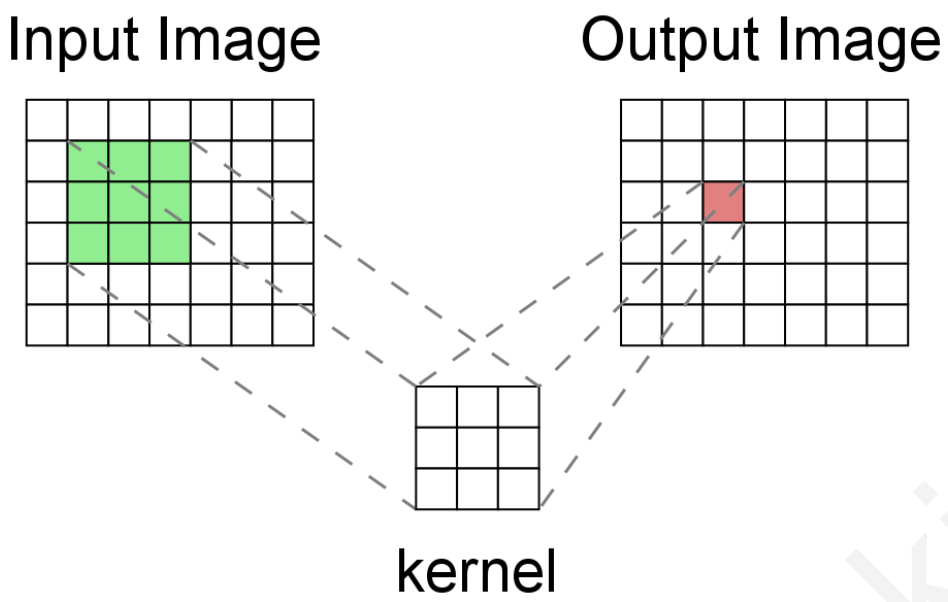

Figure 4.5: Converting an input image to an output image using a kernel.

convolution filter). For example, the convolution kernel described by the matrix in Figure 4.6a is called an "on-center" convolution kernel because it emphasizes the regions in the input image which have a bright pixel surrounded by dark pixels. The off-center kernel, of which an example is shown in Figure 4.6b, does the opposite. It emphasizes regions of the input image which have a dark pixel surrounded by bright pixels. The retina of the human eye has cells which perform similar operations as these two convolution kernels. The next level of the human visual system uses orientation maps which detect edges in various directions in order to process the images. For this reason, attempts to create image recognition systems which imitate the human visual system employ on-center, off-center, and orientation convolution kernels when processing visual images [119]. A schematic of the process is shown in Figure 4.7 and Figure 4.8 shows the results of this process when it is applied to an image of a face. When trying to recognize faces in an image, the faces can appear in different sizes in the image. This is handled by using different scales of convolution kernels [113].

Using kernel functions on spectrograms

The approach of using convolution kernels on visual images can also be applied to spectrograms. Instead of using a visual image as input to the process shown in

-1	-2	-1
-2	12	-2
-1	-2	-1

(a) On-center

1	2	1
2	-12	2
1	2	1

(b) Off-center

Figure 4.6: Convolution kernels

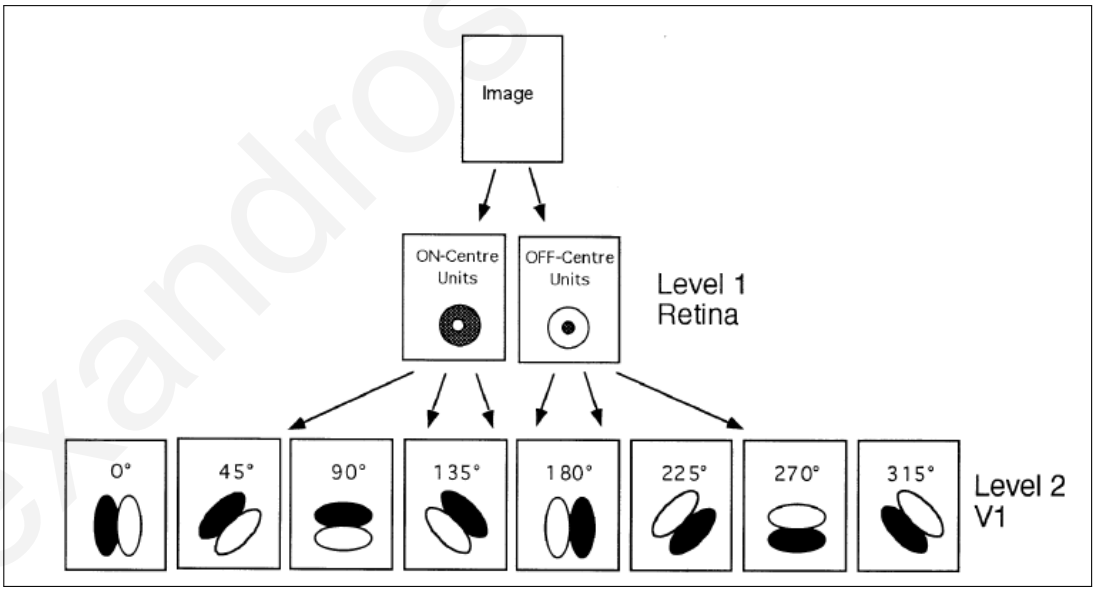

Figure 4.7: Processing an image using convolution kernels. The first level uses on-center and off-center kernels. The second level uses orientation kernels.(Figure taken from [119])

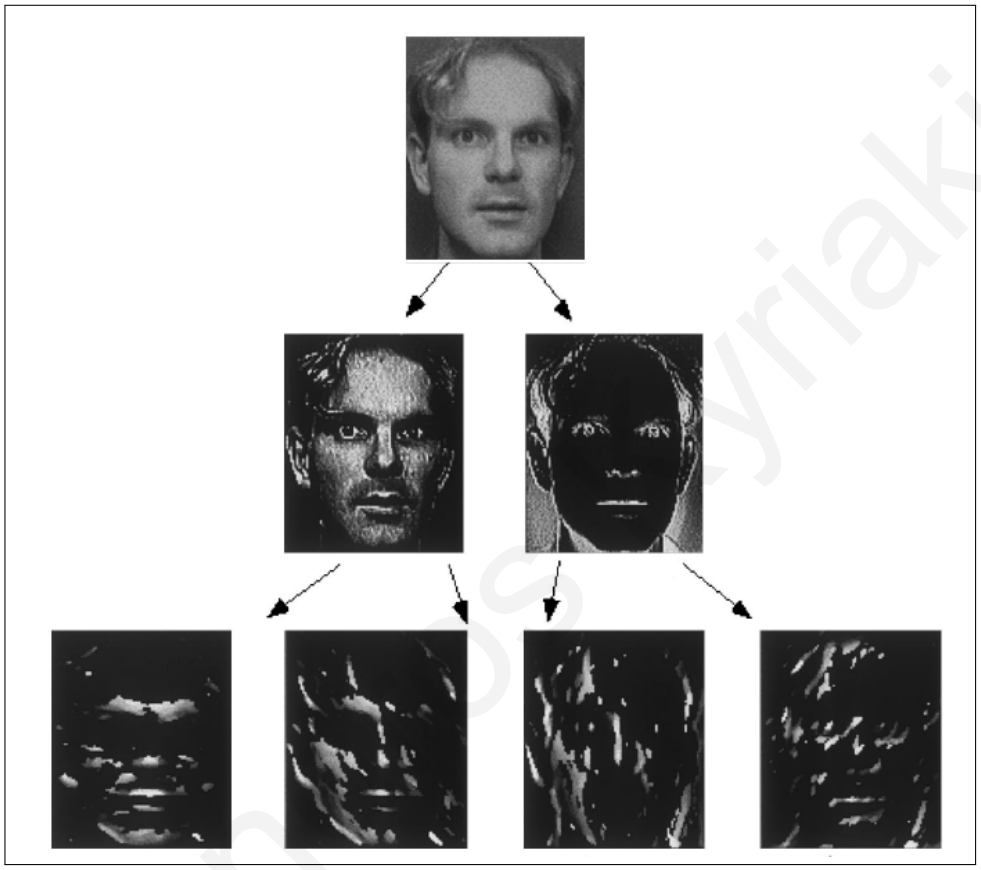

Figure 4.8: Processing an image of a face using convolution kernels. The first level uses on-center and off-center kernels. The second level uses orientation kernels.(Figure taken from [119])

Figure 4.7, we can use a spectrogram as input.

Although these convolution kernels (on-center, off-center, orientation) work well for visual images, it is not clear what kernels work best for time-frequency images originating from sound signals. The number of possible convolution kernels is infinite, so trying a brute-force approach to find appropriate convolution kernels is not feasible. Almost any approach would be prohibitively expensive computationally. It is a challenge to find appropriate kernels to use for processing time-frequency images of sound.

An approach presented by Viola and Jones [123] uses kernels which they call "rectangle features". These features are shown in Figure 4.9. They are a special kind of kernel function. The output of the kernel function is the difference of two sums. The kernel separates the pixel area under consideration into two regions. The sum of the pixel values of one region is subtracted from the sum of the pixel values of the other region. The following is an explanation of these features (or kernels) taken from their paper [123]:

> The simple features used are reminiscent of Haar basis functions which have been used by Papageorgiou et al. [77]. More specifically, we use three kinds of features. The value of a *two-rectangle feature* is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. A *three-rectangle feature* computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a *four-rectangle feature* computes the difference between diagonal pairs of rectangles.

Viola and Jones used these rectangle features for face detection. The same type of features were then used by Ke et al. [49] for music identification. They converted music signals to spectrograms and then used the rectangle features to capture important characteristics of the sound. It is important to note that these rectangle features were not created specifically for detecting characteristics in spectrograms. The features were generated by Viola and Jones for visual images and then the same features were then used successfully by Ke and al. for music identification.

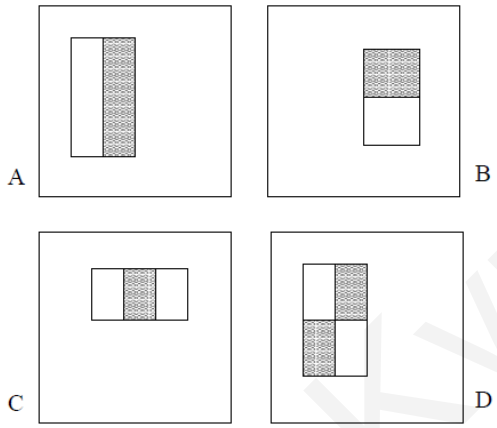

Figure 4.9: The rectangle features are a kernel function which sums the pixel values in two different regions (light regions and dark regions, as shown in the figure) and then subtracts one sum from the other. In this figure, four rectangle features are shown relative to an enclosing detection window. The sum of the pixel values which lie within the white rectangles are subtracted from the sum of the pixel values which lie within the dark rectangles. This is the output of the kernel function. (A) and (B) show "two-rectangle features". (C) shows a "three-rectangle feature". (D) shows a "four-rectangle feature". (Figure taken from [123]).

4.3.3 Rank order coding

Rank order coding is a proposition on how neurons transmit information. The classic model describing the transmission of information in neurons is based on rate coding. Meanwhile, Thorpe et al. [34, 113] have argued that rate coding is not sufficient to explain the speed with which a primate's visual system can process information. A simple model of a neuron is one which has one or more inputs, a function to process the inputs, and one output. In a neural network the output of one neuron is connected to one of the inputs of another neuron. The output information is transmitted along the axon of the neuron as a series of spikes. Rate coding is based on the rate, or frequency, of these spikes. The higher the rate of the spikes, the higher the output value. Regarding rate coding, Gautrais and Thorpe [34] state the following:

> A simple mathematical analysis reveals that, due to the stochastic nature of spike generation, even transmitting the simplest signals reliably would require either: (1) excessively long observation periods incompatible with the speed of sensory processing or (2) excessively large numbers of redundant neurons, incompatible with the anatomical constraints imposed by the sensory pathways.

Temporal codes are an alternative to rate codes and they overcome these problems. A temporal code is based on a population of neurons. The neurons fire asynchronously. The temporal code is determined by the relative time difference between the arrival of spikes across the population of neurons. What is important in temporal coding is not the rate of the spikes on each neuron, but the time of arrival of a spike on a certain neuron relative to the time of arrival of the spikes on the other neurons.

Rank order coding is a type of temporal coding. With rank order coding only the first spike on each neuron is important. This has the advantage that only one spike is required per neuron. The latency of each spike is determined by the intensity of the input to the neuron. An input with higher intensity gives a lower latency. Spikes with lower latency arrive first at the output. This is illustrated in Figure 4.10 where a population of 8 neurons is used to encode an input signal. The intensity of the signal varies in space. Each neuron is connected to a different position on

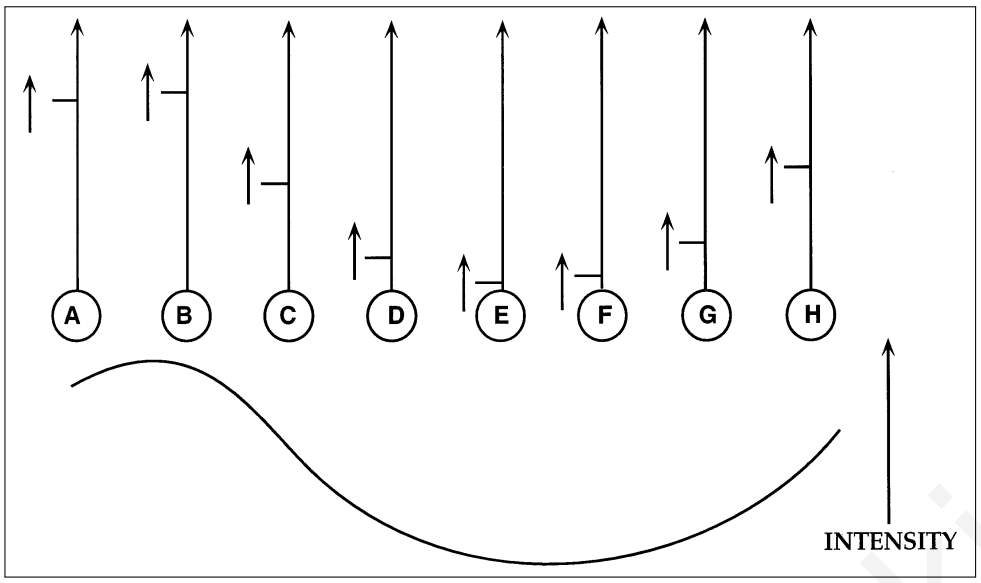

Figure 4.10: A population of 8 neurons. The intensity at the input of each neuron determines the latency of firing. Each neuron has fired one spike. More strongly activated neurons fire first. The rank order code is the order of firing: B,A,H,C,G,D,F,E. (Figure taken from [119])

the signal and, as a result, each neuron receives a different intensity value as input. A higher intensity leads to a lower latency. Neuron B is the one connected to the location with the highest intensity. Accordingly, neuron B has the lowest latency and consequently it fires first. This is represented in the diagram with the spike on neuron B having traveled the longest distance along the axon of the neuron. Neuron A fired second because it is connected to the second-highest intensity value. The rest of the neurons fire accordingly based on the input intensity at their location. Neuron E is the last one to fire because it receives the lowest intensity. The order of firing is B,A,H,C,G,D,F,E. This is the rank order code. With eight neurons, there are 8! different possible orderings, or codes.

4.3.4 Applications of Rank order coding

Thorpe et al. have created a software program called SpikeNet which simulates large networks of asynchronously firing neurons which use Rank Order coding [20, 22, 113,114]. They have used their model successfully for face identification [21,119] and scene categorization [112]. The idea of rank order coding has also been applied by others for efficient image reconstruction and encoding [10,101,102]. More relevant to

this thesis is the fact that rank order coding has been used in various ways for speech recognition. Loiselle et al. [59] have experimented with a small speech database of French digits. They used recordings from 5 male and 4 female speakers and showed that rank order coding can lead to low error rates even when the training set is small. One male and one female speaker were used for training and recognition was performed on the other 7 speakers. Uysal et al. experimented with rank order coding in order to perform speech recognition on five vowel sounds [117]. They showed that rank order coding is robust to noise [116] and also compared rank order coding to other types of spike coding techniques [118].

4.3.5 Advantages of Rank order coding

Rank order coding depends only on the relative order of the first spikes and not on their precise timing. This offers certain important advantages.

Short response times: Response times are short because processing can take place with only one spike per neuron. Contrast this to rate coding where a relatively longer time is needed for processing the code due to the necessity to sample the spikes on each neuron over a long enough period in order to estimate the frequency of the spikes.

High information capacity: A population of n neurons can encode $n!$ different codes. For example, there are 40320 different possible codes with just 8 neurons. Therefore, when creating applications which use rank order coding, only a few neurons are usually needed. This reduces the memory and hardware requirements.

Robust to noise and changes: Even if the spike timings of each neuron change by a relatively small amount, the rank order code remains the same, as long as the order of the spikes does not change. This makes the code robust to noise. Noise can randomly alter spike timings by small amounts. Additionally, large uniform changes in the overall intensity of the input do not change the rank order code. This is easily illustrated by an example, as shown in Figures 4.11 and 4.12. Take a set of 9 gray-scale intensity pixels. Each pixel is connected to one neuron. Each neuron fires one spike. The intensity of each pixel determines the latency of each spike. The bottom part of Figure 4.11

shows the pixel intensities. The top part of the figure shows the neurons with the fired spikes. The neuron connected to the pixel with the highest intensity fired first, and hence its spike has propagated the longest distance along the axon of the neuron. Figure 4.12 shows what happens when one of two changes is made to these pixels. The top of the figure shows three different sets of pixel intensities. The one in the center is the same as the one in Figure 4.11. The two pixel sets on the left and right have lower contrast. That is, there is a smaller difference between the highest intensity pixel value and lowest intensity pixel value. The one on the left has low luminance, whereas the one on the right has high luminance. What is important to note is that all three sets of pixels have exactly the same rank order code. This is shown in the bottom part of the figure. The changes in contrast and luminance do not affect the rank order code in any way, making it robust to such changes. For example, say these pixels were part of an input image taken from a room. If the lights in the room were suddenly dimmed, the rank order code which would be used to encode the image would not change.

Fast training: Although there are various ways in which rank order codes can be used for learning, it is usually the case that training is simple and fast. This is mainly due to the finite number of codes which are possible for each population of neurons, and their discrete nature.

Simple implementation: Compared to other types of neural codes, implementing rank order coding is simple because the exact timings of the spikes are not needed. Only the order is important.

Parallel implementation: The processing can be performed in parallel because of the architecture of the neural network. Each neuron can process the input independently from any other neuron. Also, in a hierarchical architecture, each level can process information independently from the other levels. This has been shown using the SpikeNet implementation where several machines were used in parallel over the network to process images using rank order coding [22].

Sparse coding: In most cases, enough information is available at the output as soon as the first spikes arrive. In a recognition task for example, it is possible

to recognize the image without waiting for the spikes to arrive from all the neurons. By using just the spikes from the first few neurons, the recognition can be made. The input image can therefore be sparsely coded by using only a small number of neurons from the total number. The advantages of this sparse coding are twofold: speed and robustness. It is faster because recognition can take place without having to wait for all the neurons to fire. It is more robust because only the salient features of the image are considered (first spikes) without considering the less important features (later spikes). It can also be argued that the first spikes encode more general features of the input, while the later spikes encode more specific details. In our formulation of rank order kernels in Section 4.4.2, we use this fact to define a *degree* for the kernel. The kernel degree places a limit on the number of first spikes to consider, therefore influencing how general or specific a kernel will be.

4.3.6 Other rank-based methods

Although the idea of rank order coding for neural codes was first proposed by Thorpe et al. [113], the ideas of rank and order have also been used in other fields. In signal detection theory the goal is to separate the information from the noise. Non-parametric detection schemes can be used which do not make assumptions about the statistics of the signal. Impressive results have been obtained using nonlinear rank statistics [111]. Nonparametric rank-order statistics have been used successfully to classify regions of sound signals into voiced, unvoiced, and silence regions [17]. Order statistic filters have been used in speech processing for endpoint detection [92].

Similarity measures have been proposed for comparing ranked lists which are encountered in daily life [125]. Such lists include the list of results returned by a search engine. For location-aware mobile applications, a rank based fingerprinting algorithm has been proposed for indoor positioning [62].

In image processing, rank-order spatial filters are a type of nonlinear filter. They have been successfully applied for the restoration of images, and have been shown to overcome certain problems which were not solvable by the use of linear filters [94]. In image processing terms, the pixel neighborhood on which the filter operates is called the "mask". The output of a rank-order filter is determined by ordering the pixels under the mask using their values. The median filter is the best-known

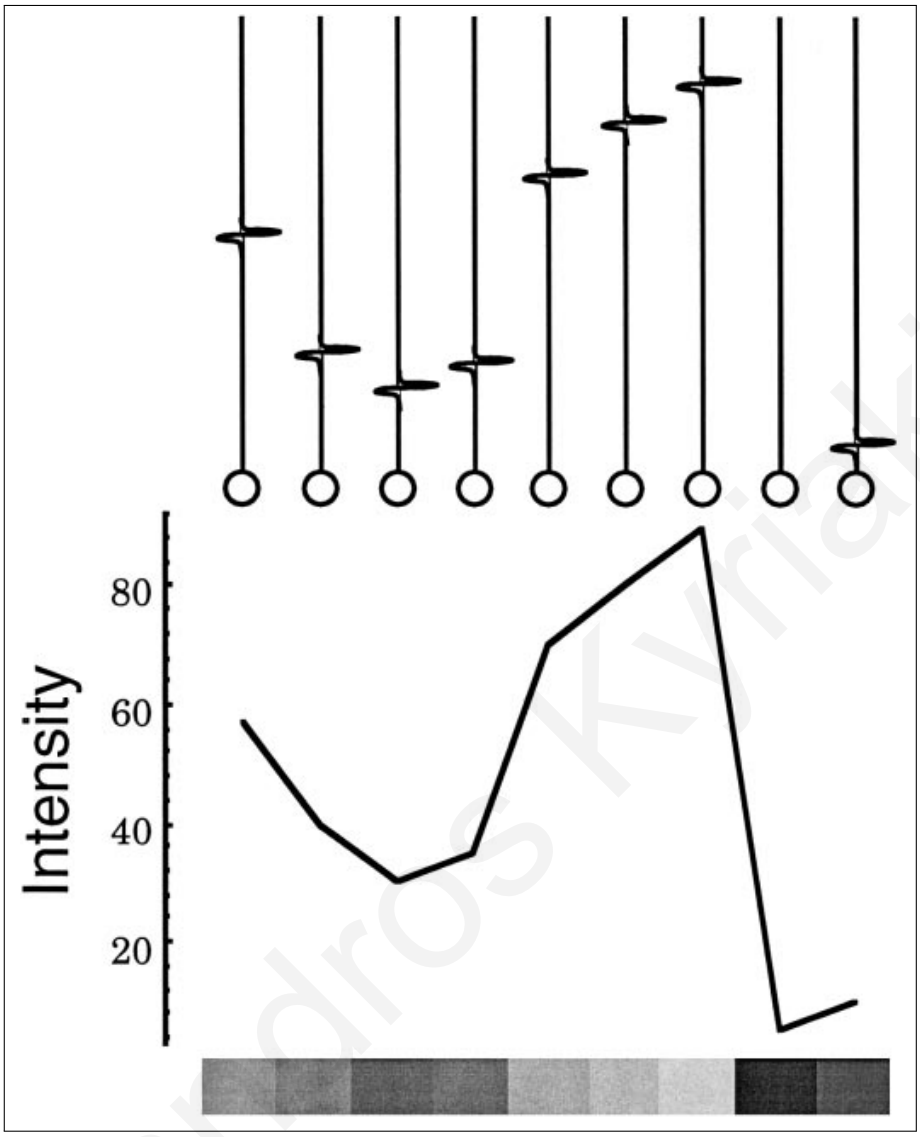

Figure 4.11: Spiking neurons connected to a set of pixels. At the bottom of the figure there are 9 gray-scale pixels representing an image. At the top of the figure there are 9 neurons. The input of each neuron is connected to each of the pixels. Each pixel has a different intensity. The intensity of each pixel determines the latency of firing of each neuron. The neuron connected to the pixel with the highest intensity (brightest pixel) fires first. The neuron connected to the pixel with the lowest intensity (darkest pixel) will fire last. (Figure taken from [34])

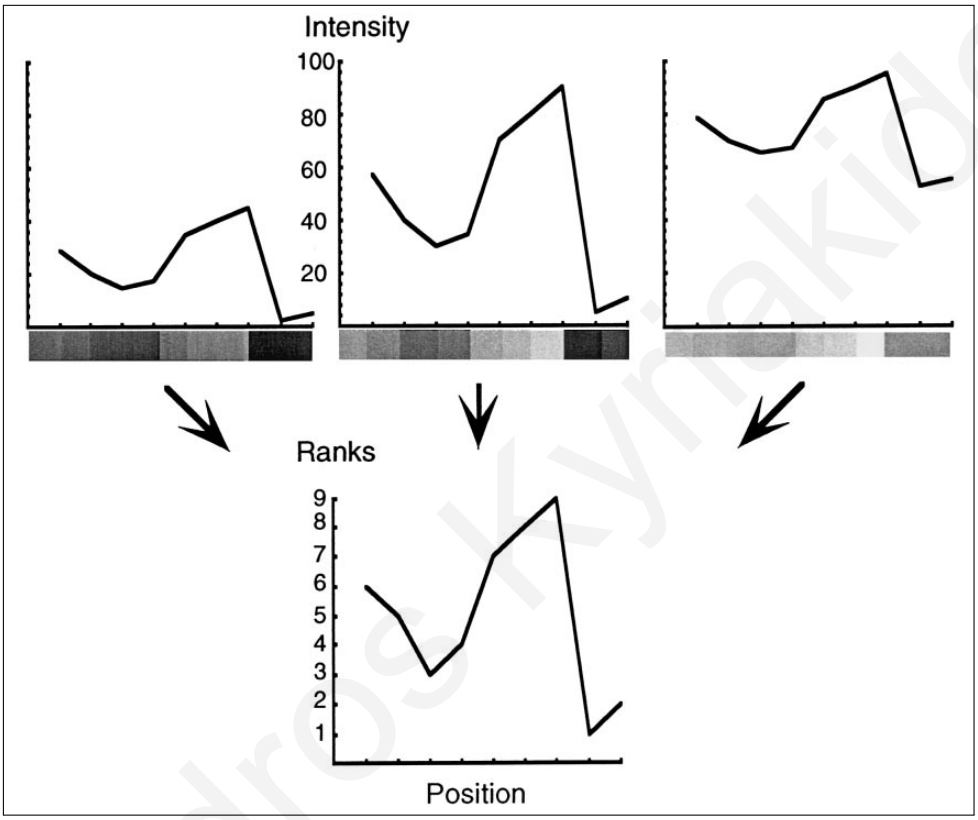

Figure 4.12: Rank order coding is robust to changes in contrast and mean luminance. At the top center of the figure are the same 9 pixels from Figure 4.11. At the top left, the contrast of the pixels is smaller and the mean luminance is lower. At the top right, the contrast of the pixels is again smaller but the mean luminance is higher. In all three cases the rank order coding is the same. This is shown at the bottom of the figure. (Figure taken from [34])

example of such a rank-order filter. The output of the median filter is the value of the pixel appearing in the middle of the order. It is crucial to emphasize that the output of these image processing filters is a real number. The rank order kernels we describe in the next section differ from these classic filters because the output of the rank order kernel is not a real number. The output of a rank order kernel is a rank order code.

4.4 Rank order kernels

In this section we will describe rank order kernels, which are one of the main contributions of this thesis. These kernels were inspired by basis functions, kernel functions, and rank order coding, as described earlier in this chapter. Rank order kernels will be used for constructing an image similarity metric which is robust to noise. In the next chapter, this similarity metric will be used to compare spectrograms of spoken words in order to perform speech recognition. Before defining rank order kernels, some critical distinctions need to be made between basis functions and kernel functions.

- Basis functions can be regarded as *synthesis functions* whose main purpose is to *reconstruct* an image. They can also be used to discriminate between images by finding a sparse code of basis functions which are needed to reconstruct specific images. Images which use the same code can be considered to be similar.

- Kernel functions can be regarded as *analysis functions* whose main purpose is to *filter* an image. They have an input and an output. They can be used to discriminate between images by transforming images to representations which emphasize the important discriminative features between the images.

- Basis functions are obtained by using a set of training images and an optimization procedure. The basis functions arise directly from the images used for training. That is, the process is "data-driven".

- Kernel functions can be pre-defined before any data is processed. For example, on-center and off-center convolution kernels are defined irrespective of the data they will process.

- Kernel functions can also be learned from the data. The challenge is to find a method to accomplish this. Viola and Jones [123] presented a method by using what they call "rectangle features". The training was done on visual images. To the best of our knowledge there is no published work which presents kernel functions trained on spectrogram images.

- Computational complexity is always a concern when learning from data. Learning basis functions is costly. Learning kernel functions can be even more costly. So in order to learn kernel functions, it is important to find a way which is not computationally expensive. Learning convolution kernels for example is prohibitively expensive. That is one reason Viola and Jones used "rectangle feature" kernels instead of convolution kernels.

- When using basis functions, it is desirable that the coding is sparse and that the dictionary is over-complete. There is an infinite number of possible basis functions and so an optimization procedure is needed to find the best ones. Methods used in the literature optimize by minimizing the reconstruction error and by maximizing the sparsity at the same time.

- With kernel functions, if one tries to find optimal convolution kernels, the number of possibilities would again be infinite.

- With kernel functions, using the "rectangle features", the possibilities are finite, but again they are quite numerous. For example, as the authors state [123], when using a 24×24 pixel size rectangle, the number of rectangle features is over $180,000$. It is important however that the number is finite. They use a weak learning algorithm (AdaBoost [31]) to select a small number of significant features. The significant features are the ones which best separate positive from negative examples.

Our approach consists of using a method which is inspired by both basis functions and kernel functions. We will create kernel functions using rank order coding. For a given kernel size, the number of possible kernels will be finite. The kernel functions are learned from the data.

4.4.1 Rank order kernels defined

A rank order kernel operates on a two dimensional image, which can be represented as a two-dimensional matrix of intensity values. The kernel is applied in a similar manner to that of image spatial filters. It is "slid" across the image to create an output that is associated to the particular kernel. Whereas in image processing the output is a single real value, in the rank order kernel case, the output is a vector containing kernel-related positional indices, ranked in terms of pixel intensity. We decided to use characters to represent the positional indices and therefore the rank order kernel output is a character string. The kernel size ($M \times N$) is defined by the kernel's height (M) and width (N) in pixels. The kernel needs to have a well-defined center point. For this reason, we restrict M and N to have only odd values. In general, a kernel transforms an input image to an output image. The rank order kernel transforms an input image into a two-dimensional array of rank order codes. Each element of the output array is a character string which is the rank order code. The kernel performs an operation on the input image on a neighborhood of pixels around the center point. This neighborhood is defined by the size of the kernel. The result of the rank order kernel operation becomes the value of the element in the output array at that center point. The kernel operation is performed for every point of the input image, by moving the center point to every pixel in the input image. An illustration of this process is shown in Figure 4.13. In the figure, the kernel has a size of 3×3 pixels. The neighborhood in the input image around the center pixel on which the kernel operates is shown in green. The output of the kernel defines the value of the corresponding center pixel in the output array, which is shown in red. The figure shows one single operation around a specific center pixel. To complete the transformation from input image to output array, the kernel operation has to be performed on all the pixels of the input image.

The output of the rank order kernel operation is a rank order code. Each pixel location of the kernel is given a label. An example can be seen in Figure 4.14 where each of the 9 pixel locations of the 3×3 kernel is given a label using the letters from A to I. The output of the kernel is a character string which represents the order of the corresponding pixel values on which the kernel operates on the input image. In the figure, the pixel values of the 3×3 neighborhood in the input image are shown inside the green squares. For gray-scale intensity images, the values can range from

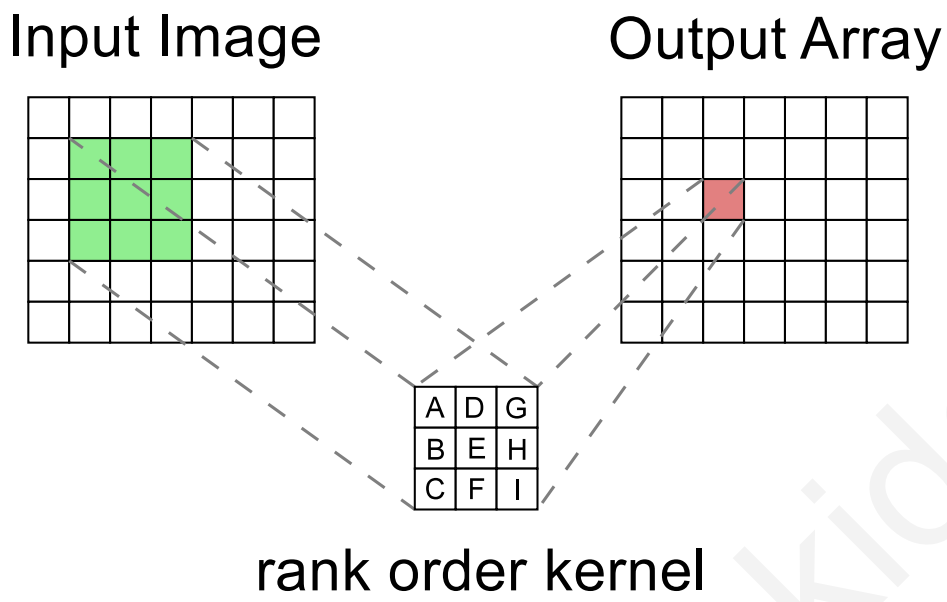

Figure 4.13: Converting an input image to an output array using a rank order kernel. Each element of the output array will be a rank order code, represented by a character string.

0.0 to 1.0. The highest pixel value in this example is 0.99, and it appears at location D. Hence, the rank order code starts with D. The second-highest pixel value is 0.93 and it is located at position A. The lowest pixel value is 0.12, at position G. The full rank order code for this example is D,A,B,F,H,C,I,E,G. This rank order code is the output of the kernel operation. In the case of ties (when two or more pixels have the same value), we arbitrarily choose to order the labels in alphabetic order.

4.4.2 Degree of Rank order kernels

When using rank order coding (described in Section 4.3.3) it is not necessary to wait for all the neurons to fire. As explained earlier, it is usually advantageous to consider only the neurons which fire first and to ignore the rest which fire later. This is accommodated by the rank order kernels by defining a degree for the kernel. The degree of the kernel is the number of top pixel values which will be used for the output. It is analogous to specifying the number of neurons for which we will wait to fire in the rank order coding paradigm. For a kernel of degree n, only the top n pixel values of the input will be used for the rank order code. This is equivalent to saying that only the first n neurons to fire will be considered.

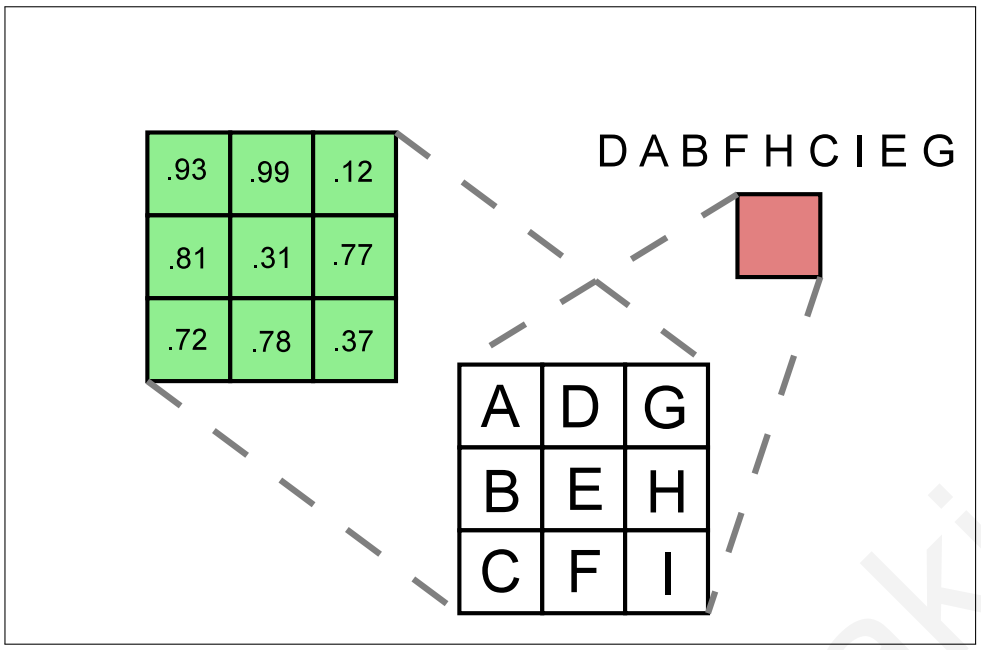

Figure 4.14: An example of a rank order kernel operation. The kernel has size 3×3. The input pixel values are shown on the top left, with green background color. The pixel locations of the kernel are labeled from A to I. The output of the kernel is shown at the top right. It is the rank order code of the 9 input pixel values.

An example of a rank order kernel of degree $n = 1$ is shown in Figure 4.15. In this case only the top pixel value is considered. It is located at position D, and as a result the output of the kernel is just D. In Figure 4.16 a rank order kernel of degree $n = 2$ is shown. Only the top two pixel values are considered. The output of the kernel is their rank order which is D,A. Figure 4.17 shows a rank order kernel of degree $n = 3$ and Figure 4.18 shows a rank order kernel of degree $n = 4$. In our experiments using spectrograms for speech recognition, which will be described in the next chapter, we have found that 3×3 rank order kernels with degrees 1 to 4 give the best results. A kernel with a lower degree captures more general features, whereas a kernel with a higher degree is more specific. The degree of the kernel provides a way to control the generality of the kernel.

4.4.3 Image similarity metric using rank order kernels

Our goal is to calculate an image similarity metric between two images. For this particular discussion we restrict ourselves to gray-scale images. Let's start with two gray-scale images G_1 and G_2. The first step in calculating the image distance

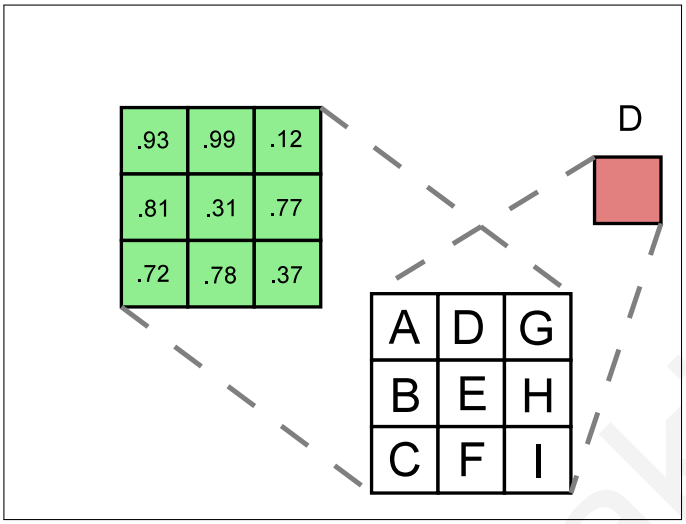

Figure 4.15: A rank order kernel of degree $n = 1$ operating on a 3×3 pixel area. Only the top pixel value is considered. The output of the kernel is: D.

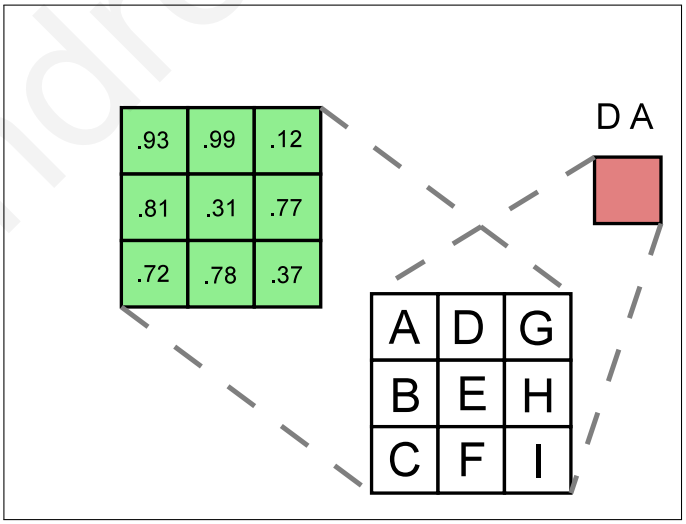

Figure 4.16: A rank order kernel of degree $n = 2$ operating on a 3×3 pixel area. Only the top two pixel values are considered. The output of the kernel is: D,A.

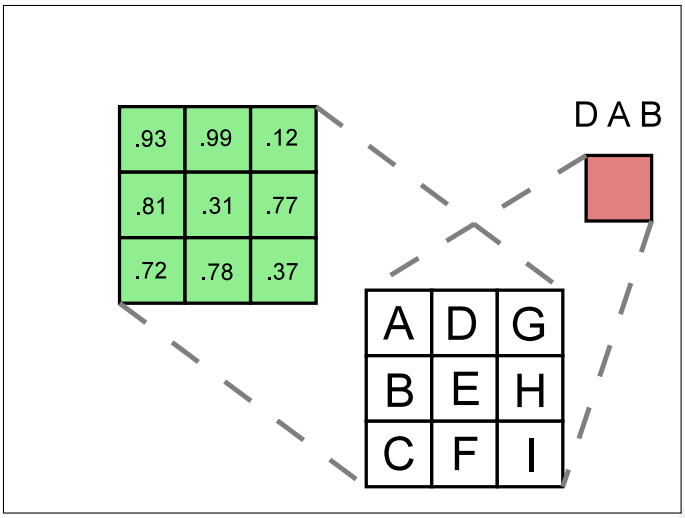

Figure 4.17: A rank order kernel of degree $n = 3$ operating on a 3×3 pixel area. Only the top three pixel values are considered. The output of the kernel is: D,A,B.

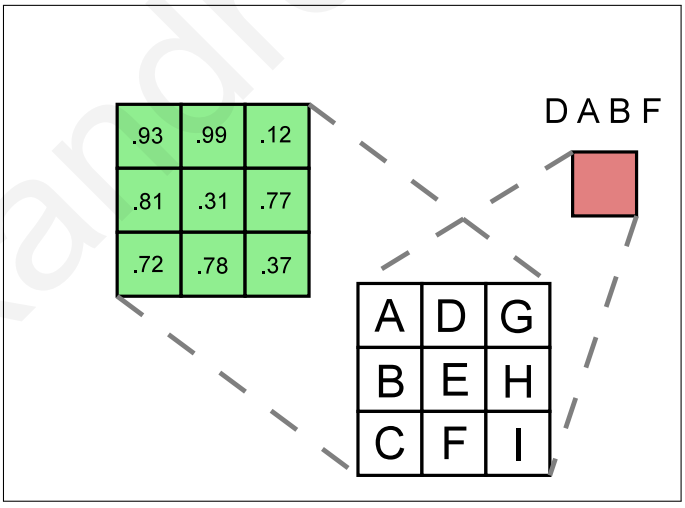

Figure 4.18: A rank order kernel of degree $n = 4$ operating on a 3×3 pixel area. Only the top four pixel values are considered. The output of the kernel is: D,A,B,F.

between the two images is to transform both images using rank order kernels. Three parameters have to be chosen: the height of the kernel (M), the width of the kernel (N), and the degree of the kernel (n). The image distance depends on these three parameters. In our experiments with spectrogram images we found that 3×3 kernels work the best, with degrees ranging from 1 to 4.

The original gray-scale images have an intensity value for each pixel. Once both images have been transformed, the two resulting "images" are actually two-dimensional arrays containing rank order codes. Processing gray-scale image G_1 with the rank order kernel gives output array R_1. Processing gray-scale image G_2 with the same rank order kernel gives output array R_2. The elements in R_1 and R_2 are each represented by a rank order code, which is the result of the rank order kernel operation. The next step is to compare each element in R_1 and R_2 to find how many corresponding elements have the same rank order code. Two corresponding elements are ones which have the same (x, y) location in array R_1 and array R_2. Equations 4.1 and 4.2 describe the transformation from gray-scale images to rank order code arrays. G_1 and G_2 are the two gray-scale images. The function $k(p)$ is the rank order kernel which operates on the pixel neighborhood centered around pixel p. The result of the two transformations are the two rank order code arrays R_1 and R_2.

$$R_1(x, y) = k(G_1(x, y)) \tag{4.1}$$

$$R_2(x, y) = k(G_2(x, y)) \tag{4.2}$$

Equation 4.3 describes the distance metric (d) between two rank order code arrays R_1 and R_2 of size $X\times Y$. The binary operator $\overset{ROC}{=\!=}$ operates on two rank order codes. If the rank order codes are exactly the same, the operator returns the value 1, otherwise it returns the value of 0. This can be visualized as a binary image. The binary image has a value of 1 at locations where the rank order codes match, and a value of 0 where they do not match. An example of this can be seen in Figure 5.3 on page 140 where two gray-scale spectrogram images are compared using rank order kernels. The two gray-scale images for which the image similarity metric is to be calculated are shown in subfigures (a1) and (b1). The binary image is shown in subfigures (a2) and (b2). The red pixels indicate the locations where the binary image has a value

of 1. These are the locations where the rank order kernels gave the same output for both spectrograms.

$$d = 1 - \frac{1}{XY}\left[\sum_{y=1}^{Y}\sum_{x=1}^{X}\left(R_1(x,y) \overset{ROC}{=\!=} R_2(x,y)\right)\right] \tag{4.3}$$

With the distance metric equation, if all the corresponding rank order codes between R_1 and R_2 are the same, the result will be that d is 0.0. That is, the distance between the two original images is zero: they are exactly the same as far as the rank order code is concerned. If all the corresponding rank order codes between R_1 and R_2 are different, the value of d will be 1.0, which is the maximum value that d can take.

4.4.4 Advantages of rank order kernels

The advantages of rank order kernels stem from the advantages of rank order coding (see Section 4.3.5).

Robust to noise: Rank order kernels are robust to noise because small changes in the input pixel values to the kernel do not change the rank order code output of the kernel.

Fast processing: A simple sort operation on the input pixel values is enough to determine the rank order code output of each kernel. The absolute values of the pixels are not important, just their relative order. Sort operations are fast compared to most other types of operations.

Simple implementation: Implementing a rank order kernel in software or hardware is simple. It is just a sort operation.

Massively parallel: Transforming an image of size $M \times N$ requires $M \times N$ kernel operations. Each of these operations can be performed independently from the other. For this reason, the image transformation can be massively parallelized. Each kernel operation can be a process of its own. Each process will need the pixel values of the neighborhood on which the kernel operates. This neighborhood is usually small, and so very little memory will be needed for each parallel thread. Memory requirements can be an issue for parallel processing when a large amount of memory is needed for each thread.

4.5 Discussion

Images can be found in various formats. There are vector graphic images, which are represented geometrically by shapes such as curves and polygons. There are also raster graphics images (or bitmap images) which are represented with pixels placed on a square grid. There is one pixel for each coordinate value pair on the grid. Raster images can be color images, gray-scale images, or binary images. For color images each pixel has color values. There are various color models. One such model is the RGB model which specifies the red, green, and blue levels of each pixel. For gray-scale images each pixel has only one value: the intensity. For binary images, each pixel has one of two values: 0 or 1. There are many different image distance metrics depending on the type of the image [24, 25].

In our image similarity formulation we have only dealt with gray-scale raster images. That is, each image is a set of pixels each having an (x, y) coordinate and an intensity value I. For digital images, the intensity can take only a finite set of values. For an 8-bit gray-scale image for example, the intensity can have an integer value between 0 and 255. In our description of image distance using rank order kernels we assumed that the intensity can take any real value between 0.0 and 1.0. We leave the formulation of a distance metric on color images as possible future work.

4.6 Summary

In this chapter we have introduced the concept of rank order kernels. A rank order kernel is defined by its size and degree. An image distance metric can be calculated using rank order kernels. The distance value is specific to a rank order kernel of specific size and degree. In the next chapter we use rank order kernels of size 3×3 and degrees ranging from 1 to 4 in order to perform Automatic Speech Recognition.

Chapter 5

Speech Recognition

5.1 Overview

Automatic Speech Recognition systems take a speech recording as input and produce a sequence of spoken words as output. The input is in the form of a sound signal and the output can be in the form of text. In this thesis we concentrate on isolated word recognition in order to demonstrate how a speech recognition system which uses rank order kernels is highly robust to noise. For isolated word recognition, each input is a sound recording containing a single spoken word.

Our approach consists of converting each sound recording into a time-frequency image representation, or spectrogram. The spectrogram is created using Linear Predictive Coding which allows us to capture the important characteristics of speech. Each recording is therefore represented by an image. A simple nearest neighbor classification algorithm is then used for prediction. The distance metric between two images used by the classification algorithm is based on rank order kernels.

For our experiments we used our own corpus of recorded speech, as presented in Chapter 2. Each instance consists of a 2-second sound recording containing a single spoken word. A training set of recordings is chosen for building the classifier, and a test set of recordings is chosen for evaluation purposes. Our results show that by using rank order kernels a low error rate can be achieved even under high levels of added noise.

5.2 Motivation

Automatic Speech Recognition (ASR) is becoming increasingly important in applications. In the near future, voice-controlled devices will be an essential part of our lives. This is evident from the recent developments in mobile applications which are now using ASR to interface with the user. Although the problem of ASR has been studied for several decades [18,86], current ASR systems are still lacking when compared to a human's ability to recognize speech. The two main challenges for ASR are the large number of variations found in speech and the presence background noise. Speech recognition becomes even more difficult when the noise correlates to speech. Co-speaker noise for example seems to have the worst effect on ASR systems.

A spoken word can vary a great deal depending on the speaker. There are differences caused by pronunciation and in the voice of each speaker, such as the tone and fundamental frequency of the voice. Background noise is always present when performing ASR in real world applications. The accuracy of ASR systems drops dramatically when the noise levels are high [58]. It is therefore impressive that humans have the ability to recognize speech under many different conditions. This ability is only slightly affected by changes in pronunciation, voice characteristics, and noise [97].

In our attempt to make a robust speech recognition system, we take inspiration from the human auditory system and from the human brain. Sounds are spectrotemporally processed by the primary auditory cortex [105]. We therefore use a time-frequency image representation of speech, or spectrogram. Rank order coding has been presented as a plausible explanation of how the brain can efficiently process information [34]. It has the advantages of being robust to noise and it allows for fast processing. We therefore employ rank order kernels, as we have defined them in Section 4.4, in order to process the spectrograms. The use of rank order kernels allows for noise-robust processing of spectrogram images, as well as the possibility for fast massively parallel implementations in both software and hardware.

5.3 Background

The classic method for performing Automatic Speech Recognition (ASR) is by modeling speech as a discrete sequence of states. Each state is described by a probability

distribution based on short-time spectral characteristics of speech. State of the art speech recognition systems use Hidden Markov Models (HMM) to model units of speech. In isolated word recognition for example, each word can be modeled by a distinct HMM [84]. Recognition is performed by finding which HMM has the highest probability of describing an observation.

More recently, researchers have turned to using spectrogram representations for ASR [89, 99]. Such approaches have shown to be robust to noise and also have the ability to perform well even when using small training sets. The spectrogram representation allows the ASR system to capture image-based features which are not greatly affected by noise. Also, image reconstruction methods [89] can overcome the problem of missing features. Missing features can result from high levels of noise which corrupt the spectrogram. When processing images, the human brain can fill in missing parts of images which are occluded. It is reasonable therefore to consider the possibility that the human brain uses similar "reconstruction" techniques when processing sounds.

5.3.1 Classic methods

Classic Automatic Speech Recognition (ASR) systems rely on Hidden Markov Models (HMM). Most state-of-the-art ASR systems are of this type. For example, Sphinx-4 is a state-of-the-art speech recognition system based on HMMs [124]. The process used by classic ASR systems can be summarized as follows:

1. Feature extraction, using Mel-Frequency Cepstral Coefficients (MFCC)
2. Acoustic modeling, using Gaussian Mixture Models (GMM)
3. Sequence modeling, using Hidden Markov Models (HMM)
4. Recognition, using the Viterbi algorithm for search

Feature extraction

Automatic Speech Recognition (ASR) is fundamentally a pattern classification task [82]. Therefore, as in all pattern classification tasks, adequate feature extraction plays a central and crucial role. It is usually not clear which features best describe the data in order for classification to be successful. The best features are the ones which allow

for maximum discriminability between instances without sacrificing generality. In ASR, the standard feature extraction method are Mel-frequency cepstral coefficients (MFCC) [19]. To calculate the values of the features, the input sound signal is broken down into short time frames and the short-time Fourier transform is computed for each frame in order to find the spectrum. The spectrum of each frame is then passed through a Mel Filter bank which is a bank of triangular filters spaced according to the Mel-frequency scale [133]. The mel scale does not space the frequency bands linearly, but rather it attempts to model the human auditory system. Human auditory perception has a higher resolution at low frequencies than at high frequencies. The final step in the calculation of the MFCCs is a discrete cosine transform which produces a set of coefficients. Typically, only the first few coefficients are kept.

Perceptual Linear Prediction (PLP) is another method which can be used for feature extraction [5, 6, 41]. It is also inspired by the human auditory system and is sometimes used as an alternative to MFCC. An extension to PLP is RASTA-PLP [42] which attempts to remove background noise that varies slowly compared to variations in the speech signal.

Acoustic modeling

Hidden Markov Models (HMM) consist of a sequence of states. Each state is modeled by a probability distribution. In ASR this is usually modeled by Gaussians or mixtures of Gaussians [45]. This Gaussian mixture model (GMM) models the likelihood of the extracted features being generated by a given state. It is the probability of having a specific feature vector given an HMM state. Instead of GMMs, multi-layer perceptrons can also be used to calculate this probability [11].

Sequence modeling

In classic ASR systems, speech is modeled as sequence of states. A Hidden Markov Model (HMM) is a probabilistic model which consists of a sequence of "hidden" states [84]. It is a probabilistic model because transitions are possible from one state to another based on transition probabilities. In a Markov model, such as this, the probabilities of transitioning from one state to the next depend only on the current state. The transition probabilities do not depend on any of the previously-visited states. In HMMs, the states are "hidden" because they cannot be directly observed.

We cannot be sure which state is the current state. Each state however has an output, based on a probability distribution (such as the GMM described above). This output is directly observable. In the case of GMMs, this output is the log-likelihood.

Recognition

Speech is modeled as a sequence of states with certain constraints on the state transitions. The Hidden Markov Model (HMM) is what places these constraints. When performing speech recognition, the speech has to be broken down into units. For example, it can be broken down into small units, such as phonemes, or larger units, such as words. The selection of the type of unit to use is a design decision. For each unit, one HMM is trained. In isolated word speech recognition it would be reasonable to create one HMM for each word. HMMs place constraints on the acoustic features based on the training set. Further constraints can be placed during speech recognition based on a language model which can define constraints on the sequence of words.

HMMs are generative models. They define a probabilistic method for generating data. In the ASR case, the data are the feature vectors. When trying to recognize a word therefore, the recognition task is one of finding the HMM which would most likely generate the sequence of feature vectors extracted from the speech input. This is also referred to as "decoding". Given a sequence of observations one needs to estimate the underlying HMM. A search procedure finds the best HMM based on the model outputs at each time step. A dynamic programming algorithm, called the Viterbi algorithm, can be used to do this efficiently [84].

5.3.2 The problem of noise

In classic ASR systems the recognition accuracy depends on how well the probability distributions of the acoustic features match between training data and test data. This presents a problem when the test data includes noise while the training data does not. In this case, the distributions of the acoustic features in the training data and test data are different [69]. As a result, the recognition accuracy drops significantly. This problem can be mitigated by transforming either the training set distribution or the test set distribution in order to decrease the mismatch. In the first case it is called data compensation, and in the second case it is called classifier compensation [89]. Some

of the data compensation methods are CDCN [2], VTS [69], RATZ [69], and POF [72]. Classifier compensation methods include PMC [32], model composition [122], and MLLR [54].

Both the data compensation methods and classifier compensation methods assume that the noise is stationary. Based on this assumption, it follows that the acoustic feature distributions would be affected in exactly the same way, irrespective of input signal. For non-stationary noise, this does not hold, and so for non-stationary noise these two methods are not successful [87].

Human beings are able to understand speech corrupted by either stationary or non-stationary noise [58, 67]. It is also interesting to note that humans can still understand speech when it has been either high-pass or low-pass filtered with a cutoff frequency of 1800Hz [30]. In the context of this thesis however, the *capture effect* [68] exhibited by the human auditory system is the most interesting: locally more intense signal components dominate the neural response, suppressing weaker components, sometimes completely [89]. The degree parameter of the rank order kernels presented in Section 4.4.2 imitate this characteristic of the human auditory system.

5.3.3 Patterns in spectrograms

Modeling speech as a discrete sequence of states is the classic approach for Automatic Speech Recognition (ASR) systems. It has been acknowledged however, that this model presents problems [75]. Breaking down speech into a sequence of time frames and processing on frame-based features is restrictive. The human brain has amazing pattern matching capabilities. A frame-based approach restricts the possible patterns available for recognition. A spectrogram representation however, opens up a wide realm of pattern matching possibilities. A great example of this is presented by Shutte in his PhD thesis [99]. He uses a *parts-based model*, based on work in machine vision [28], to capture local patterns in small time-frequency regions of the spectrogram. These patterns can represent phonetic cues such as formant transitions, bursts in particular frequency bands, and voicing information. This is illustrated in Figure 5.1. This approach allows for a more flexible and powerful approach to pattern recognition.

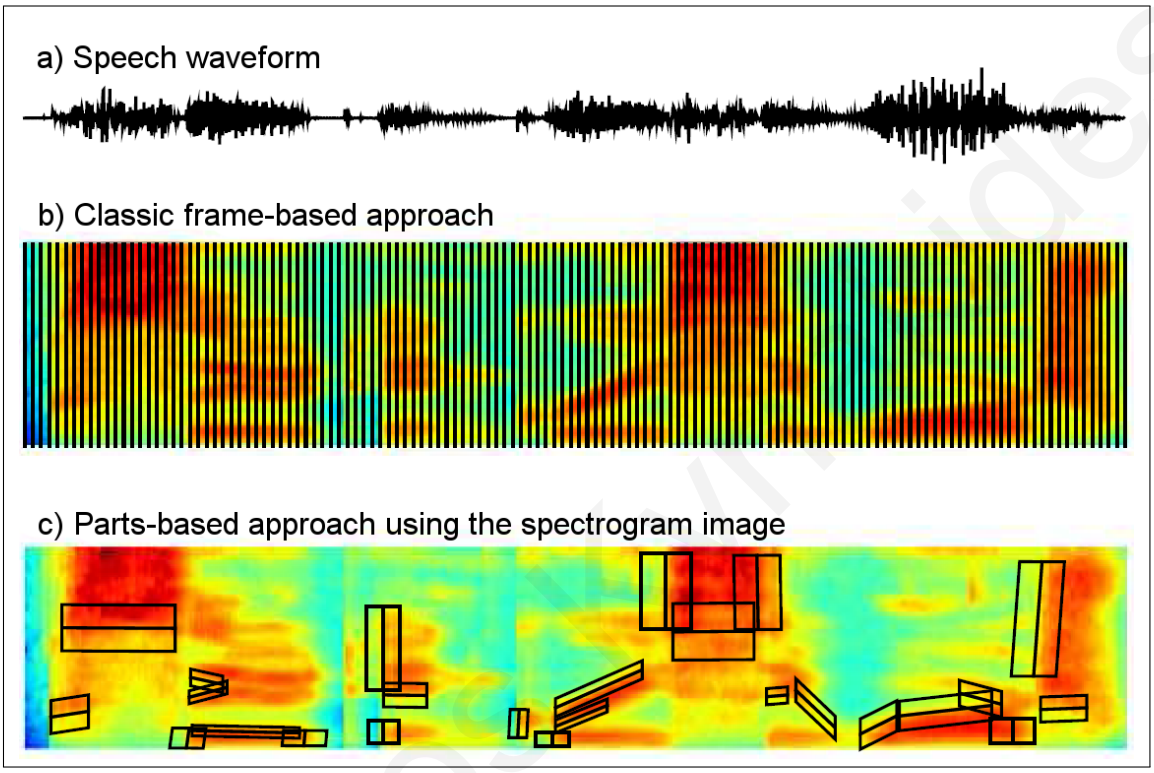

Figure 5.1: In the classic approach to speech recognition, the speech signal (sub-figure a) is broken down into a sequence of short time frames (sub-figure b). Features are then extracted individually for each time frame. In the parts-based approach however, proposed by Shutte [99], patterns are extracted from the spectrogram (sub-figure c). The parts-based approach allows for a much more flexible and powerful approach to pattern recognition. Parts of the spectrogram which represent phonetic cues, such as formant transitions, can be captured by local time-frequency patterns which are only available in the spectrogram representation. These local pattern regions are indicated with black polygons in the figure. (Figure taken from [99])

5.4 Experiments

Our approach for performing speech recognition is to use rank order kernels as described in Section 4.4. We carried out experiments using our own speech corpus of isolated word recordings. The recordings are described in Section 2.1. For the speech recognition experiments we restricted ourselves to the ten digits: "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", and "nine." It is common to restrict experiments to a small vocabulary for initial testing [60,61]. Our speech corpus contains 10 utterances of each word from each speaker. We separated our data into a training set and a test set. The training set consisted of 10 male speakers (male01-male10) and 10 female speakers (female01-female10). The test set consisted of 5 male speakers (male11-male15) and 5 female speakers (female11-female15). Therefore, the total number of training instances was 2000, and the total number of test instances was 1000. The original recordings were recorded at a sampling rate of 100kHz. For our speech recognition experiments, the recordings were down-sampled to 8kHz.

The first step was to convert each speech recording into a spectrogram representation. We found that it was advantageous to use a spectrogram representation created with Linear Predictive Coding (described in Section 3.2.5). In order to find optimal spectrogram parameters we carried out a cross-validation exercise on a separate dataset with different words. Using these parameters, we converted each recording in our training and test sets into spectrogram images. After this step therefore, the training set consisted of 2000 images, and the test set of 1000 images. Each image belongs to one of 10 classes: the 10 spoken words. In order to perform recognition, a simple nearest neighbor classification was used based on the rank order kernel distance metric. Each of the images in the test set was compared to each of the images in the training set in order to find the closest-matching image in the training set. The closest-matching image is the one with the smallest distance, based on the rank order kernel distance metric defined in Equation 4.3 on page 124. Using this method therefore, the predicted class of an image in the test set is the class of the closest-matching image in the training set.

Each of the recordings in our speech corpus has a length of 2 seconds. A single spoken word is contained within those 2 seconds. The time positions at which the speech begins and ends during those 2 seconds however, are arbitrary. This presents

a problem for our methodology because the spectrogram images of the words need be properly aligned in order to be directly comparable. It is necessary therefore to find the endpoints of the spoken word within each recording. This way, the spectrogram images will represent only the part of the recording which contains speech and so all spectrogram images will be aligned. An additional problem is that the duration of each spoken word varies. Even for the same word, different speakers may speak at different speeds. Finding the endpoints of the word solves these problems. It allows us to normalize each word in terms of time. Normalization takes place by resizing all the spectrogram images to the same width, as explained later. It is evident therefore that endpoint detection plays a crucial role in our methodology. For the following speech recognition experiments with rank order kernels we use our endpoint detection system described in Chapter 3. In all cases, the endpoint detection is carried out on noise-free recordings so that the performance of the rank order kernel methodology can be ascertained independently from the endpoint detection performance. For the endpoint detection step the recordings were down-sampled to 16kHz, whereas for the subsequent speech recognition step they were down-sampled to 8kHz.

5.4.1 Using Rank Order Kernels

Rank Order Kernels, as described in Section 4.4, require two-dimensional intensity images as input. Each of our speech recordings was therefore converted into a time-frequency representation, or spectrogram. The standard method for creating a spectrogram is to use the Short Time Fourier Transform (STFT). We found however that for spectrograms used for speech recognition it was advantageous to use Linear Predictive Coding (LPC) analysis instead. Experiments with the nearest neighbor method and a simple Mean Square distance metric showed a significant difference in error rate between spectrograms generated using STFT and spectrograms generated using LPC. These experiments using the mean square distance were completed as part of an undergraduate thesis [35]. The top two line plots in Figure 5.2 show the error rates when STFT spectrograms are used (black line plot) and when LPC spectrograms are used (red line plot). In both cases, a time window of 40ms with an overlap of 75% was used to create the spectrograms and each image was resized to 75×60 pixels. The formula for the mean square distance metric between two images

(I_1 and I_2) is shown in Equation 5.1, where N is the number of pixels in each image.

$$d = \frac{1}{N}\left[\sum_y \sum_x \Big(I_1(x,y) - I_2(x,y)\Big)^2\right] \tag{5.1}$$

For the creation of the spectrograms, certain parameters need to be decided. The spectrogram is generated from the speech signal by taking short time windows and performing a frequency analysis on each window. One parameter therefore is the size of the time window. Another parameter is the amount of overlap between successive time windows. The order used for the LPC analysis is another important parameter. The time window parameters as well as the LPC analysis resolution determine the image size of the spectrogram. The rank order kernel distance metric requires that two images have exactly the same size, in order to calculate the distance between them. For this reason, all spectrogram images are resized to a standard image size using bicubic interpolation. Once the height and width of the standard image size is decided, all images in the training set and test set have this standard size. Finally, the size and degree of the rank order kernels also needs to be decided.

In order to find the optimal values of the above parameters, a cross-validation exercise was performed using a dataset of manually-endpointed spoken words. This is the same dataset of 450 recordings that was used for the endpoint detection experiments described in Section 3.5.1 on page 82. A wide range of parameters were tried using several runs. Based on our runs, we determined that the optimal parameters for the spectrogram creation were a time window of 80ms, a window overlap of 75%, and an LPC order of 14. It is interesting that our cross-validation runs indicated that an LPC order of 14 was the optimal because it is well-established in the literature that LPC orders of 10 and 14 are well-suited for speech recognition applications. It was found that an LPC order of 10 also works well, but that an order of 14 was better. From the cross-validation runs it was concluded that a kernel size of 3×3 gave the best results and that kernel degrees above 4 did not give good results. It was also determined that the most appropriate spectrogram image size was 75×60. Table 5.1 shows the range of values tried for the different parameters during cross-validation and the optimal values found. It is interesting that the optimal value found for the the time window was 80ms. Most speech processing applications use smaller windows, but for the rank order kernels a relatively longer window is more appropriate. This may show that the rank order kernels are well-suited for capturing

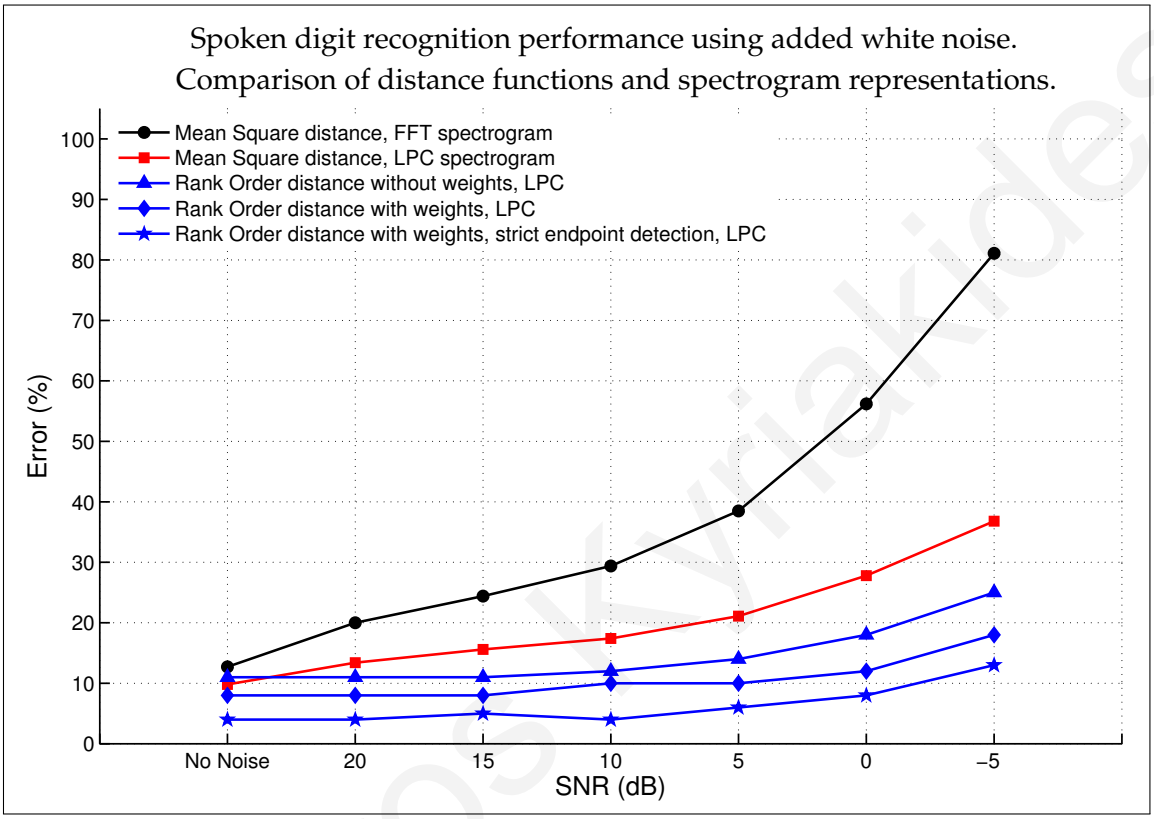

Figure 5.2: A comparison of two different image distance metrics (mean square distance and rank order kernel distance) and of two different spectrogram generation methods (FFT and LPC). Speech recognition is performed by first converting speech recordings to spectrograms. Classification is then carried out with a simple nearest neighbor algorithm using the image distance metric. The plots show how the error rate on the test set increases when various levels of white Gaussian noise is added to the test set instances. The blue plots show the results when using a rank order kernel of degree n=2. The rank order kernel distance metric is robust to high levels of noise.

Table 5.1: The range of parameter values tried in the cross validation exercise in order to find the optimal parameters. The dataset used for cross-validation consisted of manually-endpointed words and was separate from the training and test sets used in our other speech recognition experiments.

Parameter	**Range of values tried**	**Optimal value found**
LPC order	2, 4, 10, 14, 20	14
Length of time window	10, 20, 30, . . . , 130, 140, 150	80ms
Time window overlap	0, 10, 25, 75, 90	75%
Spectrogram image height	10, 15, 20, . . . , 65, 70, 75	75 pixels
Spectrogram image width	10, 15, 20, . . . , 65, 70, 75	60 pixels
Kernel height	3, 5, 7, 9	3 pixels
Kernel width	3, 5, 7, 9	3 pixels
Kernel degree	1, 2, 3, 4, 5, 6, 7	1, 2, 3, 4

features in speech which have a relatively longer time duration.

A spectrogram image was created for each of the recordings in the training set and test set by using the optimal values found from the cross-validation runs. Endpoint detection was used as a pre-processing step in order to select the section of the recording which contains speech. The spectrogram images therefore represent only the section of the recording which contains speech and not the complete 2-second recording. The selected section of input signal was then normalized to have an RMS value of 1. A nearest neighbor classification was then used to classify all the spectrograms in the test set by comparing each one to the all spectrograms in the training set. The distance metric used was the rank order kernel distance metric defined in Equation 4.3 on page 124. White noise was added at various SNRs as well as babble noise (see Section 2.2) in order to test the performance under noisy conditions. Table 5.2 shows the detailed results of the experiments. From the results it can be seen that the best performance was obtained using a rank order kernel of degree n=2. The results for added white noise with a rank order kernel of degree n=2 are shown as a line plot in Figure 5.2 (blue line plot with triangle points). For highly noisy conditions there is a significant improvement when the rank order kernel distance metric is used instead of the simple mean square distance (red line

plot).

The rank order kernel distance metric compares two spectrogram images by essentially counting how many kernels, at corresponding locations in the two images, have the same rank order code. An example of this is shown in Figure 5.3 where two spectrogram images are compared using rank order kernels. The top left image in the figure is of the word "eight" taken from the training set, from speaker "male03". The top right image in the figure is of the word "eight" taken from the test set, from speaker "male14". The rank order kernel distance between these two images is calculated as follows. For both spectrogram images, the rank order code for each 3×3 pixel area (kernel) is found by simply sorting the pixel intensity values in each pixel area. In this example we use a kernel degree of n=2. Therefore the *order* of only the top 2 pixels is considered. The bottom two images in Figure 5.3 indicate (in red) the kernel locations for which the order of the top 2 pixels is exactly the same for both spectrogram images. There are a total of 980 such matching kernels. Most of the matching kernels do not lie at time-frequency locations of high energy, but rather at locations where there are energy transitions. It is interesting to see that most of the matching kernels form areas which "envelop" the formant frequencies of the vowel sounds. This important characteristic of the speech is therefore captured by the rank order kernels without us having explicitly defined a way to consider formant frequencies. The total number of pixels in each spectrogram image is 75×60. The distance between these two specific images for rank order kernels of degree n=2 is therefore 0.7822. The calculation is shown in Equation 5.2. A smaller distance indicates that the two images are more similar. If a rank order kernel of degree n=1 is used, instead of n=2, the number of matching kernels for this example increases from 980 to 1647, giving a distance of 0.6340. When using a rank order kernel of degree n=3, the number of matching kernels for this example decreases to 642, giving a distance of 0.8573. Kernels with lower degrees are less specific and therefore give a lower distance, because a higher number of kernels match.

$$d = 1 - \frac{1}{XY}\left[\sum_{y=1}^{Y}\sum_{x=1}^{X}\left(R_1(x,y) \overset{ROC}{=\!=} R_2(x,y)\right)\right] = 1 - \frac{1}{75 \times 60} \times 980 = 0.7822 \qquad (5.2)$$

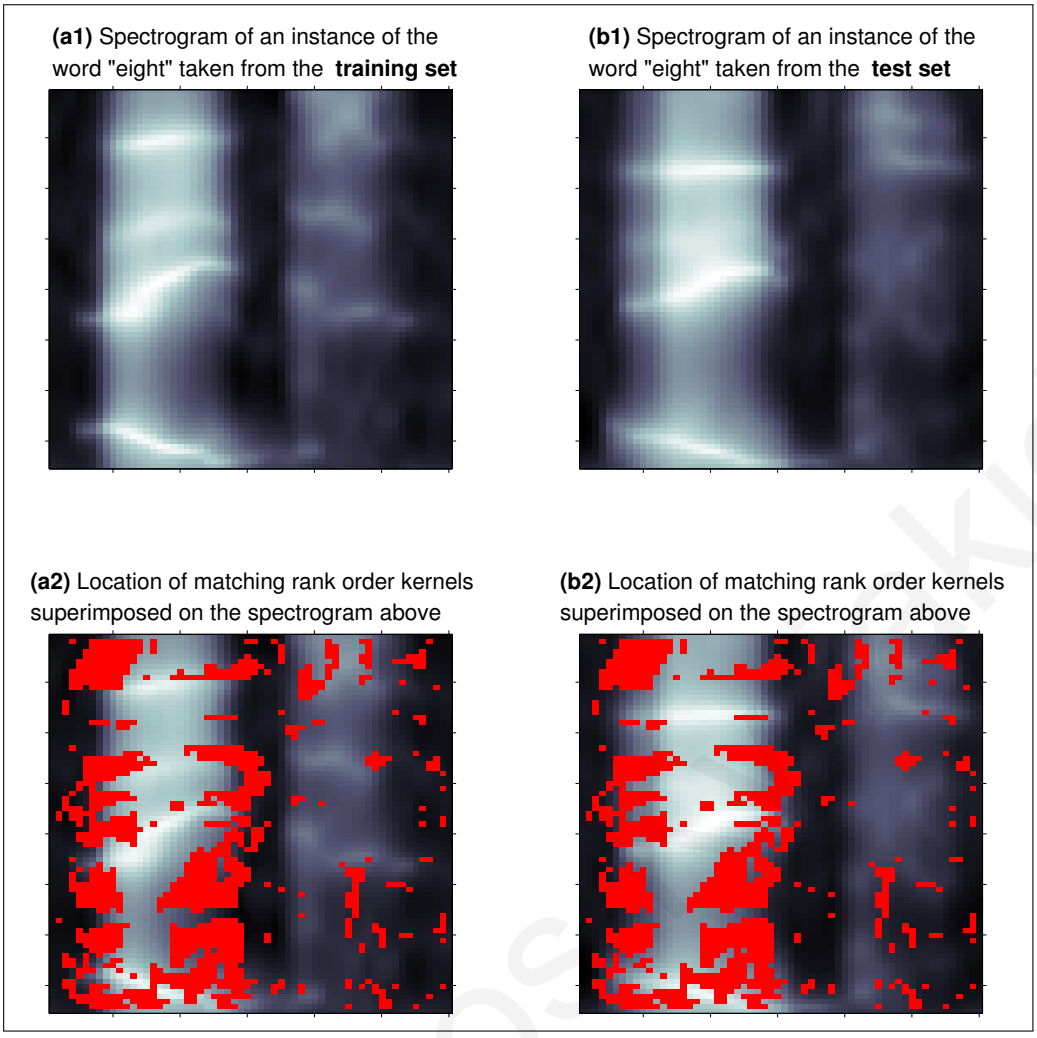

Figure 5.3: An example of how the rank order kernel distance is calculated between two spectrogram images. In this example, both spectrograms are of the word "eight". The spectrogram of an instance taken from the training set is shown in (a1). The spectrogram of an instance taken from the test set is shown in (b1). The distance is calculated by counting the number of corresponding kernel locations which have the same rank order code, based on the degree of the kernel. In this example we use a kernel degree of n=2. There are a total of 980 kernel locations which match. These locations are shown with red color in the two images at the bottom. The locations are superimposed on the training set spectrogram (a2) and on the test set spectrogram (b2) for comparison. It is interesting to see that the matching kernels form areas around the important features of the spectrogram, such as the formats of the vowels. The value of the distance between these two images is 0.7822, as shown in Equation 5.2.

Table 5.2: The performance of the rank order kernel method for speech recognition. The table shows the error rates on the test set when white gaussian noise and babble noise are added at various levels. Kernels with degree n=2 have lower error rates.

		Error (%)						
Noise	**Kernel**	**No**	**SNR**					
type	**degree**	**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
white	**n=1**	13	11	12	15	18	21	31
	n=2	11	11	11	12	14	18	25
	n=3	11	11	12	13	15	18	25
	n=4	14	13	13	14	17	20	27
babble	**n=1**	13	15	17	20	28	36	52
	n=2	11	13	15	17	22	30	46
	n=3	11	14	15	17	23	31	46
	n=4	14	16	17	20	26	34	49

5.4.2 Weights for Rank Order Kernels

When using spectrograms to discriminate between words, certain areas of the speech spectrogram are more important than other areas. In order to capture this fact, a weighting scheme was devised. We make the assumption that the most important local time-frequency regions in the spectrogram are the ones which have pixels with large differences in intensity values between them. Such areas represent features which are more distinctive. Rank order kernels can capture these features in an elegant way. When the sound signal is corrupted with noise, the rank order code is less likely to change for regions with large differences in intensity values. A simple example with numbers can make this clear. Take two pixel areas, A_1 and A_2, of size 2×2 pixels. The pixel values in A_1 are $p_1 = 0.010$, $p_2 = 0.020$, $p_3 = 0.012$, and $p_4 = 0.030$. The pixel values in A_2 are $p_1 = 0.14$, $p_2 = 0.37$, $p_3 = 0.58$, and $p_4 = 0.80$. The differences in intensity values for area A_1 are small, whereas the differences in intensity values between the different values in A_2 are large. When random noise is added to the pixels of both areas, the intensity values will change. Let's assume that after some noise is added, the new pixel values in A_1 become $p_1 = 0.030$, $p_2 = 0.015$, $p_3 = 0.013$, $p_4 = 0.012$ and in A_2 the pixel values become $p_1 = 0.05$, $p_2 = 0.29$,

Table 5.3: An example illustrating that local pixel areas with pixels which have large differences in intensity values have a rank order code which is more robust to noise than pixel areas which have small differences in intensity values. Pixel areas A_1 and A_2 of size 2×2 have 4 pixels each. The intensity values in pixel area A_1 have small differences between them whereas the pixel values in A_2 have large differences between them. When noise is added, the pixel values in both areas change. The example shows how the pixel values can change hypothetically when random white noise is added. After noise is added, the rank order code of A_1 changes, but the rank order code of A_2 remains the same.

	Before (no noise)		**After (with noise)**	
Pixel area	**pixel values**	**rank order**	**pixel values**	**rank order**
A_1	$p_1 = 0.010$	p_4	$p_1 = 0.030$	p_1
	$p_2 = 0.020$	p_2	$p_2 = 0.015$	p_2
	$p_3 = 0.012$	p_3	$p_3 = 0.013$	p_3
	$p_4 = 0.030$	p_1	$p_4 = 0.012$	p_4
A_2	$p_1 = 0.14$	p_4	$p_1 = 0.05$	p_4
	$p_2 = 0.37$	p_3	$p_2 = 0.29$	p_3
	$p_3 = 0.58$	p_2	$p_3 = 0.30$	p_2
	$p_4 = 0.80$	p_1	$p_4 = 1.00$	p_1

$p_3 = 0.30$, $p_4 = 1.00$. Before adding noise, the rank order of the pixel values in A_1 was p_4, p_2, p_3, p_1. When noise was added however, the rank order for A_1 changed to p_1, p_2, p_3, p_4. For pixel area A_2 the rank order remained p_4, p_3, p_2, p_1 both before and after the noise was added. This example is summarized in Table 5.3.

The weighting scheme we devised attempts to assign a higher weight to kernel locations which are more robust to noise. Each kernel location of a spectrogram image in the training set is assigned a weight depending on the level of noise required to change the rank order code of the kernel at that specific location. The training procedure which finds the weights for each spectrogram in the training set is the following:

1. Corrupt the input sound recording with random white noise at successive levels of noise with SNR 30dB, 20dB, 10dB, and 0dB, and create the spectrogram

in each case.

2. For each level of noise consider each kernel location and find the locations where the rank order code does not change (based on the kernel degree).

3. Initialize all kernel locations with a weight of zero.

4. For locations where the rank order code does not change for noise level 30dB, assign a weight of 1.

5. For locations where the rank order code does not change for noise level 20dB, add 1 to the weight.

6. For locations where the rank order code does not change for noise level 10dB, add 1 to the weight.

7. For locations where the rank order code does not change for noise level 0dB, add 1 to the weight.

The above procedure will produce a weight matrix $W(x, y)$ for each kernel location (x, y). The minimum possible weight is zero, which indicates kernel locations which are not robust to noise. The maximum weight is four, which indicates kernel locations which are very robust to noise. In our experiments, we carried out the above procedure 10 times for each training image, taking the average, in order to even out the randomness of the white noise. It is important to note that the weights depend on the kernel degree. Kernels with lower degree are more robust to noise, and will therefore have higher weights. In our speech recognition experiments, we used kernel degrees of 1, 2, 3, and 4. We therefore created a weight matrix for each of those degrees. In Appendix B we show the weight matrices graphically as images for ten different words from the training set spoken by "male03". We believe that this weighting scheme can generalize to many types of images, not just spectrograms, in order to designate the areas of an image with the most important features.

In the distance calculation, the weights are used as coefficients. Equation 4.3 is now modified to incorporate the weights $W(x, y)$, as shown in Equation 5.3. The weighted distance metric improves the performance of the speech recognition. The detailed results using this weighted distance metric are shown in Table 5.4. A comparison to previous results is shown in Figure 5.2 where the blue line plot with

Table 5.4: The performance of the rank order kernel method for speech recognition *with the use of weights*. The table shows the error rates on the test set when white gaussian noise and babble noise are added at various levels. Kernels with degree n=2 have lower error rates.

		Error (%)						
Noise	**Kernel**	**No**	**SNR**					
type	**degree**	**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
white	**n=1**	9	8	8	9	11	14	22
	n=2	8	8	8	10	10	12	18
	n=3	10	9	10	11	12	13	20
	n=4	10	11	11	13	13	15	24
babble	**n=1**	9	11	11	13	19	28	42
	n=2	8	9	10	12	17	22	36
	n=3	10	11	11	13	17	23	35
	n=4	10	12	12	15	18	25	37

diamond marks shows the results when weights are used for the rank order kernels. The figure shows the performance of rank order kernels with degree n=2.

$$d = 1 - \frac{1}{XY}\left[\sum_{y=1}^{Y}\sum_{x=1}^{X} W(x,y)\Big(R_1(x,y) \overset{ROC}{=\!=} R_2(x,y)\Big)\right] \tag{5.3}$$

5.4.3 Strict Endpoint Detection and Rank Order Kernels

The performance of our speech recognition algorithm greatly depends on the accuracy of the endpoint detection step. The rank order kernel distance metric compares corresponding locations of two spectrogram images. A large error in the locations of the calculated endpoints results in the shifting of the spectrogram image on the time axis as well as a change in the width of the image region which represents speech. In the experiments described previously, the endpoint detection step was performed using the endpoint detection algorithm presented in Chapter 3. In order to reduce the effect of endpoint detections which have a great deal of error, we decided to employ "strict" endpoint detection which uses cutoffs. Recordings for which the

calculated endpoints determined a highly unlikely time length were discarded. For each word, a minimum and maximum cutoff was put in place. If the length of the spoken word was below the minimum cutoff, or above the maximum cutoff, then the recording was discarded. These cutoffs were used for both the training set and test set.

To determine the cutoff values for each word, we used the distribution of the endpointed lengths of all the recordings from each word. For each word ("zero", "one", . . . , "nine") the endpoints of all the recordings, in both training and test set, were calculated. Based on the endpoints, the time length of the speech region in each recording was found. The 30th percentile of this distribution of time lengths was used as the minimum cutoff, and the 70th percentile as the maximum cutoff. Table 5.5 shows the cutoffs we used for each word. From the 2000 recordings in the training set, 1033 recordings were discarded because the speech length determined by the endpoint detection exceeded the cutoffs. From the 1000 recordings in the test set, 557 recordings were discarded because the speech length determined by the endpoint detection exceeded the cutoffs. By using the remaining recordings to carry out the speech recognition test, the results improved, indicating that a higher endpoint detection accuracy has a positive effect on the speech recognition accuracy. The detailed results based on endpoint detection with cutoffs are shown in Table 5.6. A comparison to previous results is seen in Figure 5.2 where the blue line plot with star markers shows the results based on endpoint detection with cutoffs. The plots in the figure for the rank order kernel distance metric are all for rank order kernels of degree n=2.

5.4.4 Multi-degree Voting for Rank Order Kernels

For all the previous experiments described in this section we used rank order kernels with degrees 1, 2, 3, and 4. The results we presented were for the performance of each kernel degree separately. From the results it can be seen that kernel degree n=2 has lower error rates. We now describe a method for combining several kernel degrees together in order to reduce the error rate.

When a specific kernel degree is used, the nearest neighbor algorithm finds a single spectrogram in the training set which has the smallest distance to the test set image, based on that specific kernel degree. The predicted class is the class of the

Table 5.5: The minimum and maximum cutoffs used for "strict" endpoint detection. If the calculated endpoints of a word resulted in a speech region with a time length less than the minimum cutoff or greater than the maximum cutoff, then the recording was discarded from the dataset.

Word	**Minimum cutoff (ms)**	**Maximum cutoff (ms)**
zero	682	814
one	525	683
two	472	578
three	577	683
four	630	788
five	682	841
six	787	893
seven	682	841
eight	682	841
nine	577	762

Table 5.6: The performance of the rank order kernel method for speech recognition *with the use of weights and endpoint cutoffs*. The table shows the error rates on the test set when white gaussian noise and babble noise are added at various levels. In general, kernels with degree n=2 have lower error rates.

Noise type	**Kernel degree**	**Error (%)**						
		No noise	**SNR**					
			20dB	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
white	**n=1**	5	5	5	5	7	10	17
	n=2	4	4	5	4	6	8	13
	n=3	4	4	4	5	7	10	15
	n=4	5	5	5	6	9	11	17
babble	**n=1**	5	6	10	11	11	23	36
	n=2	4	4	9	8	10	16	25
	n=3	4	4	11	8	10	15	25
	n=4	5	6	11	10	13	17	29

Table 5.7: The performance of rank order kernels on speech recognition when using multi-degree voting. The prediction is made by combining the predictions of kernels with degree 1, 2, and 3. If the three predictions do not all agree, the instance is classified as "unknown" and the prediction is considered a "miss". The results shown were obtained with kernel weights and endpoint cutoffs.

Noise		**No**	**SNR**					
type		**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
	correct(%):	92	92	82	83	83	69	54
white	wrong(%):	1	1	3	3	3	6	7
	miss(%):	7	7	15	14	14	25	39
	correct(%):	92	92	82	83	83	71	55
babble	wrong(%):	1	1	3	3	3	4	7
	miss(%):	7	7	15	14	14	26	38

training set instance and it will be either "correct" or "wrong". From our experiments we have found that kernel degrees 1, 2, and 3, give better results than kernel degree 4. If we consider kernel degrees 1, 2, and 3, we will have three predictions. One prediction for each kernel degree. We can combine these three predictions in a voting scheme so that the three predictions become one prediction. If all three predictions agree, then the final prediction will be the class of the three predictions. If not all the predictions agree, then the class will be considered as "unknown", and the prediction will be regarded as a "miss". So each prediction can now be "correct", "wrong", or "miss". This has the result of decreasing the number of errors, at the expense of also decreasing the number of correct predictions. The detailed results of this multi-degree voting method, using both weights and endpoint cutoffs, are shown in Table 5.7. The error rate ("wrong") is greatly decreased compared to the error rates reported in previous experiments which used only a single kernel degree.

5.4.5 Comparison to Sphinx

We compared the performance of our speech recognition system to the Sphinx-4 system [124], which is a state-of-the-art speech recognition system that uses Hidden Markov Models. In order to train Sphinx-4, a large collection of speech data is

needed, including text transcriptions of the sound files. For this reason, we decided to use Sphinx with models which were already trained on the TIDIGITS [55] speech corpus. The TIDIGITS speech corpus consists of 326 speakers (111 men, 114 women, 50 boys, and 51 girls) each pronouncing 77 digit sequences[1]. For these Sphinx models, the expected error rate[2] on out-of-sample spoken digits is less than 1%, when no noise is added. When we tested the Sphinx system on the 3000 utterances of spoken digits in our own speech corpus however, without adding any noise, we obtained an error rate of 16%. On average, the error rate was higher for female speakers than male speakers. One reason for the unexpectedly high error rate is probably the fact that the speakers used to create our own speech corpus had pronunciations which were significantly different than those of the speakers used to create the TIDIGITS corpus. When we inspected the error rates for each speaker individually, we found that the error rate for some speakers was significantly lower than for other speakers. We ran experiments using added white noise and babble noise. The detailed results are shown in Tables 5.8, 5.9, 5.10, and 5.11. For four specific speakers ("male03", "male10", "male14", and "male15") the error rates were close to zero when no noise was added. To compare the results of our speech recognition system with those of Sphinx we therefore decided to restrict the test set for Sphinx to use only those four speakers.

Figure 5.4 on page 154 shows the comparison of our rank order kernel method to that of Sphinx when white Gaussian noise is added at various levels. The results for Sphinx are based on a test set of 400 recordings: 10 utterances of each of the 10 digits from each of four male speakers. The results for our rank order kernel method are based on the test set we used for all our previous experiments. We used weighted rank order kernels with multi-degree voting and endpoint cutoffs. The results of our method which we used for comparison purposes are those of Table 5.7. At high SNRs the Sphinx system has a very high accuracy with almost zero wrong classifications, and zero misses. At SNR 5dB the performance of the Sphinx system starts to degrade. At SNR 0dB, the number of wrongly classified instances reaches 22.5%. When the SNR is -5dB, the Sphinx system cannot recognize most of the words

[1]Unfortunately the TIDIGITS corpus is not freely-available: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S10`

[2]For expected error rates, see: `http://cmusphinx.sourceforge.net/sphinx4/#speed_and_accuracy`

Table 5.8: Digit recognition performance of the Sphinx system on 15 male speakers when using added white noise. The counts in the table are from a test set of 100 recordings for each speaker (10 utterances of each of the 10 digits).

		No	SNR (white noise)					
Speaker		**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
	correct:	76	77	81	65	57	36	1
male01	wrong:	24	23	19	32	37	39	8
	miss:	0	0	0	3	6	25	91
	correct:	96	94	92	87	74	47	15
male02	wrong:	4	6	8	12	25	49	13
	miss:	0	0	0	1	1	4	72
	correct:	97	100	99	96	91	76	31
male03	wrong:	3	0	1	4	9	19	11
	miss:	0	0	0	0	0	5	58
	correct:	91	97	97	91	81	52	19
male04	wrong:	9	3	3	8	17	39	12
	miss:	0	0	0	1	2	9	69
	correct:	86	87	84	80	74	62	15
male05	wrong:	14	13	16	19	24	25	13
	miss:	0	0	0	1	2	13	72
	correct:	90	90	88	84	84	67	17
male06	wrong:	10	10	12	16	15	26	11
	miss:	0	0	0	0	1	7	72
	correct:	94	94	95	93	85	67	15
male07	wrong:	6	6	5	6	12	22	14
	miss:	0	0	0	1	3	11	71
	correct:	87	89	83	80	74	57	9
male08	wrong:	13	11	17	20	25	33	17
	miss:	0	0	0	0	1	10	74
	correct:	94	98	96	97	91	89	37
male09	wrong:	6	2	4	3	9	7	16
	miss:	0	0	0	0	0	4	47
	correct:	98	100	98	97	83	52	2
male10	wrong:	2	0	2	3	13	28	10
	miss:	0	0	0	0	4	20	88
	correct:	84	84	80	86	70	49	15
male11	wrong:	16	16	20	13	21	34	13
	miss:	0	0	0	1	9	17	72
	correct:	95	92	90	89	78	55	13
male12	wrong:	5	8	10	11	19	26	9
	miss:	0	0	0	0	3	19	78
	correct:	94	92	96	96	91	78	21
male13	wrong:	6	8	4	4	9	21	21
	miss:	0	0	0	0	0	1	58
	correct:	100	100	99	100	94	74	17
male14	wrong:	0	0	1	0	6	16	9
	miss:	0	0	0	0	0	10	74
	correct:	98	98	96	99	92	64	13
male15	wrong:	2	2	4	1	6	27	22
	miss:	0	0	0	0	2	9	65

Table 5.9: Digit recognition performance of the Sphinx system on 15 female speakers when using added white noise. The counts in the table are from a test set of 100 recordings for each speaker (10 utterances of each of the 10 digits).

		No	SNR (white noise)					
Speaker		**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
	correct:	87	92	93	86	67	49	5
female01	wrong:	13	8	7	14	27	29	5
	miss:	0	0	0	0	6	22	90
	correct:	67	79	71	69	52	35	6
female02	wrong:	33	21	29	29	40	43	17
	miss:	0	0	0	2	8	22	77
	correct:	81	79	76	77	68	55	15
female03	wrong:	19	21	24	22	29	36	17
	miss:	0	0	0	1	3	9	68
	correct:	75	76	71	68	56	26	0
female04	wrong:	25	24	29	32	38	54	1
	miss:	0	0	0	0	6	20	99
	correct:	71	77	74	73	61	44	7
female05	wrong:	29	23	26	27	39	44	2
	miss:	0	0	0	0	0	12	91
	correct:	77	69	66	63	51	35	12
female06	wrong:	23	31	34	35	41	45	9
	miss:	0	0	0	2	8	20	79
	correct:	73	72	69	65	48	11	1
female07	wrong:	27	28	31	35	48	23	1
	miss:	0	0	0	0	4	66	98
	correct:	76	70	70	69	70	50	8
female08	wrong:	24	30	30	31	30	47	11
	miss:	0	0	0	0	0	3	81
	correct:	72	62	62	55	46	30	1
female09	wrong:	28	38	37	44	49	38	6
	miss:	0	0	1	1	5	32	93
	correct:	81	73	73	70	55	35	4
female10	wrong:	19	27	27	30	40	43	8
	miss:	0	0	0	0	5	22	88
	correct:	78	71	73	74	54	26	2
female11	wrong:	22	29	27	25	38	45	1
	miss:	0	0	0	1	8	29	97
	correct:	74	74	72	73	68	48	8
female12	wrong:	26	26	28	27	31	48	17
	miss:	0	0	0	0	1	4	75
	correct:	68	67	58	58	47	26	1
female13	wrong:	32	33	42	42	51	38	1
	miss:	0	0	0	0	2	36	98
	correct:	77	78	78	60	52	27	1
female14	wrong:	23	22	22	40	43	49	8
	miss:	0	0	0	0	5	24	91
	correct:	78	75	77	75	76	57	3
female15	wrong:	22	25	23	25	24	37	9
	miss:	0	0	0	0	0	6	88

Table 5.10: Digit recognition performance of the Sphinx system on 15 male speakers when using added babble noise. The counts in the table are from a test set of 100 recordings for each speaker (10 utterances of each of the 10 digits).

		No	SNR (babble noise)					
Speaker		**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
	correct:	76	81	82	79	72	41	4
male01	wrong:	24	19	18	19	20	25	2
	miss:	0	0	0	2	8	34	94
	correct:	96	98	97	96	84	61	18
male02	wrong:	4	2	3	4	15	28	10
	miss:	0	0	0	0	1	11	72
	correct:	97	95	93	95	92	79	30
male03	wrong:	3	5	7	5	8	15	12
	miss:	0	0	0	0	0	6	58
	correct:	91	92	92	89	85	61	23
male04	wrong:	9	8	8	11	14	33	27
	miss:	0	0	0	0	1	6	50
	correct:	86	89	88	84	74	66	26
male05	wrong:	14	11	11	15	24	24	14
	miss:	0	0	1	1	2	10	60
	correct:	90	85	86	83	79	70	25
male06	wrong:	10	15	14	17	21	25	20
	miss:	0	0	0	0	0	5	55
	correct:	94	95	93	86	76	69	33
male07	wrong:	6	5	7	14	22	24	22
	miss:	0	0	0	0	2	7	45
	correct:	87	91	87	86	71	57	8
male08	wrong:	13	9	13	14	28	35	22
	miss:	0	0	0	0	1	8	70
	correct:	94	96	96	95	84	77	55
male09	wrong:	6	4	4	5	14	18	23
	miss:	0	0	0	0	2	5	22
	correct:	98	98	98	97	79	59	21
male10	wrong:	2	2	2	3	20	30	15
	miss:	0	0	0	0	1	11	64
	correct:	84	92	91	93	78	57	17
male11	wrong:	16	8	9	6	14	20	5
	miss:	0	0	0	1	8	23	78
	correct:	95	95	95	88	74	50	10
male12	wrong:	5	5	5	12	23	35	10
	miss:	0	0	0	0	3	15	80
	correct:	94	96	92	84	77	67	18
male13	wrong:	6	4	8	16	23	28	22
	miss:	0	0	0	0	0	5	60
	correct:	100	100	100	99	98	77	30
male14	wrong:	0	0	0	1	2	14	7
	miss:	0	0	0	0	0	9	63
	correct:	98	96	95	89	79	67	20
male15	wrong:	2	4	5	11	19	22	20
	miss:	0	0	0	0	2	11	60

Table 5.11: Digit recognition performance of the Sphinx system on 15 female speakers when using added babble noise. The counts in the table are from a test set of 100 recordings for each speaker (10 utterances of each of the 10 digits).

		No	SNR (babble noise)					
Speaker		**noise**	**20dB**	**15dB**	**10dB**	**5dB**	**0dB**	**-5dB**
	correct:	87	90	87	84	70	47	7
female01	wrong:	13	10	13	16	28	32	12
	miss:	0	0	0	0	2	21	81
	correct:	67	71	72	65	51	29	3
female02	wrong:	33	29	28	32	39	43	14
	miss:	0	0	0	3	10	28	83
	correct:	81	78	75	76	59	47	15
female03	wrong:	19	22	25	24	39	46	12
	miss:	0	0	0	0	2	7	73
	correct:	75	71	67	66	53	30	0
female04	wrong:	25	29	33	33	43	41	5
	miss:	0	0	0	1	4	29	95
	correct:	71	72	69	64	44	29	3
female05	wrong:	29	28	31	36	55	47	14
	miss:	0	0	0	0	1	24	83
	correct:	77	79	78	75	61	39	5
female06	wrong:	23	21	22	23	30	30	0
	miss:	0	0	0	2	9	31	95
	correct:	73	74	74	68	54	13	7
female07	wrong:	27	26	26	31	43	20	2
	miss:	0	0	0	1	3	67	91
	correct:	76	72	71	67	58	53	12
female08	wrong:	24	28	29	33	42	44	20
	miss:	0	0	0	0	0	3	68
	correct:	72	67	60	47	35	14	2
female09	wrong:	28	33	40	49	56	46	9
	miss:	0	0	0	4	9	40	89
	correct:	81	77	73	70	59	43	10
female10	wrong:	19	23	27	30	34	29	6
	miss:	0	0	0	0	7	28	84
	correct:	78	76	71	61	51	28	4
female11	wrong:	22	24	29	38	46	38	12
	miss:	0	0	0	1	3	34	84
	correct:	74	78	78	72	64	46	10
female12	wrong:	26	22	22	28	36	47	19
	miss:	0	0	0	0	0	7	71
	correct:	68	63	61	54	46	22	0
female13	wrong:	32	37	39	46	50	27	3
	miss:	0	0	0	0	4	51	97
	correct:	77	78	70	62	57	40	3
female14	wrong:	23	22	30	38	38	46	11
	miss:	0	0	0	0	5	14	86
	correct:	78	79	80	74	68	47	8
female15	wrong:	22	21	20	26	31	47	13
	miss:	0	0	0	0	1	6	79

and the miss rate becomes 71%, compared to the miss rate of 39% of our rank order kernel method. At this same noise level, the Sphinx system can only recognize 16% of the instances correctly, whereas our system recognizes 54% of the words correctly. We also carried out statistical significance tests using Fisher's exact test [29] to see if the performance of the rank order kernel method is significantly different than the performance of Sphinx. Figure C.1 in Appendix C shows the p-values obtained. For high levels of noise, when the SNR is 15dB or less, the performance of the rank order kernel is significantly different than the performance of Sphinx.

Figure 5.5 on page 155 shows the same comparison between the two systems, this time with added babble noise. Again, the same pattern is observed, where the Sphinx system has a better performance at low levels of noise, but at high levels of noise, our rank order kernel method has a lower miss rate, and higher accuracy.

5.5 Discussion

Using our own speech corpus of isolated words we have shown that a simple nearest neighbor classification algorithm which uses the rank order kernel distance metric can outperform even state-of-the-art speech recognition systems at high levels of noise. The outstanding performance is due to the spectrogram image representation used and to the robustness of rank order kernels. We have shown how a weighting scheme, which gives higher emphasis to more important areas of the spectrogram, improves results. Our speech recognition system relies on the direct comparison of spectrogram images. It relies on the correct alignment of the spectrograms, which itself relies on accurate endpoint detection. If the endpoints are found with greater accuracy at the pre-processing step, then the performance of our speech recognition algorithm improves. We demonstrated this by applying "strict" endpoint detection which discarded recordings which were highly likely to have been wrongly end-pointed. This is acceptable, as most speech recognition experiments in the literature are conducted with the premise that the endpoints of the input speech are known in advance, even in noisy speech recognition conditions [134]. Our goal in this chapter was to independently asses the performance of the speech recognition aspect of the rank order kernel method, without the influence of the endpoint detection accuracy.

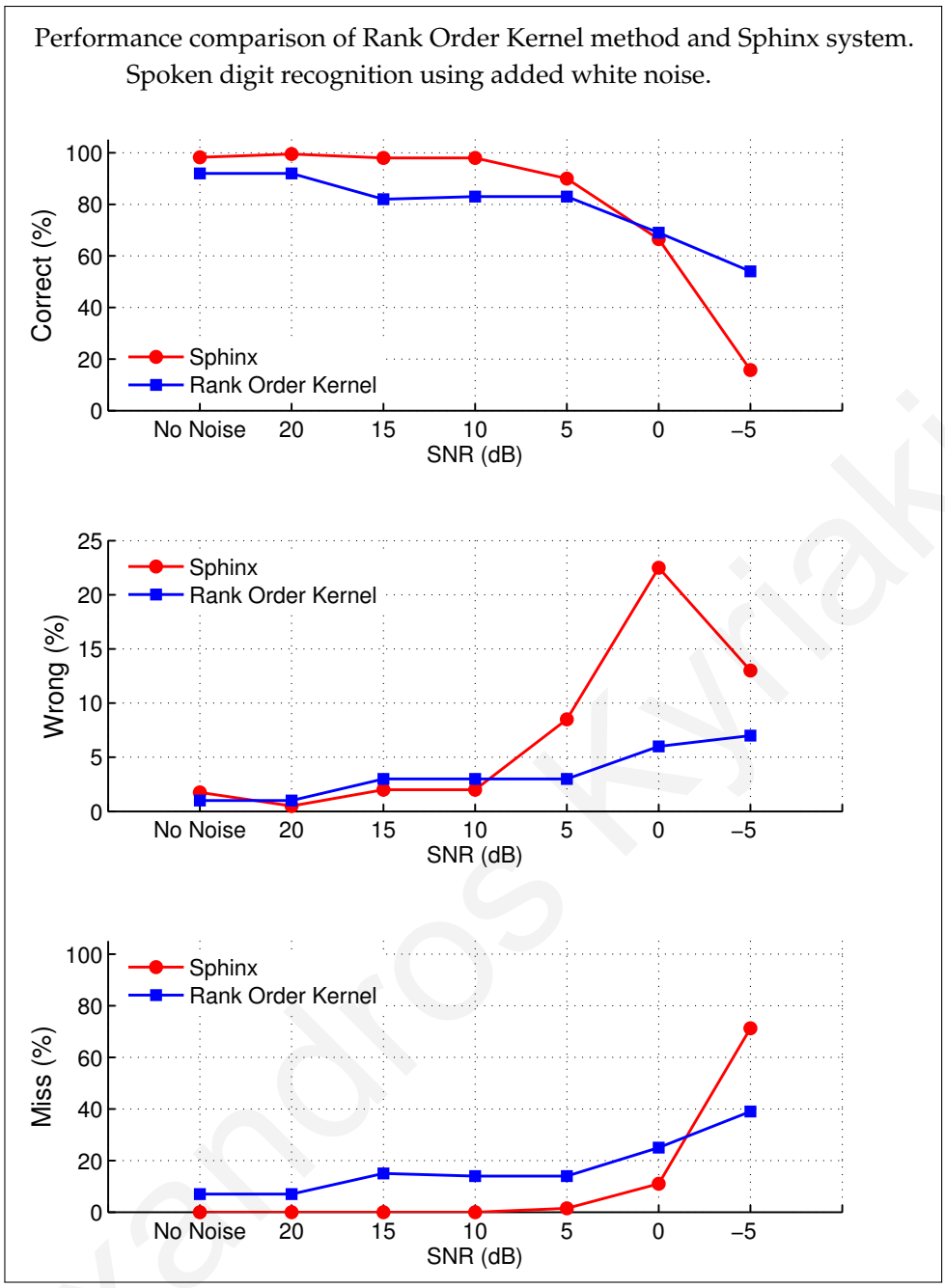

Figure 5.4: Comparison of the performance of our rank order kernel method to that of the Sphinx speech recognition system, using added white Gaussian noise. The Sphinx system has a better performance when the levels of noise are low. At high levels of noise however, the rank order kernel has an advantage. At SNR -5dB, the Sphinx system cannot recognize the words and therefore the miss rate increases significantly. At that noise level, the rank order kernel method correctly recognizes 54% of the words, whereas the Sphinx system only recognizes 16% of them correctly.

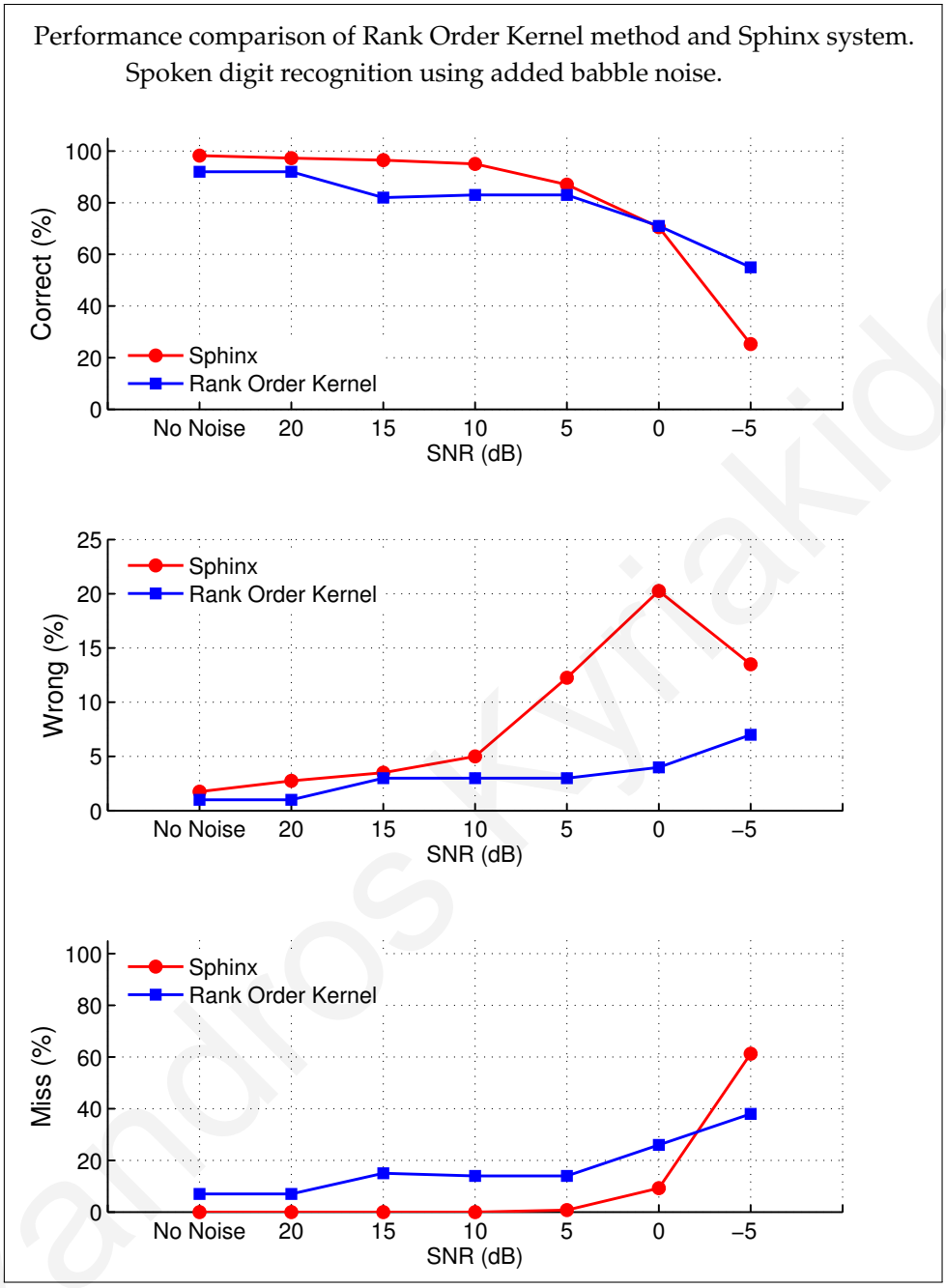

Figure 5.5: Comparison of the performance of our rank order kernel method to that of the Sphinx speech recognition system, using added babble noise. The Sphinx system has a better performance when the levels of noise are low. At high levels of noise however, the rank order kernel has an advantage. At SNR -5dB, the Sphinx system cannot recognize the words and therefore the miss rate increases significantly. At that noise level, the rank order kernel method correctly recognizes 55% of the words, whereas the Sphinx system only recognizes 25% of them correctly.

Chapter 6

Other Applications

6.1 A general framework

The endpoint detection algorithm introduced in Chapter 3 was used for finding the endpoints of spoken words in sound recordings. The rank order kernel method described in Section 4.4 was used for speech recognition and the results were presented in Chapter 5. The general framework we propose is one which transforms the input signal to a two-dimensional image representation and then applies an appropriate image processing algorithm. For example, we found that the same endpoint detection algorithm and the same rank order kernel method can also be used for other applications with data other than speech. In this short chapter we present two applications for which the endpoint detection algorithm and the rank order kernel method were successful.

6.2 Endpoint detection algorithm applied to ultrasound signals

Garreau et al. [33] have developed a method to distinguish the mode of transport for human beings (e.g. walking, running, skating, cycling). They used a micro-Doppler (mD) system to classify the mode of transport based on the time-frequency signatures obtained from ultrasound signals. Figure 6.1 shows some examples of spectrograms for four different modes of transport: walking, running, skating, cycling. Garreau et al. report accuracies as high as 97% when these spectrograms are used to predict the mode of transport. The ultrasound recordings obtained from their system however,

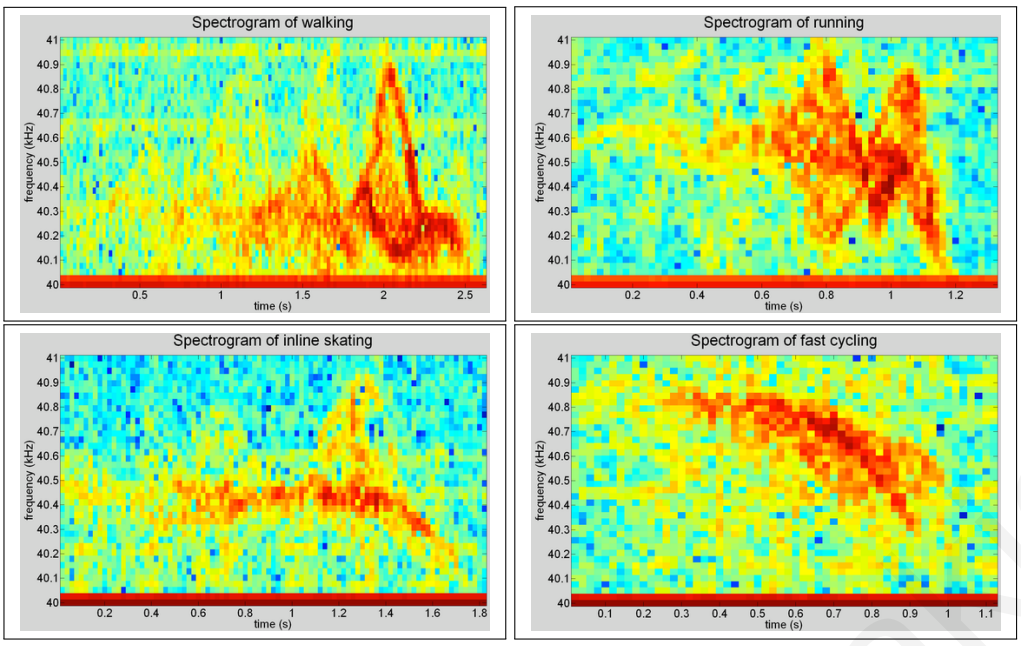

Figure 6.1: Spectrograms generated from ultrasound signals recorded from humans moving in front of the sensor under four different modes of transport. The horizontal axis shows the time in seconds and the vertical axis the frequency from 40kHz to 41kHz. Each of the four modes of transport shown (walking, running, skating, cycling) has a characteristic spectrogram which can be used to classify the mode. (Figure taken from [33].)

had a great deal of unwanted noise. It became a challenge therefore to find the locations of the mD signatures in each recording. In a recording of 12 seconds containing one mD signature the region of interest (ROI) was a small time segment of 1 to 3 seconds. This ROI was determined manually by a human expert before passing it to the classification algorithm. The spectrograms shown in Figure 6.1 show only the ROI which was selected manually for each recording.

In order to automate the detection of the ROI, we applied our variance kernel endpoint detection system. The endpoint detection system was able to detect the endpoints (start and end) of each ROI. For speech we used an LPC order of 4 and kernels of size 5×5. For the ultrasound signals we found that an LPC order of 25 and kernels of size 7×7 were more appropriate. Figure 6.2 shows an example of how the endpoint detection system located the endpoints of one of the ultrasound recordings. The top left diagram shows the waveform of an ultrasound recording of length 12 seconds. The high level of noise is readily apparent. During those 12 seconds, a human subject walked in front of the ultrasound transceiver for a few

seconds. Therefore only a short segment of the total recording represents the ROI which is needed for classification. The endpoint detection system converts the signal into a spectrogram representation using LPC analysis, then applies a variance kernel transformation, and then converts the variance image to a binary image using the automatically calculated threshold. This procedure is shown in the diagrams on the left column of the figure. The right column of the figure shows the endpoints which were automatically calculated. The diagram on the bottom right compares the spectrograms of the ROI which was automatically selected by the endpoint detection algorithm with the ROI selected manually by a human expert. The two ROIs agree.

6.3 Rank order kernels for the classification of Raman signals

The classification of Raman spectra can be very useful in a wide range of diagnostic applications including bacterial identification [52]. By being able to predict the species of the bacteria present in a urine sample of a patient for example, a medical doctor can promptly decide on a suitable course of treatment. When inexpensive equipment is used to acquire the Raman spectra however, a great deal of noise is present, making the classification task particularly challenging. An example of 90 such spectra, taken from three different species of bacteria, are shown in Figure 6.3. For classification purposes, this data was separated into a training set and a test set. The test set spectra were acquired at a later time period than the training set data. Kyriakides et al. [52] reported a high level of accuracy on the test set (87%) using support vector machines with a correlation kernel.

We decided to use the rank order kernel approach to classify this same set of Raman spectra. The spectra were each converted to a two-dimensional image representation by using segment ratios in the following way. Each spectrum has wavenumbers ranging from 300cm^{-1} to 2200cm^{-1}. We segmented each spectrum into overlapping frames of width 50cm^{-1}, using 50% overlap. This resulted in 77 frames. For each frame, we took the mean of the intensity values in that frame. A matrix of ratios was then created by taking the ratio of each mean intensity to every other mean intensity. This resulted in a two-dimensional matrix of size 77×77. We then processed this matrix as an image. The image was resized to a size of 25×75 pixels

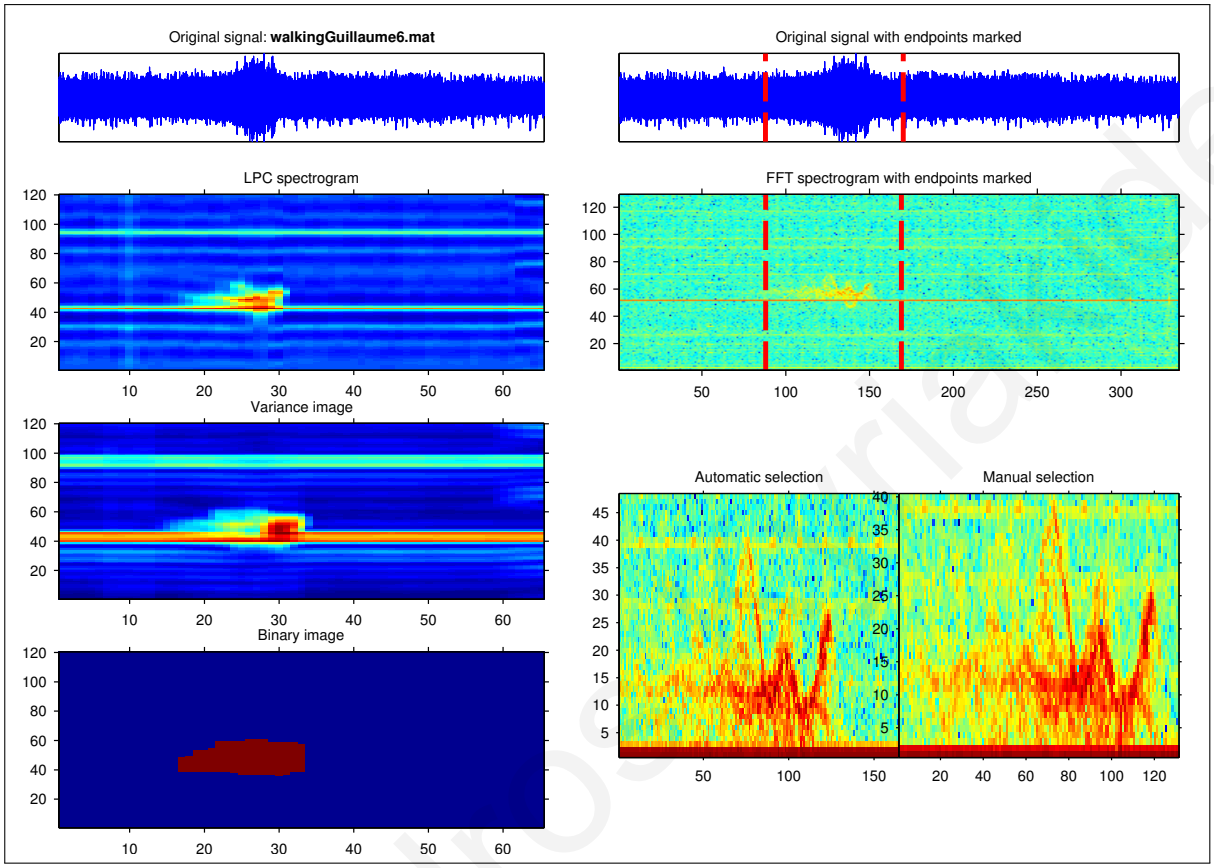

Figure 6.2: An example of our endpoint detection algorithm applied to ultrasound signals in order to find the region of interest (ROI) which represents the micro-Doppler signature of a human walking in front of the transceiver. The diagram on the top left shows the noisy input signal. The diagrams on the left column describe the endpoint detection procedure. The diagrams on the right column show the locations of the automatically calculated endpoints. The diagram on the bottom right compares the spectrograms of the ROI which was automatically selected by the endpoint detection algorithm with the ROI selected manually by a human expert.

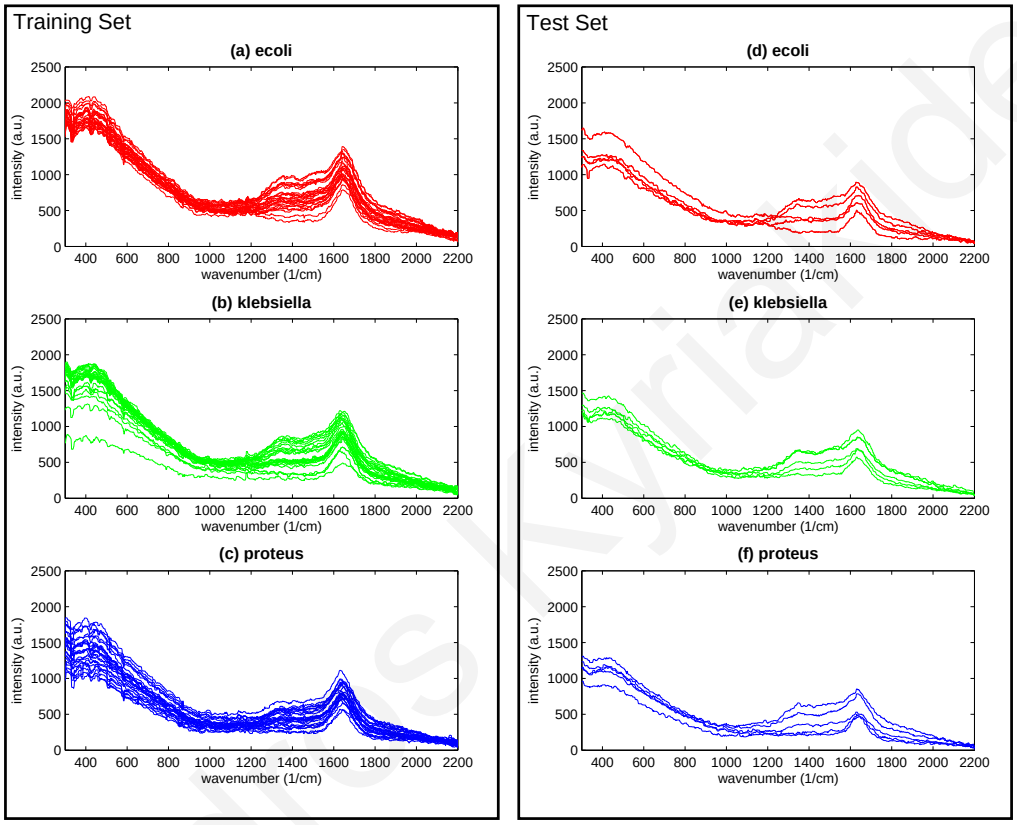

Figure 6.3: A set of 90 Raman spectra. Of these, 75 were used for training and 15 for testing. The spectra shown in the figure are obtained after median filtering was applied. In the subplots: **(a)** 25 *E.coli* spectra used for training. **(b)** 25 *Klebsiella* spectra used for training. **(c)** 25 *Proteus* spectra used for training. **(d)** 5 *E.coli* spectra used for testing. **(e)** 5 *Klebsiella* spectra used for testing. **(f)** 5 *Proteus* spectra used for testing.

Table 6.1: The actual class and predicted class of the 15 instances in the test set. Twelve of the instances were classified correctly using a nearest neighbor method with the rank order kernel distance metric.

Instance	**Actual class**	**Predicted class**
1	E.coli	E.coli
2	E.coli	E.coli
3	E.coli	Klebsiella
4	E.coli	E.coli
5	E.coli	E.coli
6	Klebsiella	Klebsiella
7	Klebsiella	Klebsiella
8	Klebsiella	Klebsiella
9	Klebsiella	Klebsiella
10	Klebsiella	Klebsiella
11	Proteus	Proteus
12	Proteus	Proteus
13	Proteus	Proteus
14	Proteus	E.coli
15	Proteus	Klebsiella

using bicubic interpolation. The 75 images in the training set were then used to classify the 15 images in the test set using a simple nearest neighbor algorithm and the rank order kernel distance metric. We found that a rank order kernel size of 7×3 and of degree n=1 gave the best results. The instances in the test set were classified with 80% accuracy (12/15). Table 6.1 shows the predictions.

Chapter 7

Conclusion

7.1 Summary

In our effort to find a noise-robust classification algorithm which will be able to process signals from various modalities, we have turned to biology for inspiration. In this thesis we have presented how an appropriate transformation of a signal to a two-dimensional image and subsequent processing with rank order kernels can lead to superior recognition performance even under high levels of noise. We have focused on the problem of automatic speech recognition (ASR) as an example of an application where our method shows its advantages. While tackling this problem, we have also devised a necessary noise-robust endpoint detection algorithm, which is also biologically-inspired. Our proposed ASR system does not aim to outperform current state-of-the-art ASR systems which have been built with years of knowledge, experience, and fine tuning. Nonetheless, we introduce a novel and noise-robust speech recognition system, which is based on models derived from biological evidence. Biologically-inspired approaches, based on pattern matching, require less training than conventional statistical approaches, which are based on Bayesian learning methods. It would be possible to combine the two approaches depending on the circumstances and on the availability of training data. This thesis presents a transformation and subsequent pattern matching which is a clear departure from the traditional cepstral-based vector features. We have shown that the biologically-inspired spectral imaging features used by our algorithm are robust to background noise.

We have proposed the use of Rank order kernels as a new way to generate

noise-resistant features. These features make the recognition system insensitive to undesired variability in the input signal and allow for discrimination even under high levels of noise. This is achieved by an image distance metric which is insensitive to changes. We have used two-dimensional image representations because, based on the literature on auditory perception, there is evidence that the auditory system generates spatio-temporal representations of the one dimensional sound signal. The principal result of our research is a noise-robust image similarity metric. We expect this similarity metric to also perform well in other applications, such as for image retrieval. Additionally, the weighting scheme we have proposed based on rank order kernels is a possible method for finding the most important parts of an image. In general, the performance of machine learning algorithms is highly dependent on the distance metric used. New proposals for distance metrics are therefore of great importance to machine learning applications.

7.2 Future Work

The classification model we have proposed is based on a two-dimensional image representation and subsequent processing with rank order kernels for finding the distance between images. For speech recognition we have restricted ourselves to a specific spectrogram representation. It would be interesting to explore new representations, such as ones based on Mel Frequency Cepstral Coefficients. The spectrogram images we have used are intensity images, where each pixel is represented by a single value. The images are essentially gray-scale images. Humans can only differ between approximately 30 different levels of gray, but they can differ between around 350,000 different colors. Color plays an important part in human visual system. Research has shown that it is advantageous to use color for image segmentation and pattern matching [27]. It would therefore be worthwhile to attempt to incorporate color information in the two-dimensional image representations processed by the rank order kernel.

In our experiments, we have used a simple nearest neighbor algorithm. Other, more powerful classification algorithms exist, such as Support Vector Machines (SVM), which might be able to further improve the results. An SVM model classifies future instances by creating a discrimination boundary between classes in the feature space. SVMs have good generalization ability because they maximize the separating

margin between the classes. Only a selected number of the instances in the training set are used to define the separating margin. These are called support vectors. Our speech recognition system would greatly benefit from an SVM model which uses a small number of support vectors because during the recognition step it would just need to compare the instance to be classified with the support vectors only and not with all the instances in the training set, which is the case with the nearest neighbor algorithm.

In this thesis, we used an empirical method to find weights for the kernels. An analytical method based on probability and the statistics of noise would be another approach. The weight of each kernel would be proportional to the probability that the output of a kernel (rank order code) changes, based on the statistics of the noise.

The weighting scheme we have proposed finds the important areas in the spectrogram image. It would be interesting to apply the same weighting scheme to other types of images, for example natural images, to see if areas with high weights also indicate important areas in those images.

It would be beneficial to implement rank order kernels in hardware, for faster processing. The software implementation we used in this thesis, "slides" the kernel around the input image in order to process each location one after the other, in a sequential manner. In hardware however, each kernel can form an independent hardware unit, and therefore all kernels can process information simultaneously and in parallel, greatly improving performance.

It could be beneficial to use a hybrid approach to speech recognition where the classic speech recognition algorithms are used when the noise levels are low, and then a switch-over to our algorithm is made when the noise levels are high.

7.3 Contributions

1. We have introduced rank order kernels. Rank order kernels use rank order coding and are defined by a kernel size and a degree. Rank order coding is not a new concept. Our formulation, implementation, and application of rank order kernels however, is completely novel.

2. We have devised a weighting scheme which can be used for rank order kernels. Locations with high weight are ones with rank order codes which are

insensitive to noise. It is possible that this weighting method can be used as a general method to pick out the important areas of images.

3. We have developed an endpoint detection algorithm to separate speech from non-speech. The algorithm uses an appropriate spectrogram representation of sound and makes the assumption that high-variance regions of the spectrogram contain speech. We have shown that our algorithm has very good performance even with high levels of added background noise.

4. We have created our own speech corpus of isolated words by recording 100 different words from 15 male and 15 female speakers. Each word was spoken 10 times by each speaker. This corpus was used for our experiments.

5. We have shown that Automatic Speech Recognition (ASR) of isolated words can be performed using a distance metric for images based on rank order kernels. Our experiments show that our method outperforms state-of-the-art speech recognition systems at high levels of noise.

6. We have presented two other applications, not related to speech processing, which show that our endpoint detection system and rank order kernel distance metric can possibly be applied to many types of applications.

7. Our most important contribution is the introduction of an image similarity metric, or distance metric, based on rank order kernels. This metric can be used in image processing applications and in machine learning algorithms.

Bibliography

[1] W. Abdulla, V. Kecman, and N. Kasabov, "Speech-background classification by using SVM technique," in *Proc. of Artificial Neural Networks and Neural Information Processing ICANN/ICONIP 2003 International Conference. Istanbul, Turkey*, 2003.

[2] A. Acero, *Acoustical and environmental robustness in automatic speech recognition*. Springer, 1993, vol. 201.

[3] A. Acero, C. Crespo, C. Torre, and J. Torrecilla, "Robust HMM-based endpoint detector," in *Third European Conference on Speech Communication and Technology*, 1993.

[4] S. Al-Haddad, S. Samad, A. Hussein, K. Ishak, A. Azid, R. Ghaffar, D. Ramli, M. Zainal, and M. Abdullah, "Automatic segmentation and labeling for continuous number recognition," *WSEAS Transactions on Signal Processing*, 2006.

[5] V. Atti, "Algorithms and Software for Predictive and Perceptual Modeling of Speech," *Synthesis Lectures on Algorithms and Software Engineering*, vol. 2, no. 1, pp. 1–119, 2011.

[6] V. Atti and A. Spanias, "Speech analysis by estimating perceptually relevant pole locations," in *Proc. IEEE ICASSP*, vol. 5, 2005, pp. 217–220.

[7] M. Bear, B. Connors, and M. Paradiso, *Neuroscience: Exploring the brain*. Lippincott Williams & Wilkins, 2007.

[8] A. Benyassine, E. Shlomot, H. Su, D. Massaloux, C. Lamblin, and J. Petit, "ITU-T Recommendation G. 729 Annex B: a silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications," *Communications Magazine, IEEE*, vol. 35, no. 9, pp. 64–73, 1997.

[9] F. Beritelli, S. Casale, G. Ruggeri, and S. Serrano, "Performance evaluation and comparison of G. 729/AMR/fuzzy voice activity detectors," *Signal processing letters, IEEE*, vol. 9, no. 3, pp. 85–88, 2002.

[10] B. Bhattacharya and S. Furber, "Biologically inspired means for rank-order encoding images: A quantitative analysis," *Neural Networks, IEEE Transactions on*, vol. 21, no. 7, pp. 1087–1099, 2010.

[11] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994.

[12] J. Bradbury, "Linear predictive coding," *Florida Institute of Technology. http://my.fit.edu/~vkepuska/ece5525/lpc_paper.pdf (2/22/11)*.

[13] J. Chen, Y. Lin, and R. Cox, "A fixed-point 16 kb/s LD-CELP algorithm," in *icassp*. IEEE, 1991, pp. 21–24.

[14] R. Chengalvarayan, "Robust energy normalization using speech/nonspeech discriminator for German connected digit recognition," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[15] Y. Cho and A. Kondoz, "Analysis and improvement of a statistical model-based voice activity detector," *Signal Processing Letters, IEEE*, vol. 8, no. 10, pp. 276–278, 2001.

[16] R. Christopher deCharms, D. Blake, and M. Merzenich, "Optimizing sound features for cortical neurons," *Science*, vol. 280, no. 5368, pp. 1439–1444, 1998.

[17] B. Cox and L. Timothy, "Nonparametric rank-order statistics applied to robust voiced-unvoiced-silence classification," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 5, pp. 550–561, 1980.

[18] K. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits," *The Journal of the Acoustical Society of America*, vol. 24, p. 637, 1952.

[19] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[20] A. Delorme, J. Gautrais, R. Van Rullen, and S. Thorpe, "SpikeNET: A simulator for modeling large networks of integrate and fire neurons," *Neurocomputing*, vol. 26, pp. 989–996, 1999.

[21] A. Delorme and S. Thorpe, "Face identification using one spike per neuron: resistance to image degradations," *Neural Networks*, vol. 14, no. 6-7, pp. 795–803, 2001.

[22] ——, "SpikeNET: an event-driven simulation package for modelling large networks of spiking neurons," *Network: Computation in Neural Systems*, vol. 14, no. 4, pp. 613–627, 2003.

[23] L. Deng and D. O'Shaughnessy, *Speech processing: a dynamic and optimization-oriented approach*. CRC, 2003, vol. 17.

[24] M. Deza and E. Deza, *Encyclopedia of distances*. Springer Verlag, 2009.

[25] V. Di Gesú and V. Starovoitov, "Distance-based functions for image comparison," *Pattern Recognition Letters*, vol. 20, no. 2, pp. 207–214, 1999.

[26] E. Elia, "Speech Analysis for the Recognition of Words which can be used for Voice Control," May 2009.

[27] R. Etienne-Cummings, P. Pouliquen, and M. Lewis, "A vision chip for color segmentation and pattern matching," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 703–712, 2003.

[28] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[29] S. Fisher, *Statistical methods for research workers*. Genesis Publishing Pvt Ltd, 1932, no. 5.

[30] H. Fletcher, "Speech and hearing in communication." 1953.

[31] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[32] M. Gales and S. Young, "HMM recognition in noise using parallel model combination," in *Third European Conference on Speech Communication and Technology*, 1993.

[33] G. Garreau, N. Nicolaou, C. Andreou, C. D'Urbal, G. Stuarts, and J. Georgiou, "Computationally efficient classification of human transport mode using micro-doppler signatures," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*. IEEE, 2011, pp. 1–4.

[34] J. Gautrais and S. Thorpe, "Rate coding versus temporal order coding: a theoretical approach," *Biosystems*, vol. 48, no. 1-3, pp. 57–65, 1998.

[35] E. Giannarou, "Word recognition in voice signals based on time-frequency features," May 2012.

[36] J. Górriz, J. Ramírez, and C. Puntonet, "New Advances in Voice Activity Detection using HOS and Optimization Strategies," *Robust Speech Recognition and Understanding*, p. 460.

[37] L. Granai, "Nonlinear approximation with redundant multi-component dictionaries," Ph.D. dissertation, Institut de traitement des signaux Section de Génie Électrique et Électronique pour l'obtention du grade de docteur ès sciences par Laurea in Ingegneria delle Telecommunicazioni, Università degli Studi di Siena, 2006.

[38] L. Gu, J. Gao, and A. Harris, "Endpoint detection in noisy environment using a Poincare recurrence metric," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–428.

[39] J. Haigh and J. Mason, "Robust voice activity detection using cepstral features," in *TENCON'93. Proceedings. Computer, Communication, Control and Power Engineering. 1993 IEEE Region 10 Conference on*. IEEE, 1993, pp. 321–324.

[40] J. Hawkins and S. Blakeslee, *On intelligence*. Owl Books, 2005.

[41] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, p. 1738, 1990.

[42] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP speech analysis technique," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 121–124.

[43] J. Hoyt and H. Wechsler, "Detection of human speech in structured noise," in *icassp*. IEEE, 1994, pp. 237–240.

[44] C. Jia and B. Xu, "An improved entropy-based endpoint detection algorithm," in *International Symposium on Chinese Spoken Language Processing*, 2002.

[45] B. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of markov chains (Corresp.)," *Information Theory, IEEE Transactions on*, vol. 32, no. 2, pp. 307–309, 1986.

[46] J. Junqua, "Robustness and cooperative multimodal man-machine communication applications," in *The Structure of Multimodal Dialogue; Second VENACO Workshop*, 1991.

[47] J. Junqua, B. Mak, and B. Reaves, "A robust algorithm for word boundary detection in the presence of noise," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 3, pp. 406–412, 1994.

[48] J. Junqua, B. Reaves, and B. Mak, "A study of endpoint detection algorithms in adverse conditions: incidence on a DTW and HMM recognizer," in *Second European Conference on Speech Communication and Technology*, 1991.

[49] Y. Ke, D. Hoiem, and R. Sukthankar, "Computer Vision for Music Identification," in *CVPR*. IEEE Computer Society, 2005, pp. 597–604. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.105

[50] B. Kingsbury, "Perceptually inspired signal-processing strategies for robust speech recognition in reverberant environments," Ph.D. dissertation, UNIVERSITY of CALIFORNIA, 1998.

[51] S. Kullback, *Information theory and statistics*. Dover Pubns, 1997.

[52] A. Kyriakides, E. Kastanos, K. Hadjigeorgiou, and C. Pitris, "Classification of Raman spectra using the correlation kernel," *Journal of Raman Spectroscopy*, vol. 42, no. 5, pp. 904–909, 2011.

[53] L. Lamel, L. Rabiner, A. Rosenberg, and J. Wilpon, "An improved endpoint detector for isolated word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 4, pp. 777–785, 1981.

[54] C. Leggetter and P. Woodland, "Speaker adaptation of HMMs using linear regression," *Cambridge University, Cambridge, UK, Tech. Rep. CUED/F-INFENG/TR*, vol. 181, 1994.

[55] R. Leonard and G. Doddington, "TIDIGITS speech corpus," *Texas Instruments, Inc*, 1993.

[56] Q. Li, J. Zheng, A. Tsai, and Q. Zhou, "Robust endpoint detection and energy normalization for real-time speech and speaker recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, no. 3, pp. 146–157, 2002.

[57] A. Lipeika and J. Lipeikienė, "Word Endpoint Detection Using Dynamic Programming," *Informatica*, vol. 14, no. 4, pp. 487–496, 2003.

[58] R. Lippmann, "Speech recognition by machines and humans," *Speech communication*, vol. 22, no. 1, pp. 1–15, 1997.

[59] S. Loiselle, J. Rouat, D. Pressnitzer, and S. Thorpe, "Exploration of rank order coding with spiking neural networks for speech recognition," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 4. IEEE, 2005, pp. 2076–2080.

[60] P. Loizou and A. Spanias, "Improving discrimination of confusable words using the divergence measure," *The Journal of the Acoustical Society of America*, vol. 101, p. 1106, 1997.

[61] ——, "High-performance alphabet recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 6, pp. 430–445, 1996.

[62] J. Machaj, P. Brida, and R. Piche, "Rank based fingerprinting algorithm for indoor positioning," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*. IEEE, 2011, pp. 1–6.

[63] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised Dictionary Learning," in *NIPS*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2008, pp. 1033–1040.

[64] M. Markaki, A. Karpov, E. Apostolopoulos, M. Astrinaki, Y. Stylianou, and A. Ronzhin, "A hybrid system for Audio segmentation and speech-endpoint detection of broadcast news." SPECOM, 2007.

[65] J. Markel, "The SIFT algorithm for fundamental frequency estimation," *Audio and Electroacoustics, IEEE Transactions on*, vol. 20, no. 5, pp. 367–377, 1972.

[66] M. Marzinzik and B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, no. 2, pp. 109–118, 2002.

[67] G. Miller and J. Licklider, "The intelligibility of interrupted speech," *The Journal of the Acoustical Society of America*, vol. 22, p. 167, 1950.

[68] B. Moore, "Introduction to the psychology of hearing." 1977.

[69] P. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon University, 1996.

[70] J. Navarro-Mesa, A. Moreno-Bilbao, and E. Lleida-Solano, "An improved speech endpoint detection system in noisy environments by means of third-order spectra," *Signal Processing Letters, IEEE*, vol. 6, no. 9, pp. 224–226, 1999.

[71] E. Nemer, R. Goubran, and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 217–231, 2001.

[72] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. 1. IEEE, 1994, pp. I–417.

[73] H. Ney, "An optimization algorithm for determining the endpoints of isolated utterances," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'81.*, vol. 6. IEEE, 1981, pp. 720–723.

[74] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[75] M. Ostendorf, "Moving beyond the "beads-on-a-string" model of speech," in *Proc. IEEE ASRU Workshop*, 1999, pp. 79–84.

[76] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, pp. 62–66, Mar. 1979, minimize intra and inter class variance.

[77] C. Papageorgiou, M. Oren, and T. Poggio, "A General Framework for Object Detection," in *ICCV*, 1998, pp. 555–562.

[78] B. Pinkowski, "LPC spectral moments for clustering acoustic transients," *Speech and Audio Processing, IEEE Transactions on*, vol. 1, no. 3, pp. 362–368, 1993.

[79] Y. Qi and B. Hunt, "Voiced-unvoiced-silence classifications of speech using hybrid features and a network classifier," *Speech and Audio Processing, IEEE Transactions on*, vol. 1, no. 2, pp. 250–255, 1993.

[80] H. Qiang and Z. Youwei, "On prefiltering and endpoint detection of speech signal," in *Signal Processing Proceedings, 1998. ICSP'98. 1998 Fourth International Conference on*. IEEE, 1998, pp. 749–752.

[81] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 5, pp. 399–418, 1976.

[82] L. Rabiner and B. Juang, *Fundamentals of speech recognition*. Prentice Hall, 1993.

[83] L. Rabiner and M. Sambur, "Voiced-unvoiced-silence detection using the Itakura LPC distance measure," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'77.*, vol. 2. IEEE, 1977, pp. 323–326.

[84] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[85] L. Rabiner and M. Sambur, "An algorithm for determining the endpoints of isolated utterances," *The Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.

[86] L. Rabiner and R. Schafer, *Digital processing of speech signals*. Prentice-hall Englewood Cliffs, NJ, 1978, vol. 100.

[87] B. Raj, V. Parikh, and R. Stern, "The effects of background music on speech recognition accuracy," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 851–854.

[88] B. Raj and R. Singh, "Classifier-based non-linear projection for adaptive end-pointing of continuous speech," *Computer Speech & Language*, vol. 17, no. 1, pp. 5–26, 2003.

[89] B. Ramakrishnan, "Reconstruction of incomplete spectrograms for robust speech recognition," Ph.D. dissertation, Carnegie Mellon University, 2000.

[90] J. Ramirez, J. Górriz, and J. Segura, "Voice activity detection. Fundamentals and speech recognition system robustness," *Robust Speech Recognition and Understanding*, pp. 1–22, 2007.

[91] J. Ramirez, J. Segura, C. Benitez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[92] J. Ramírez, J. Segura, C. Benítez, Á. de la Torre, and A. Rubio, "An effective subband OSF-based VAD with noise reduction for robust speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 6, pp. 1119–1129, 2005.

[93] J. Ramírez, J. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *Signal Processing Letters, IEEE*, vol. 12, no. 10, pp. 689–692, 2005.

[94] A. Restrepo, G. Hincapie, and A. Parra, "On the detection of edges using order statistic filters," in *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, vol. 1. IEEE, 1994, pp. 308–312.

[95] J. Rouat, S. Loiselle, and R. Pichevar, "Acoustic Representation and Processing: It is time!"

[96] M. Savoji, "A robust algorithm for accurate endpointing of speech signals," *Speech Communication*, vol. 8, no. 1, pp. 45–60, 1989.

[97] O. Scharenborg, "Reaching over the gap: A review of efforts to link human and automatic speech recognition research," *Speech Communication*, vol. 49, no. 5, pp. 336–347, 2007.

[98] M. Schönle and V. Gilg, "An Algorithm For Cancellation Of Sidetone Oscillations."

[99] K. Schutte, "Parts-based Models and Local Features for Automatic Speech Recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[100] N. Seman, Z. Bakar, N. Bakar, H. Mohamed, N. Abdullah, P. Ramakrisnan, and S. Ahmad, "Evaluating endpoint detection algorithms for isolated word from Malay parliamentary speech," in *Information Retrieval & Knowledge Management,(CAMP), 2010 International Conference on*. IEEE, pp. 291–296.

[101] B. Sen and S. Furber, "Information recovery from rank-order encoded images," *Proceedings of Biologically Inspired Information Fusion*, pp. 8–13, 2006.

[102] ——, "Evaluating rank-order code performance using a biologically-derived retinal model," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 2867–2874.

[103] J. Shen, J. Hung, and L. Lee, "Robust entropy-based endpoint detection for speech recognition in noisy environments," in *Fifth International Conference on Spoken Language Processing*, 1998.

[104] M. Sima, D. Burileanu, C. Burileanu, and V. Croitoru, "Full-Custom Software for Start/End Point Detection of Isolated-Spoken Words," in *Proceedings of the 12th International Conference on Control System and Computer Science, Bucharest*, vol. 2, 1999, pp. 19–24.

[105] W. Smit and E. Barnard, "Efficient coding leads to novel features for speech recognition," in *Fifteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2004, p. 99.

[106] ——, "Continuous speech recognition with sparse coding," *Computer Speech & Language*, vol. 23, no. 2, pp. 200–219, 2009.

[107] J. Sohn, N. Kim, and W. Sung, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, 1999.

[108] H. Steeneken and F. Geurtsen, "Description of the RSG. 10 noise database," *report IZF*, vol. 3, 1988.

[109] A. Syrdal, R. Bennett, and S. Greenspan, *Applied speech technology*. CRC, 1995.

[110] S. Tanyer and H. Ozer, "Voice activity detection in nonstationary noise," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 478–482, 2000.

[111] J. Thomas, "Nonparametric detection," *Proceedings of the IEEE*, vol. 58, no. 5, pp. 623–631, 1970.

[112] S. Thorpe, "Ultra-rapid scene categorization with a wave of spikes," in *Biologically Motivated Computer Vision*. Springer, 2002, pp. 335–351.

[113] S. Thorpe, A. Delorme, and R. Van Rullen, "Spike-based strategies for rapid processing," *Neural networks*, vol. 14, no. 6-7, pp. 715–725, 2001.

[114] S. Thorpe, R. Guyonneau, N. Guilbaud, J. Allegraud, and R. VanRullen, "SpikeNet: Real-time visual processing with one spike per neuron," *Neurocomputing*, vol. 58, pp. 857–864, 2004.

[115] R. Tucker, "Voice activity detection using a periodicity measure," in *Communications, Speech and Vision, IEE Proceedings I*, vol. 139. IET, 1992, pp. 377–380.

[116] I. Uysal, H. Sathyendra, and J. Harris, "A biologically plausible system approach for noise robust vowel recognition," in *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, vol. 1. IEEE, 2006, pp. 245–249.

[117] ——, "Spike-based feature extraction for noise robust speech recognition using phase synchrony coding," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1529–1532.

[118] ——, "Towards Spike-Based Speech Processing: A Biologically Plausible Approach to Simple Acoustic Classification," *International Journal of Applied Mathematics and Computer Science*, vol. 18, no. 2, pp. 129–137, 2008.

[119] R. Van Rullen, J. Gautrais, A. Delorme, and S. Thorpe, "Face processing using one spike per neurone," *Biosystems*, vol. 48, no. 1-3, pp. 229–239, 1998.

[120] R. VanRullen, R. Guyonneau, and S. Thorpe, "Spike times make sense," *Trends in neurosciences*, vol. 28, no. 1, pp. 1–4, 2005.

[121] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[122] A. Varga and R. Moore, "Hidden Markov model decomposition of speech and noise," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 845–848.

[123] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. CVPR*, vol. 1, pp. 511–518, 2001.

[124] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," *Sun Microsystems, Inc. Mountain View, CA, USA*, p. 18, 2004.

[125] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 4, p. 20, 2010.

[126] B. Wells, "Voiced/unvoiced decision based on the bispectrum," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, vol. 10. IEEE, 1985, pp. 1589–1592.

[127] J. G. Wilpon, L. R. Rabiner, and T. Martin, "An Improved Word-Detection Algorithm for Telephone-Quality Speech Incorporating Both Syntactic and Semantic Constraints," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 3, pp. 479–498, Mar., 1984.

[128] J. Wilpon and L. Rabiner, "Application of hidden Markov models to automatic speech endpoint detection," *Computer Speech & Language*, vol. 2, no. 3-4, pp. 321–341, 1987.

[129] K. Woo, T. Yang, K. Park, and C. Lee, "Robust voice activity detection algorithm for estimating noise spectrum," *Electronics Letters*, vol. 36, no. 2, pp. 180–181, 2000.

[130] B. Wu and K. Wang, "Robust endpoint detection algorithm based on the adaptive band-partitioning spectral entropy in adverse environments," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 762–775, 2005.

[131] D. Wu, M. Tanaka, R. Chen, L. Olorenshaw, M. Amador, and X. Menendez-Pidal, "A robust speech detection algorithm for speech activated hands-free applications," in *Acoustics, Speech, and Signal Processing, 1999. ICASSP'99. Proceedings., 1999 IEEE International Conference on*, vol. 4. IEEE, 1999, pp. 2407–2410.

[132] G. Wu and C. Lin, "Word boundary detection with mel-scale frequency bank in noisy environment," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 5, pp. 541–554, 2000.

[133] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The HTK book (for HTK version 3.4)," 2006.

[134] Z. Zhang and S. Furui, "Noisy speech recognition based on robust end-point detection and model adaptation," in *Proc. ICASSP*, 2005, pp. 981–984.

[135] J. Zhu and F. Chen, "The Analysis and Application of a New Endpoint Detection Method Based on Distance of Autocorrelated Similarity," in *Sixth European Conference on Speech Communication and Technology*, 1999.

Appendix A

Complete Endpoint Detection Results

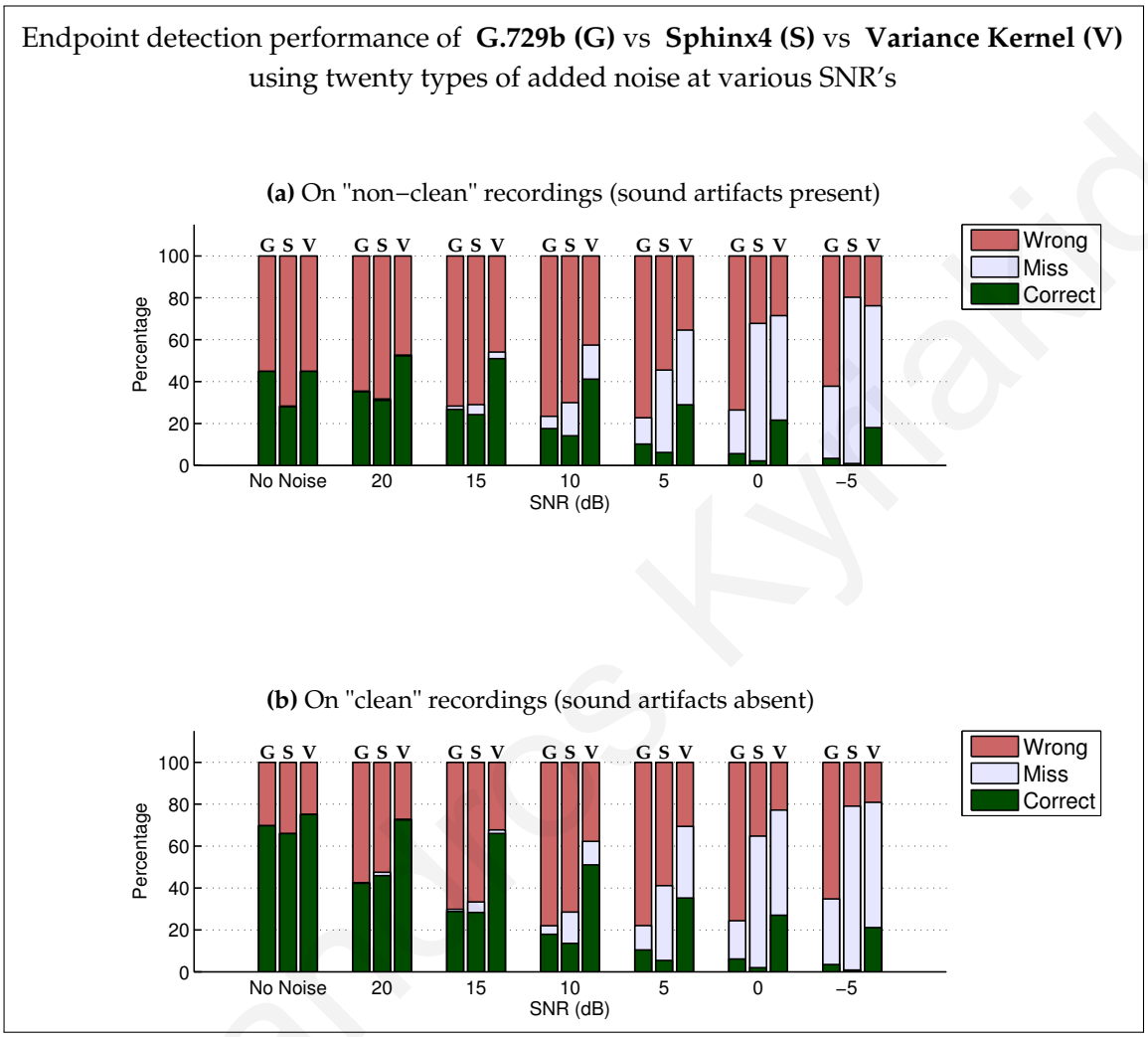

Figure A.1: A comparison of the endpoint detection performance of three different methods, using twenty types of added noise at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.1: Endpoint detection results for three different methods using *twenty types of added noise* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	35.19	31.02	52.26	42.27	45.87	72.49
	15 dB	26.75	24.23	51.00	28.76	28.32	66.05
	10 dB	17.60	14.17	41.21	17.89	13.57	51.05
	5 dB	10.21	6.21	28.94	10.43	5.43	35.22
	0 dB	5.60	2.15	21.62	6.11	1.95	27.03
	-5 dB	3.38	0.92	18.06	3.46	0.84	21.14
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.32	0.68	0.38	0.30	1.65	0.30
	15 dB	1.58	4.77	3.11	1.08	5.00	1.68
	10 dB	5.83	15.77	16.21	4.11	14.92	11.24
	5 dB	12.53	39.30	35.64	11.60	35.68	34.19
	0 dB	20.81	65.62	49.87	18.27	62.81	50.19
	-5 dB	34.41	79.32	58.17	31.32	78.22	59.78
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	64.49	68.30	47.36	57.43	52.49	27.22
	15 dB	71.66	71.00	45.89	70.16	66.68	32.27
	10 dB	76.57	70.06	42.59	78.00	71.51	37.70
	5 dB	77.26	54.49	35.41	77.97	58.89	30.59
	0 dB	73.58	32.23	28.51	75.62	35.24	22.78
	-5 dB	62.21	19.75	23.77	65.22	20.95	19.08

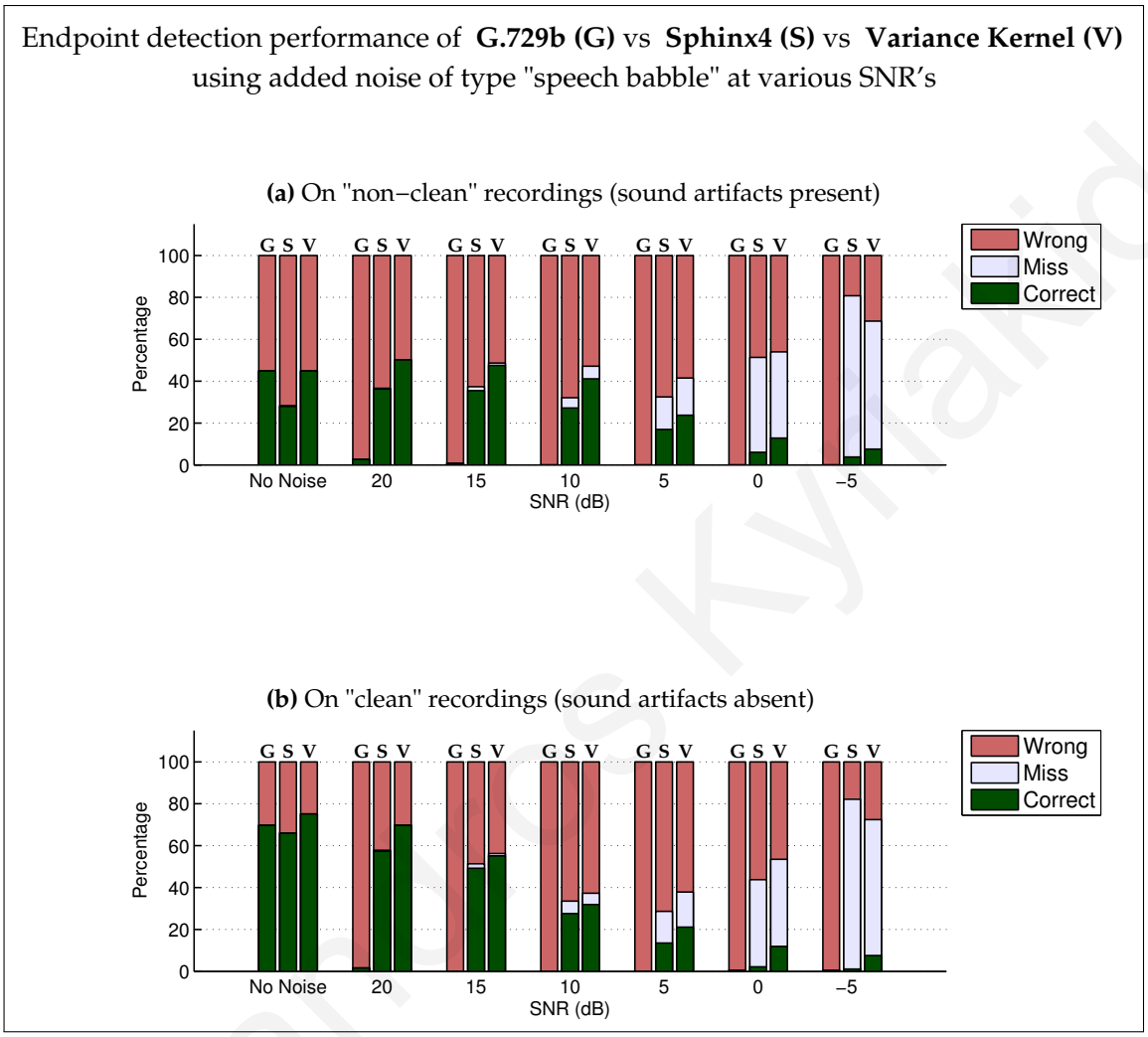

Figure A.2: A comparison of the endpoint detection performance of three different methods, using added noise of type "speech babble" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.2: Endpoint detection results for three different methods using *added noise of type "speech babble"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	2.64	36.23	50.19	1.62	57.30	69.73
	15 dB	0.75	35.47	47.55	0.00	49.19	55.13
	10 dB	0.00	27.17	41.13	0.00	27.57	31.89
	5 dB	0.00	16.98	23.77	0.00	13.51	21.08
	0 dB	0.00	6.04	12.83	0.54	2.16	11.89
	-5 dB	0.00	3.77	7.55	0.54	1.08	7.57
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.38	0.00	0.00	0.54	0.00
	15 dB	0.00	1.89	1.13	0.00	2.16	1.08
	10 dB	0.00	4.91	6.04	0.00	5.95	5.41
	5 dB	0.00	15.47	17.74	0.00	15.13	16.76
	0 dB	0.00	45.28	41.13	0.00	41.62	41.62
	-5 dB	0.00	76.98	61.13	0.00	81.08	64.86
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	97.36	63.40	49.81	98.38	42.16	30.27
	15 dB	99.25	62.64	51.32	100.00	48.65	43.78
	10 dB	100.00	67.92	52.83	100.00	66.49	62.70
	5 dB	100.00	67.55	58.49	100.00	71.35	62.16
	0 dB	100.00	48.68	46.04	99.46	56.22	46.49
	-5 dB	100.00	19.25	31.32	99.46	17.84	27.57

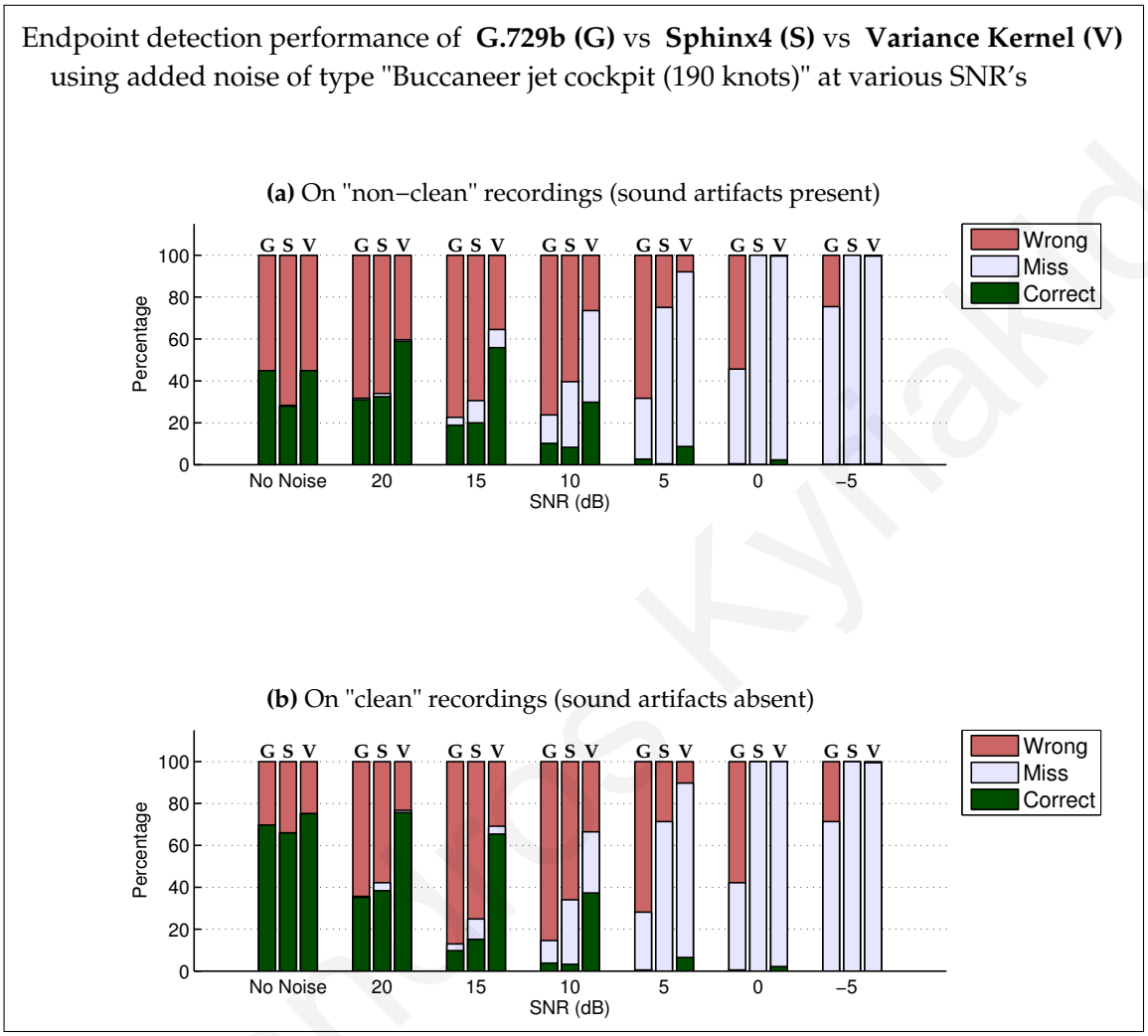

Figure A.3: A comparison of the endpoint detection performance of three different methods, using added noise of type "Buccaneer jet cockpit (190 knots)" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.3: Endpoint detection results for three different methods using *added noise of type "Buccaneer jet cockpit (190 knots)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	30.94	32.45	58.87	35.13	38.38	75.68
	15 dB	18.87	20.00	55.85	9.73	15.13	65.41
	10 dB	10.19	8.30	29.81	3.78	3.24	37.30
	5 dB	2.64	0.38	8.68	0.54	0.00	6.49
	0 dB	0.38	0.00	2.26	0.54	0.00	2.16
	-5 dB	0.00	0.00	0.38	0.00	0.00	0.00
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.75	1.51	0.75	0.54	3.78	1.08
	15 dB	3.77	10.57	8.68	3.24	9.73	3.78
	10 dB	13.59	31.32	43.77	10.81	30.81	29.19
	5 dB	29.06	74.72	83.40	27.57	71.35	83.24
	0 dB	45.28	100.00	97.36	41.62	100.00	97.84
	-5 dB	75.47	100.00	99.25	71.35	100.00	99.46
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	68.30	66.04	40.38	64.32	57.84	23.24
	15 dB	77.36	69.43	35.47	87.03	75.14	30.81
	10 dB	76.23	60.38	26.41	85.41	65.95	33.51
	5 dB	68.30	24.91	7.92	71.89	28.65	10.27
	0 dB	54.34	0.00	0.38	57.84	0.00	0.00
	-5 dB	24.53	0.00	0.38	28.65	0.00	0.54

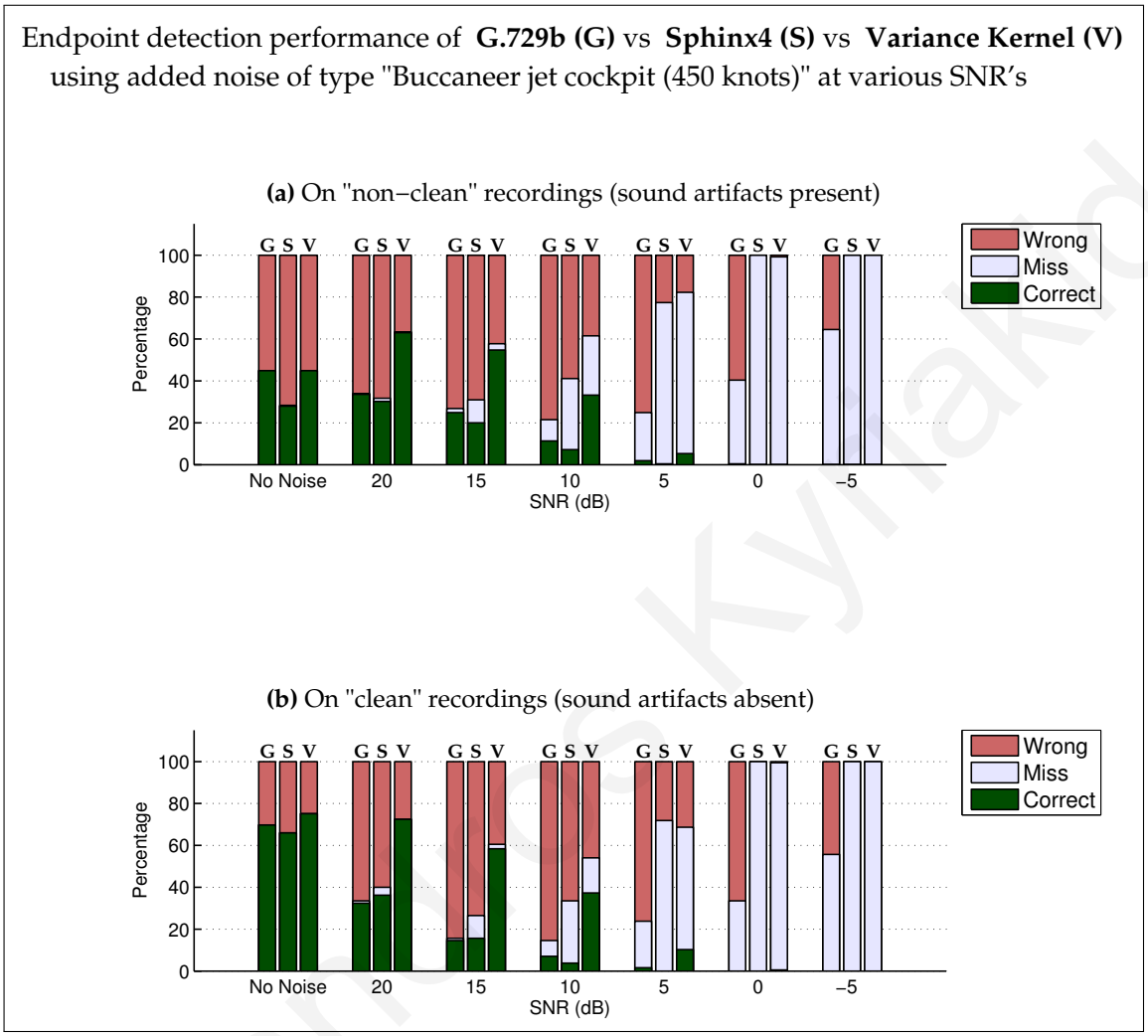

Figure A.4: A comparison of the endpoint detection performance of three different methods, using added noise of type "Buccaneer jet cockpit (450 knots)" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.4: Endpoint detection results for three different methods using *added noise of type "Buccaneer jet cockpit (450 knots)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	33.59	30.19	63.02	32.43	36.22	72.43
	15 dB	24.91	20.00	54.72	14.60	15.68	58.38
	10 dB	11.32	7.17	33.21	7.03	3.78	37.30
	5 dB	1.89	0.38	5.28	1.62	0.00	10.27
	0 dB	0.38	0.00	0.00	0.00	0.00	0.54
	-5 dB	0.00	0.00	0.00	0.00	0.00	0.00
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.38	1.51	0.38	1.08	3.78	0.00
	15 dB	1.89	10.94	3.02	1.08	10.81	2.16
	10 dB	10.19	33.96	28.30	7.57	29.73	16.76
	5 dB	23.02	76.98	76.98	22.16	71.89	58.38
	0 dB	40.00	100.00	99.25	33.51	100.00	98.92
	-5 dB	64.53	100.00	100.00	55.68	100.00	100.00
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	66.04	68.30	36.60	66.49	60.00	27.57
	15 dB	73.21	69.06	42.26	84.32	73.51	39.46
	10 dB	78.49	58.87	38.49	85.41	66.49	45.95
	5 dB	75.09	22.64	17.74	76.22	28.11	31.35
	0 dB	59.62	0.00	0.75	66.49	0.00	0.54
	-5 dB	35.47	0.00	0.00	44.32	0.00	0.00

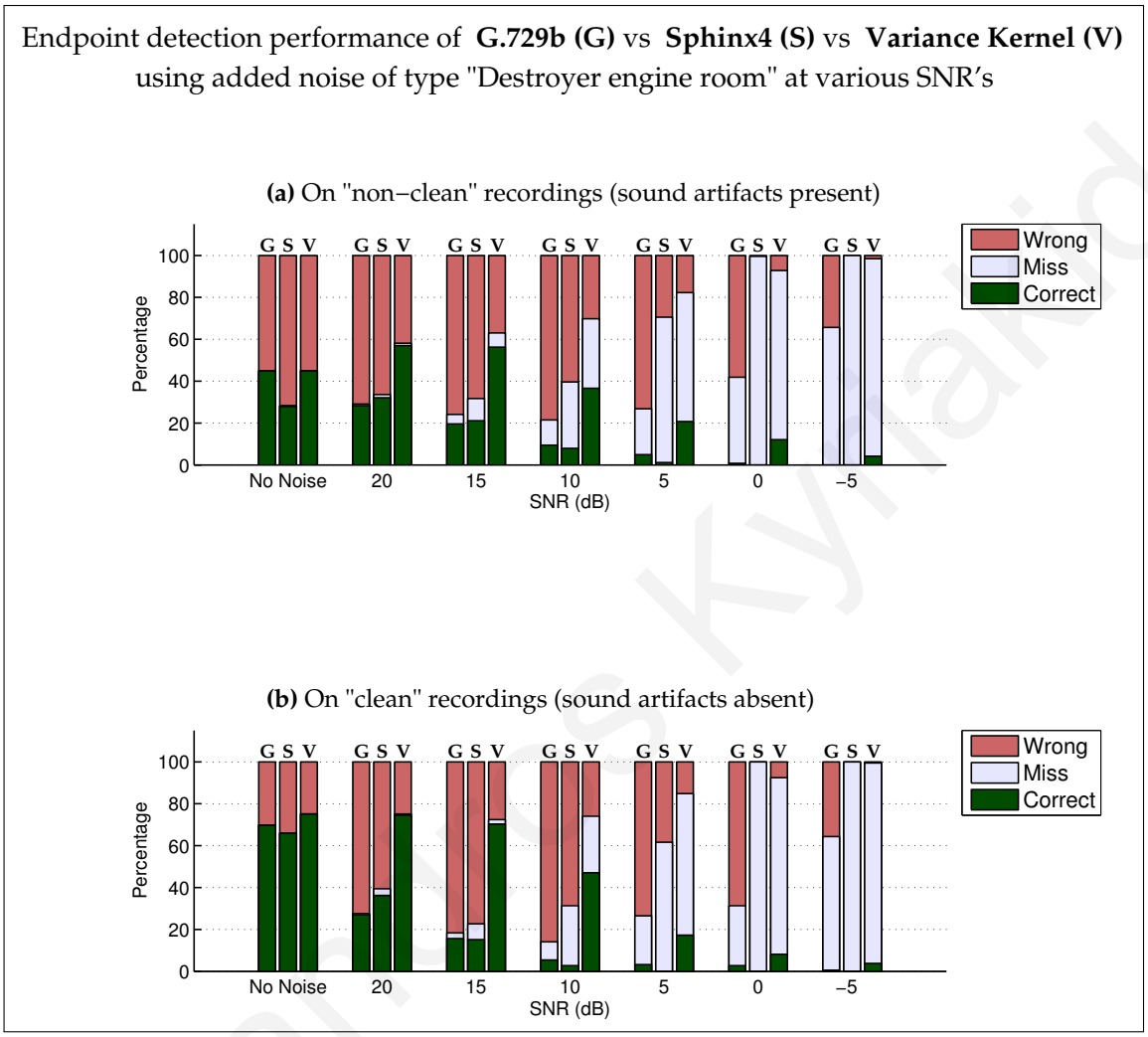

Figure A.5: A comparison of the endpoint detection performance of three different methods, using added noise of type "Destroyer engine room" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.5: Endpoint detection results for three different methods using *added noise of type "Destroyer engine room"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	28.30	32.08	56.98	27.03	36.22	74.59
	15 dB	19.62	21.13	56.23	15.68	15.13	70.27
	10 dB	9.43	7.92	36.60	5.41	2.70	47.03
	5 dB	4.91	1.13	20.75	3.24	0.00	17.30
	0 dB	0.75	0.00	12.07	2.70	0.00	8.11
	-5 dB	0.00	0.00	4.15	0.54	0.00	3.78
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.75	1.51	1.13	0.54	3.24	0.54
	15 dB	4.53	10.57	6.79	2.70	7.57	2.16
	10 dB	12.07	31.70	33.21	8.65	28.65	27.03
	5 dB	21.89	69.43	61.51	23.24	61.62	67.57
	0 dB	41.13	99.62	80.75	28.65	100.00	84.32
	-5 dB	65.66	100.00	94.34	63.78	100.00	95.68
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	70.94	66.42	41.89	72.43	60.54	24.86
	15 dB	75.85	68.30	36.98	81.62	77.30	27.57
	10 dB	78.49	60.38	30.19	85.95	68.65	25.95
	5 dB	73.21	29.43	17.74	73.51	38.38	15.13
	0 dB	58.11	0.38	7.17	68.65	0.00	7.57
	-5 dB	34.34	0.00	1.51	35.68	0.00	0.54

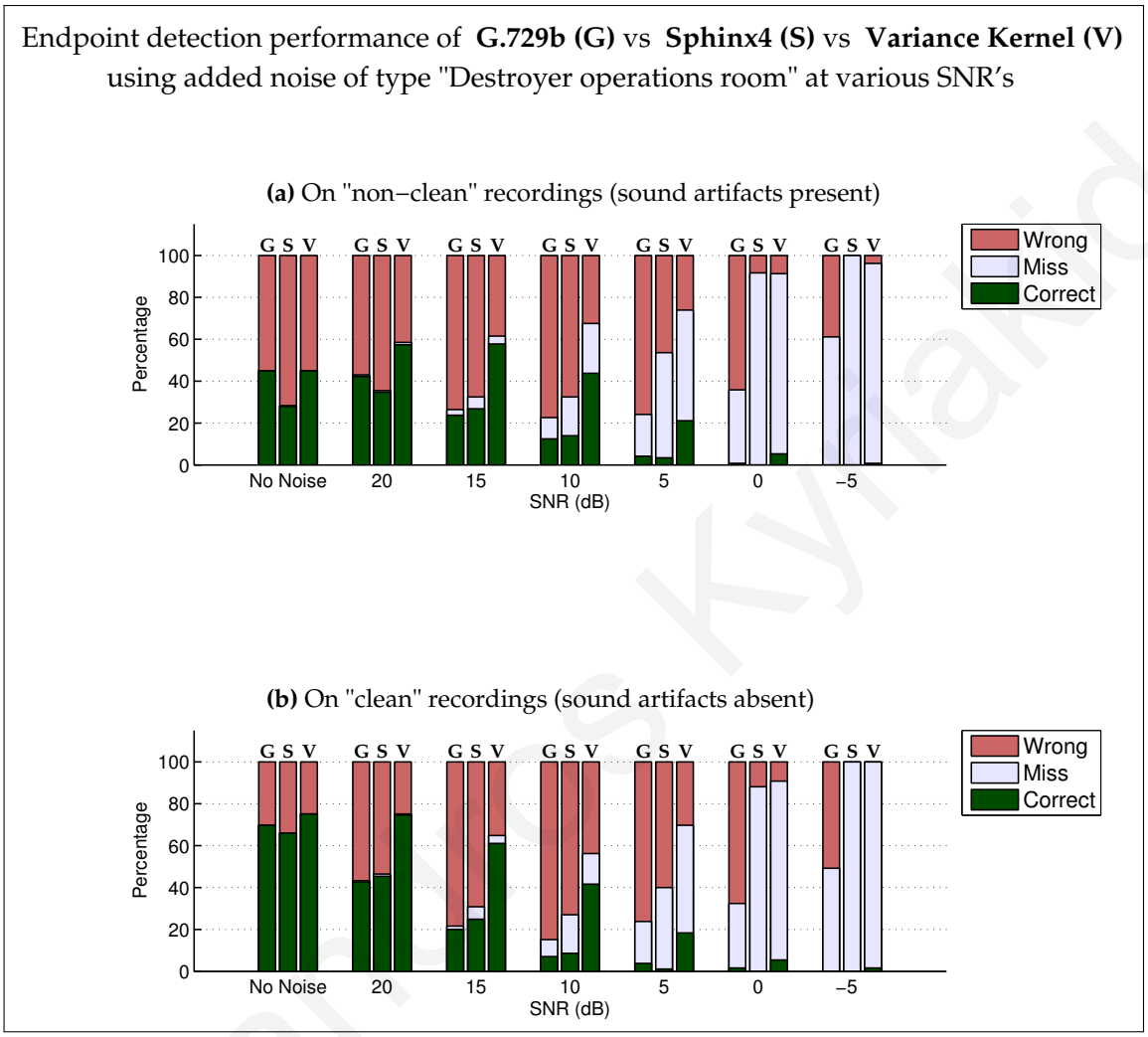

Figure A.6: A comparison of the endpoint detection performance of three different methods, using added noise of type "Destroyer operations room" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.6: Endpoint detection results for three different methods using *added noise of type "Destroyer operations room"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	42.26	34.72	57.36	42.70	45.41	74.59
	15 dB	23.77	26.79	57.74	20.00	24.86	61.08
	10 dB	12.45	13.96	43.77	7.03	8.65	41.62
	5 dB	4.15	3.40	21.13	3.78	1.08	18.38
	0 dB	0.75	0.00	5.28	1.62	0.00	5.41
	-5 dB	0.00	0.00	0.75	0.00	0.00	1.62
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.75	0.75	1.13	0.54	1.08	0.54
	15 dB	2.64	5.66	3.77	1.62	5.95	3.78
	10 dB	10.19	18.49	23.77	8.11	18.38	14.60
	5 dB	20.00	50.19	52.83	20.00	38.92	51.35
	0 dB	35.09	91.70	86.04	30.81	88.11	85.41
	-5 dB	61.13	100.00	95.47	49.19	100.00	98.38
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	56.98	64.53	41.51	56.76	53.51	24.86
	15 dB	73.58	67.55	38.49	78.38	69.19	35.13
	10 dB	77.36	67.55	32.45	84.86	72.97	43.78
	5 dB	75.85	46.41	26.04	76.22	60.00	30.27
	0 dB	64.15	8.30	8.68	67.57	11.89	9.19
	-5 dB	38.87	0.00	3.77	50.81	0.00	0.00

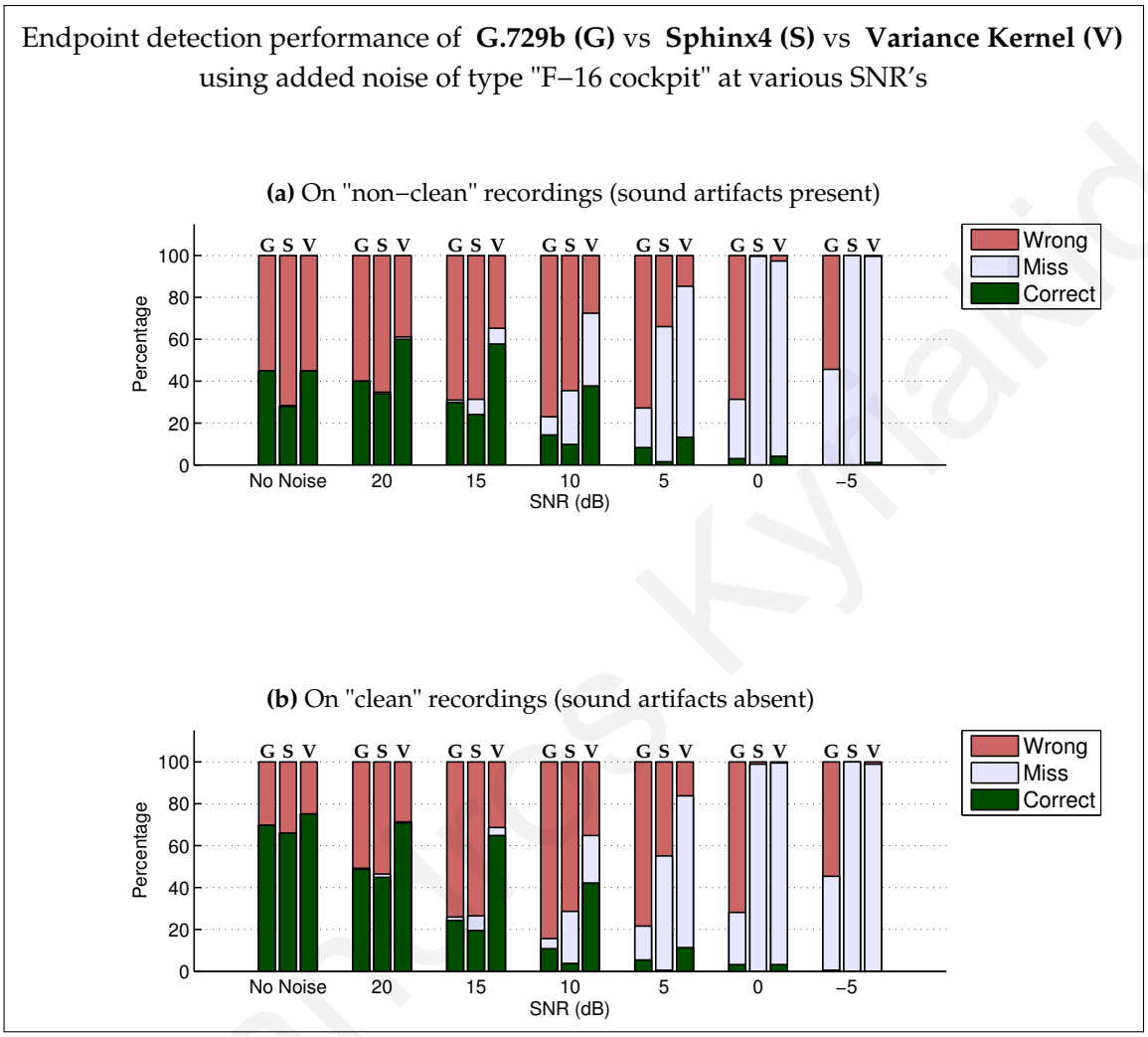

Figure A.7: A comparison of the endpoint detection performance of three different methods, using added noise of type "F-16 cockpit" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.7: Endpoint detection results for three different methods using *added noise of type "F-16 cockpit"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	40.00	33.96	60.00	48.65	44.87	70.81
	15 dB	29.81	24.15	57.74	24.32	19.46	64.86
	10 dB	14.34	9.81	37.74	10.81	3.78	42.16
	5 dB	8.30	1.51	13.21	5.41	0.54	11.35
	0 dB	3.02	0.00	4.15	3.24	0.00	3.24
	-5 dB	0.00	0.00	1.13	0.54	0.00	0.00
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.75	1.13	0.54	1.62	0.54
	15 dB	1.13	7.17	7.55	1.62	7.03	3.78
	10 dB	8.68	25.66	34.72	4.86	24.86	22.70
	5 dB	18.87	64.53	72.08	16.22	54.59	72.43
	0 dB	28.30	99.62	93.21	24.86	98.92	96.22
	-5 dB	45.66	100.00	98.49	44.87	100.00	98.92
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	60.00	65.28	38.87	50.81	53.51	28.65
	15 dB	69.06	68.68	34.72	74.05	73.51	31.35
	10 dB	76.98	64.53	27.55	84.32	71.35	35.13
	5 dB	72.83	33.96	14.72	78.38	44.87	16.22
	0 dB	68.68	0.38	2.64	71.89	1.08	0.54
	-5 dB	54.34	0.00	0.38	54.59	0.00	1.08

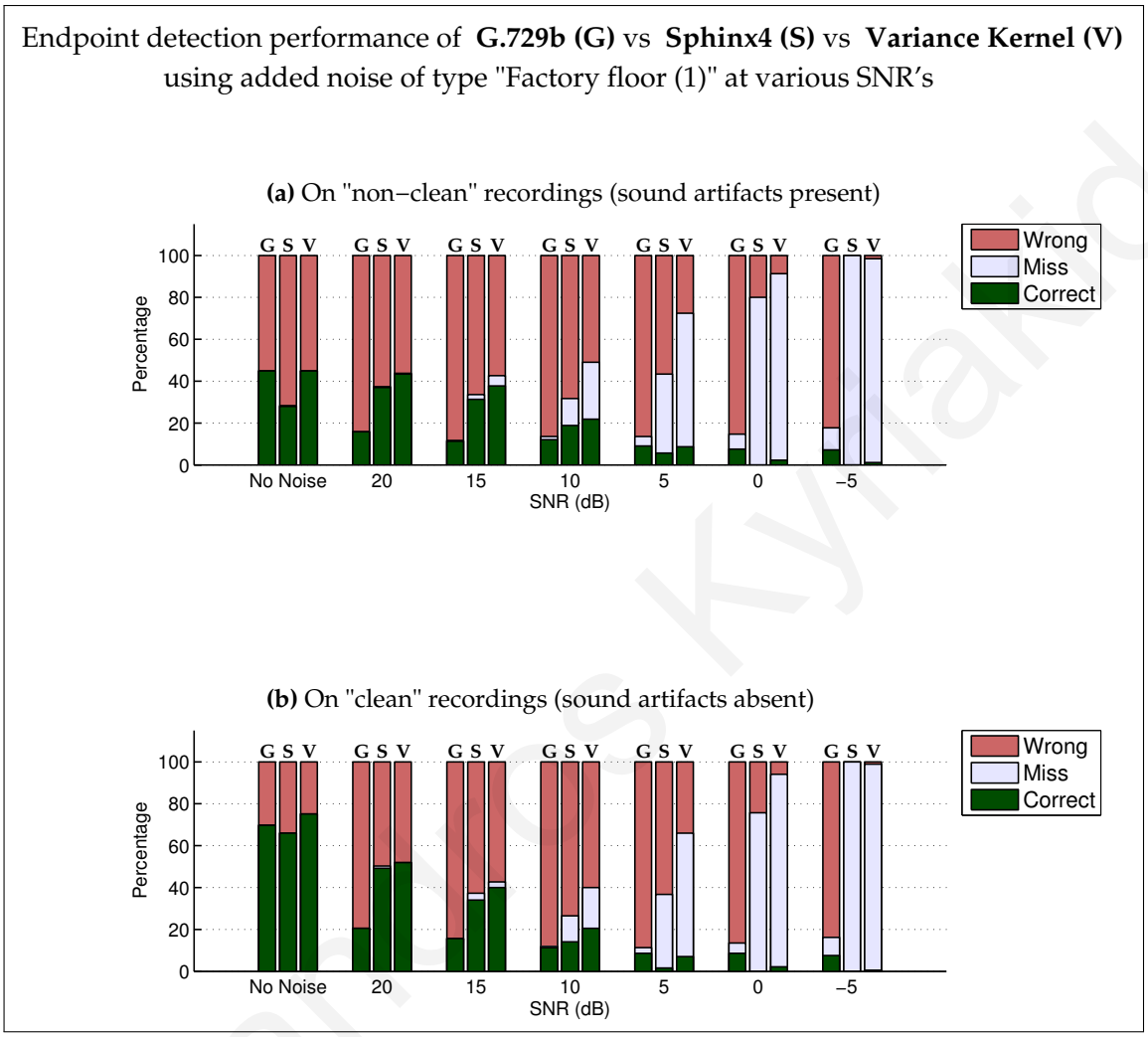

Figure A.8: A comparison of the endpoint detection performance of three different methods, using added noise of type "Factory floor (1)" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.8: Endpoint detection results for three different methods using *added noise of type "Factory floor (1)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	15.85	36.98	43.40	20.54	49.19	51.89
	15 dB	11.32	31.32	37.74	15.68	34.05	40.00
	10 dB	12.07	18.87	21.89	11.35	14.05	20.54
	5 dB	9.06	5.66	8.68	8.65	1.62	7.03
	0 dB	7.55	0.00	2.26	8.65	0.00	2.16
	-5 dB	7.17	0.00	1.13	7.57	0.00	0.54
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.38	0.38	0.00	1.08	0.00
	15 dB	0.38	2.26	4.91	0.00	3.24	2.70
	10 dB	1.51	12.83	27.17	0.54	12.43	19.46
	5 dB	4.53	37.74	63.77	2.70	35.13	58.92
	0 dB	7.17	80.00	89.06	4.86	75.68	91.89
	-5 dB	10.57	100.00	97.36	8.65	100.00	98.38
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	84.15	62.64	56.23	79.46	49.73	48.11
	15 dB	88.30	66.42	57.36	84.32	62.70	57.30
	10 dB	86.42	68.30	50.94	88.11	73.51	60.00
	5 dB	86.42	56.60	27.55	88.65	63.24	34.05
	0 dB	85.28	20.00	8.68	86.49	24.32	5.95
	-5 dB	82.26	0.00	1.51	83.78	0.00	1.08

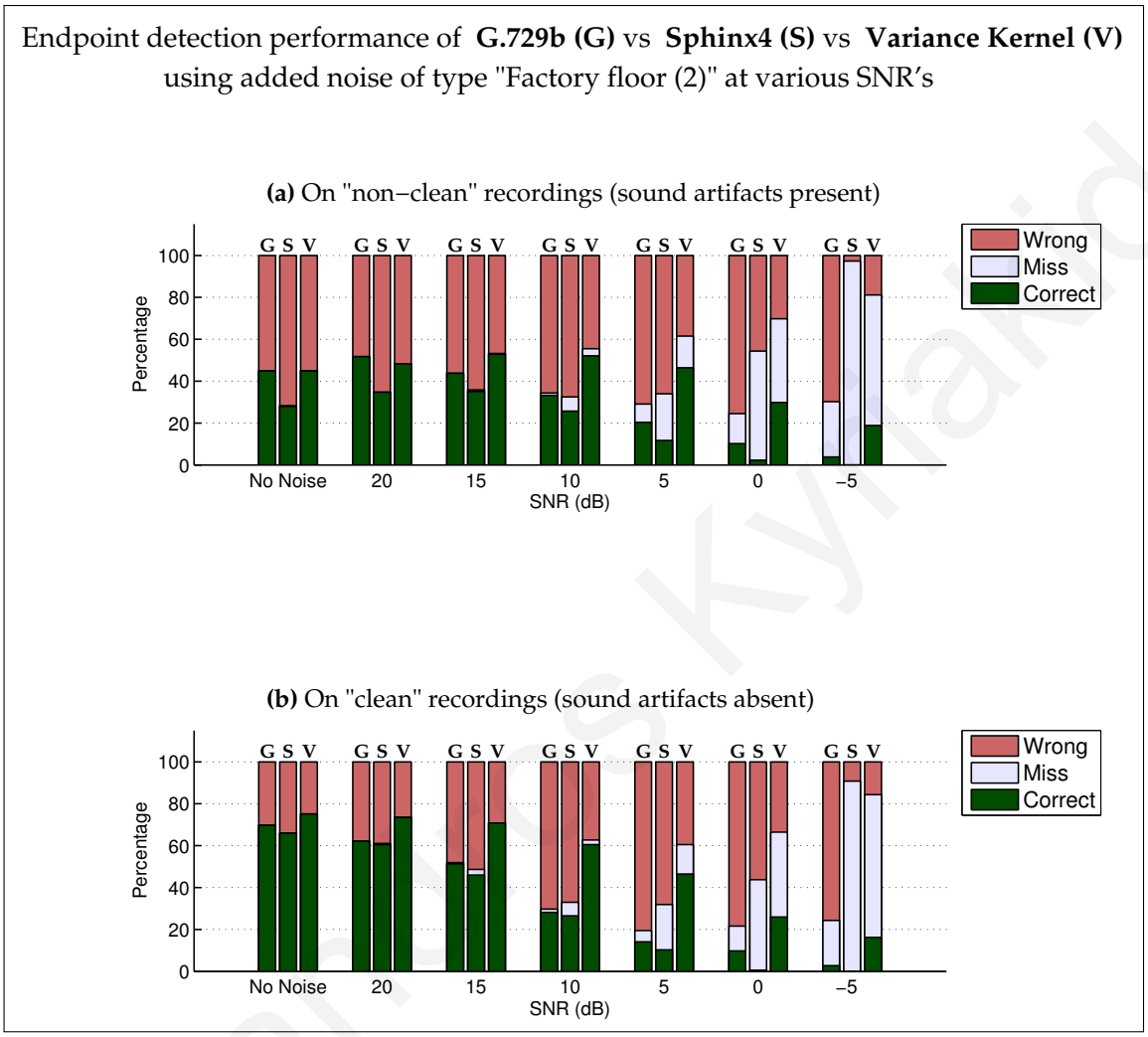

Figure A.9: A comparison of the endpoint detection performance of three different methods, using added noise of type "Factory floor (2)" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.9: Endpoint detection results for three different methods using *added noise of type "Factory floor (2)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	51.70	34.72	48.30	62.16	60.54	73.51
	15 dB	43.77	35.09	52.83	51.35	45.95	70.81
	10 dB	33.21	25.66	52.08	28.11	26.49	60.54
	5 dB	20.38	11.70	46.41	14.05	10.27	46.49
	0 dB	10.19	2.26	29.81	9.73	0.54	25.95
	-5 dB	3.77	0.00	18.87	2.70	0.00	16.22
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.00	0.00	0.00	0.54	0.00
	15 dB	0.00	0.75	0.38	0.54	2.70	0.00
	10 dB	1.13	6.79	3.40	1.62	6.49	2.16
	5 dB	8.68	22.26	15.09	5.41	21.62	14.05
	0 dB	14.34	52.08	40.00	11.89	43.24	40.54
	-5 dB	26.41	97.36	62.26	21.62	90.81	68.11
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	48.30	65.28	51.70	37.84	38.92	26.49
	15 dB	56.23	64.15	46.79	48.11	51.35	29.19
	10 dB	65.66	67.55	44.53	70.27	67.03	37.30
	5 dB	70.94	66.04	38.49	80.54	68.11	39.46
	0 dB	75.47	45.66	30.19	78.38	56.22	33.51
	-5 dB	69.81	2.64	18.87	75.68	9.19	15.68

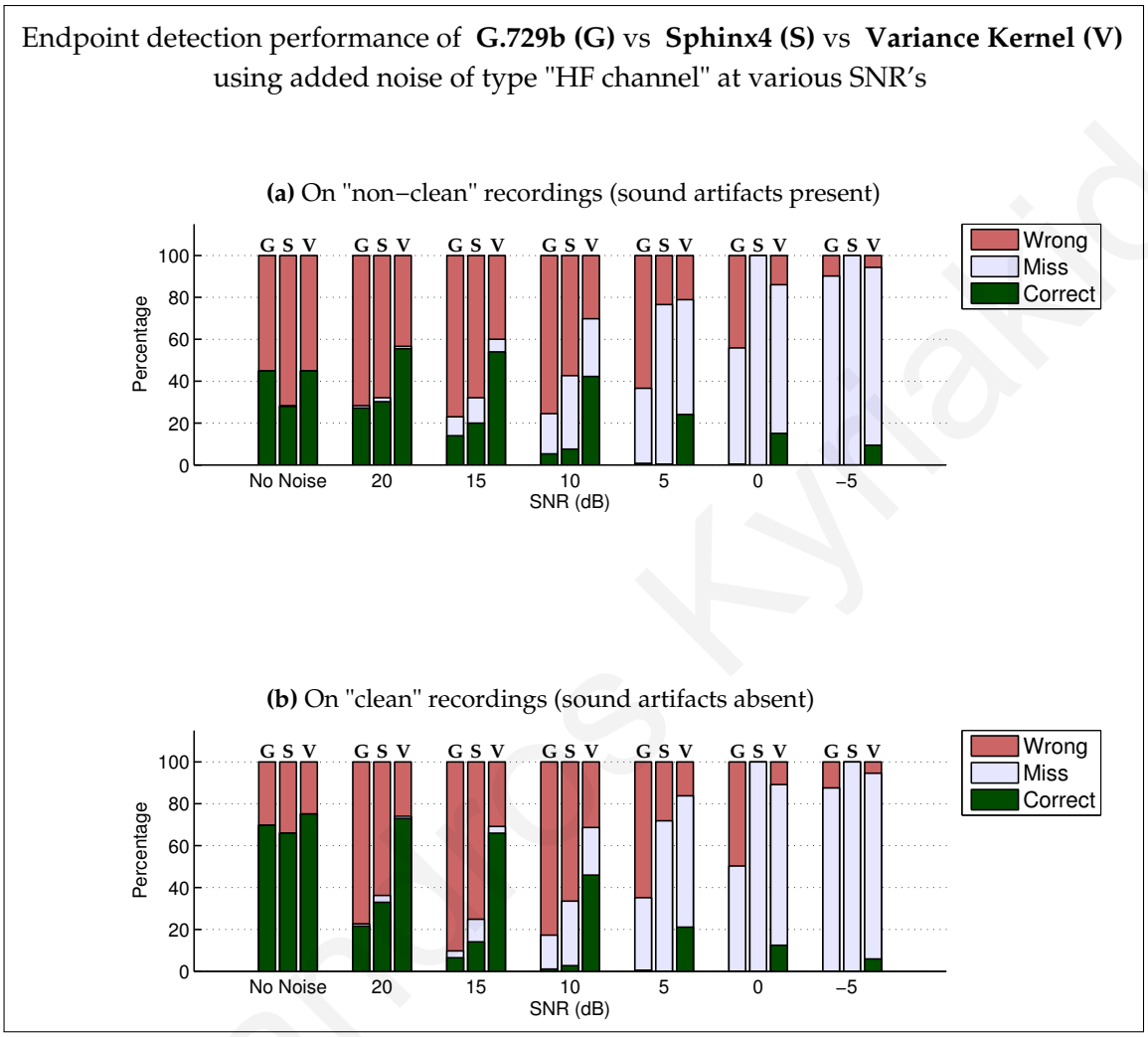

Figure A.10: A comparison of the endpoint detection performance of three different methods, using added noise of type "HF channel" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.10: Endpoint detection results for three different methods using *added noise of type "HF channel"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	27.17	30.19	55.47	21.62	32.97	72.97
	15 dB	13.96	20.00	53.96	6.49	14.05	65.95
	10 dB	5.28	7.55	42.26	1.08	2.70	45.95
	5 dB	0.75	0.38	24.15	0.54	0.00	21.08
	0 dB	0.38	0.00	15.09	0.00	0.00	12.43
	-5 dB	0.00	0.00	9.43	0.00	0.00	5.95
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	1.13	1.89	1.13	1.08	3.24	1.08
	15 dB	9.06	12.07	6.04	3.24	10.81	3.24
	10 dB	19.25	35.09	27.55	16.22	30.81	22.70
	5 dB	35.85	76.23	54.72	34.59	71.89	62.70
	0 dB	55.47	100.00	70.94	50.27	100.00	76.76
	-5 dB	90.19	100.00	84.91	87.57	100.00	88.65
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	71.70	67.92	43.40	77.30	63.78	25.95
	15 dB	76.98	67.92	40.00	90.27	75.14	30.81
	10 dB	75.47	57.36	30.19	82.70	66.49	31.35
	5 dB	63.40	23.40	21.13	64.86	28.11	16.22
	0 dB	44.15	0.00	13.96	49.73	0.00	10.81
	-5 dB	9.81	0.00	5.66	12.43	0.00	5.41

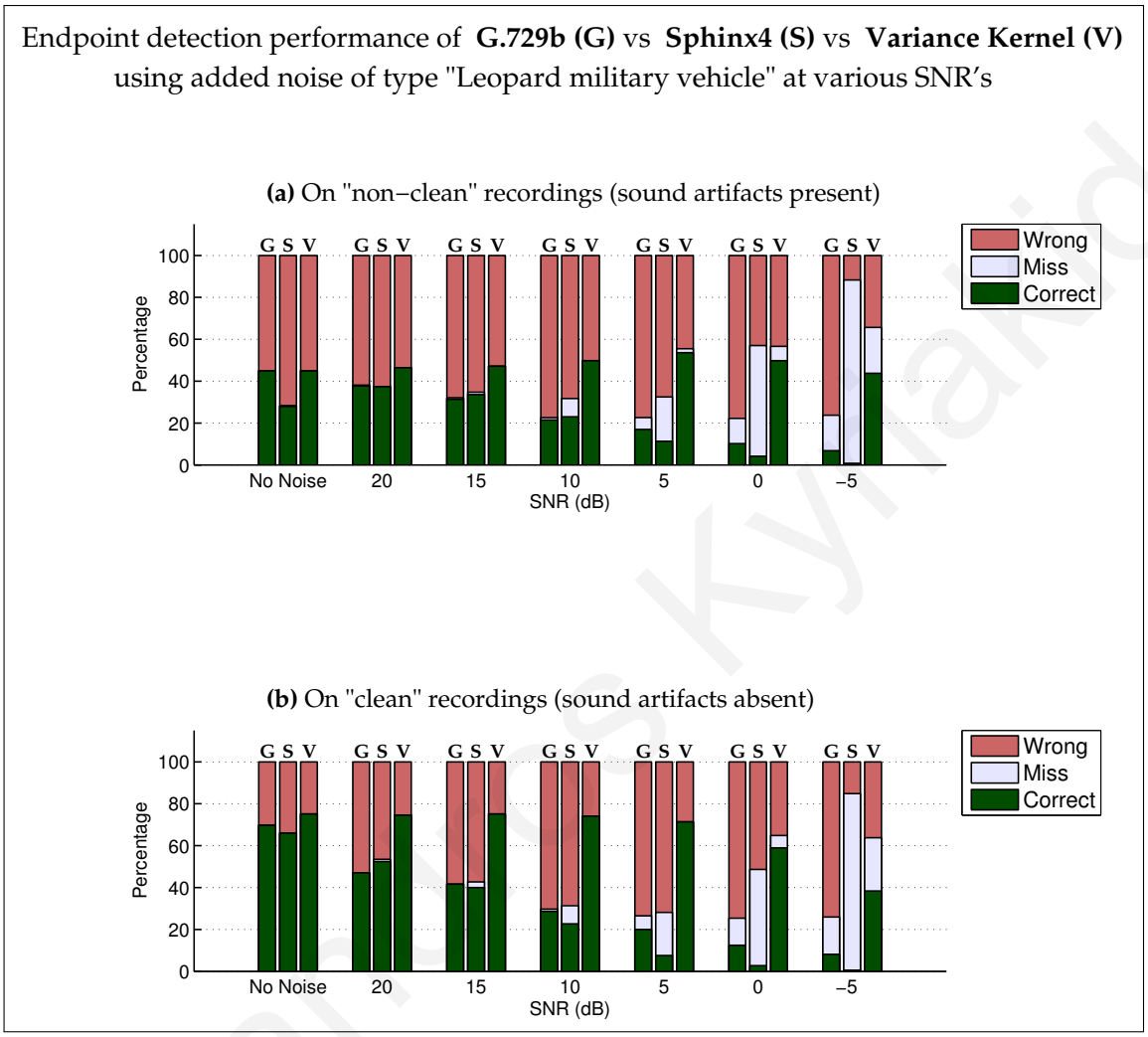

Figure A.11: A comparison of the endpoint detection performance of three different methods, using added noise of type "Leopard military vehicle" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.11: Endpoint detection results for three different methods using *added noise of type "Leopard military vehicle"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	37.74	37.36	46.41	47.03	52.43	74.59
	15 dB	31.32	33.59	47.17	41.62	40.00	75.14
	10 dB	21.51	23.02	49.81	28.65	22.70	74.05
	5 dB	16.98	11.32	53.59	20.00	7.57	71.35
	0 dB	10.19	4.15	49.81	12.43	2.70	58.92
	-5 dB	6.79	0.75	43.77	8.11	0.54	38.38
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.38	0.00	0.00	0.00	1.08	0.00
	15 dB	0.75	1.13	0.00	0.00	2.70	0.00
	10 dB	1.13	8.68	0.00	1.08	8.65	0.00
	5 dB	5.66	21.13	1.89	6.49	20.54	0.00
	0 dB	12.07	52.83	6.79	12.97	45.95	5.95
	-5 dB	16.98	87.55	21.89	17.84	84.32	25.41
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	61.89	62.64	53.59	52.97	46.49	25.41
	15 dB	67.92	65.28	52.83	58.38	57.30	24.86
	10 dB	77.36	68.30	50.19	70.27	68.65	25.95
	5 dB	77.36	67.55	44.53	73.51	71.89	28.65
	0 dB	77.74	43.02	43.40	74.59	51.35	35.13
	-5 dB	76.23	11.70	34.34	74.05	15.13	36.22

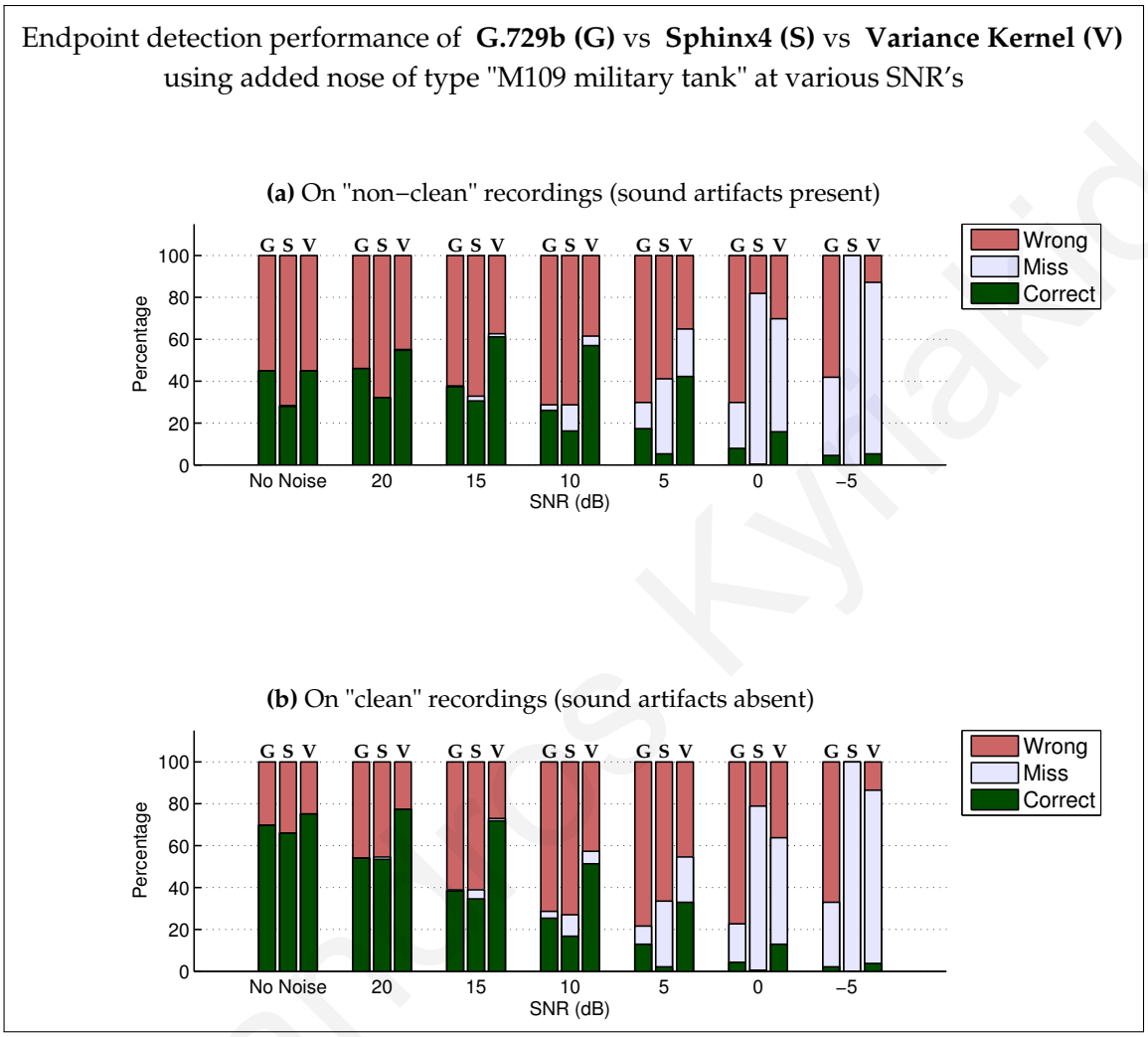

Figure A.12: A comparison of the endpoint detection performance of three different methods, using added nose of type "M109 military tank" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.12: Endpoint detection results for three different methods using *added nose of type "M109 military tank"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	46.04	32.08	54.72	54.05	53.51	77.30
	15 dB	37.36	30.57	61.13	38.38	34.59	71.89
	10 dB	26.04	16.23	56.98	25.41	16.76	51.35
	5 dB	17.36	5.28	42.26	12.97	2.16	32.97
	0 dB	7.92	0.38	15.85	4.32	0.54	12.97
	-5 dB	4.53	0.00	5.28	2.16	0.00	3.78
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.00	0.38	0.00	1.08	0.00
	15 dB	0.38	2.26	1.51	0.54	4.32	1.08
	10 dB	2.64	12.45	4.53	3.24	10.27	5.95
	5 dB	12.45	35.85	22.64	8.65	31.35	21.62
	0 dB	21.89	81.51	53.96	18.38	78.38	50.81
	-5 dB	37.36	100.00	81.89	30.81	100.00	82.70
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	53.96	67.92	44.91	45.95	45.41	22.70
	15 dB	62.26	67.17	37.36	61.08	61.08	27.03
	10 dB	71.32	71.32	38.49	71.35	72.97	42.70
	5 dB	70.19	58.87	35.09	78.38	66.49	45.41
	0 dB	70.19	18.11	30.19	77.30	21.08	36.22
	-5 dB	58.11	0.00	12.83	67.03	0.00	13.51

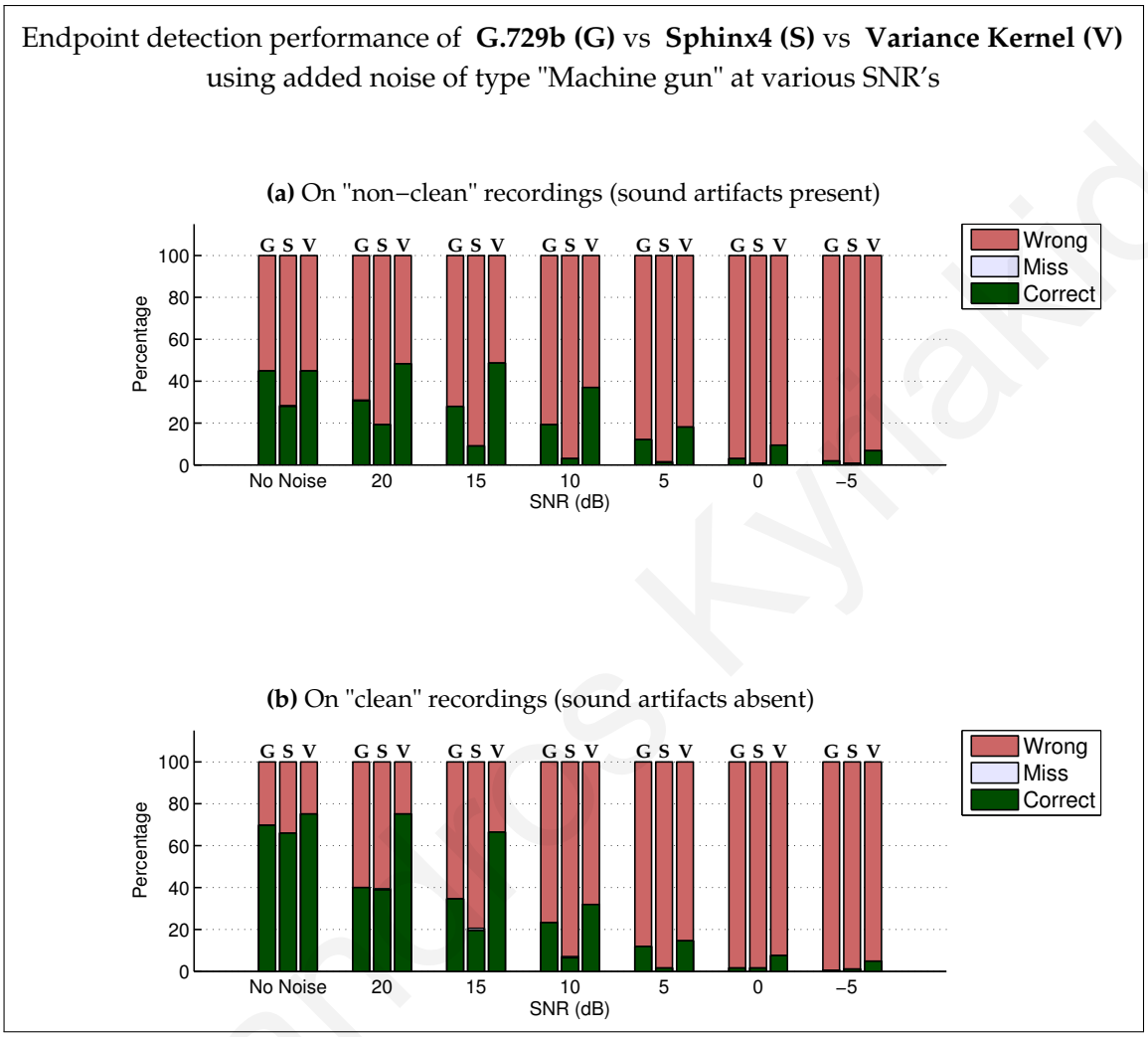

Figure A.13: A comparison of the endpoint detection performance of three different methods, using added noise of type "Machine gun" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.13: Endpoint detection results for three different methods using *added noise of type "Machine gun"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	30.57	19.25	48.30	40.00	38.92	75.14
	15 dB	27.93	9.06	48.68	34.59	19.46	66.49
	10 dB	19.25	3.02	36.98	23.24	6.49	31.89
	5 dB	12.07	1.13	18.11	11.89	1.62	14.60
	0 dB	3.02	0.75	9.43	1.62	1.62	7.57
	-5 dB	1.89	0.75	6.79	0.54	1.08	4.86
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.38	0.00	0.00	0.00	0.54	0.00
	15 dB	0.00	0.00	0.00	0.00	1.08	0.00
	10 dB	0.00	0.00	0.00	0.00	0.54	0.00
	5 dB	0.00	0.38	0.00	0.00	0.00	0.00
	0 dB	0.00	0.00	0.00	0.00	0.00	0.00
	-5 dB	0.00	0.00	0.00	0.00	0.00	0.00
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	69.06	80.75	51.70	60.00	60.54	24.86
	15 dB	72.08	90.94	51.32	65.41	79.46	33.51
	10 dB	80.75	96.98	63.02	76.76	92.97	68.11
	5 dB	87.92	98.49	81.89	88.11	98.38	85.41
	0 dB	96.98	99.25	90.57	98.38	98.38	92.43
	-5 dB	98.11	99.25	93.21	99.46	98.92	95.14

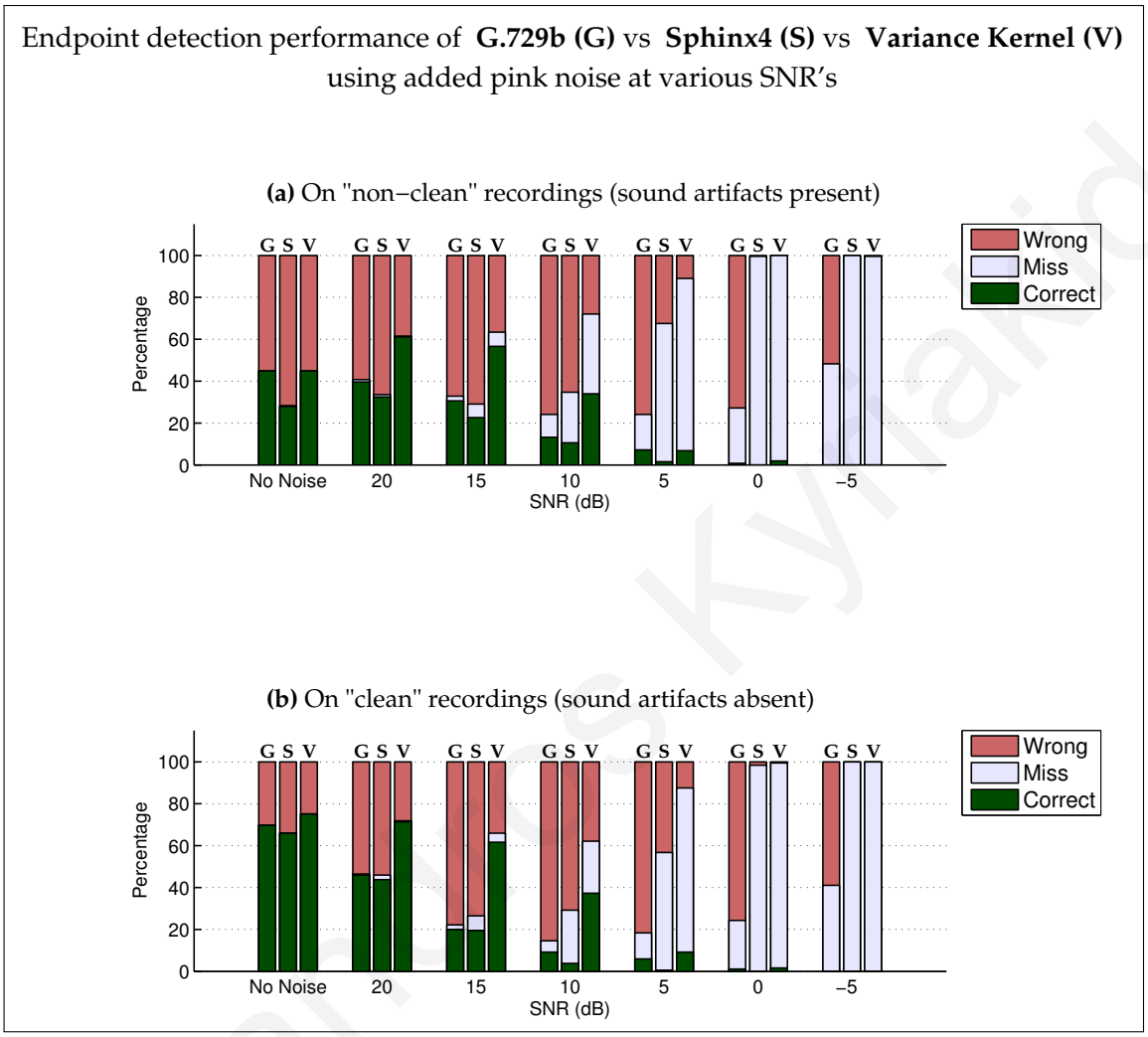

Figure A.14: A comparison of the endpoint detection performance of three different methods, using added pink noise at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.14: Endpoint detection results for three different methods using *added pink noise* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	39.62	32.45	61.13	45.95	43.78	71.35
	15 dB	30.57	22.64	56.60	20.00	19.46	61.62
	10 dB	13.21	10.57	33.96	9.19	3.78	37.30
	5 dB	7.17	1.51	6.79	5.95	0.54	9.19
	0 dB	0.75	0.00	1.89	1.08	0.00	1.62
	-5 dB	0.00	0.00	0.00	0.00	0.00	0.00
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	1.13	1.13	0.38	0.54	2.16	0.54
	15 dB	2.26	6.42	6.79	2.16	7.03	4.32
	10 dB	10.94	24.15	38.11	5.41	25.41	24.86
	5 dB	16.98	66.04	82.26	12.43	56.22	78.38
	0 dB	26.41	99.62	98.11	23.24	98.38	97.84
	-5 dB	48.30	100.00	99.62	41.08	100.00	100.00
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	59.24	66.42	38.49	53.51	54.05	28.11
	15 dB	67.17	70.94	36.60	77.84	73.51	34.05
	10 dB	75.85	65.28	27.93	85.41	70.81	37.84
	5 dB	75.85	32.45	10.94	81.62	43.24	12.43
	0 dB	72.83	0.38	0.00	75.68	1.62	0.54
	-5 dB	51.70	0.00	0.38	58.92	0.00	0.00

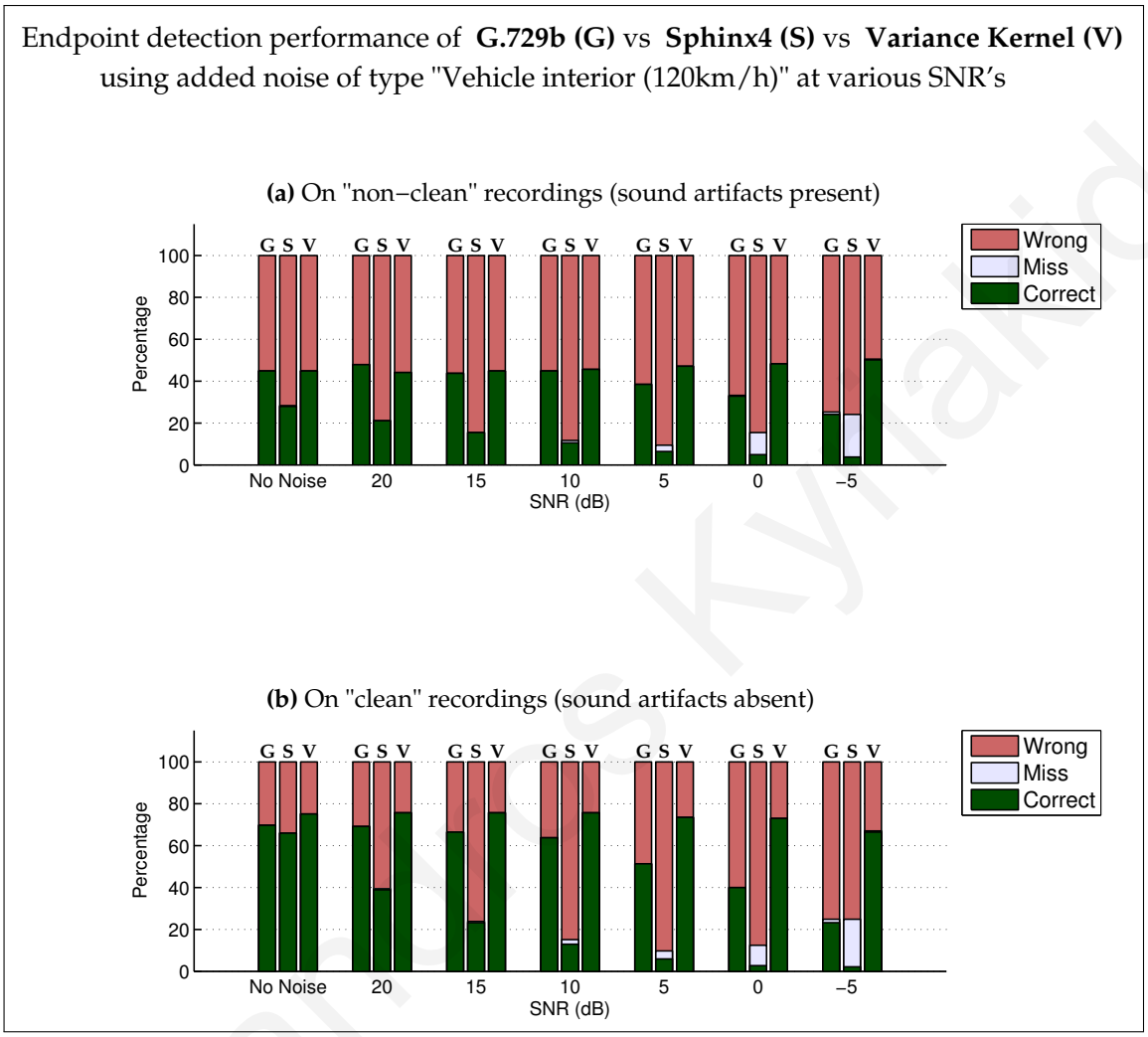

Figure A.15: A comparison of the endpoint detection performance of three different methods, using added noise of type "Vehicle interior (120km/h)" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.15: Endpoint detection results for three different methods using *added noise of type "Vehicle interior (120km/h)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	47.92	21.13	44.15	69.19	38.92	75.68
	15 dB	43.77	15.47	44.91	66.49	23.24	75.68
	10 dB	44.91	10.57	45.66	63.78	12.97	75.68
	5 dB	38.49	6.42	47.17	51.35	5.95	73.51
	0 dB	32.83	4.91	48.30	40.00	2.70	72.97
	-5 dB	24.15	3.77	50.19	23.24	2.16	66.49
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.00	0.00	0.00	0.54	0.00
	15 dB	0.00	0.00	0.00	0.00	0.54	0.00
	10 dB	0.00	1.13	0.00	0.00	2.16	0.00
	5 dB	0.00	3.02	0.00	0.00	3.78	0.00
	0 dB	0.38	10.57	0.00	0.00	9.73	0.00
	-5 dB	1.13	20.38	0.38	1.62	22.70	0.54
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	52.08	78.87	55.85	30.81	60.54	24.32
	15 dB	56.23	84.53	55.09	33.51	76.22	24.32
	10 dB	55.09	88.30	54.34	36.22	84.86	24.32
	5 dB	61.51	90.57	52.83	48.65	90.27	26.49
	0 dB	66.79	84.53	51.70	60.00	87.57	27.03
	-5 dB	74.72	75.85	49.43	75.14	75.14	32.97

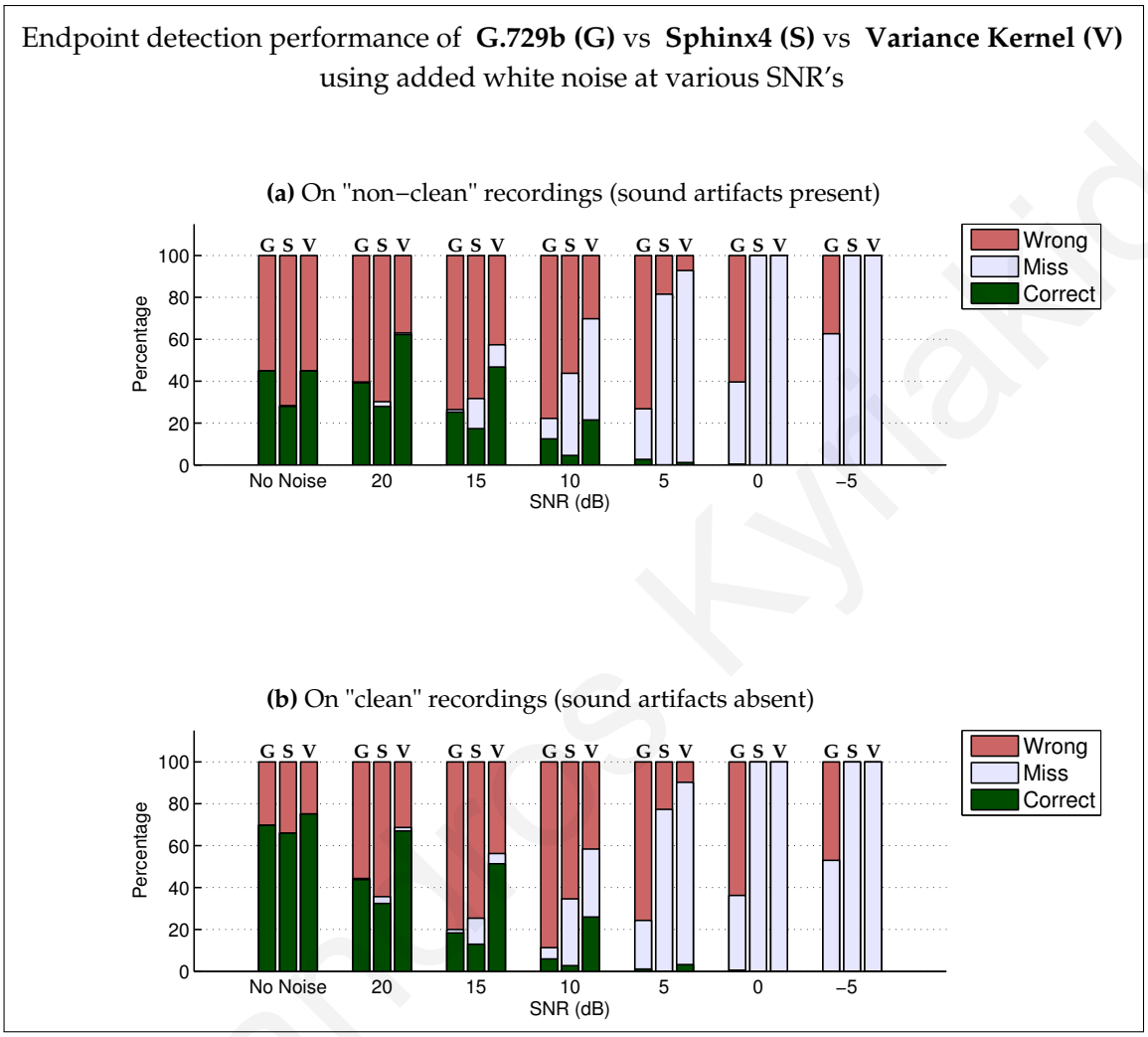

Figure A.16: A comparison of the endpoint detection performance of three different methods, using added white noise at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.16: Endpoint detection results for three different methods using *added white noise* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	39.24	27.93	62.26	43.78	32.43	67.03
	15 dB	25.28	17.36	46.79	18.38	12.97	51.35
	10 dB	12.45	4.53	21.51	5.95	2.70	25.95
	5 dB	2.64	0.00	1.13	1.08	0.00	3.24
	0 dB	0.38	0.00	0.00	0.54	0.00	0.00
	-5 dB	0.00	0.00	0.00	0.00	0.00	0.00
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.38	2.26	0.75	0.54	3.24	1.62
	15 dB	1.13	14.34	10.57	1.62	12.43	4.86
	10 dB	9.81	39.24	48.30	5.41	31.89	32.43
	5 dB	24.15	81.51	91.70	23.24	77.30	87.03
	0 dB	39.24	100.00	100.00	35.68	100.00	100.00
	-5 dB	62.64	100.00	100.00	52.97	100.00	100.00
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	60.38	69.81	36.98	55.68	64.32	31.35
	15 dB	73.58	68.30	42.64	80.00	74.59	43.78
	10 dB	77.74	56.23	30.19	88.65	65.41	41.62
	5 dB	73.21	18.49	7.17	75.68	22.70	9.73
	0 dB	60.38	0.00	0.00	63.78	0.00	0.00
	-5 dB	37.36	0.00	0.00	47.03	0.00	0.00

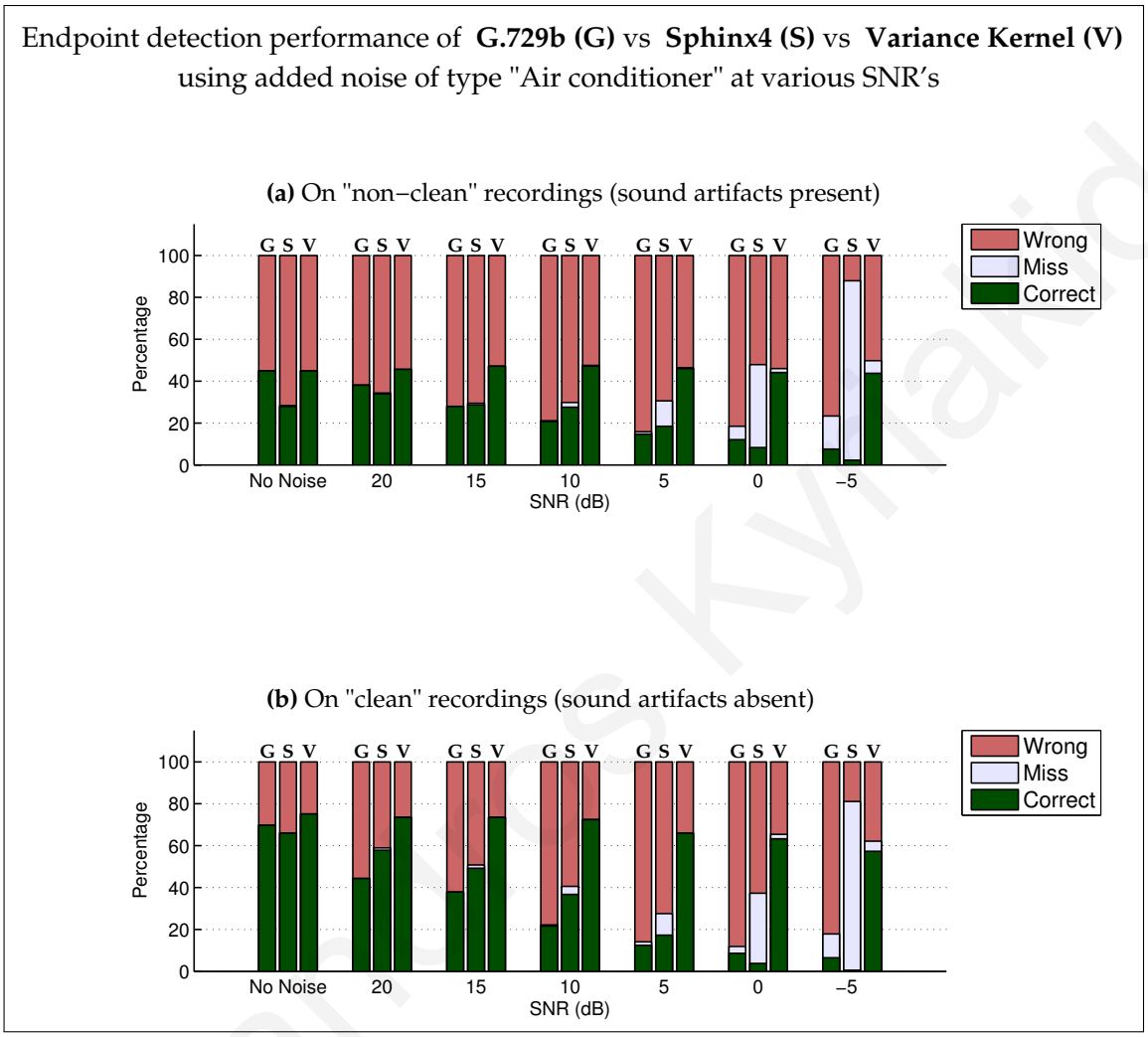

Figure A.17: A comparison of the endpoint detection performance of three different methods, using added noise of type "Air conditioner" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.17: Endpoint detection results for three different methods using *added noise of type "Air conditioner"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	38.11	33.96	45.66	44.32	57.84	73.51
	15 dB	27.93	28.68	47.17	37.84	49.19	73.51
	10 dB	20.75	27.55	47.17	21.62	36.76	72.43
	5 dB	14.72	18.49	46.04	12.43	17.30	65.95
	0 dB	12.07	8.30	44.15	8.65	3.78	63.24
	-5 dB	7.55	2.26	43.77	6.49	0.54	57.30
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.38	0.00	0.00	1.08	0.00
	15 dB	0.00	0.75	0.00	0.00	1.62	0.00
	10 dB	0.38	2.26	0.38	0.54	3.78	0.00
	5 dB	1.13	12.07	0.38	1.62	10.27	0.00
	0 dB	6.42	39.62	1.89	3.24	33.51	2.16
	-5 dB	15.85	85.66	6.04	11.35	80.54	4.86
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	61.89	65.66	54.34	55.68	41.08	26.49
	15 dB	72.08	70.57	52.83	62.16	49.19	26.49
	10 dB	78.87	70.19	52.45	77.84	59.46	27.57
	5 dB	84.15	69.43	53.59	85.95	72.43	34.05
	0 dB	81.51	52.08	53.96	88.11	62.70	34.59
	-5 dB	76.60	12.07	50.19	82.16	18.92	37.84

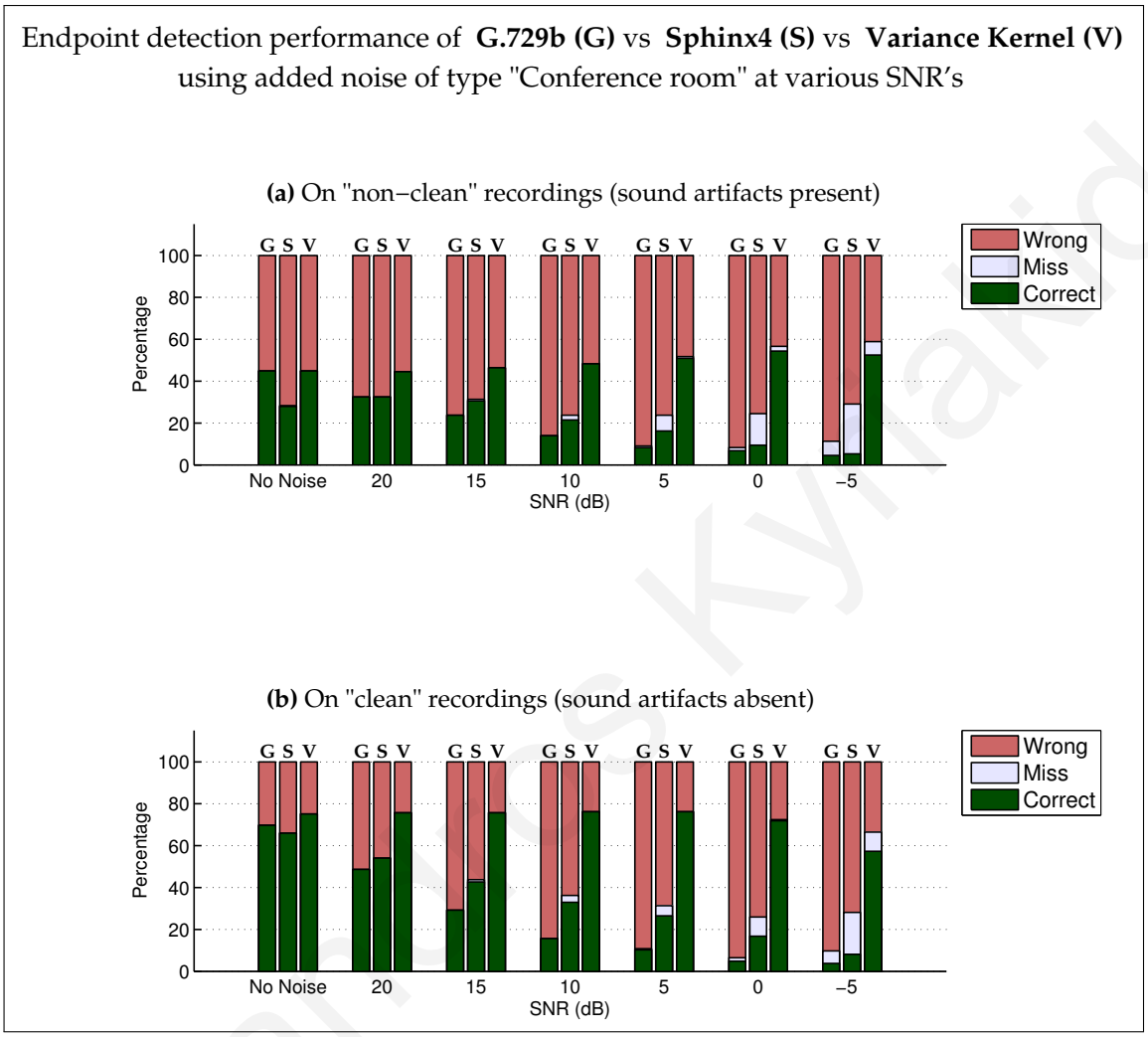

Figure A.18: A comparison of the endpoint detection performance of three different methods, using added noise of type "Conference room" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.18: Endpoint detection results for three different methods using *added noise of type "Conference room"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	32.45	32.45	44.53	48.65	54.05	75.68
	15 dB	23.77	30.57	46.41	29.19	42.70	75.68
	10 dB	13.96	21.51	48.30	15.68	32.97	76.22
	5 dB	8.30	16.23	50.94	10.27	26.49	76.22
	0 dB	6.79	9.43	54.34	4.86	16.76	71.89
	-5 dB	4.53	5.28	52.45	3.78	8.11	57.30
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.00	0.00	0.00	0.00	0.00
	15 dB	0.00	0.75	0.00	0.00	1.08	0.00
	10 dB	0.00	2.26	0.00	0.00	3.24	0.00
	5 dB	0.75	7.55	0.75	0.54	4.86	0.00
	0 dB	1.51	15.09	2.26	1.62	9.19	0.54
	-5 dB	6.79	23.77	6.42	5.95	20.00	9.19
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	67.55	67.55	55.47	51.35	45.95	24.32
	15 dB	76.23	68.68	53.59	70.81	56.22	24.32
	10 dB	86.04	76.23	51.70	84.32	63.78	23.78
	5 dB	90.94	76.23	48.30	89.19	68.65	23.78
	0 dB	91.70	75.47	43.40	93.51	74.05	27.57
	-5 dB	88.68	70.94	41.13	90.27	71.89	33.51

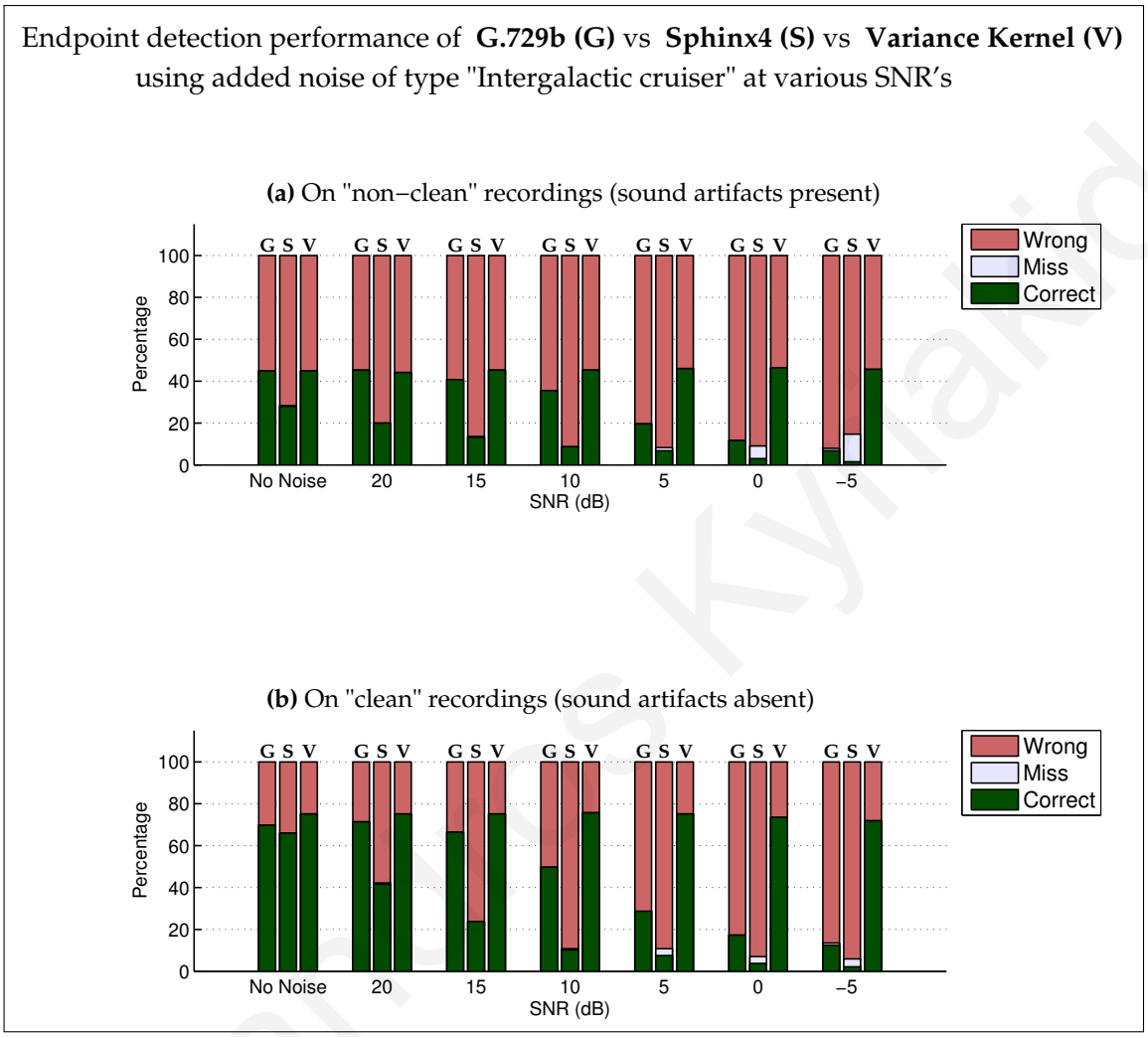

Figure A.19: A comparison of the endpoint detection performance of three different methods, using added noise of type "Intergalactic cruiser" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.19: Endpoint detection results for three different methods using *added noise of type "Intergalactic cruiser"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	45.28	20.00	44.15	71.35	41.62	75.14
	15 dB	40.76	13.21	45.28	66.49	23.78	75.14
	10 dB	35.47	8.68	45.28	49.73	10.27	75.68
	5 dB	19.62	6.79	46.04	28.65	7.57	75.14
	0 dB	11.70	3.02	46.41	17.30	3.78	73.51
	-5 dB	6.79	1.51	45.66	12.43	2.16	71.89
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.00	0.00	0.00	0.54	0.00
	15 dB	0.00	0.38	0.00	0.00	0.00	0.00
	10 dB	0.00	0.00	0.00	0.00	0.54	0.00
	5 dB	0.00	1.51	0.00	0.00	3.24	0.00
	0 dB	0.00	6.04	0.00	0.00	3.24	0.00
	-5 dB	1.13	13.21	0.00	1.08	3.78	0.00
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	54.72	80.00	55.85	28.65	57.84	24.86
	15 dB	59.24	86.42	54.72	33.51	76.22	24.86
	10 dB	64.53	91.32	54.72	50.27	89.19	24.32
	5 dB	80.38	91.70	53.96	71.35	89.19	24.86
	0 dB	88.30	90.94	53.59	82.70	92.97	26.49
	-5 dB	92.08	85.28	54.34	86.49	94.05	28.11

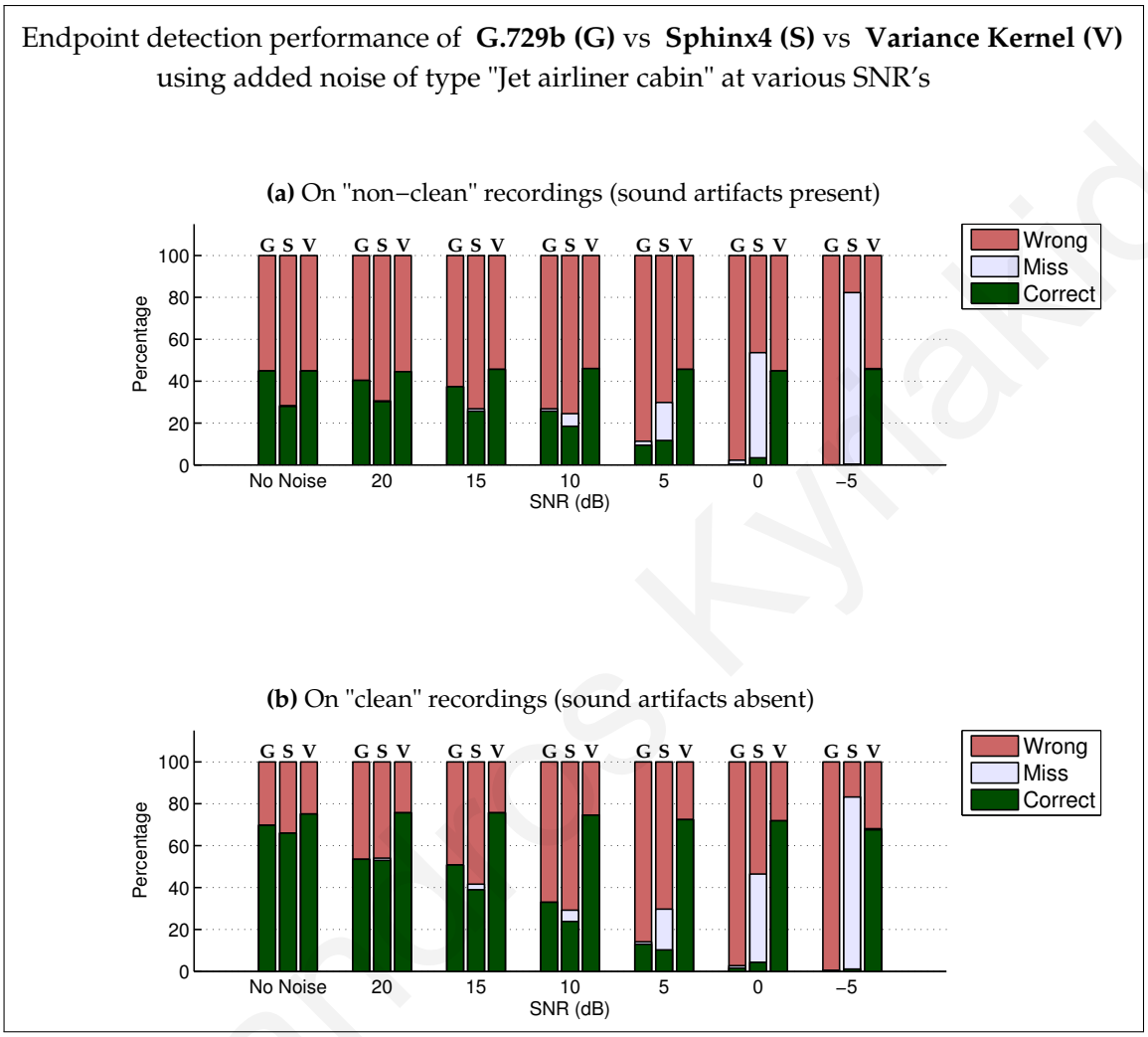

Figure A.20: A comparison of the endpoint detection performance of three different methods, using added noise of type "Jet airliner cabin" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.20: Endpoint detection results for three different methods using *added noise of type "Jet airliner cabin"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	40.38	30.19	44.53	53.51	52.97	75.68
	15 dB	37.36	25.66	45.66	50.81	38.92	75.68
	10 dB	25.66	18.49	46.04	32.97	23.78	74.59
	5 dB	9.43	11.70	45.66	12.97	10.27	72.43
	0 dB	0.38	3.40	44.91	1.62	4.32	71.89
	-5 dB	0.00	0.38	45.66	0.00	1.08	67.57
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.00	0.38	0.00	0.00	1.08	0.00
	15 dB	0.00	1.13	0.00	0.00	2.70	0.00
	10 dB	1.13	6.04	0.00	0.00	5.41	0.00
	5 dB	1.89	18.11	0.00	1.08	19.46	0.00
	0 dB	1.89	50.19	0.00	1.08	42.16	0.00
	-5 dB	0.00	81.89	0.38	0.54	82.16	0.54
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	59.62	69.43	55.47	46.49	45.95	24.32
	15 dB	62.64	73.21	54.34	49.19	58.38	24.32
	10 dB	73.21	75.47	53.96	67.03	70.81	25.41
	5 dB	88.68	70.19	54.34	85.95	70.27	27.57
	0 dB	97.74	46.41	55.09	97.30	53.51	28.11
	-5 dB	100.00	17.74	53.96	99.46	16.76	31.89

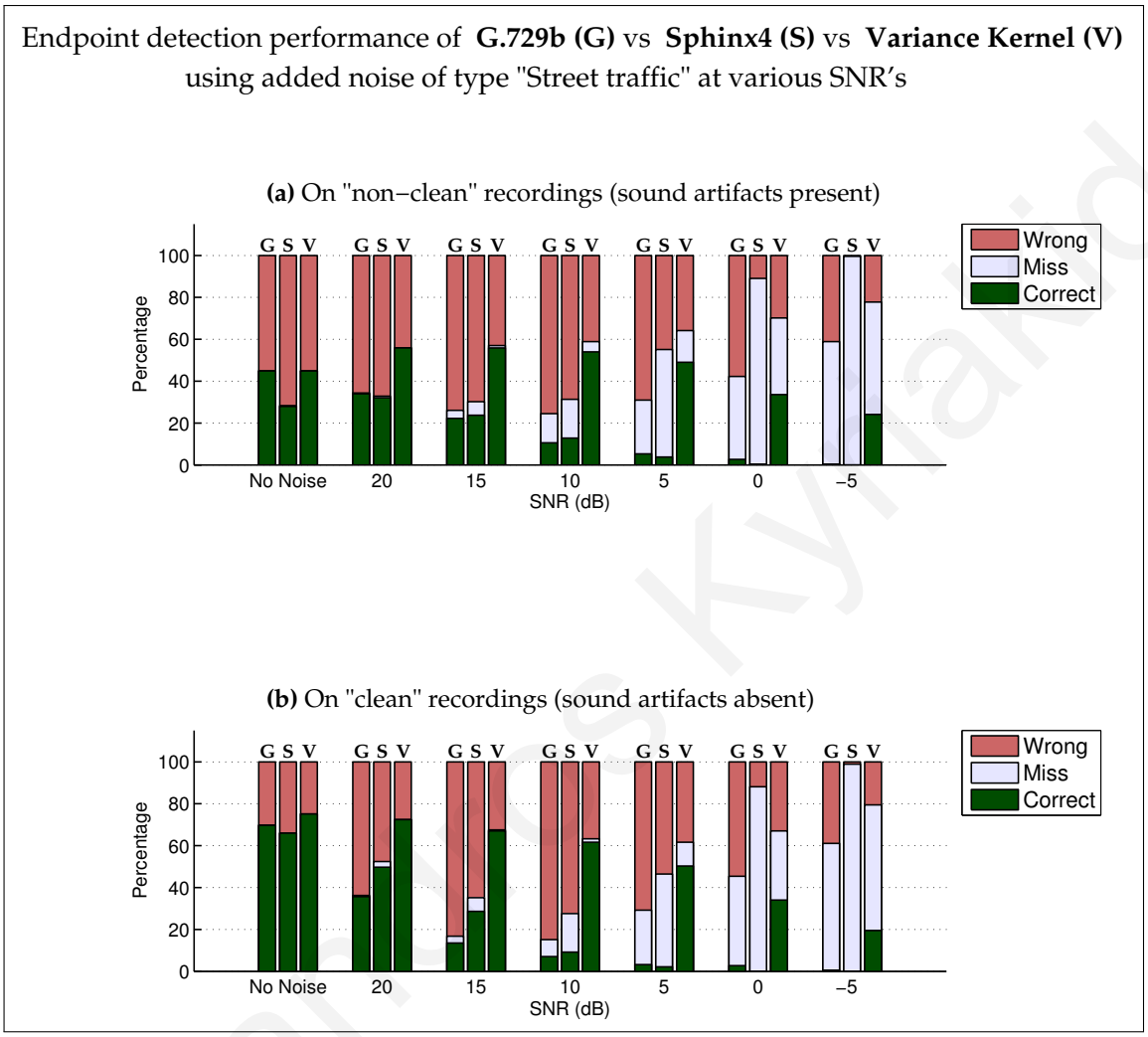

Figure A.21: A comparison of the endpoint detection performance of three different methods, using added noise of type "Street traffic" at various SNR's. The results are shown as a percentage of a total of 265 "non-clean" recordings and 185 "clean" recordings. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

Table A.21: Endpoint detection results for three different methods using *added noise of type "Street traffic"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The calculated endpoints were determined to be **Correct** or **Wrong** by comparing them to manually labeled endpoints. A **Miss** occurs when the endpoint detection system does not detect any speech in the recording.

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
Correct (%)	**no noise**	44.91	27.93	44.91	69.73	65.95	75.14
	20 dB	33.96	32.08	55.85	35.68	49.73	72.43
	15 dB	22.26	23.77	55.85	13.51	28.65	67.03
	10 dB	10.57	12.83	53.96	7.03	9.19	61.62
	5 dB	5.28	3.77	49.06	3.24	2.16	50.27
	0 dB	2.64	0.38	33.59	2.70	0.00	34.05
	-5 dB	0.38	0.00	24.15	0.54	0.00	19.46
Miss (%)	**no noise**	0.00	0.38	0.00	0.00	0.00	0.00
	20 dB	0.38	0.75	0.00	0.54	2.70	0.00
	15 dB	3.77	6.42	1.13	3.24	6.49	0.54
	10 dB	13.96	18.49	4.91	8.11	18.38	1.62
	5 dB	25.66	51.32	15.09	25.95	44.32	11.35
	0 dB	39.62	88.68	36.60	42.70	88.11	32.97
	-5 dB	58.49	99.62	53.59	60.54	98.92	60.00
Wrong (%)	**no noise**	55.09	71.70	55.09	30.27	34.05	24.86
	20 dB	65.66	67.17	44.15	63.78	47.57	27.57
	15 dB	73.96	69.81	43.02	83.24	64.86	32.43
	10 dB	75.47	68.68	41.13	84.86	72.43	36.76
	5 dB	69.06	44.91	35.85	70.81	53.51	38.38
	0 dB	57.74	10.94	29.81	54.59	11.89	32.97
	-5 dB	41.13	0.38	22.26	38.92	1.08	20.54

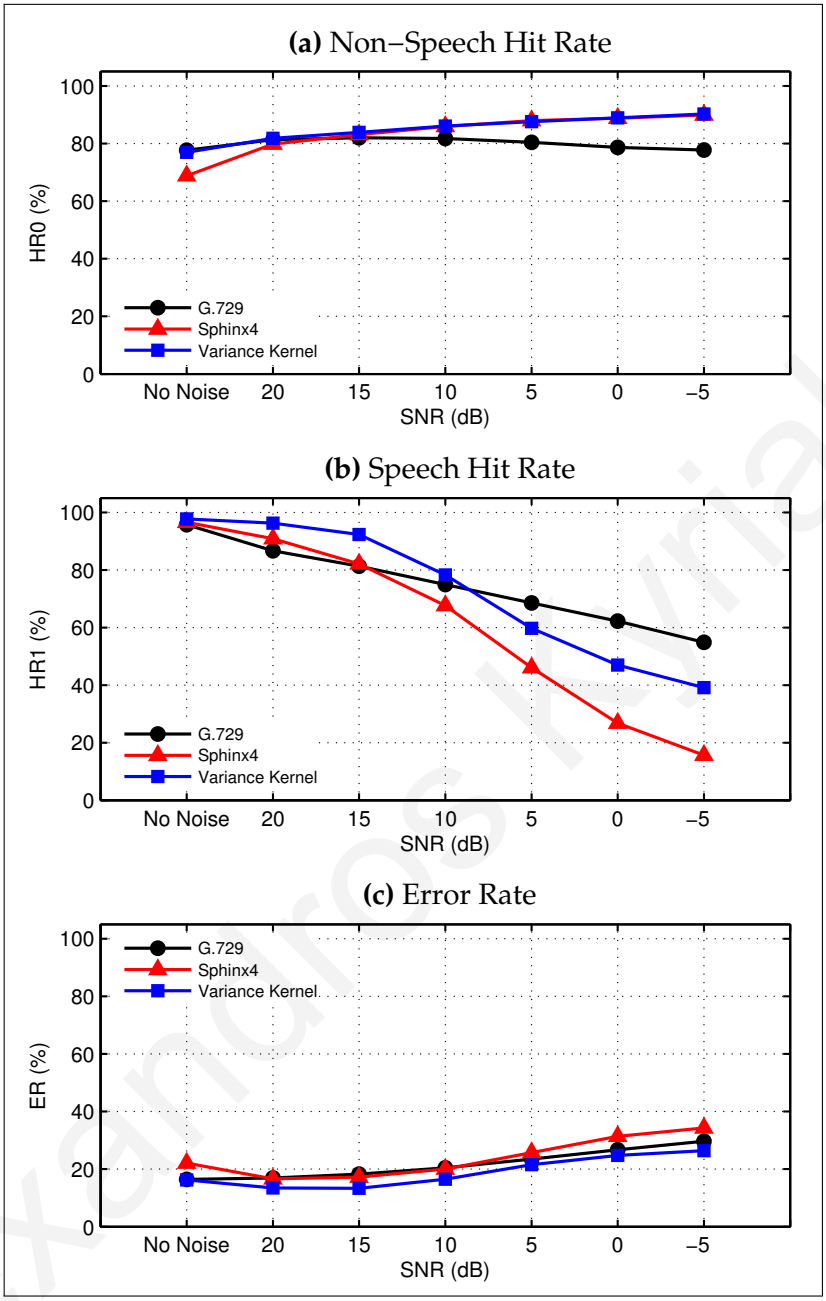

Figure A.22: A comparison of the voice activity detection performance of three different methods, using twenty types of added noise at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

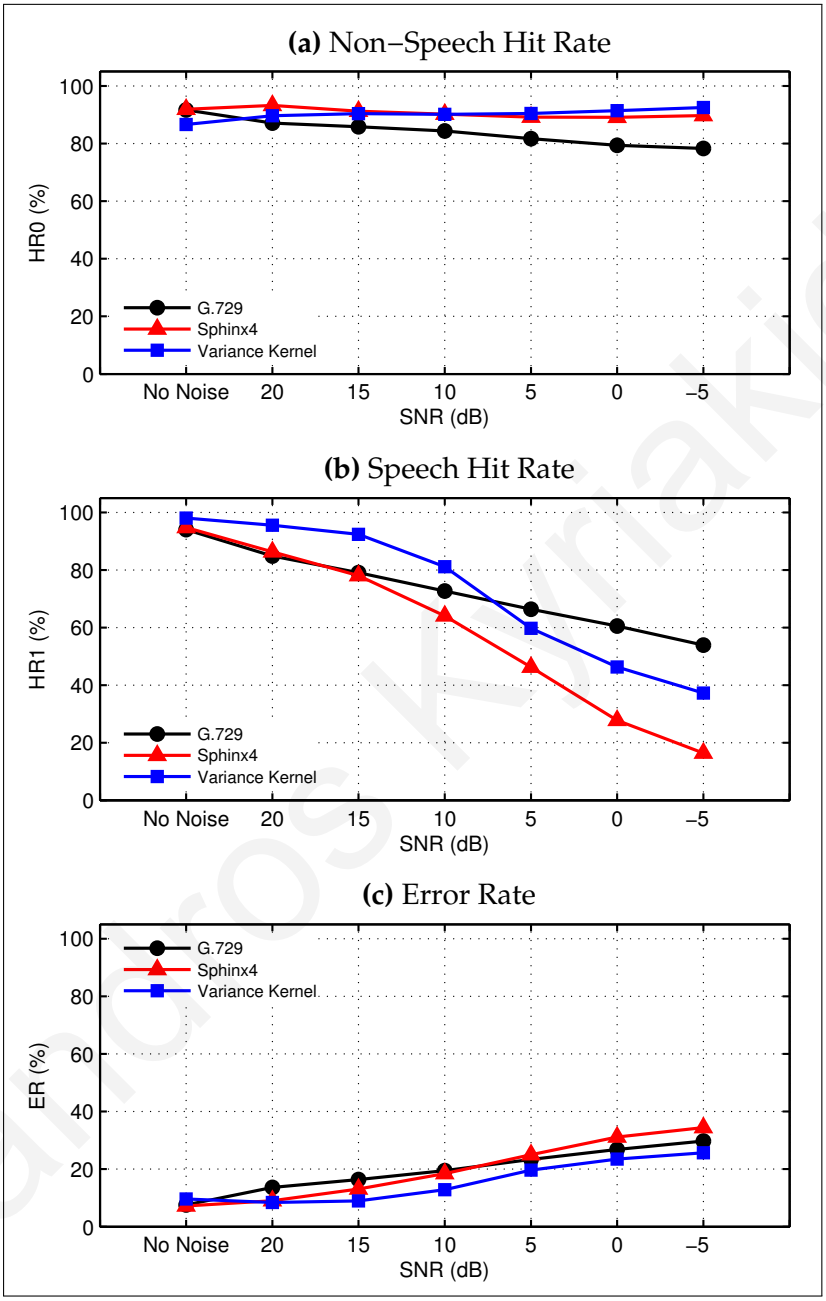

Figure A.23: A comparison of the voice activity detection performance of three different methods, using twenty types of added noise at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.22: Voice activity detection performance for three different methods using *twenty types of added noise* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	81.31	79.81	81.80	87.10	93.26	89.68
	15 dB	81.98	83.11	83.87	85.83	91.26	90.40
	10 dB	81.73	85.97	86.07	84.34	90.20	90.11
	5 dB	80.40	87.95	87.57	81.72	89.15	90.46
	0 dB	78.68	88.83	88.93	79.42	89.11	91.40
	-5 dB	77.77	89.90	90.26	78.28	89.73	92.53
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	86.70	90.83	96.25	84.86	86.28	95.53
	15 dB	81.33	82.13	92.35	79.10	78.06	92.38
	10 dB	74.96	67.63	78.25	72.69	64.06	81.16
	5 dB	68.56	46.03	59.72	66.34	46.21	59.75
	0 dB	62.25	26.79	46.99	60.55	27.75	46.30
	-5 dB	54.90	15.66	39.14	53.92	16.39	37.27
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	16.93	16.59	13.49	13.64	9.04	8.39
	15 dB	18.23	17.21	13.36	16.39	13.09	8.95
	10 dB	20.48	20.01	16.48	19.50	18.41	12.84
	5 dB	23.46	25.72	21.51	23.35	25.00	19.66
	0 dB	26.68	31.40	24.74	26.80	31.11	23.46
	-5 dB	29.68	34.30	26.40	29.75	34.44	25.68

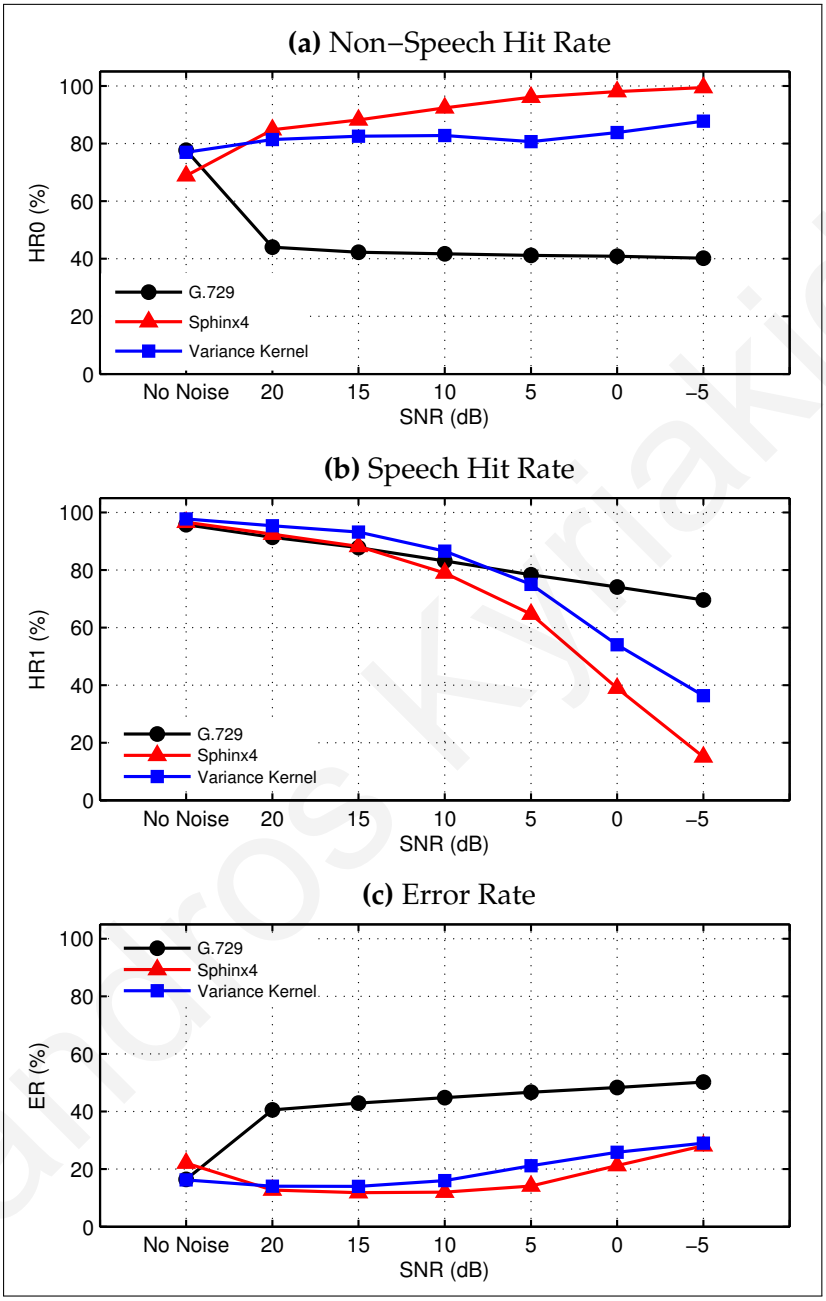

Figure A.24: A comparison of the voice activity detection performance of three different methods, using added noise of type "speech babble" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

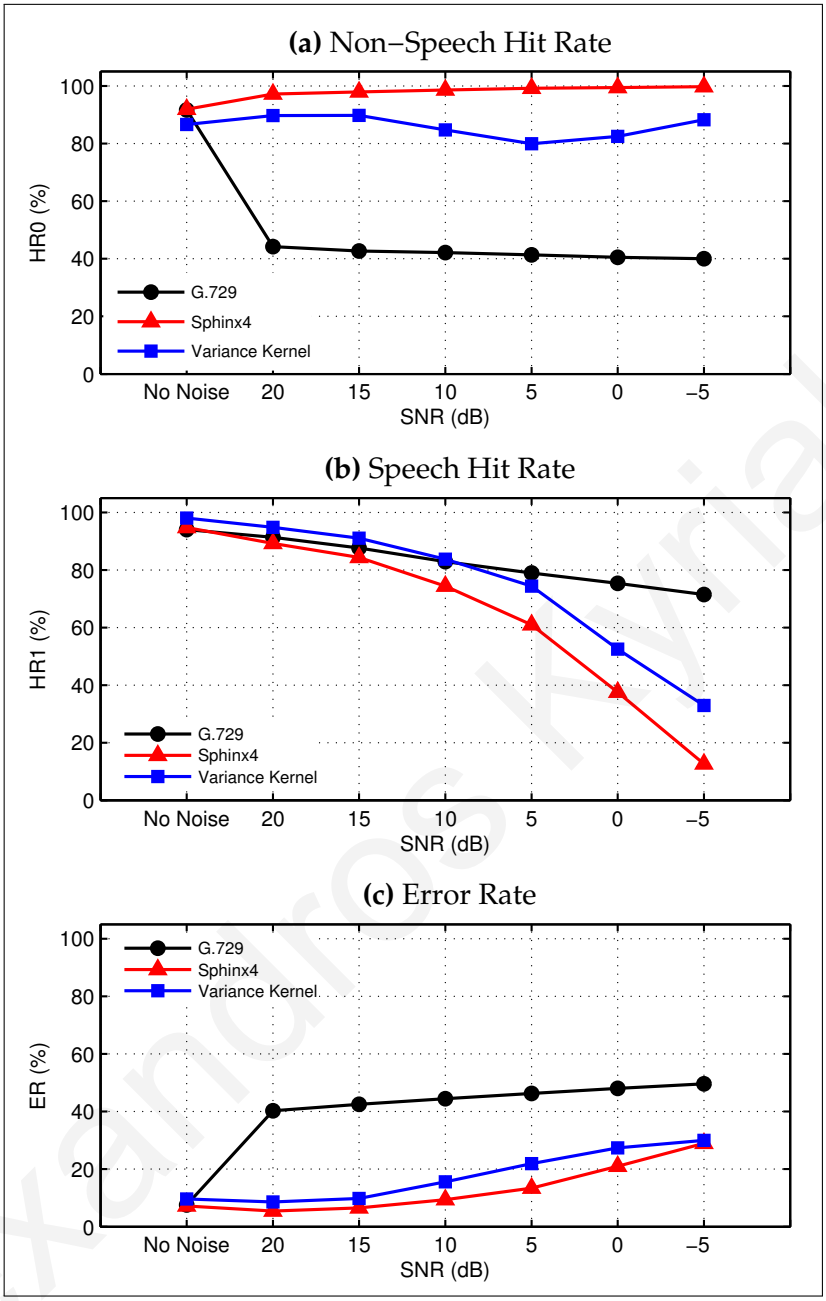

Figure A.25: A comparison of the voice activity detection performance of three different methods, using added noise of type "speech babble" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.23: Voice activity detection performance for three different methods using *added noise of type "speech babble"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	44.02	84.80	81.40	44.21	97.20	89.69
	15 dB	42.26	88.17	82.54	42.67	97.92	89.76
	10 dB	41.72	92.37	82.78	42.13	98.62	84.74
	5 dB	41.18	96.11	80.69	41.38	99.20	79.95
	0 dB	40.83	98.03	83.84	40.50	99.41	82.50
	-5 dB	40.19	99.44	87.79	40.01	99.74	88.23
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	91.37	92.44	95.34	91.33	89.23	94.81
	15 dB	87.77	88.16	93.16	87.63	84.36	91.03
	10 dB	83.12	78.97	86.52	82.92	74.40	83.73
	5 dB	78.35	64.65	74.99	78.93	60.95	74.37
	0 dB	74.07	38.92	54.02	75.34	37.57	52.50
	-5 dB	69.59	15.06	36.33	71.45	12.63	32.92
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	40.55	12.71	14.05	40.26	5.43	8.62
	15 dB	42.91	11.83	14.00	42.51	6.55	9.82
	10 dB	44.78	12.00	16.00	44.43	9.36	15.59
	5 dB	46.70	14.15	21.17	46.25	13.40	21.89
	0 dB	48.33	21.24	25.88	48.02	20.97	27.39
	-5 dB	50.22	28.07	28.99	49.63	28.97	30.00

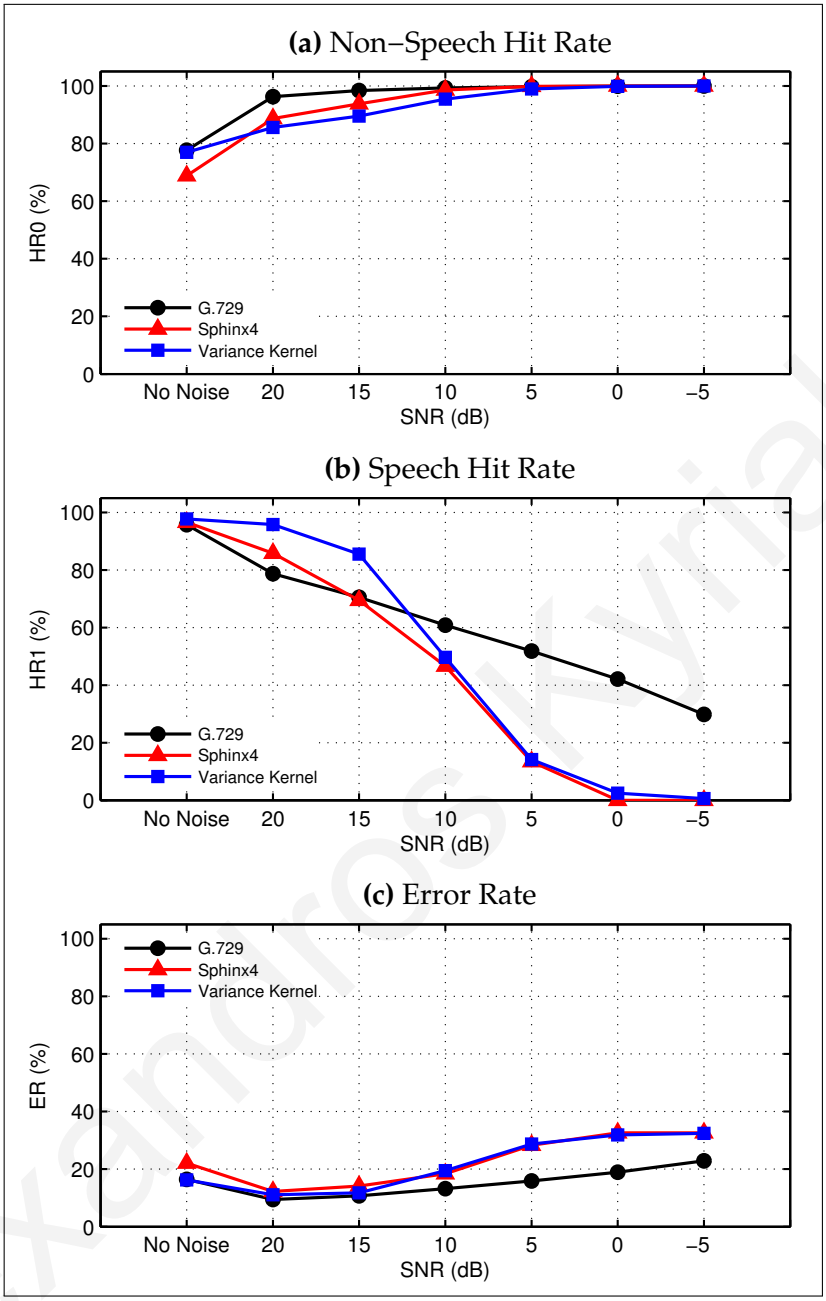

Figure A.26: A comparison of the voice activity detection performance of three different methods, using added noise of type "Buccaneer jet cockpit (190 knots)" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

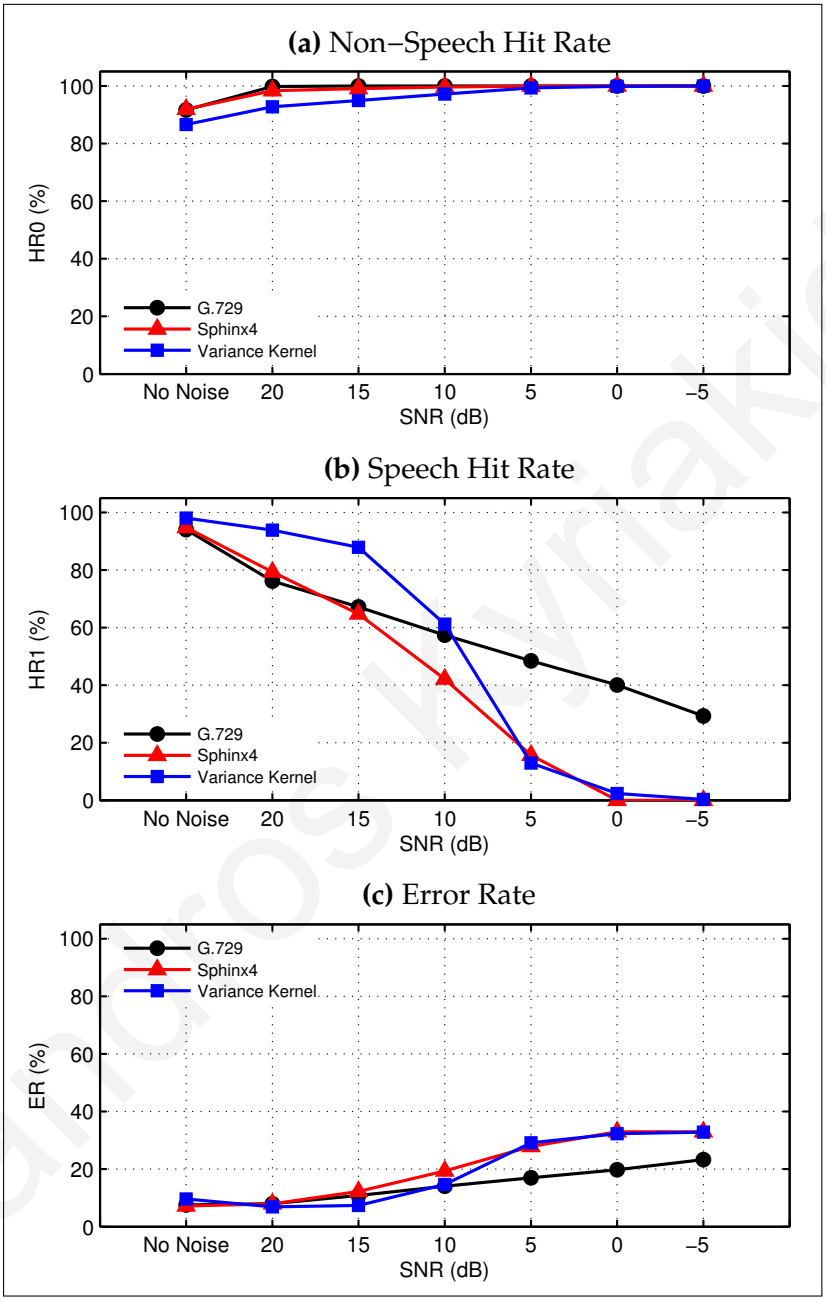

Figure A.27: A comparison of the voice activity detection performance of three different methods, using added noise of type "Buccaneer jet cockpit (190 knots)" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.24: Voice activity detection performance for three different methods using *added noise of type "Buccaneer jet cockpit (190 knots)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	96.29	88.70	85.59	99.80	98.35	92.75
	15 dB	98.39	93.80	89.56	99.97	99.07	94.95
	10 dB	99.32	98.57	95.44	100.00	99.62	97.21
	5 dB	99.76	99.89	98.96	99.99	99.95	99.33
	0 dB	99.91	100.00	99.85	100.00	100.00	99.88
	-5 dB	99.99	100.00	99.97	100.00	100.00	100.00
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	78.71	85.82	95.78	76.12	79.29	93.83
	15 dB	70.50	69.45	85.49	67.15	64.74	87.89
	10 dB	60.83	46.63	49.66	57.38	42.12	61.19
	5 dB	51.81	13.43	14.15	48.41	15.63	12.94
	0 dB	42.08	0.00	2.49	40.02	0.00	2.32
	-5 dB	29.80	0.00	0.58	29.29	0.00	0.29
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	9.44	12.24	11.09	8.01	7.93	6.90
	15 dB	10.70	14.14	11.77	10.85	12.24	7.37
	10 dB	13.23	18.37	19.49	14.05	19.33	14.66
	5 dB	15.87	28.30	28.69	17.01	27.84	29.14
	0 dB	18.95	32.60	31.89	19.77	32.96	32.27
	-5 dB	22.89	32.60	32.44	23.30	32.96	32.87

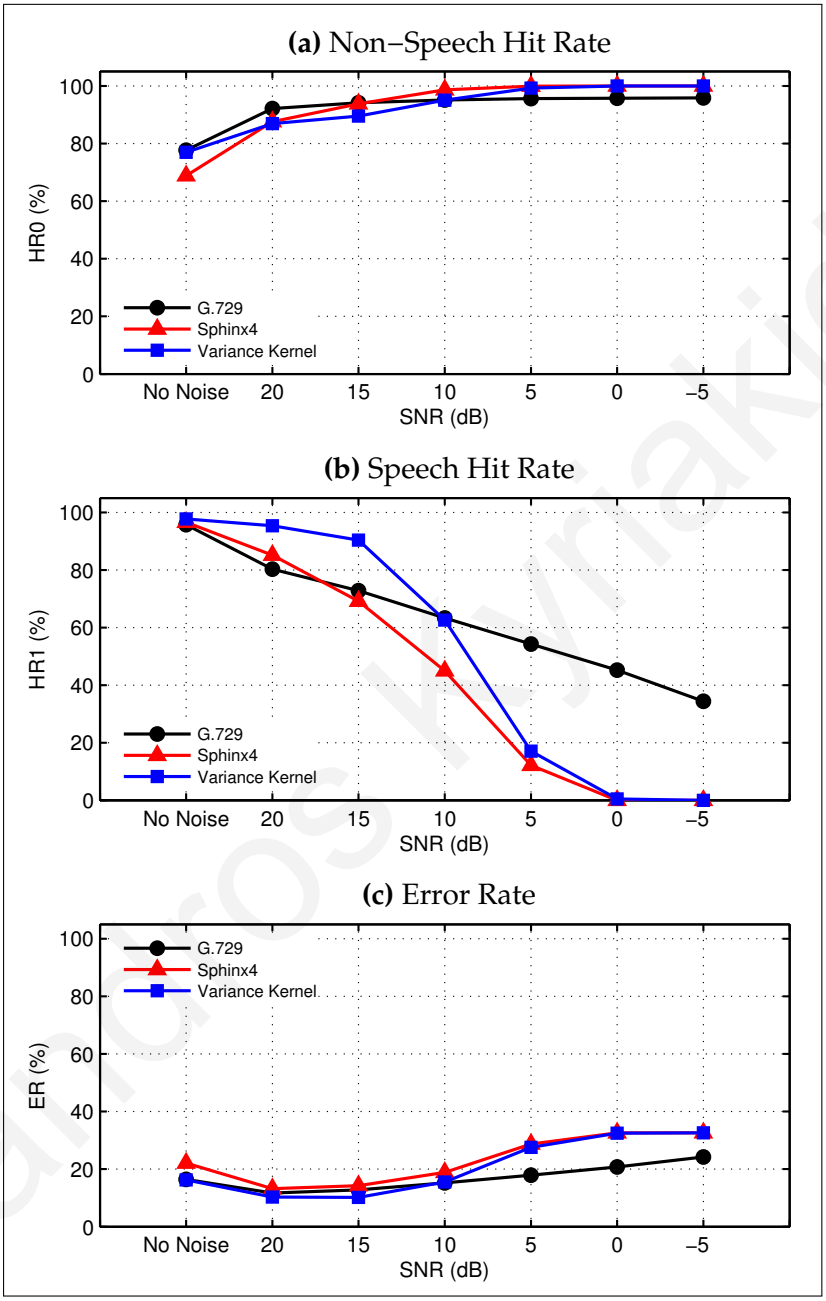

Figure A.28: A comparison of the voice activity detection performance of three different methods, using added noise of type "Buccaneer jet cockpit (450 knots)" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

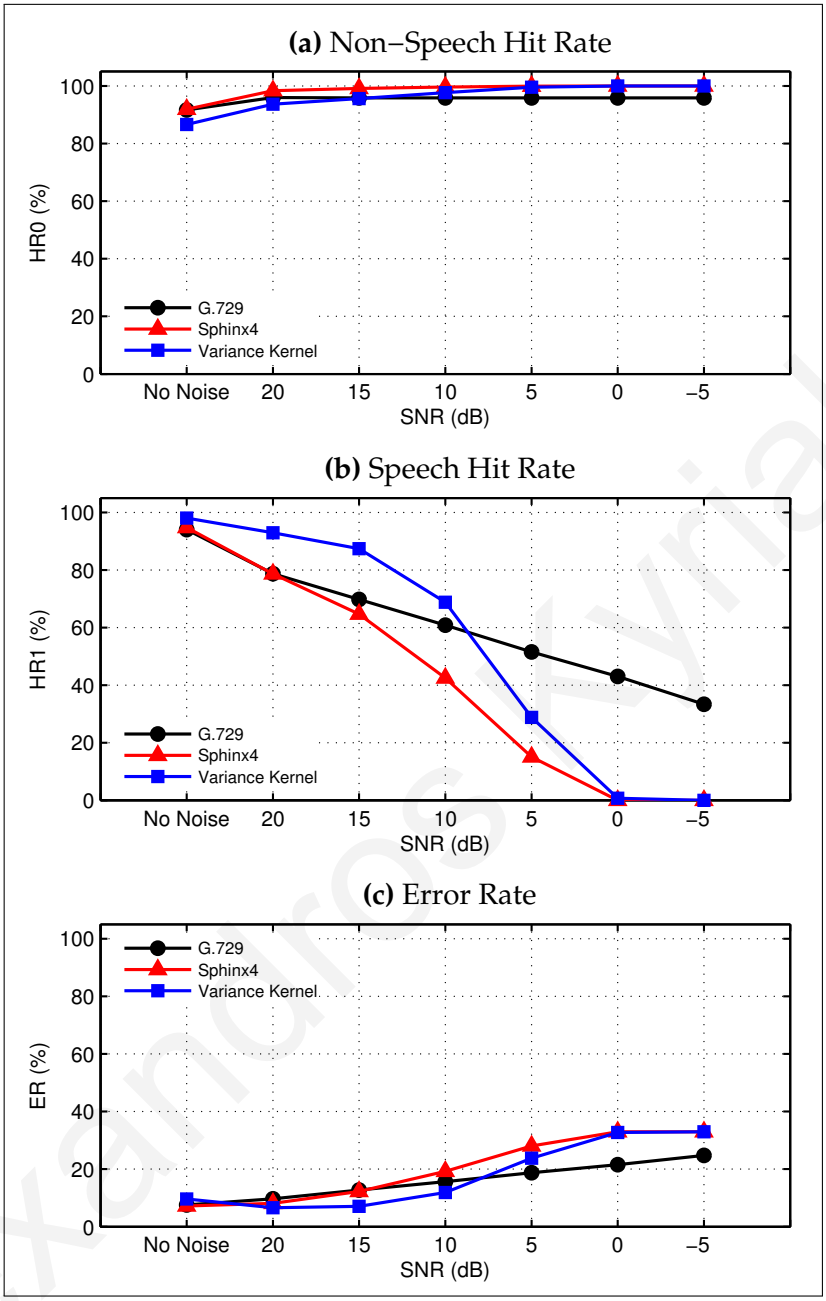

Figure A.29: A comparison of the voice activity detection performance of three different methods, using added noise of type "Buccaneer jet cockpit (450 knots)" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.25: Voice activity detection performance for three different methods using *added noise of type "Buccaneer jet cockpit (450 knots)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	92.17	87.58	86.94	95.99	98.36	93.67
	15 dB	94.16	93.77	89.56	95.84	99.14	95.61
	10 dB	95.13	98.65	95.06	95.87	99.62	97.65
	5 dB	95.58	99.92	99.25	95.86	99.96	99.55
	0 dB	95.74	100.00	100.00	95.85	100.00	99.98
	-5 dB	95.85	100.00	100.00	95.83	100.00	100.00
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	80.29	85.15	95.36	78.69	78.66	92.93
	15 dB	72.81	69.14	90.36	69.75	64.65	87.37
	10 dB	63.32	44.98	62.60	60.85	42.44	68.77
	5 dB	54.26	12.14	17.08	51.51	15.05	28.82
	0 dB	45.20	0.00	0.44	43.02	0.00	0.68
	-5 dB	34.35	0.00	0.00	33.36	0.00	0.00
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	11.71	13.21	10.32	9.71	8.13	6.57
	15 dB	12.80	14.26	10.18	12.76	12.23	7.10
	10 dB	15.24	18.84	15.52	15.67	19.23	11.87
	5 dB	17.89	28.70	27.54	18.75	28.03	23.76
	0 dB	20.74	32.60	32.46	21.56	32.96	32.74
	-5 dB	24.20	32.60	32.60	24.76	32.96	32.96

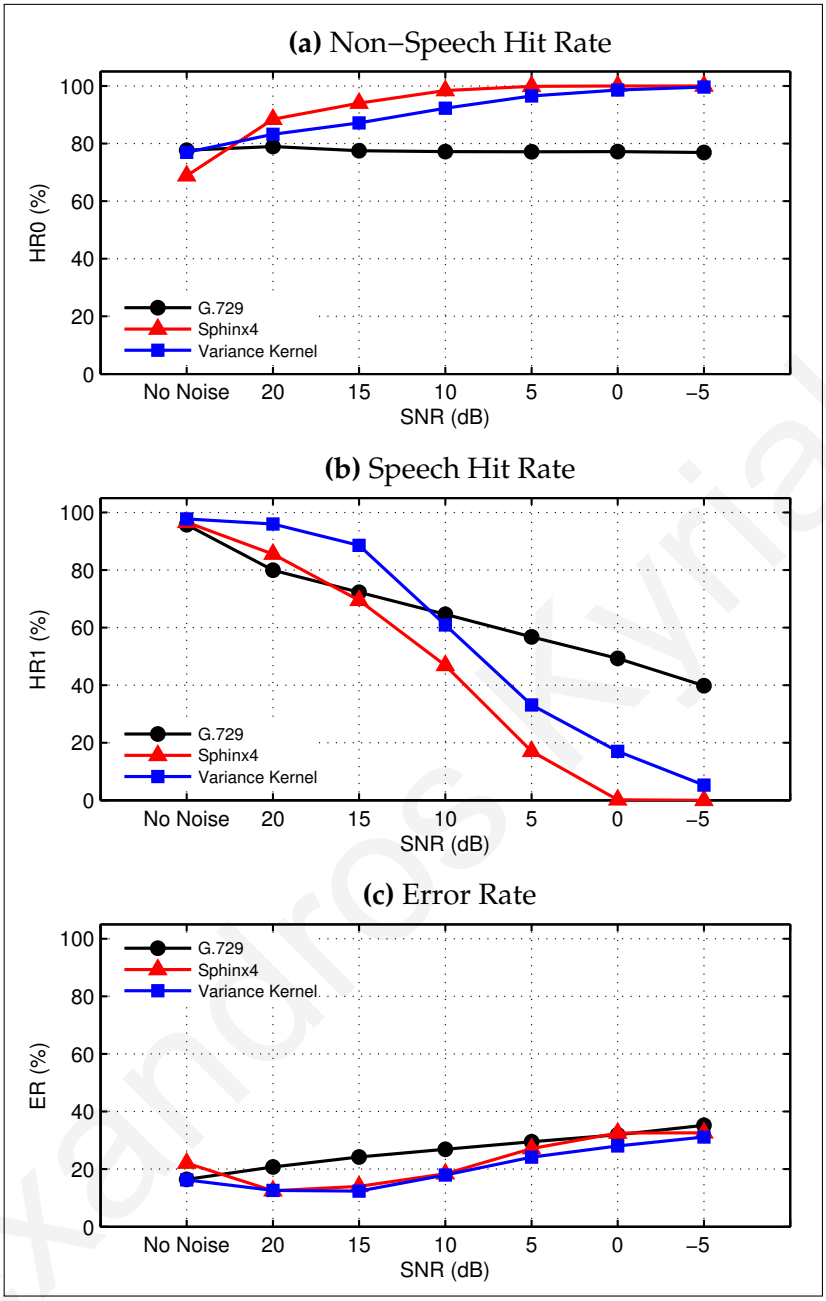

Figure A.30: A comparison of the voice activity detection performance of three different methods, using added noise of type "Destroyer engine room" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

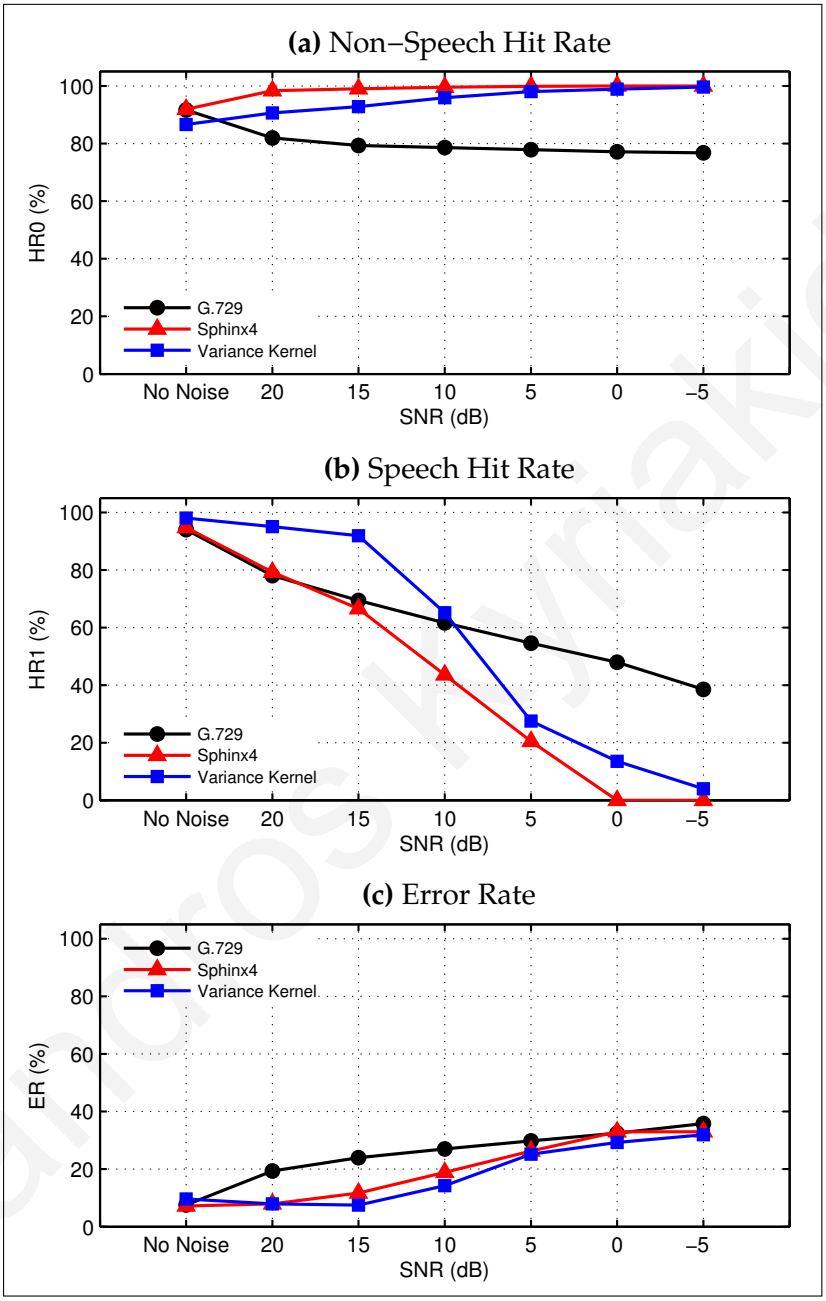

Figure A.31: A comparison of the voice activity detection performance of three different methods, using added noise of type "Destroyer engine room" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.26: Voice activity detection performance for three different methods using *added noise of type "Destroyer engine room"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	78.97	88.44	83.23	81.97	98.33	90.63
	15 dB	77.52	94.01	87.18	79.31	99.03	92.80
	10 dB	77.21	98.40	92.27	78.59	99.57	95.90
	5 dB	77.13	99.87	96.55	77.87	99.89	98.05
	0 dB	77.18	100.00	98.60	77.17	100.00	98.88
	-5 dB	76.90	100.00	99.62	76.75	100.00	99.62
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	79.89	85.52	95.94	78.03	79.26	95.08
	15 dB	72.29	69.49	88.56	69.40	66.56	91.92
	10 dB	64.60	46.84	60.88	61.61	43.54	65.14
	5 dB	56.75	16.99	33.08	54.59	20.43	27.50
	0 dB	49.26	0.17	16.96	47.94	0.00	13.50
	-5 dB	39.77	0.00	5.28	38.54	0.00	3.98
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	20.73	12.51	12.62	19.33	7.96	7.90
	15 dB	24.19	13.98	12.37	23.95	11.67	7.49
	10 dB	26.90	18.41	17.97	27.00	18.89	14.24
	5 dB	29.52	27.15	24.14	29.80	26.30	25.20
	0 dB	31.93	32.55	28.01	32.47	32.96	29.26
	-5 dB	35.21	32.60	31.14	35.84	32.96	31.90

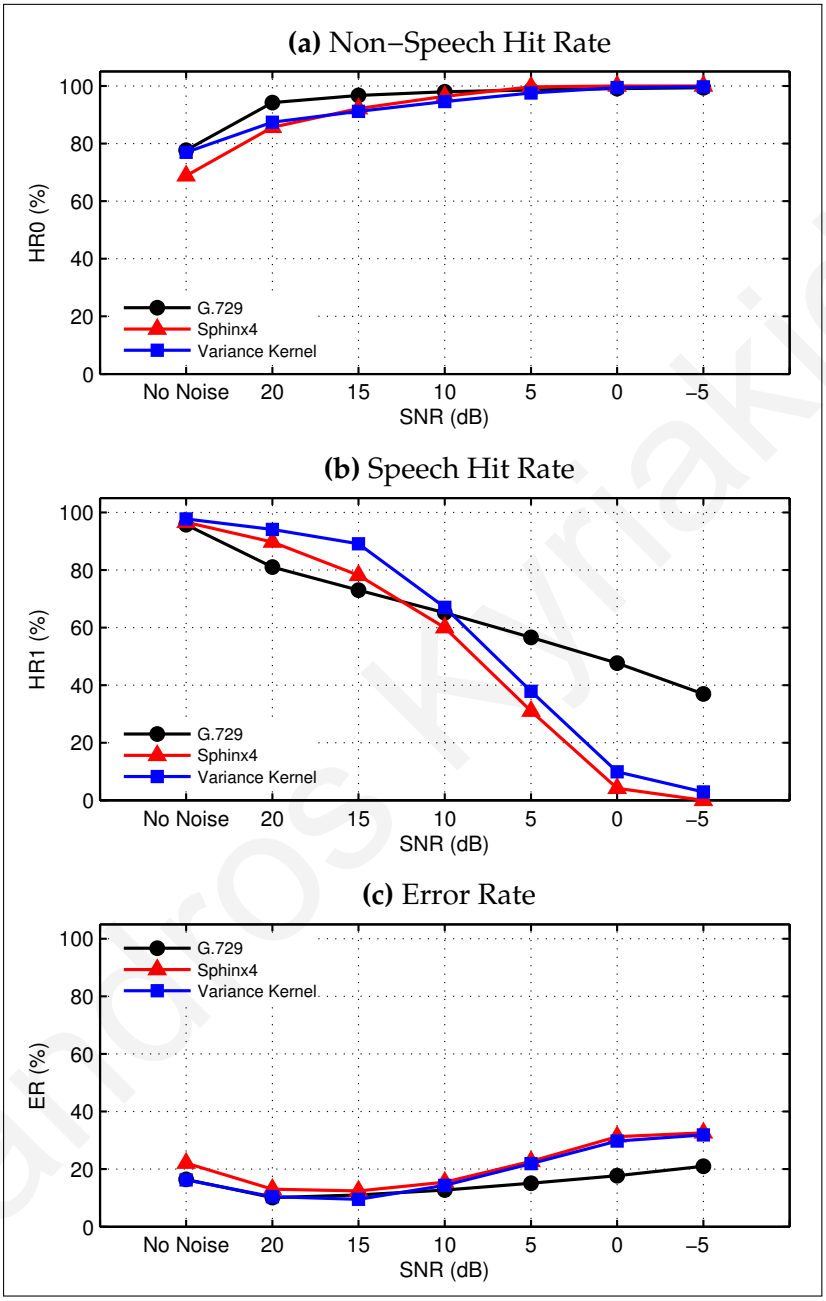

Figure A.32: A comparison of the voice activity detection performance of three different methods, using added noise of type "Destroyer operations room" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

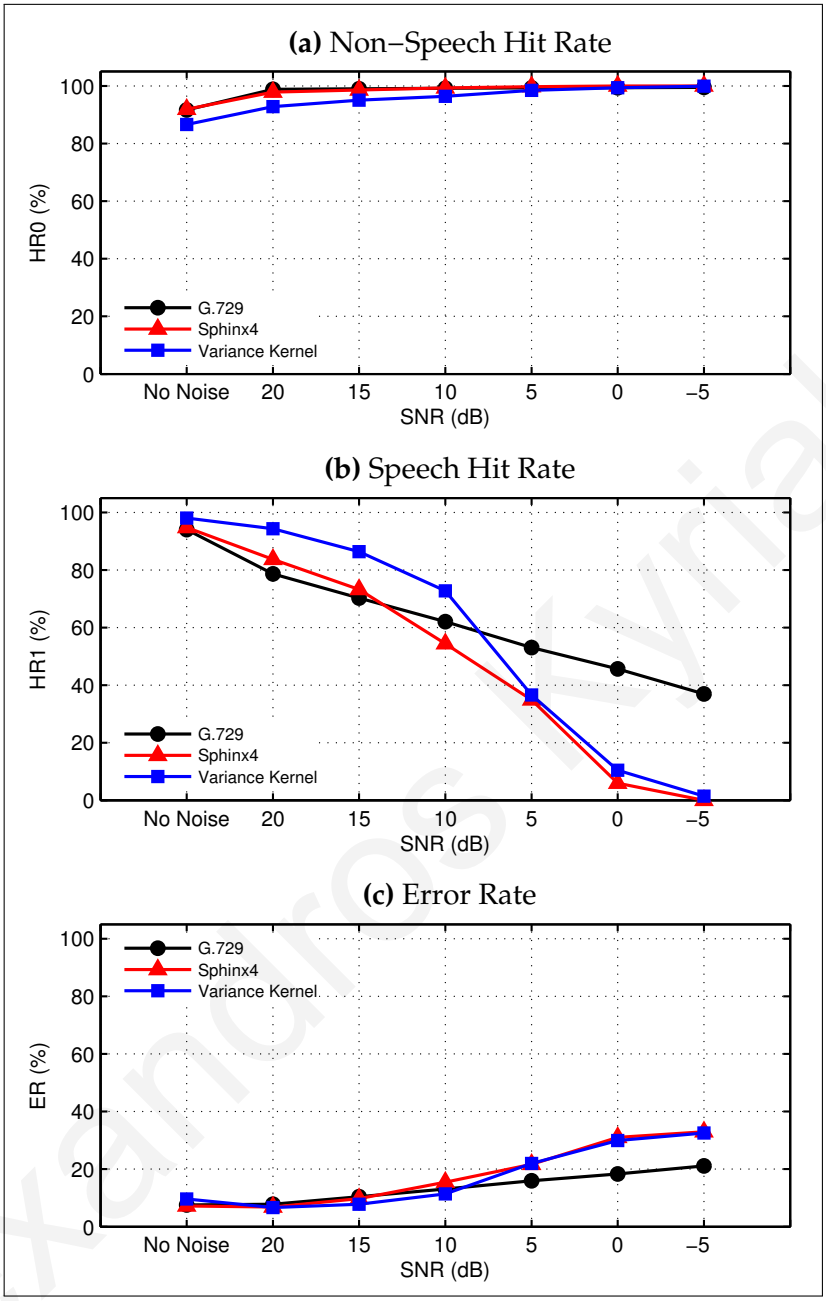

Figure A.33: A comparison of the voice activity detection performance of three different methods, using added noise of type "Destroyer operations room" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.27: Voice activity detection performance for three different methods using *added noise of type "Destroyer operations room"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	94.19	85.65	87.39	98.82	97.86	92.84
	15 dB	96.71	92.12	91.19	99.04	98.60	95.09
	10 dB	97.98	96.43	94.59	99.17	99.31	96.43
	5 dB	98.59	99.67	97.58	99.31	99.72	98.44
	0 dB	99.07	99.96	99.43	99.40	99.99	99.38
	-5 dB	99.36	100.00	99.68	99.52	100.00	99.95
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	81.04	89.69	94.10	78.63	83.68	94.32
	15 dB	73.00	78.14	89.08	70.24	73.21	86.39
	10 dB	65.15	60.00	67.05	62.02	54.40	72.72
	5 dB	56.56	31.00	37.87	53.08	34.89	36.56
	0 dB	47.66	4.15	9.87	45.61	5.90	10.44
	-5 dB	36.97	0.00	2.87	36.92	0.00	1.41
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	10.10	13.03	10.42	7.83	6.81	6.67
	15 dB	11.02	12.44	9.50	10.45	9.77	7.78
	10 dB	12.72	15.45	14.39	13.07	15.49	11.38
	5 dB	15.11	22.71	21.89	15.92	21.65	21.95
	0 dB	17.69	31.27	29.77	18.33	31.02	29.93
	-5 dB	20.98	32.60	31.88	21.11	32.96	32.53

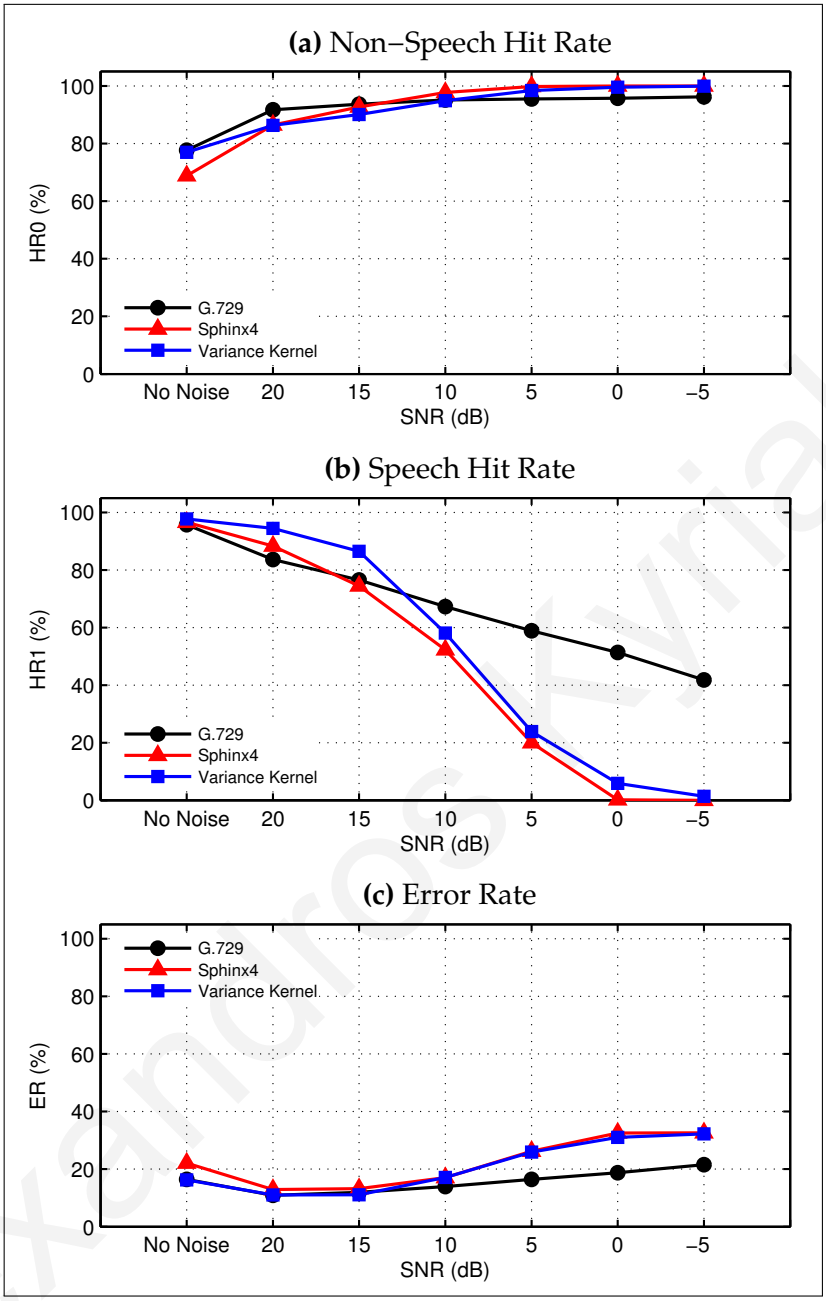

Figure A.34: A comparison of the voice activity detection performance of three different methods, using added noise of type "F-16 cockpit" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

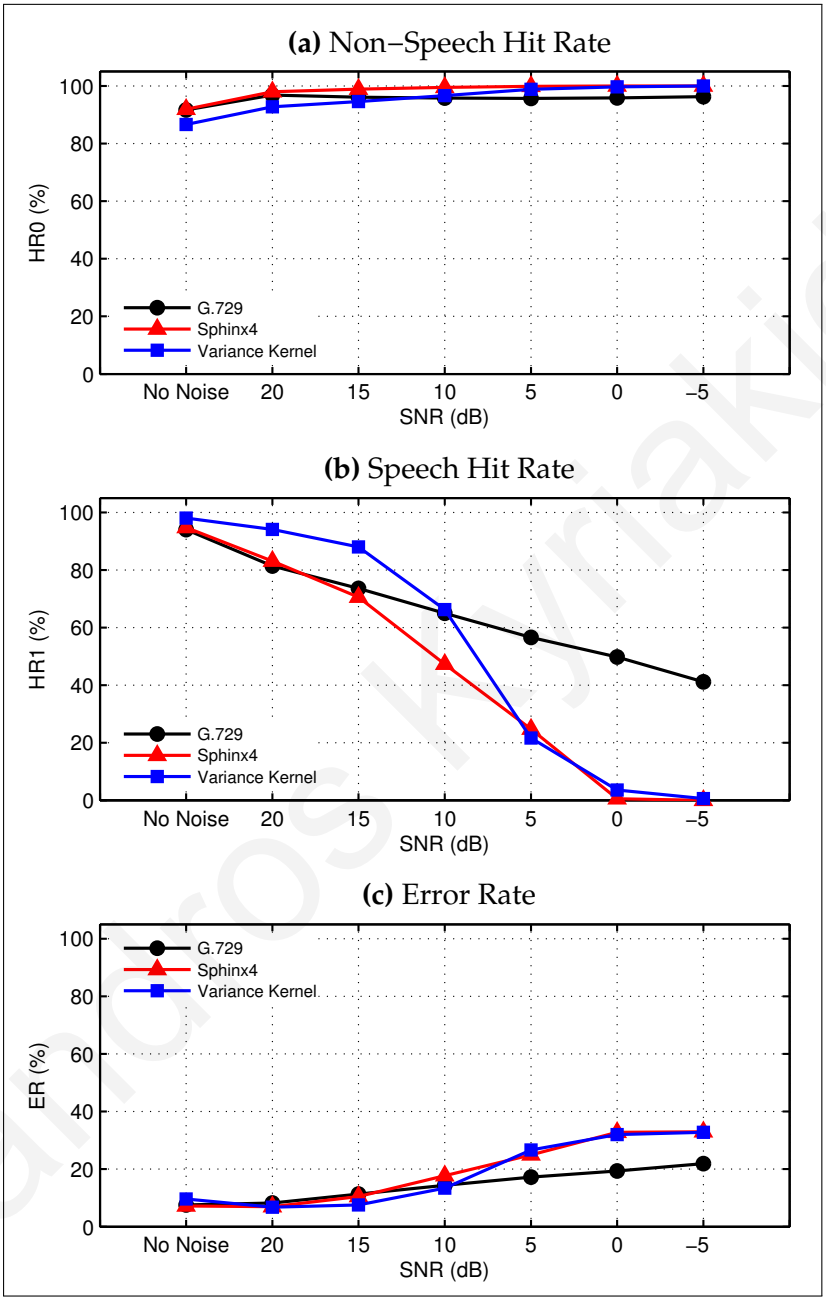

Figure A.35: A comparison of the voice activity detection performance of three different methods, using added noise of type "F-16 cockpit" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.28: Voice activity detection performance for three different methods using *added noise of type "F-16 cockpit"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	91.72	86.45	86.29	96.83	97.97	92.78
	15 dB	93.68	92.71	90.10	96.13	98.88	94.60
	10 dB	95.12	97.77	94.90	95.79	99.50	96.62
	5 dB	95.51	99.83	98.39	95.67	99.84	98.84
	0 dB	95.75	100.00	99.55	95.88	100.00	99.66
	-5 dB	96.23	100.00	99.92	96.29	100.00	99.98
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	83.61	88.34	94.47	81.38	83.08	94.07
	15 dB	76.49	74.47	86.49	73.57	70.54	88.02
	10 dB	67.30	52.28	58.10	64.99	47.30	66.22
	5 dB	58.87	19.98	23.89	56.56	24.71	21.61
	0 dB	51.37	0.17	5.87	49.75	0.49	3.56
	-5 dB	41.79	0.00	1.36	41.15	0.00	0.56
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	10.92	12.93	11.04	8.26	6.93	6.79
	15 dB	11.93	13.23	11.08	11.31	10.46	7.57
	10 dB	13.95	17.06	17.10	14.36	17.71	13.39
	5 dB	16.43	26.21	25.90	17.22	24.92	26.62
	0 dB	18.72	32.55	30.99	19.33	32.80	32.01
	-5 dB	21.52	32.60	32.21	21.88	32.96	32.78

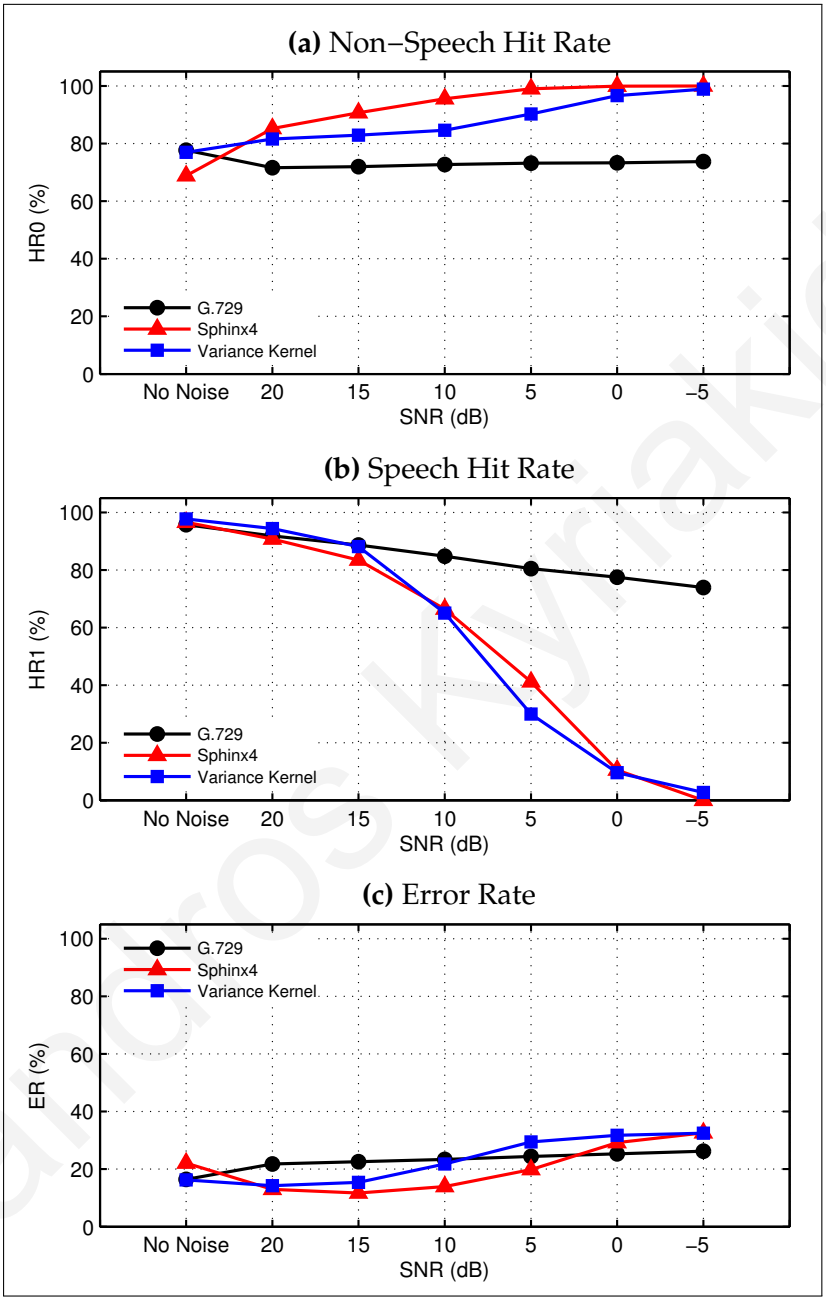

Figure A.36: A comparison of the voice activity detection performance of three different methods, using added noise of type "Factory floor (1)" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

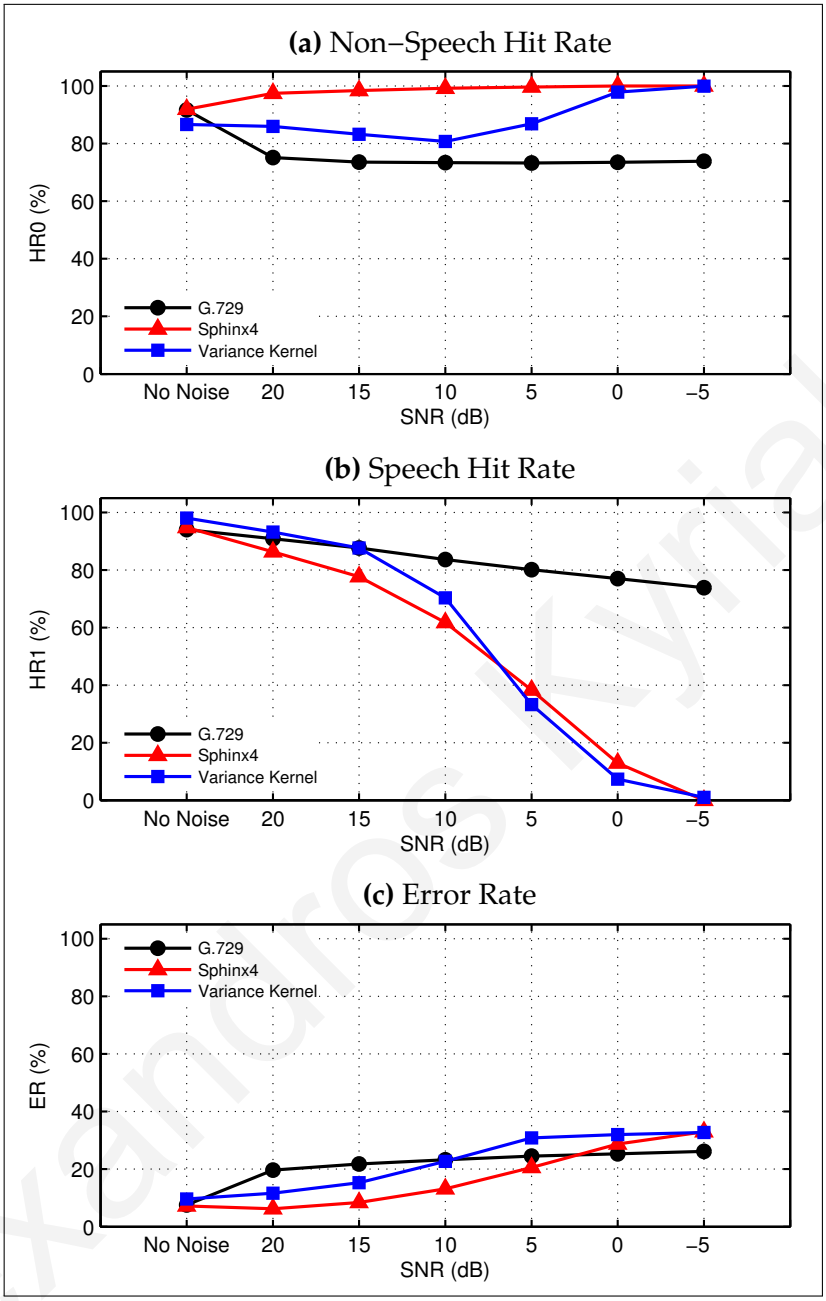

Figure A.37: A comparison of the voice activity detection performance of three different methods, using added noise of type "Factory floor (1)" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.29: Voice activity detection performance for three different methods using *added noise of type "Factory floor (1)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	71.61	85.23	81.55	75.13	97.46	85.98
	15 dB	71.96	90.68	82.89	73.55	98.41	83.22
	10 dB	72.71	95.54	84.64	73.39	99.17	80.72
	5 dB	73.22	99.00	90.26	73.23	99.64	86.84
	0 dB	73.32	99.95	96.63	73.52	99.96	97.84
	-5 dB	73.74	100.00	98.87	73.87	100.00	99.90
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	91.87	90.72	94.38	90.87	86.32	93.18
	15 dB	88.65	83.45	88.11	87.62	77.65	87.66
	10 dB	84.78	66.42	65.02	83.66	61.76	70.28
	5 dB	80.47	41.12	29.93	80.12	38.30	33.25
	0 dB	77.52	10.47	9.59	77.02	12.95	7.33
	-5 dB	73.92	0.00	2.68	73.85	0.00	1.01
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	21.78	12.97	14.27	19.68	6.21	11.65
	15 dB	22.60	11.68	15.41	21.81	8.43	15.31
	10 dB	23.36	13.95	21.76	23.23	13.16	22.72
	5 dB	24.42	19.87	29.41	24.50	20.58	30.82
	0 dB	25.31	29.22	31.75	25.32	28.72	31.99
	-5 dB	26.20	32.60	32.49	26.14	32.96	32.69

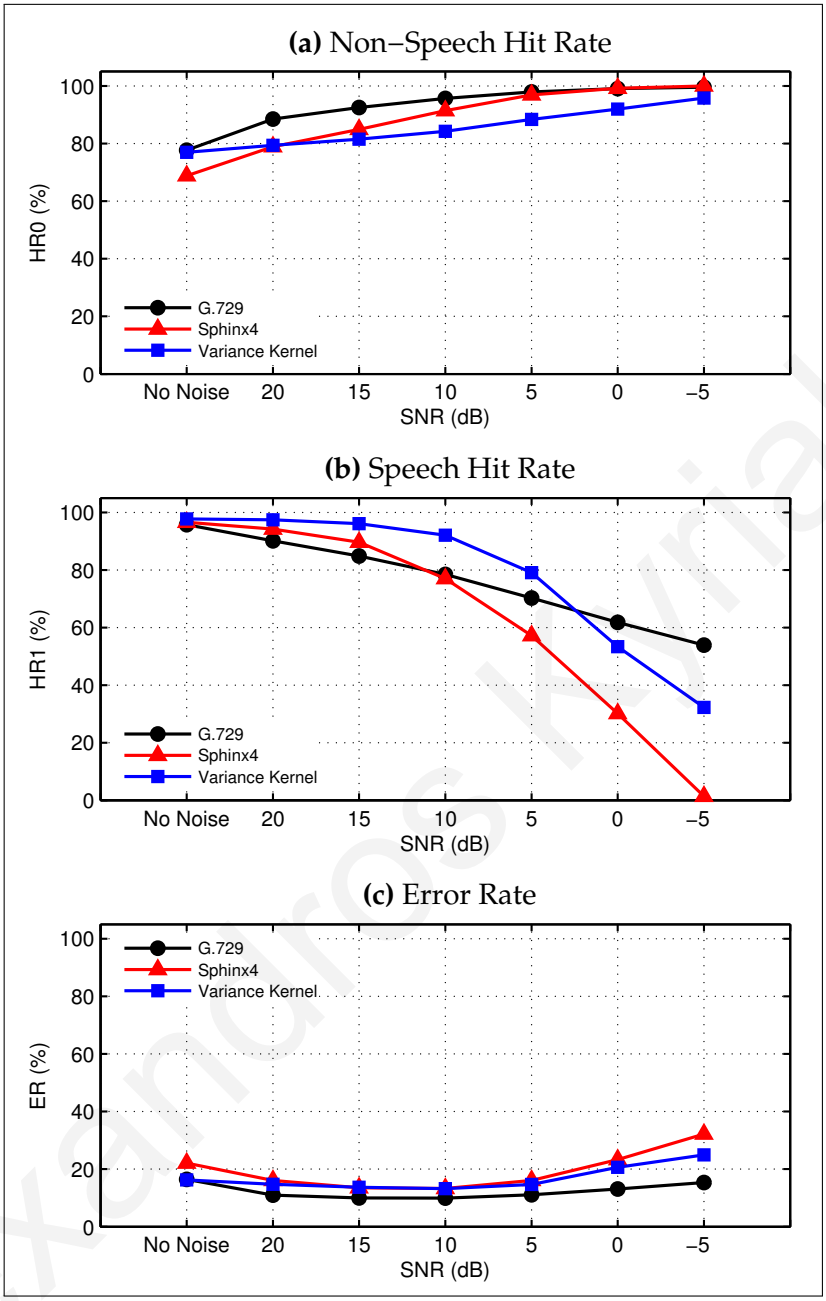

Figure A.38: A comparison of the voice activity detection performance of three different methods, using added noise of type "Factory floor (2)" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

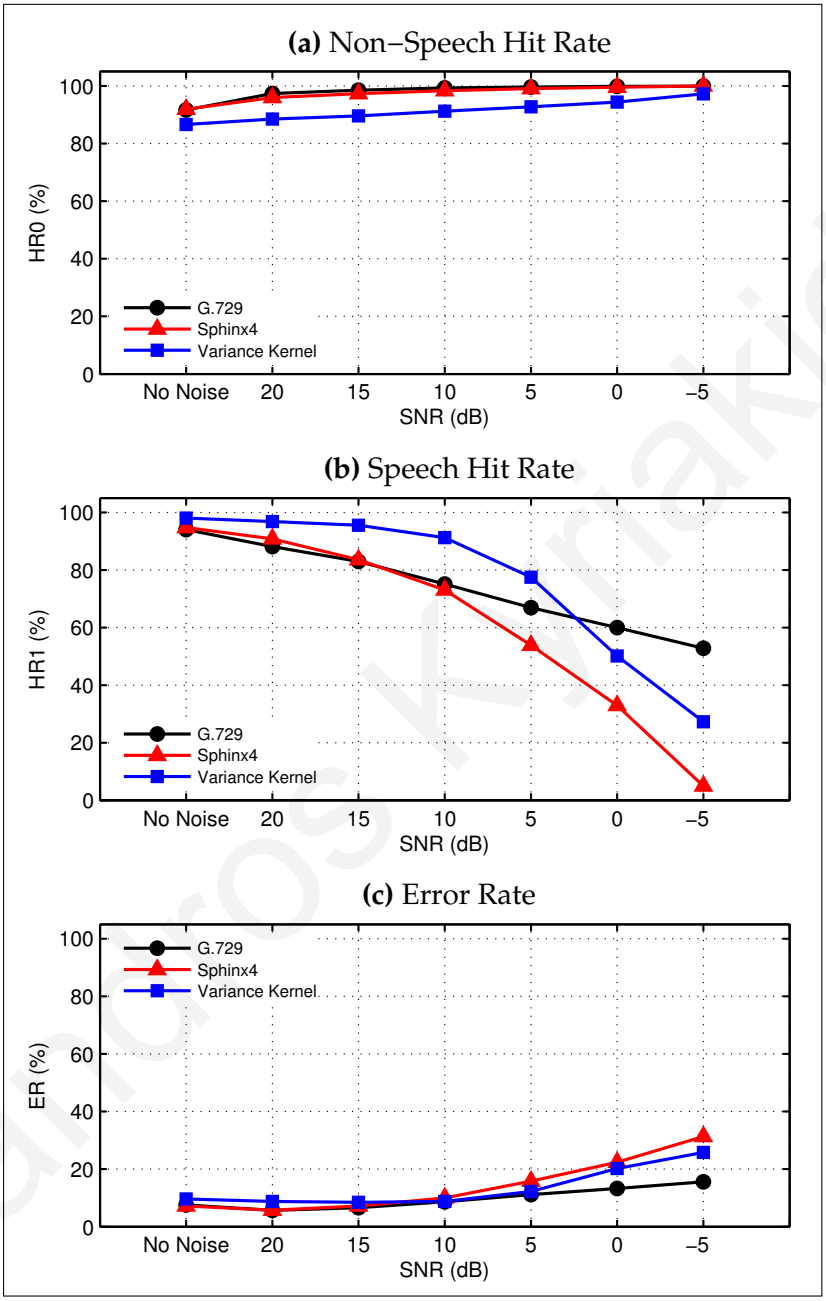

Figure A.39: A comparison of the voice activity detection performance of three different methods, using added noise of type "Factory floor (2)" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.30: Voice activity detection performance for three different methods using *added noise of type "Factory floor (2)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	88.50	78.93	79.40	97.37	95.99	88.50
	15 dB	92.53	84.95	81.52	98.53	97.31	89.57
	10 dB	95.69	91.43	84.25	99.30	98.32	91.22
	5 dB	97.94	96.90	88.36	99.61	99.10	92.78
	0 dB	99.06	99.18	91.98	99.88	99.55	94.37
	-5 dB	99.54	100.00	95.79	99.94	99.98	97.27
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	90.11	94.23	97.46	88.15	90.83	96.80
	15 dB	84.87	89.63	96.12	82.98	83.56	95.52
	10 dB	78.47	76.98	92.11	75.12	73.06	91.22
	5 dB	70.27	57.19	79.07	66.92	53.84	77.49
	0 dB	61.78	30.13	53.35	59.97	32.91	50.12
	-5 dB	53.87	1.35	32.23	52.81	4.94	27.24
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	10.97	16.08	14.71	5.67	5.71	8.77
	15 dB	9.97	13.53	13.72	6.59	7.22	8.47
	10 dB	9.93	13.28	13.19	8.66	10.01	8.78
	5 dB	11.09	16.05	14.67	11.16	15.82	12.26
	0 dB	13.09	23.33	20.61	13.28	22.41	20.21
	-5 dB	15.35	32.17	24.93	15.59	31.34	25.81

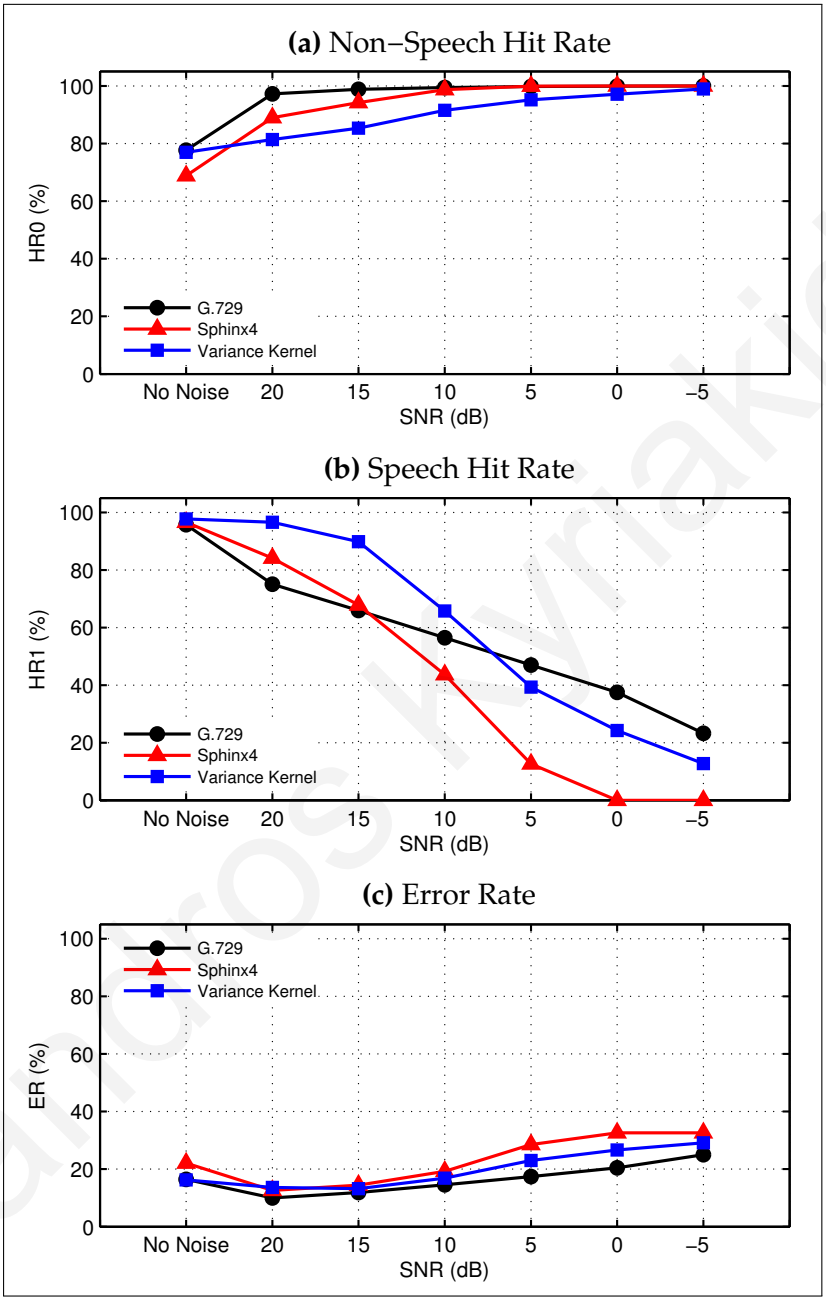

Figure A.40: A comparison of the voice activity detection performance of three different methods, using added noise of type "HF channel" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

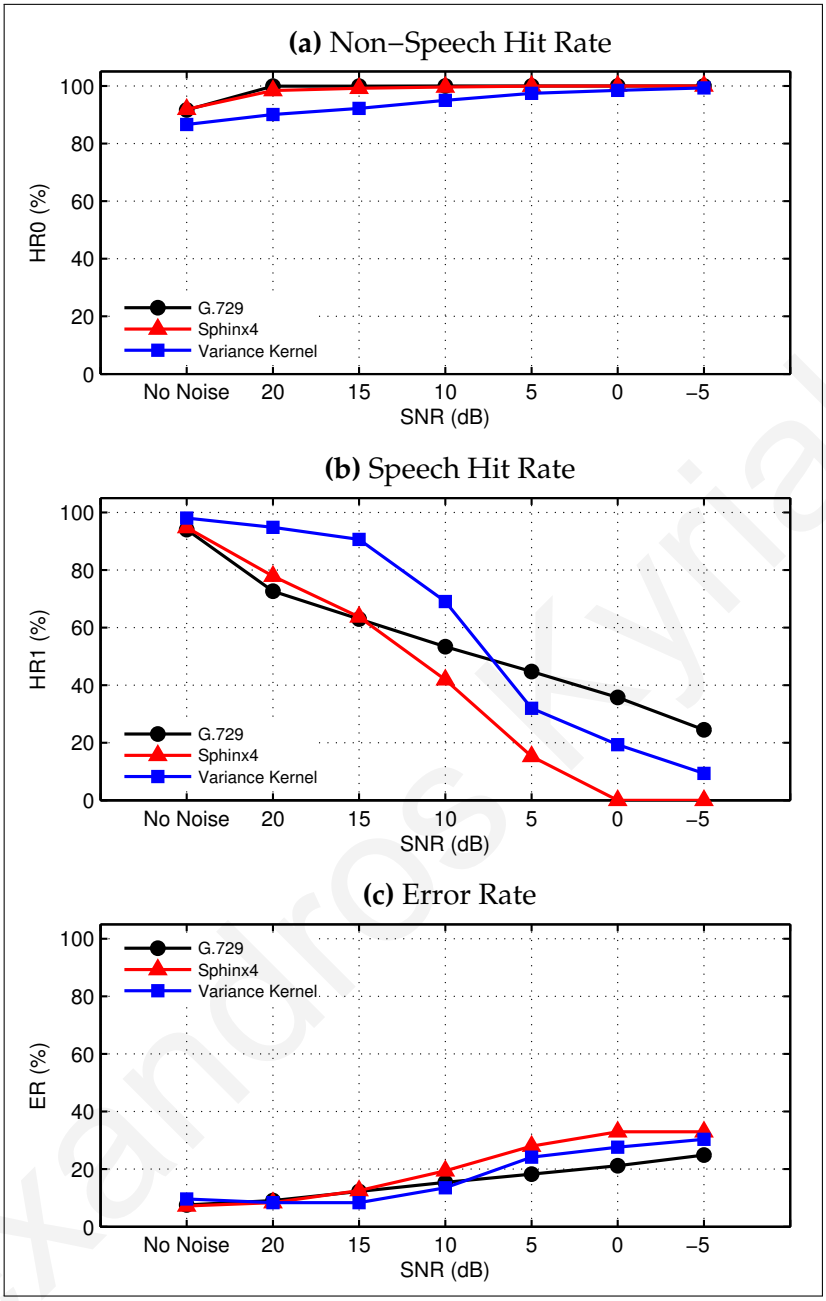

Figure A.41: A comparison of the voice activity detection performance of three different methods, using added noise of type "HF channel" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.31: Voice activity detection performance for three different methods using *added noise of type "HF channel"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	97.25	88.96	81.39	99.92	98.42	90.06
	15 dB	98.85	94.19	85.35	99.94	99.18	92.20
	10 dB	99.44	98.73	91.56	99.97	99.62	95.02
	5 dB	99.78	99.91	95.21	99.97	99.96	97.46
	0 dB	99.94	100.00	97.12	99.99	100.00	98.48
	-5 dB	99.98	100.00	98.94	99.99	100.00	99.31
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	75.06	84.10	96.59	72.60	77.85	94.81
	15 dB	65.92	67.82	89.81	62.98	63.66	90.61
	10 dB	56.43	43.62	65.75	53.34	41.84	69.07
	5 dB	46.97	12.63	39.34	44.74	15.25	31.92
	0 dB	37.47	0.00	24.22	35.72	0.00	19.33
	-5 dB	23.28	0.00	12.74	24.47	0.00	9.36
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	9.99	12.62	13.65	9.08	8.36	8.37
	15 dB	11.89	14.41	13.20	12.24	12.53	8.33
	10 dB	14.58	19.24	16.85	15.40	19.43	13.53
	5 dB	17.43	28.55	23.01	18.23	27.96	24.14
	0 dB	20.43	32.60	26.65	21.19	32.96	27.60
	-5 dB	25.02	32.60	29.16	24.90	32.96	30.34

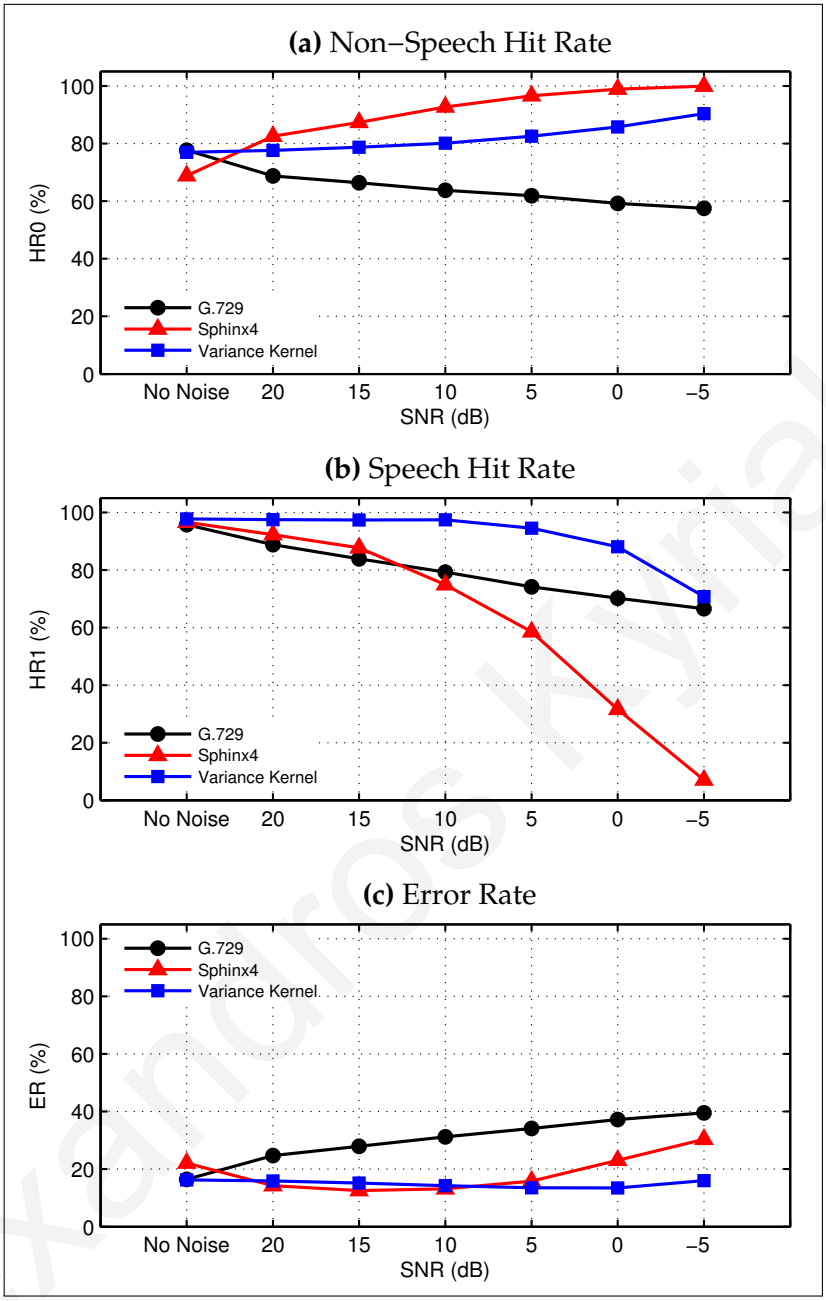

Figure A.42: A comparison of the voice activity detection performance of three different methods, using added noise of type "Leopard military vehicle" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

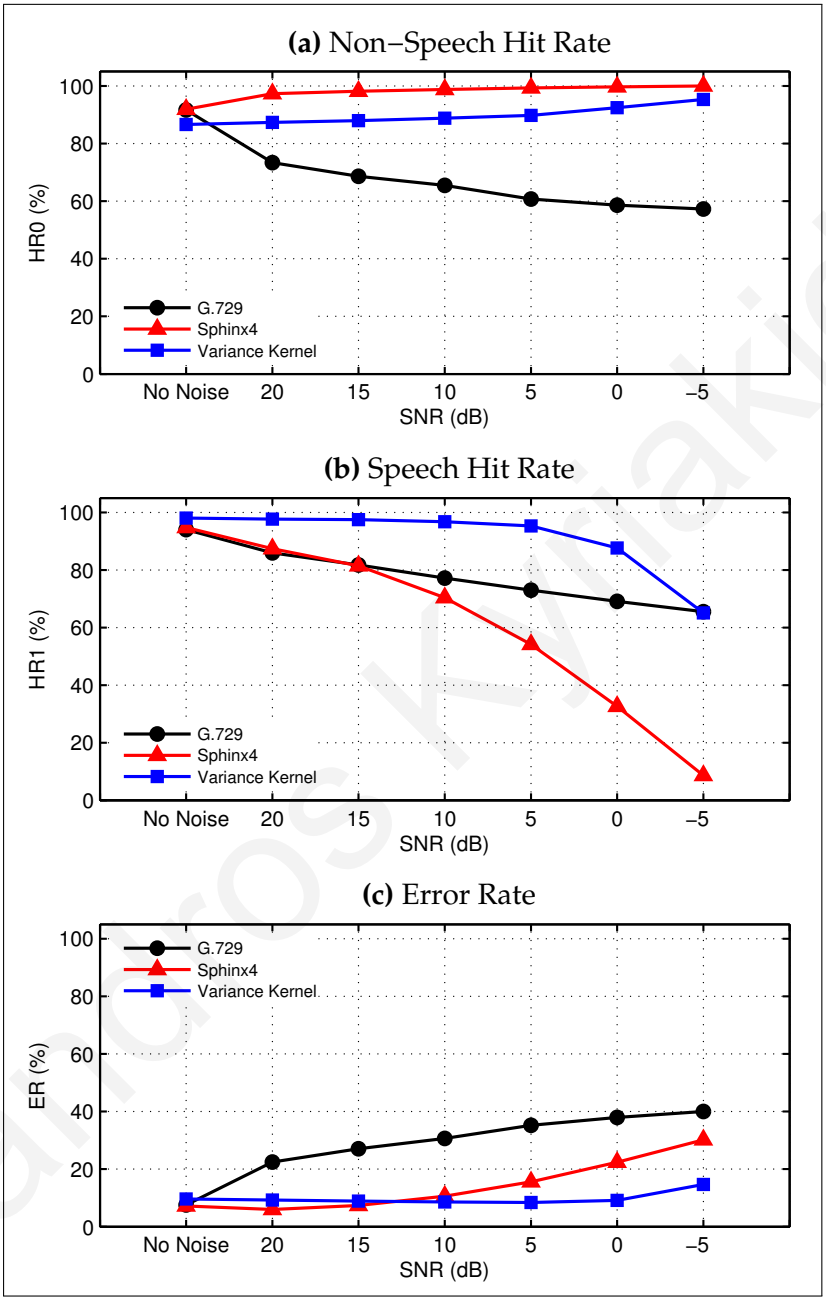

Figure A.43: A comparison of the voice activity detection performance of three different methods, using added noise of type "Leopard military vehicle" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.32: Voice activity detection performance for three different methods using *added noise of type "Leopard military vehicle"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	68.76	82.55	77.62	73.38	97.28	87.34
	15 dB	66.40	87.35	78.75	68.65	98.13	87.95
	10 dB	63.76	92.67	80.15	65.49	98.75	88.78
	5 dB	61.89	96.58	82.56	60.74	99.29	89.75
	0 dB	59.19	98.88	85.77	58.59	99.70	92.46
	-5 dB	57.54	99.93	90.37	57.27	99.97	95.29
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	88.77	92.29	97.49	85.95	87.41	97.68
	15 dB	83.86	87.69	97.36	81.73	81.46	97.49
	10 dB	79.28	74.89	97.41	77.16	70.37	96.74
	5 dB	74.15	58.51	94.51	72.99	54.14	95.31
	0 dB	70.20	31.61	88.08	69.08	32.62	87.65
	-5 dB	66.55	7.01	70.74	65.54	8.59	65.11
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	24.71	14.27	15.91	22.48	5.97	9.25
	15 dB	27.91	12.54	15.18	27.04	7.36	8.90
	10 dB	31.18	13.13	14.23	30.66	10.61	8.60
	5 dB	34.12	15.83	13.54	35.22	15.59	8.42
	0 dB	37.22	23.05	13.48	37.95	22.41	9.13
	-5 dB	39.52	30.36	16.03	40.01	30.15	14.66

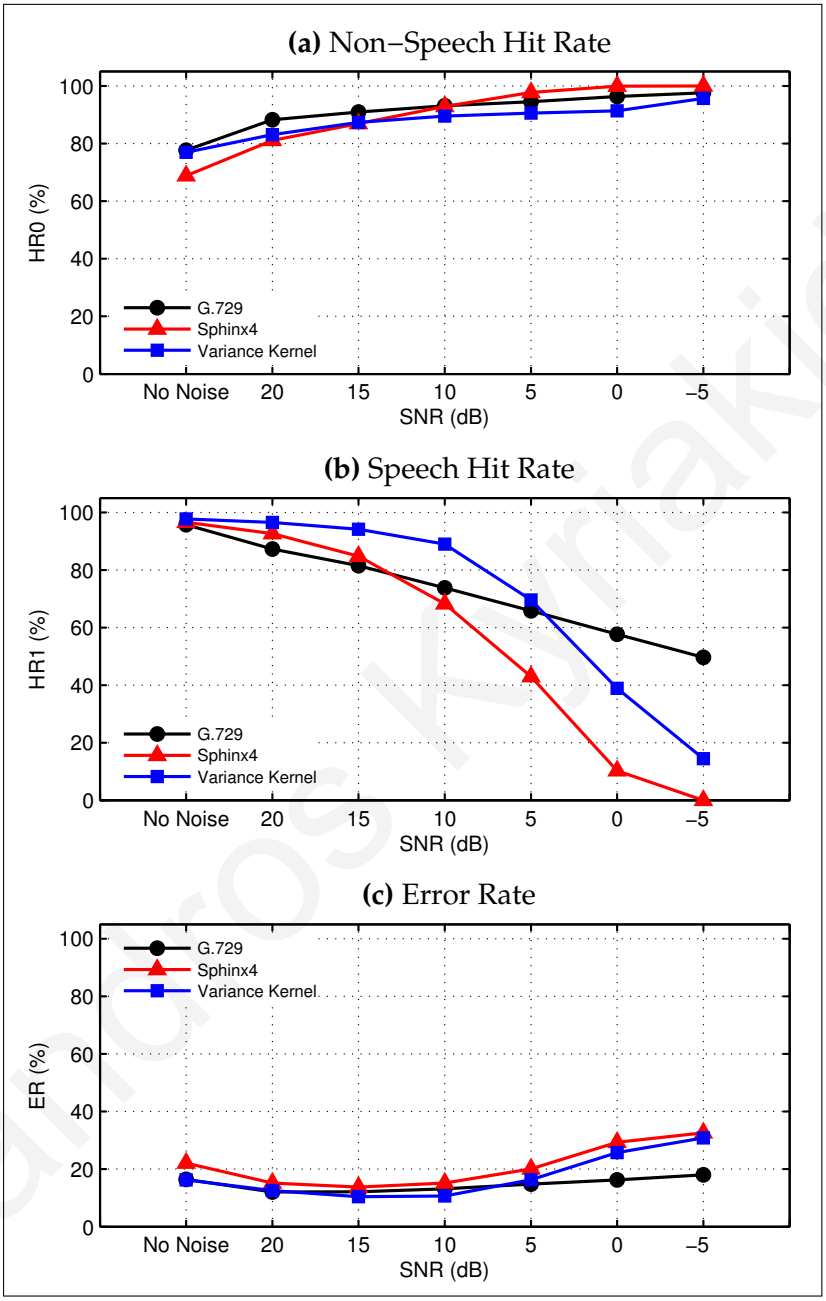

Figure A.44: A comparison of the voice activity detection performance of three different methods, using added nose of type "M109 military tank" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

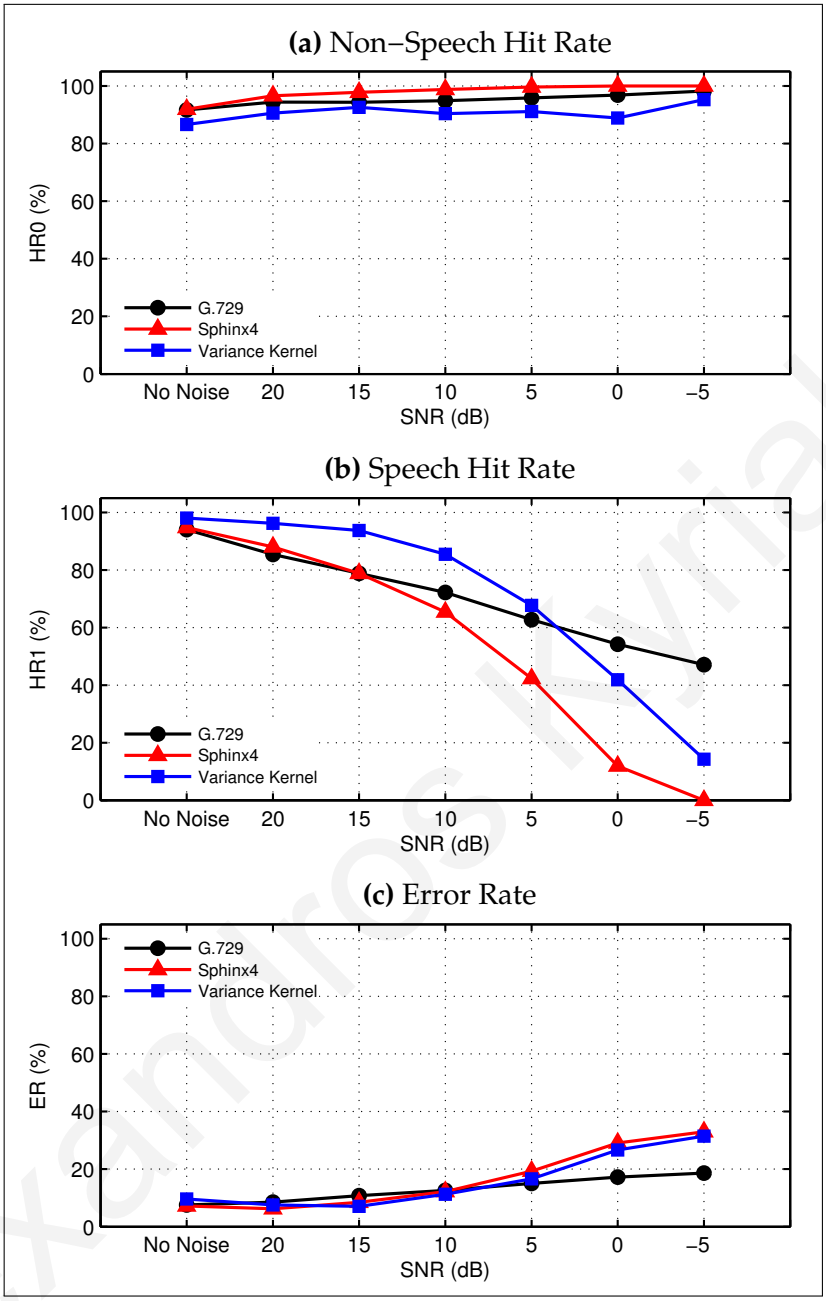

Figure A.45: A comparison of the voice activity detection performance of three different methods, using added nose of type "M109 military tank" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.33: Voice activity detection performance for three different methods using *added nose of type "M109 military tank"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	88.28	81.08	83.11	94.40	96.59	90.58
	15 dB	90.92	86.92	87.35	94.35	97.79	92.55
	10 dB	93.14	92.90	89.54	94.89	98.80	90.36
	5 dB	94.53	97.76	90.56	95.88	99.61	91.11
	0 dB	96.36	99.94	91.37	96.86	99.97	88.88
	-5 dB	97.61	100.00	95.66	98.21	100.00	95.23
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	87.28	92.62	96.49	85.39	88.03	96.24
	15 dB	81.53	84.77	94.14	78.84	78.83	93.73
	10 dB	73.77	68.26	88.98	72.21	65.39	85.48
	5 dB	65.84	42.98	69.57	62.74	42.30	67.70
	0 dB	57.68	10.23	38.91	54.20	11.90	41.86
	-5 dB	49.67	0.00	14.47	47.08	0.00	14.27
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	12.05	15.15	12.53	8.57	6.23	7.55
	15 dB	12.14	13.78	10.44	10.76	8.46	7.06
	10 dB	13.18	15.13	10.64	12.59	12.21	11.25
	5 dB	14.82	20.10	16.29	15.04	19.28	16.61
	0 dB	16.25	29.30	25.73	17.20	29.06	26.61
	-5 dB	18.02	32.60	30.81	18.64	32.96	31.45

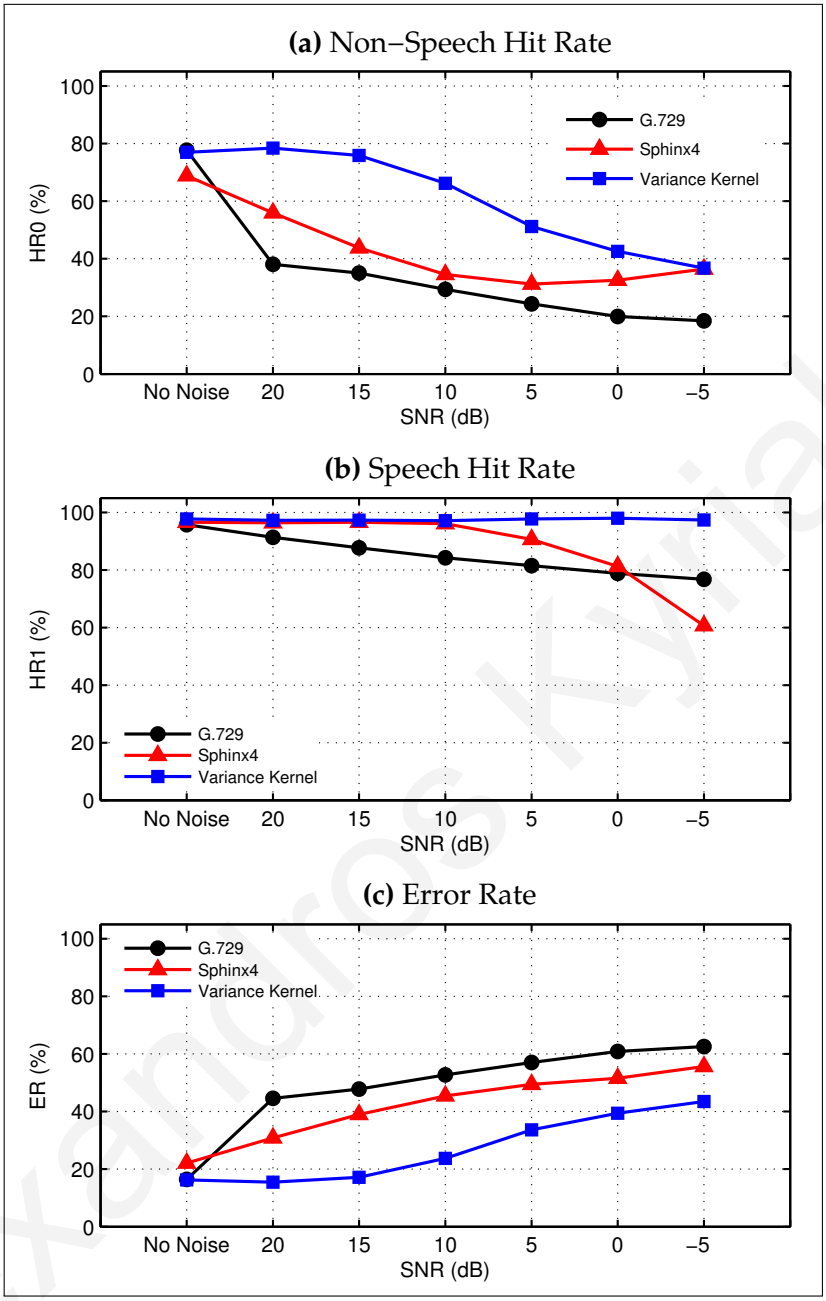

Figure A.46: A comparison of the voice activity detection performance of three different methods, using added noise of type "Machine gun" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

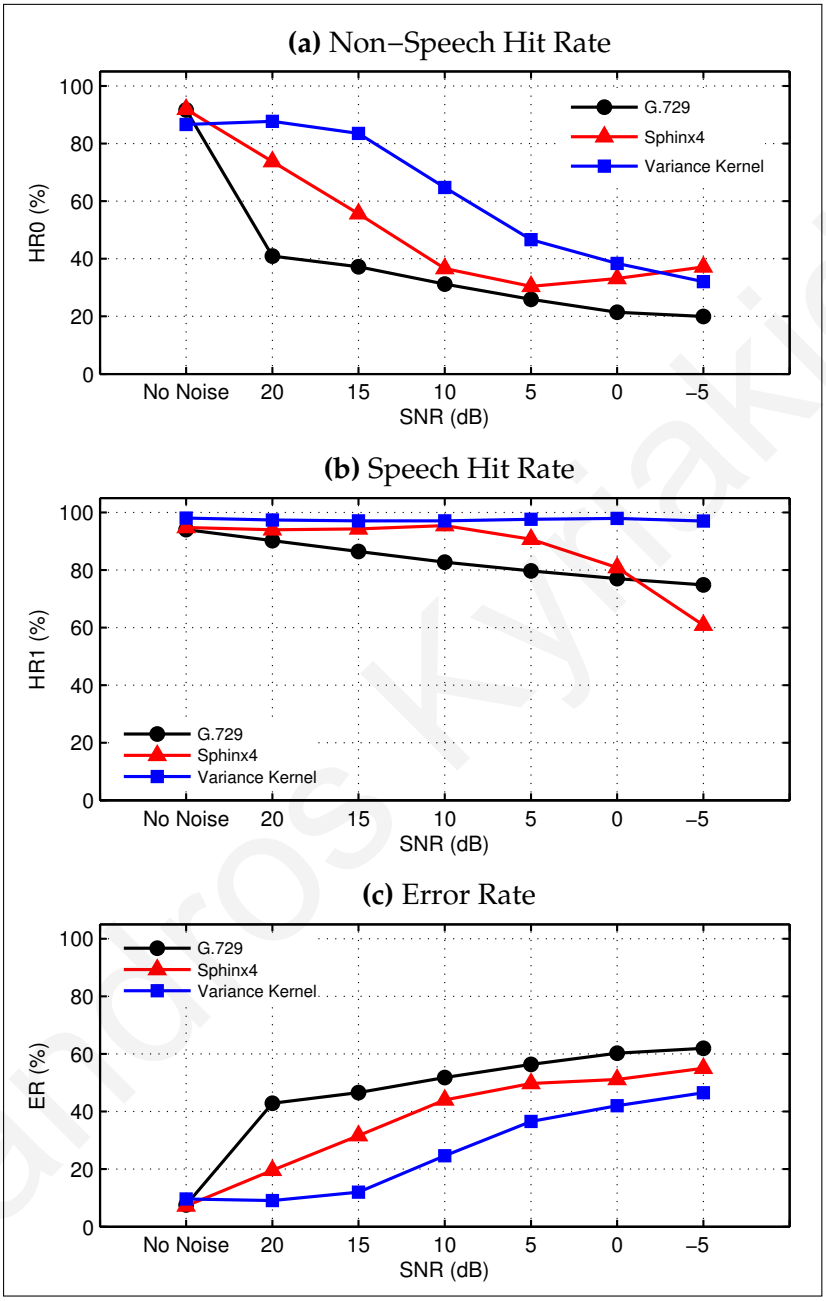

Figure A.47: A comparison of the voice activity detection performance of three different methods, using added noise of type "Machine gun" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.34: Voice activity detection performance for three different methods using *added noise of type "Machine gun"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	38.05	55.96	78.39	40.90	73.75	87.73
	15 dB	35.05	43.80	75.85	37.23	55.60	83.50
	10 dB	29.40	34.53	66.22	31.21	36.63	64.72
	5 dB	24.34	31.19	51.20	25.92	30.39	46.63
	0 dB	19.97	32.52	42.54	21.40	33.14	38.36
	-5 dB	18.44	36.44	36.74	19.97	37.14	32.05
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	91.34	96.42	97.28	90.21	93.96	97.39
	15 dB	87.73	96.55	97.31	86.41	94.27	97.09
	10 dB	84.23	96.07	97.10	82.69	95.43	97.08
	5 dB	81.48	90.64	97.76	79.67	90.70	97.62
	0 dB	78.85	81.27	97.96	77.01	80.85	97.92
	-5 dB	76.77	60.67	97.37	74.80	60.86	97.03
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	44.58	30.85	15.45	42.84	19.59	9.08
	15 dB	47.77	39.00	17.15	46.56	31.66	12.02
	10 dB	52.73	45.41	23.71	51.83	43.99	24.61
	5 dB	57.03	49.43	33.62	56.37	49.73	36.56
	0 dB	60.83	51.58	39.39	60.27	51.14	42.01
	-5 dB	62.54	55.66	43.49	61.96	55.05	46.53

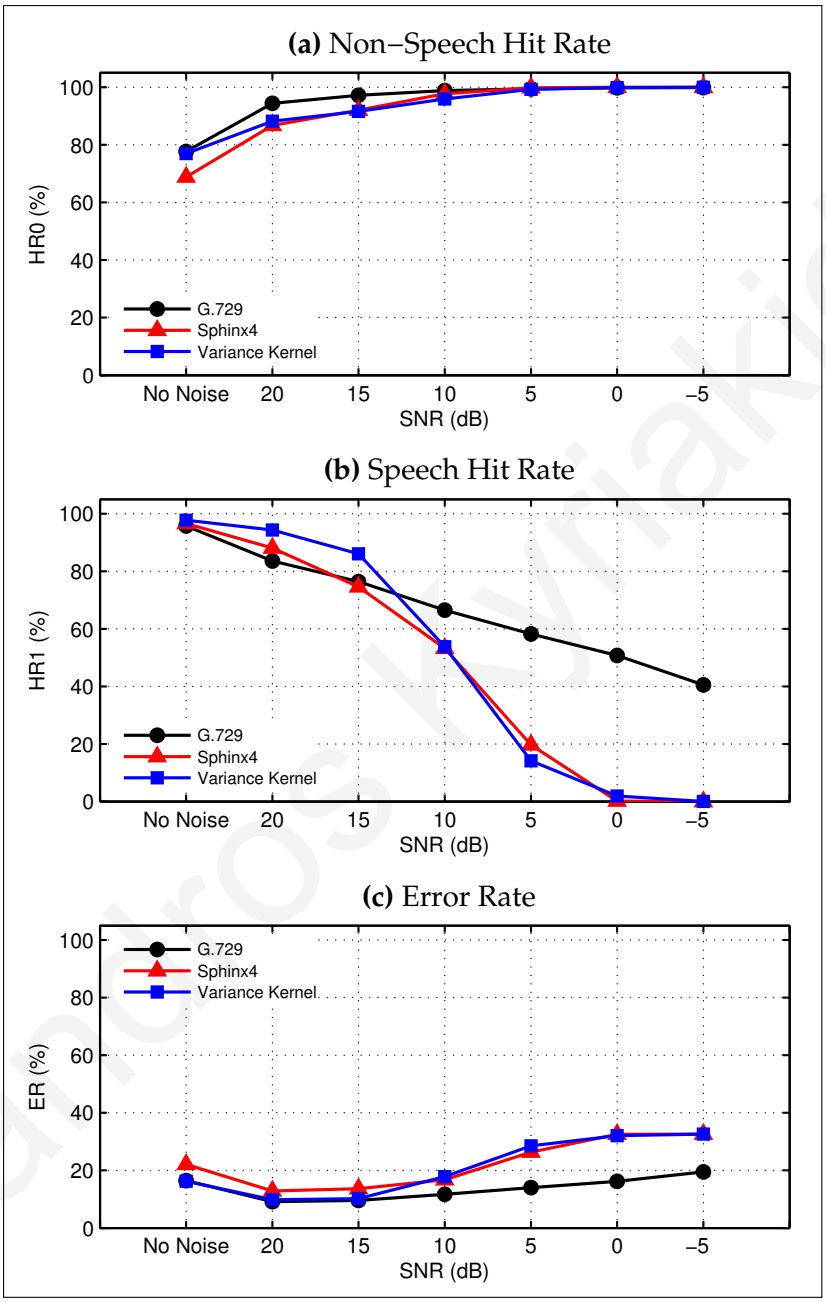

Figure A.48: A comparison of the voice activity detection performance of three different methods, using added pink noise at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

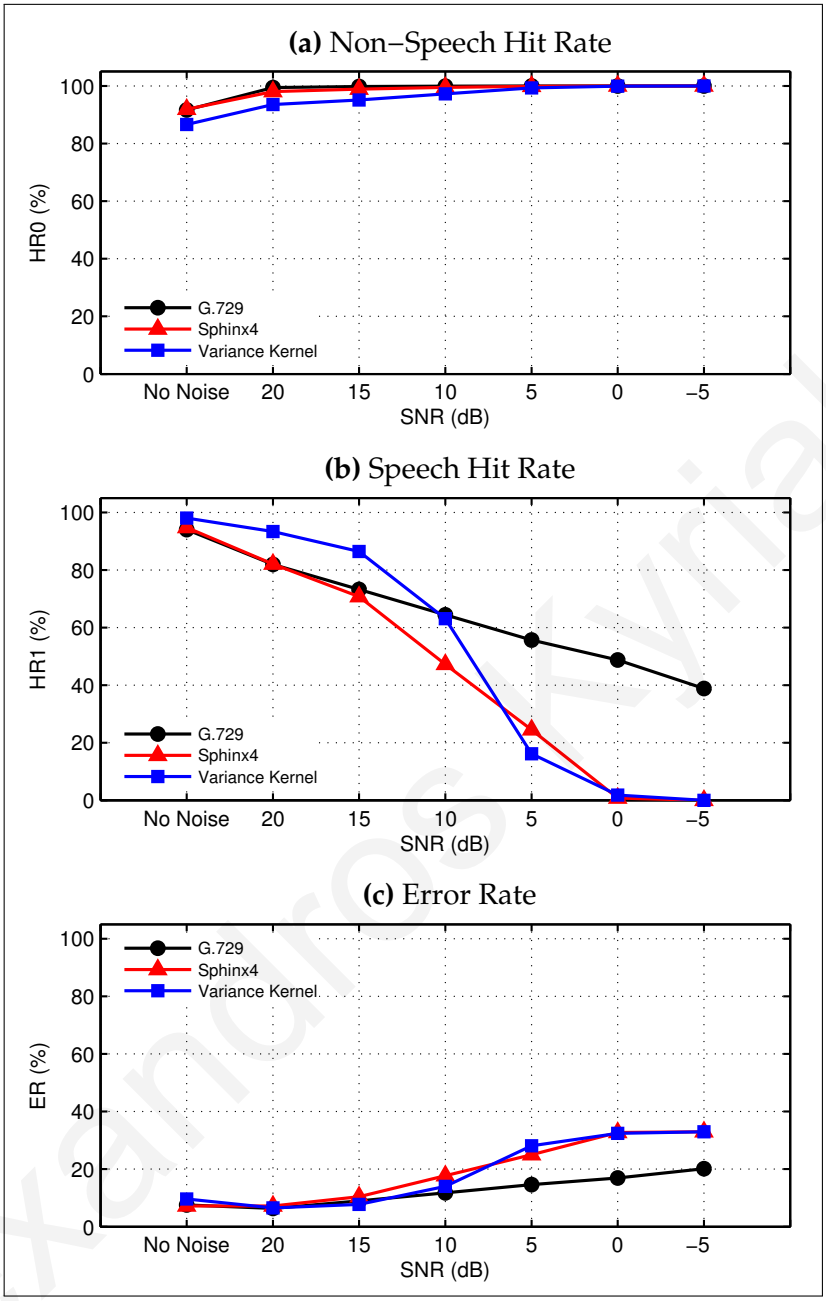

Figure A.49: A comparison of the voice activity detection performance of three different methods, using added pink noise at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.35: Voice activity detection performance for three different methods using *added pink noise* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	94.38	86.71	88.21	99.45	98.05	93.53
	15 dB	97.17	92.05	91.62	99.78	98.87	95.15
	10 dB	98.84	97.74	95.93	99.96	99.53	97.27
	5 dB	99.41	99.82	99.24	99.97	99.84	99.33
	0 dB	99.81	100.00	99.89	99.99	100.00	99.95
	-5 dB	99.89	100.00	100.00	100.00	100.00	100.00
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	83.55	88.09	94.31	81.90	82.06	93.36
	15 dB	76.40	74.58	86.06	73.26	70.75	86.40
	10 dB	66.48	53.25	53.76	64.39	47.23	63.08
	5 dB	58.20	19.64	14.13	55.63	24.47	16.13
	0 dB	50.73	0.17	1.92	48.72	0.76	1.79
	-5 dB	40.53	0.00	0.06	38.84	0.00	0.00
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	9.15	12.84	9.80	6.34	7.22	6.52
	15 dB	9.60	13.65	10.19	8.96	10.40	7.74
	10 dB	11.71	16.77	17.82	11.76	17.71	14.00
	5 dB	14.02	26.32	28.51	14.64	25.00	28.09
	0 dB	16.19	32.55	32.05	16.91	32.71	32.40
	-5 dB	19.46	32.60	32.58	20.16	32.96	32.96

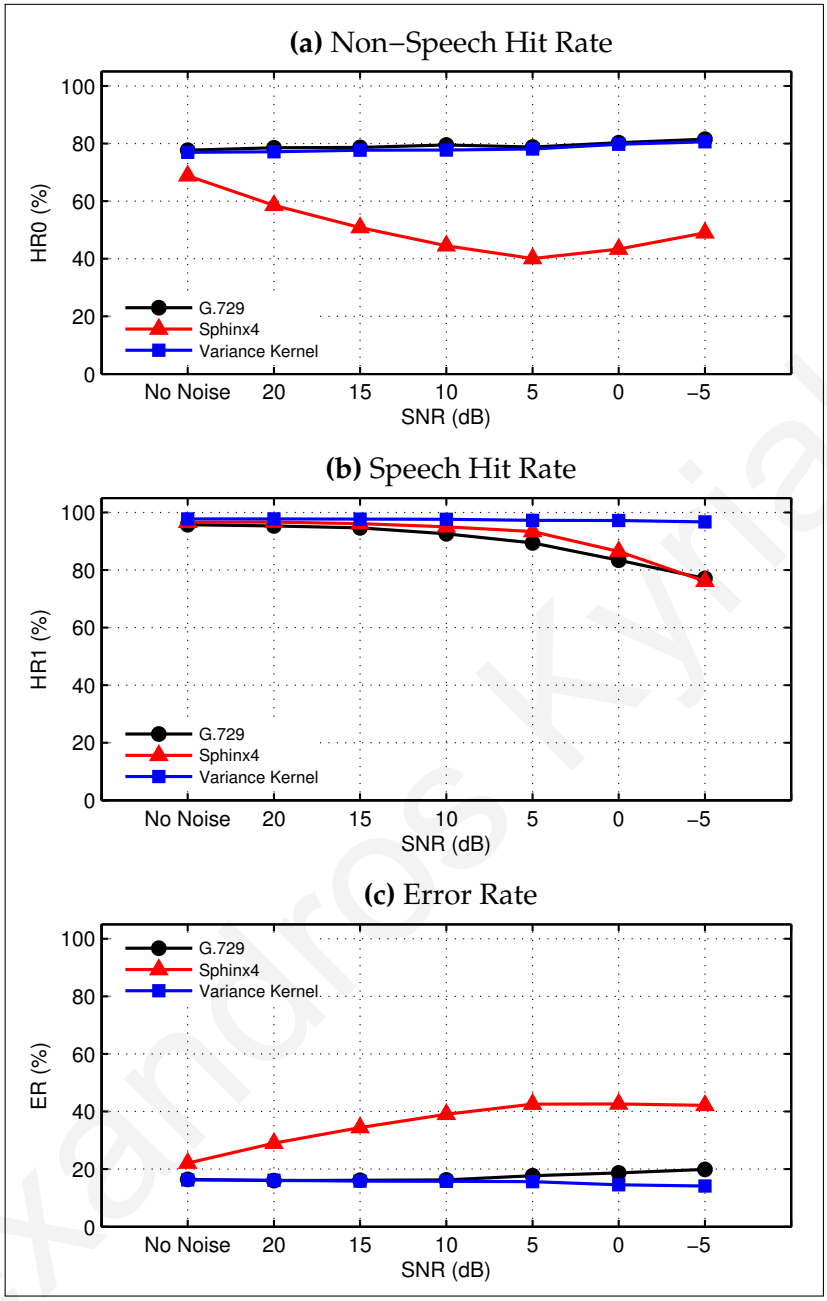

Figure A.50: A comparison of the voice activity detection performance of three different methods, using added noise of type "Vehicle interior (120km/h)" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

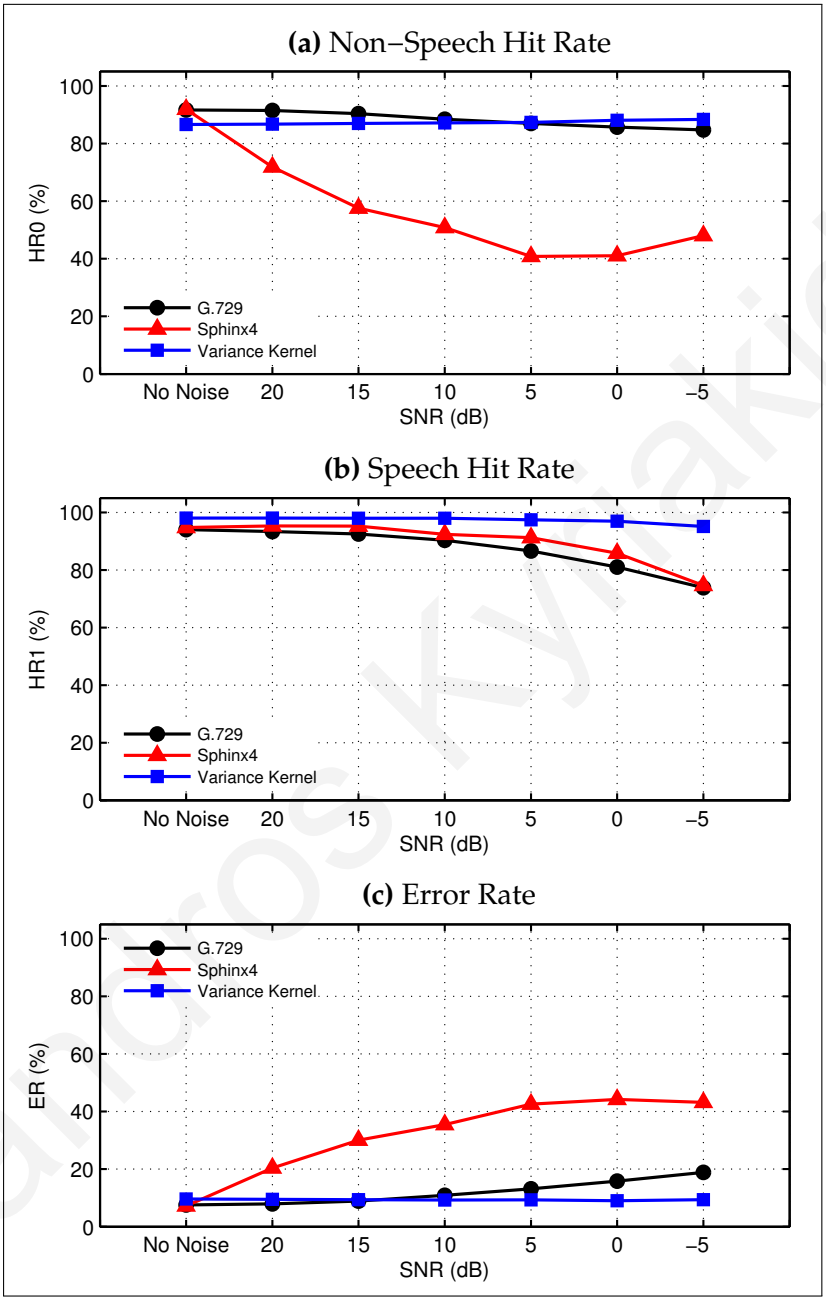

Figure A.51: A comparison of the voice activity detection performance of three different methods, using added noise of type "Vehicle interior (120km/h)" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.36: Voice activity detection performance for three different methods using *added noise of type "Vehicle interior (120km/h)"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	78.53	58.56	77.16	91.45	71.86	86.76
	15 dB	78.67	50.82	77.67	90.38	57.55	86.96
	10 dB	79.49	44.50	77.75	88.45	50.85	87.14
	5 dB	78.80	40.05	78.09	87.00	40.78	87.34
	0 dB	80.30	43.34	79.75	85.70	41.02	88.06
	-5 dB	81.54	49.01	80.61	84.75	48.06	88.41
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	95.30	96.66	97.71	93.34	95.32	98.05
	15 dB	94.63	96.09	97.69	92.52	95.26	98.00
	10 dB	92.56	95.02	97.61	90.31	92.38	97.95
	5 dB	89.41	93.33	97.24	86.59	91.25	97.43
	0 dB	83.39	86.42	97.21	81.00	85.84	96.92
	-5 dB	77.04	76.10	96.67	73.85	74.65	95.10
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	16.00	29.02	16.14	7.93	20.41	9.52
	15 dB	16.13	34.42	15.80	8.92	30.02	9.40
	10 dB	16.25	39.03	15.77	10.94	35.46	9.30
	5 dB	17.74	42.58	15.67	13.13	42.58	9.33
	0 dB	18.70	42.61	14.56	15.85	44.21	9.02
	-5 dB	19.93	42.15	14.15	18.84	43.18	9.39

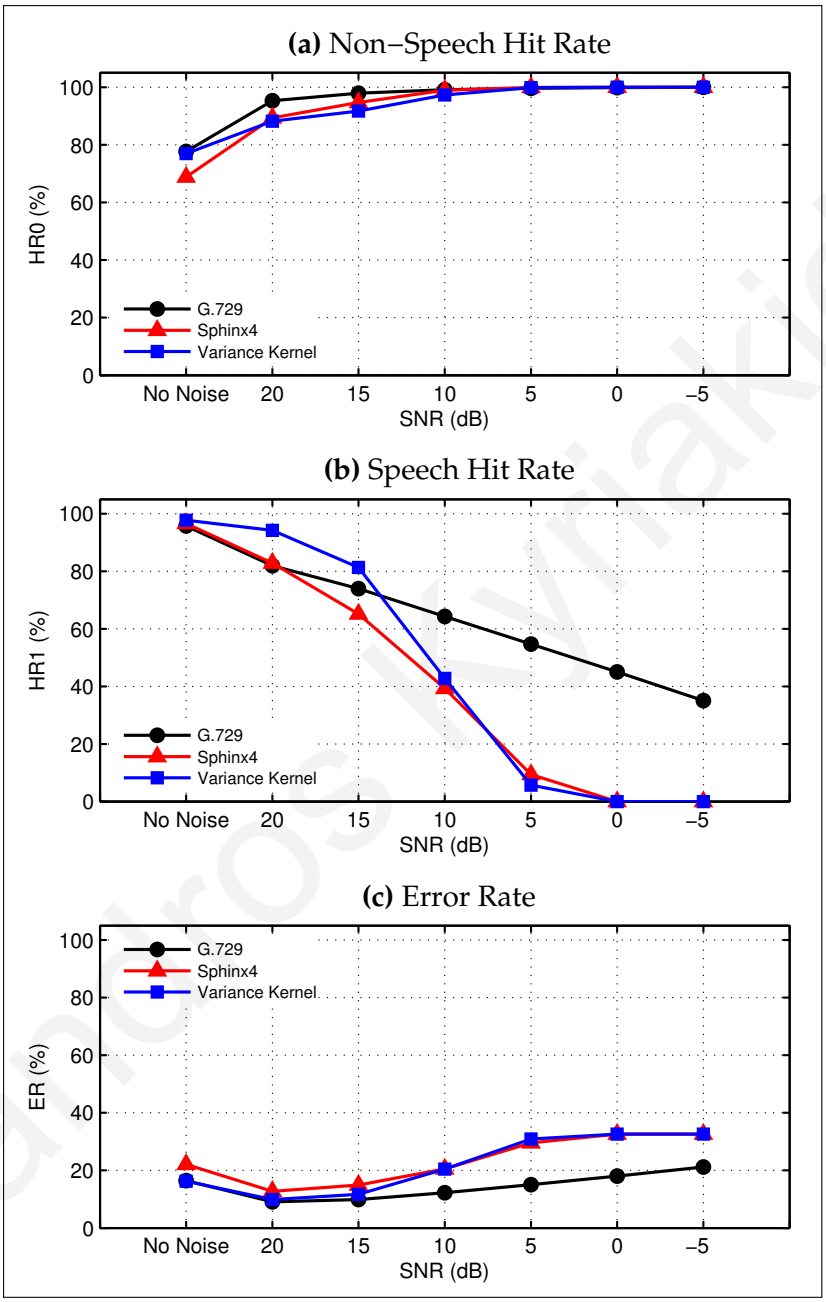

Figure A.52: A comparison of the voice activity detection performance of three different methods, using added white noise at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

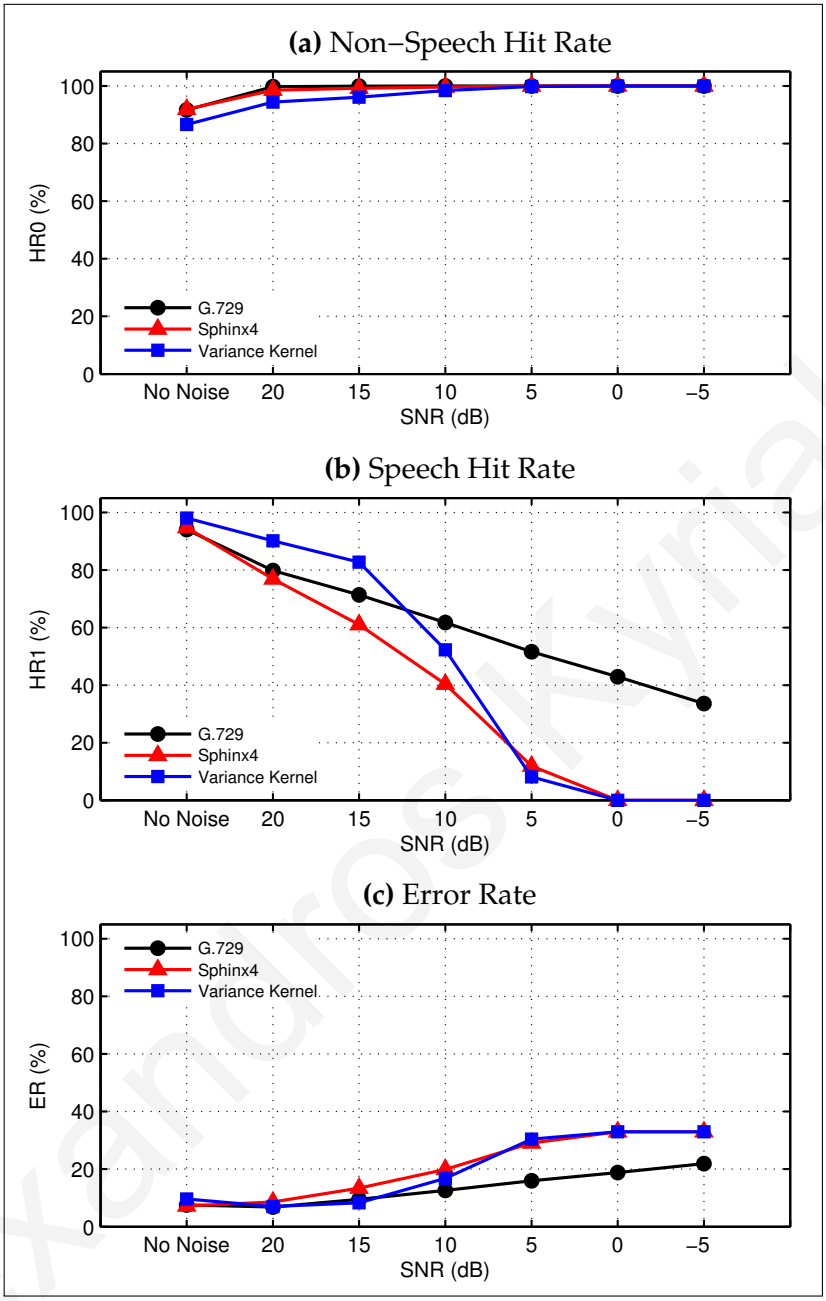

Figure A.53: A comparison of the voice activity detection performance of three different methods, using added white noise at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.37: Voice activity detection performance for three different methods using *added white noise* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	95.29	89.38	88.18	99.73	98.51	94.39
	15 dB	97.93	94.70	91.72	99.92	99.21	96.11
	10 dB	99.15	98.95	97.33	99.98	99.65	98.39
	5 dB	99.62	99.95	99.84	100.00	99.97	99.83
	0 dB	99.89	100.00	100.00	100.00	100.00	100.00
	-5 dB	99.97	100.00	100.00	100.00	100.00	100.00
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	81.92	82.85	94.19	79.78	76.86	90.16
	15 dB	73.95	65.12	81.29	71.36	60.98	82.71
	10 dB	64.29	39.33	42.76	61.71	40.33	52.24
	5 dB	54.70	9.39	5.65	51.56	11.88	8.11
	0 dB	45.02	0.00	0.00	42.92	0.00	0.00
	-5 dB	35.04	0.00	0.00	33.60	0.00	0.00
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	9.07	12.75	9.86	6.84	8.62	7.00
	15 dB	9.88	14.94	11.68	9.49	13.39	8.31
	10 dB	12.21	20.49	20.47	12.63	19.90	16.82
	5 dB	15.03	29.57	30.87	15.96	29.06	30.40
	0 dB	18.00	32.60	32.60	18.81	32.96	32.96
	-5 dB	21.20	32.60	32.60	21.88	32.96	32.96

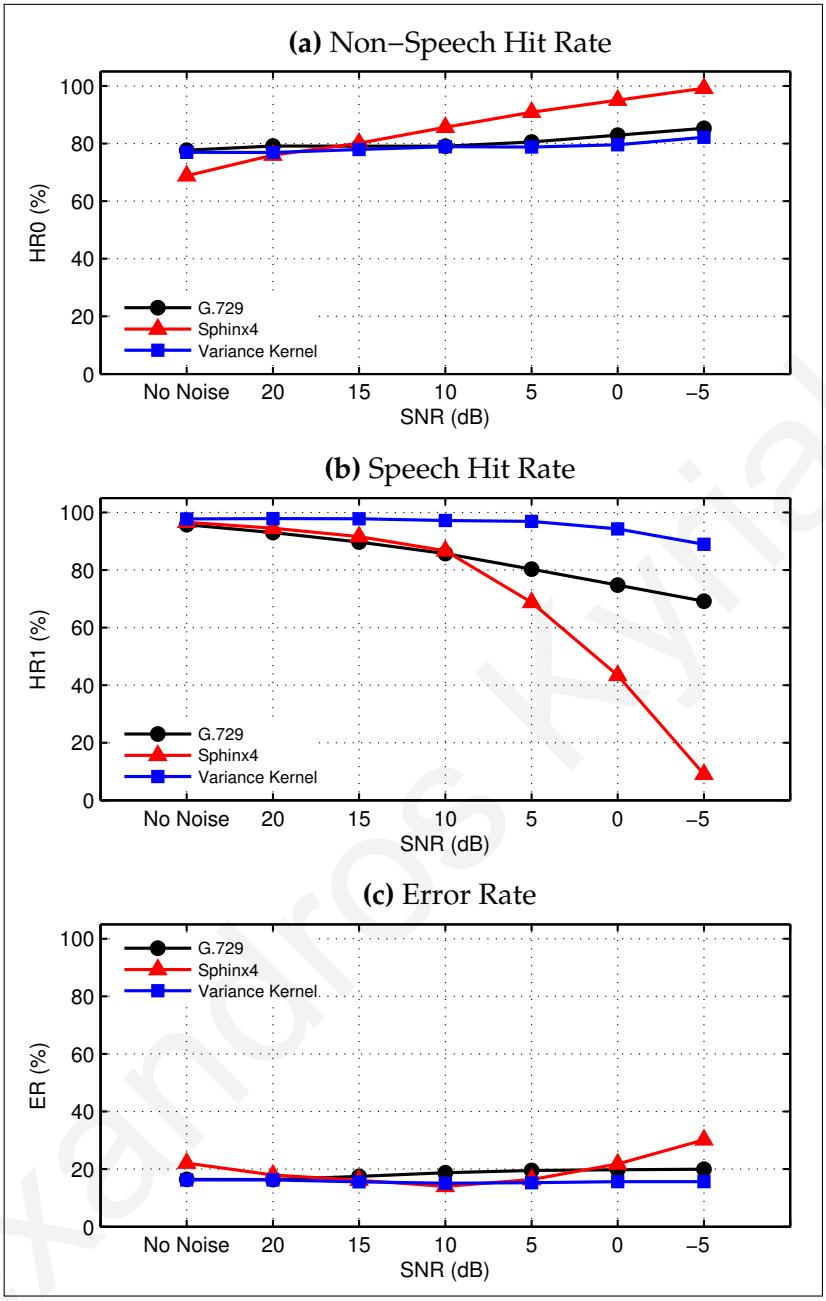

Figure A.54: A comparison of the voice activity detection performance of three different methods, using added noise of type "Air conditioner" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

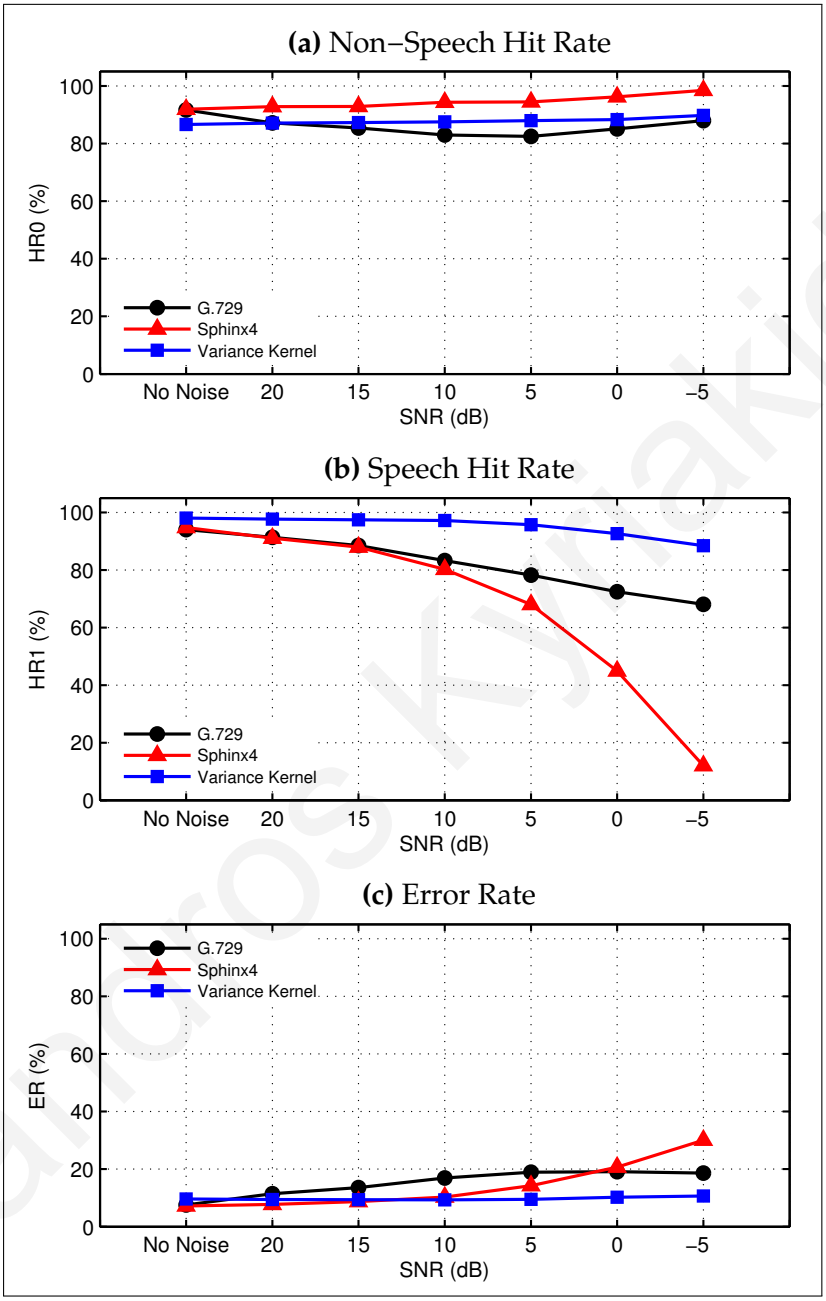

Figure A.55: A comparison of the voice activity detection performance of three different methods, using added noise of type "Air conditioner" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.38: Voice activity detection performance for three different methods using *added noise of type "Air conditioner"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	79.12	75.98	76.90	87.17	92.84	87.08
	15 dB	79.11	80.09	77.92	85.38	92.86	87.29
	10 dB	79.11	85.67	78.92	82.94	94.36	87.51
	5 dB	80.54	90.87	78.80	82.50	94.46	87.93
	0 dB	82.89	95.09	79.55	85.10	96.20	88.34
	-5 dB	85.27	99.20	82.17	87.93	98.45	89.77
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	92.97	94.50	97.85	91.38	91.03	97.65
	15 dB	89.69	91.59	97.81	88.48	87.97	97.45
	10 dB	85.71	86.65	97.19	83.26	80.21	97.17
	5 dB	80.28	68.72	96.91	78.22	68.03	95.72
	0 dB	74.78	43.37	94.24	72.42	44.89	92.64
	-5 dB	69.19	9.03	88.90	68.08	11.99	88.40
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	16.37	17.98	16.27	11.45	7.76	9.44
	15 dB	17.44	16.16	15.60	13.60	8.75	9.36
	10 dB	18.74	14.01	15.12	16.95	10.30	9.31
	5 dB	19.54	16.35	15.29	18.91	14.25	9.51
	0 dB	19.75	21.77	15.66	19.08	20.71	10.24
	-5 dB	19.97	30.19	15.64	18.61	30.05	10.68

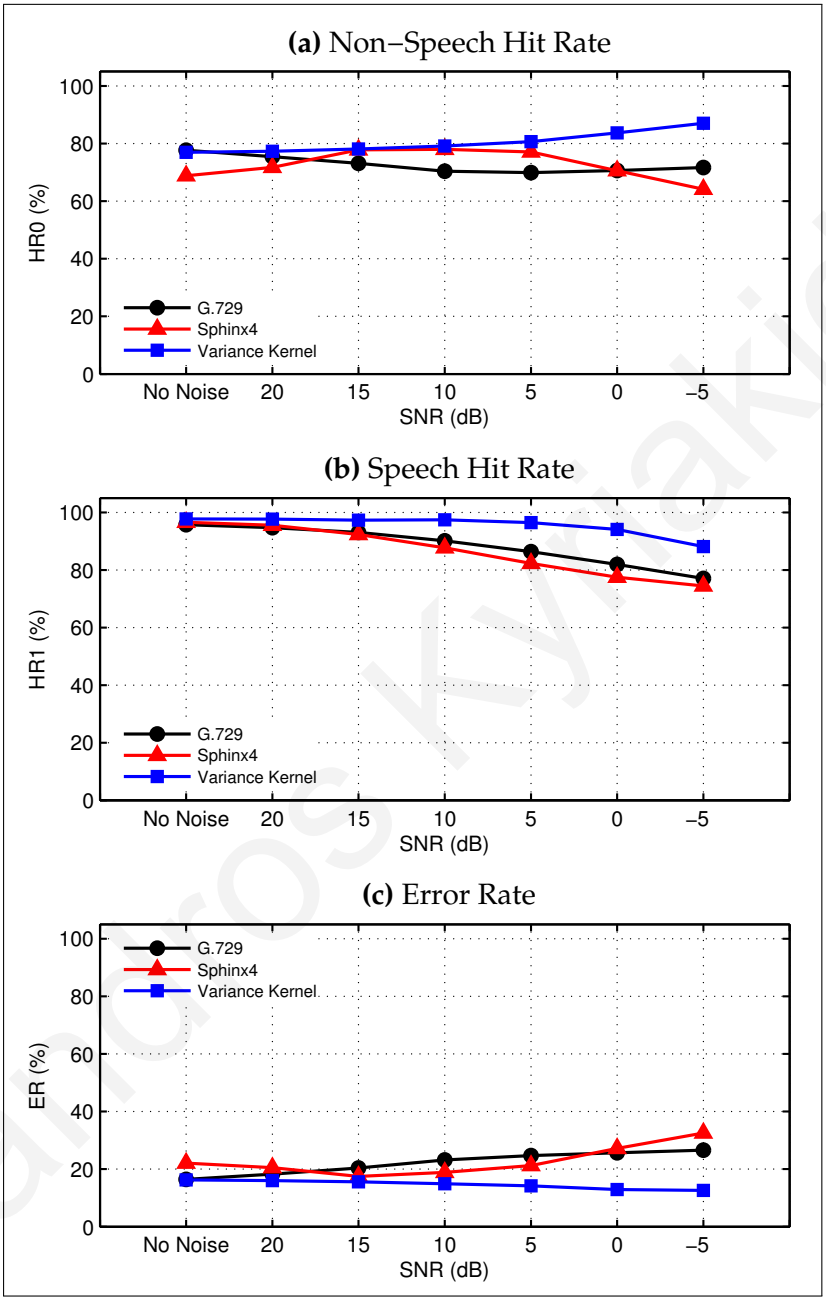

Figure A.56: A comparison of the voice activity detection performance of three different methods, using added noise of type "Conference room" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

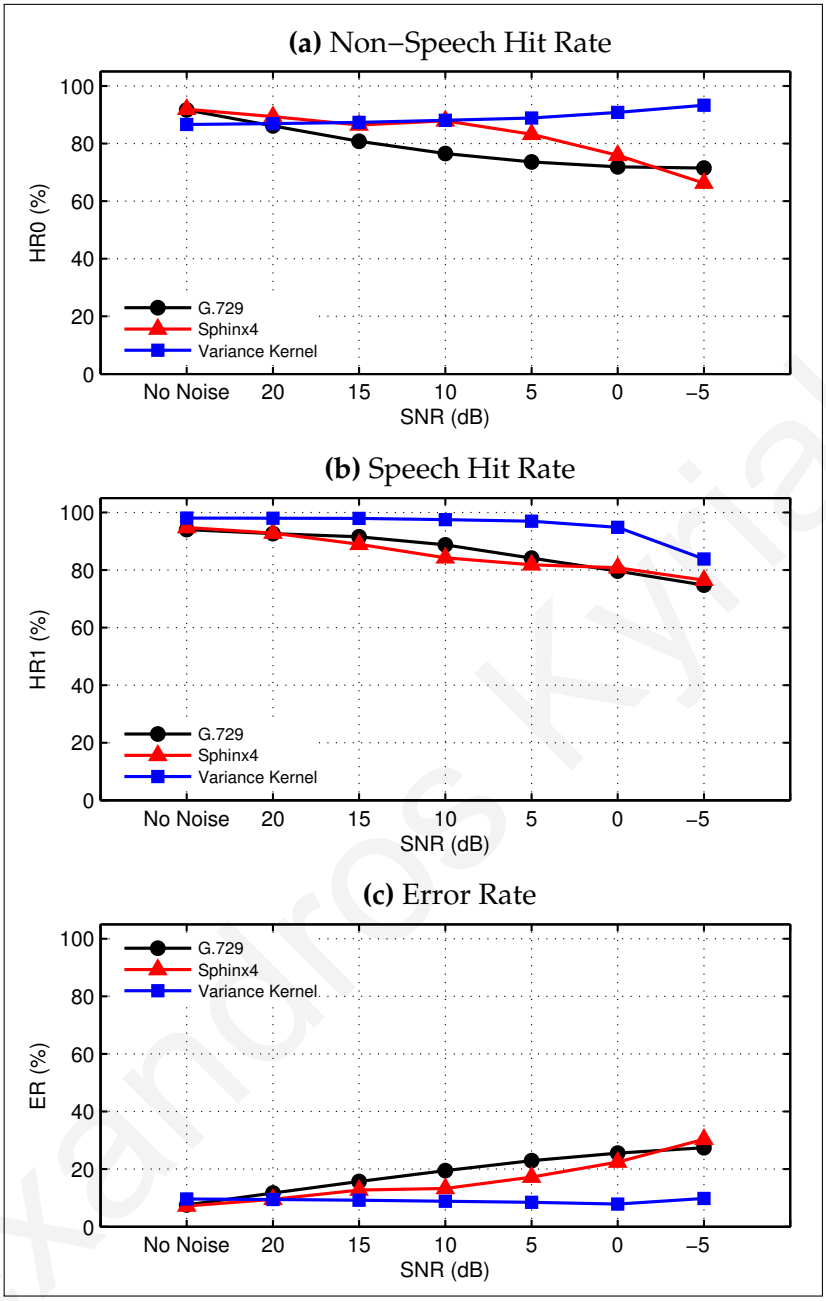

Figure A.57: A comparison of the voice activity detection performance of three different methods, using added noise of type "Conference room" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.39: Voice activity detection performance for three different methods using *added noise of type "Conference room"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	75.44	71.74	77.34	86.13	89.36	86.92
	15 dB	73.12	77.81	78.13	80.78	86.45	87.32
	10 dB	70.37	78.00	79.17	76.52	87.92	88.05
	5 dB	69.93	77.05	80.65	73.58	83.23	88.88
	0 dB	70.67	70.60	83.70	71.94	75.95	90.84
	-5 dB	71.69	64.11	87.02	71.47	66.28	93.33
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	94.70	95.52	97.69	92.66	92.80	98.00
	15 dB	93.03	92.33	97.33	91.50	88.91	97.89
	10 dB	90.13	87.68	97.40	88.73	84.23	97.48
	5 dB	86.35	82.32	96.48	84.12	81.79	96.94
	0 dB	81.92	77.47	94.09	79.60	80.75	94.79
	-5 dB	77.04	74.42	88.14	74.74	76.45	83.82
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	18.28	20.51	16.02	11.72	9.50	9.43
	15 dB	20.38	17.46	15.61	15.69	12.74	9.20
	10 dB	23.19	18.84	14.89	19.45	13.29	8.84
	5 dB	24.72	21.23	14.19	22.94	17.24	8.46
	0 dB	25.66	27.16	12.92	25.53	22.47	7.86
	-5 dB	26.56	32.53	12.62	27.45	30.37	9.81

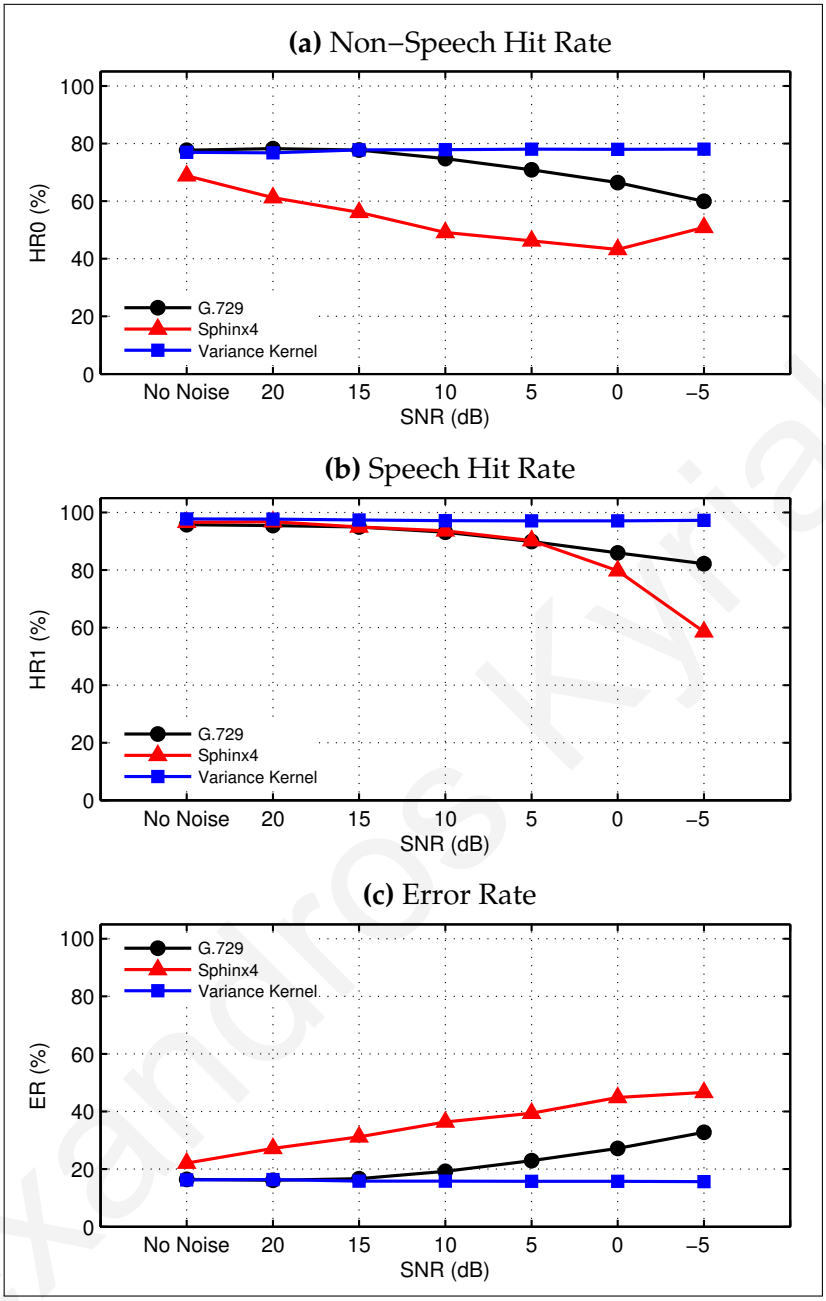

Figure A.58: A comparison of the voice activity detection performance of three different methods, using added noise of type "Intergalactic cruiser" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

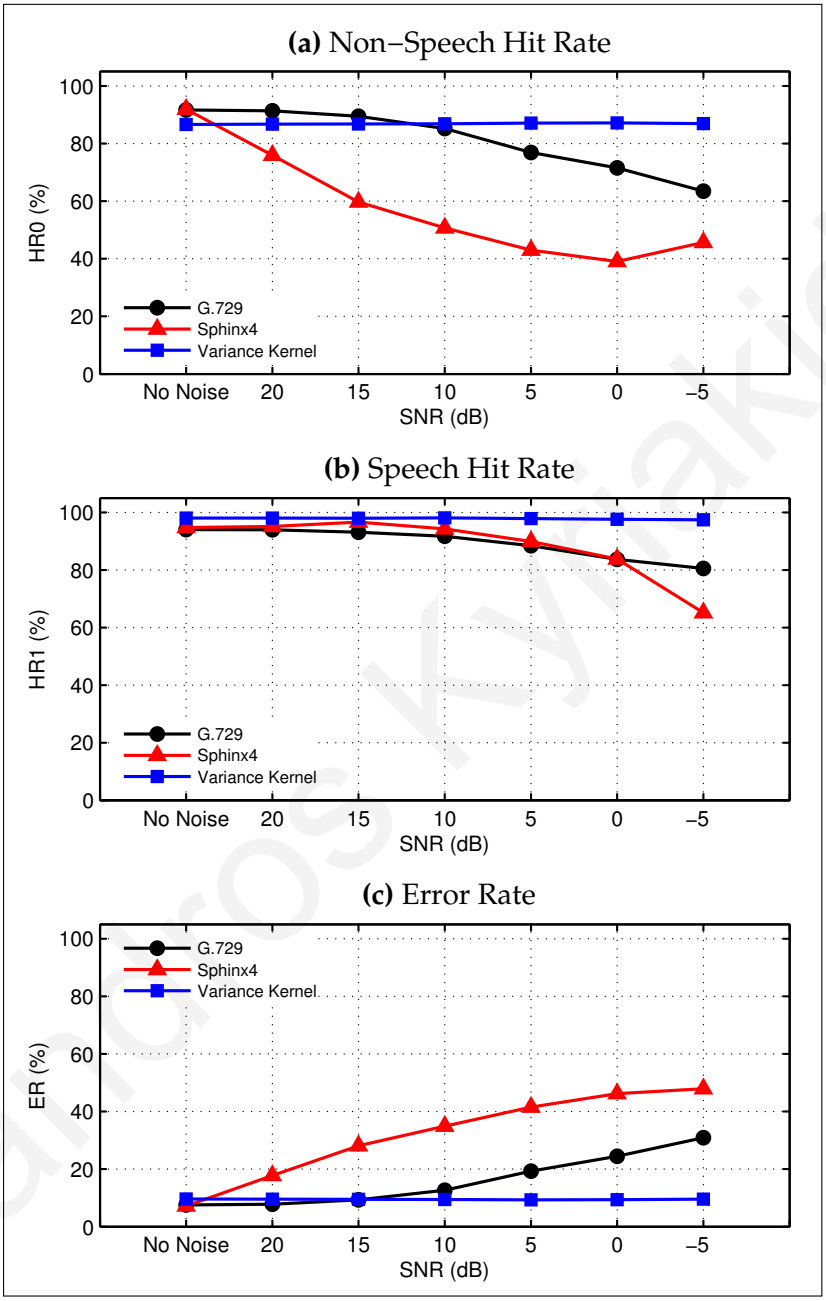

Figure A.59: A comparison of the voice activity detection performance of three different methods, using added noise of type "Intergalactic cruiser" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.40: Voice activity detection performance for three different methods using *added noise of type "Intergalactic cruiser"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	78.26	61.21	76.80	91.34	75.93	86.73
	15 dB	77.73	56.11	77.81	89.49	59.72	86.77
	10 dB	74.79	49.10	77.86	85.23	50.73	86.86
	5 dB	70.87	46.24	78.04	76.90	42.98	87.11
	0 dB	66.44	43.26	77.99	71.55	39.02	87.16
	-5 dB	59.97	50.90	78.08	63.49	45.64	86.94
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	95.43	96.76	97.66	93.96	95.12	98.04
	15 dB	94.92	94.97	97.39	93.09	96.62	98.00
	10 dB	93.13	93.62	97.15	91.69	94.25	98.09
	5 dB	89.89	90.22	97.08	88.43	89.90	97.84
	0 dB	85.93	79.72	97.08	83.71	83.80	97.60
	-5 dB	82.17	58.53	97.26	80.53	65.14	97.46
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	16.14	27.20	16.40	7.79	17.75	9.54
	15 dB	16.66	31.22	15.80	9.32	28.11	9.53
	10 dB	19.23	36.39	15.85	12.64	34.92	9.44
	5 dB	22.93	39.42	15.75	19.30	41.56	9.35
	0 dB	27.20	44.85	15.79	24.45	46.22	9.40
	-5 dB	32.79	46.61	15.67	30.89	47.93	9.59

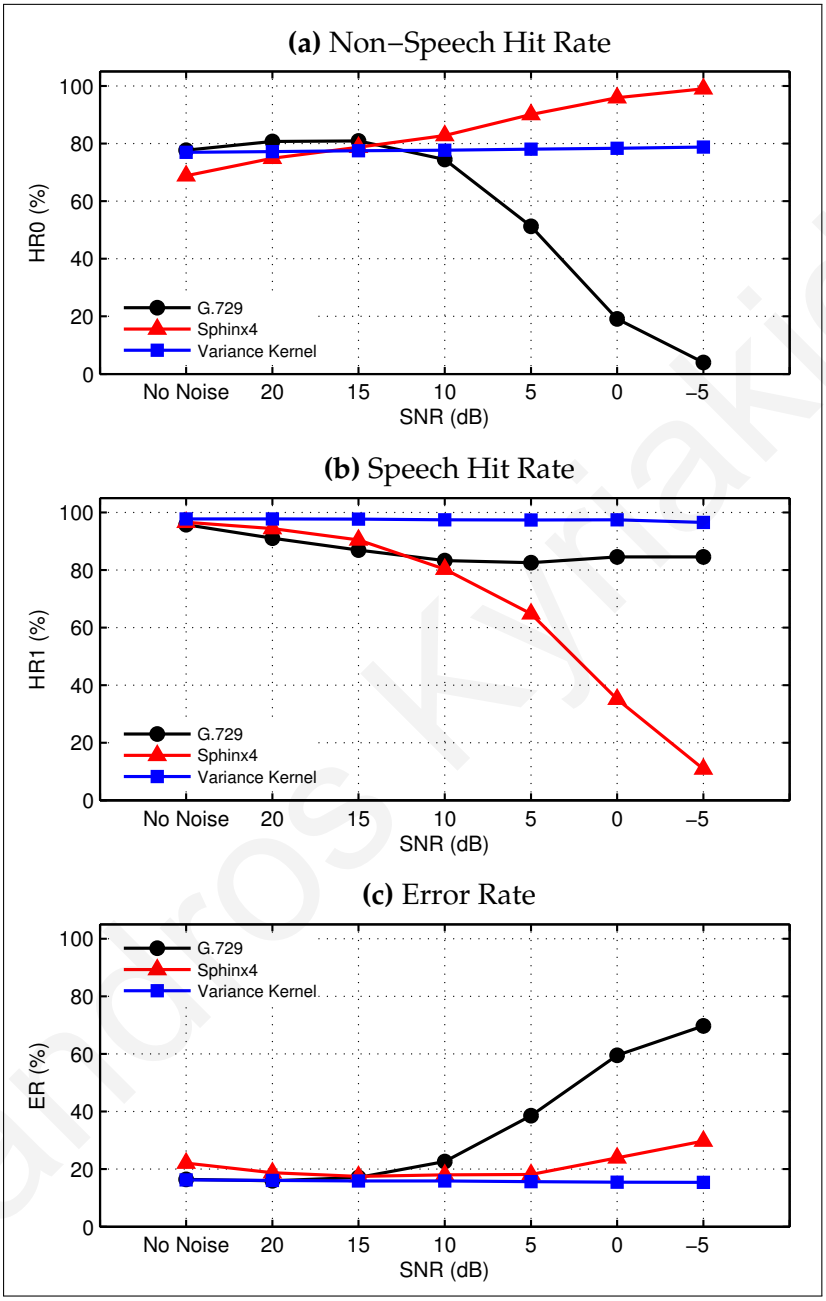

Figure A.60: A comparison of the voice activity detection performance of three different methods, using added noise of type "Jet airliner cabin" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

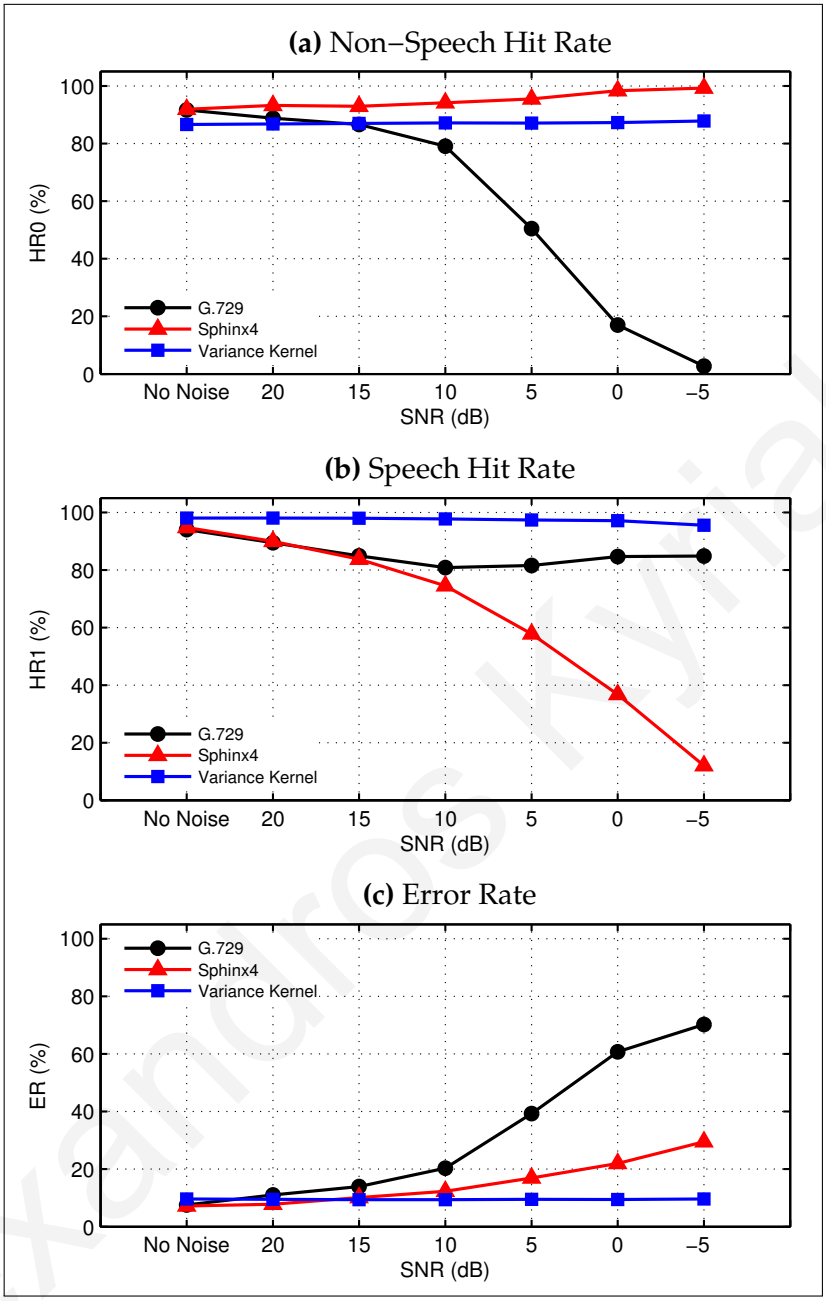

Figure A.61: A comparison of the voice activity detection performance of three different methods, using added noise of type "Jet airliner cabin" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.41: Voice activity detection performance for three different methods using *added noise of type "Jet airliner cabin"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			"clean" recordings		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	80.72	74.86	77.21	88.78	93.26	86.77
	15 dB	80.91	78.69	77.50	86.60	92.94	86.96
	10 dB	74.47	82.80	77.67	79.06	94.14	87.14
	5 dB	51.25	90.09	78.07	50.47	95.51	87.12
	0 dB	19.12	95.90	78.35	16.95	98.37	87.28
	-5 dB	3.99	99.01	78.79	2.73	99.27	87.83
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	91.02	94.42	97.72	89.45	89.98	98.04
	15 dB	86.90	90.45	97.69	84.88	83.74	98.00
	10 dB	83.28	80.31	97.43	80.85	74.50	97.74
	5 dB	82.54	64.75	97.36	81.54	57.81	97.39
	0 dB	84.56	35.09	97.41	84.65	36.76	97.13
	-5 dB	84.56	10.87	96.53	84.87	12.00	95.52
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	15.93	18.76	16.10	11.00	7.82	9.52
	15 dB	17.13	17.47	15.92	13.96	10.09	9.40
	10 dB	22.66	18.01	15.88	20.35	12.33	9.37
	5 dB	38.55	18.17	15.64	39.29	16.91	9.49
	0 dB	59.54	23.92	15.44	60.74	21.94	9.48
	-5 dB	69.74	29.72	15.43	70.20	29.49	9.63

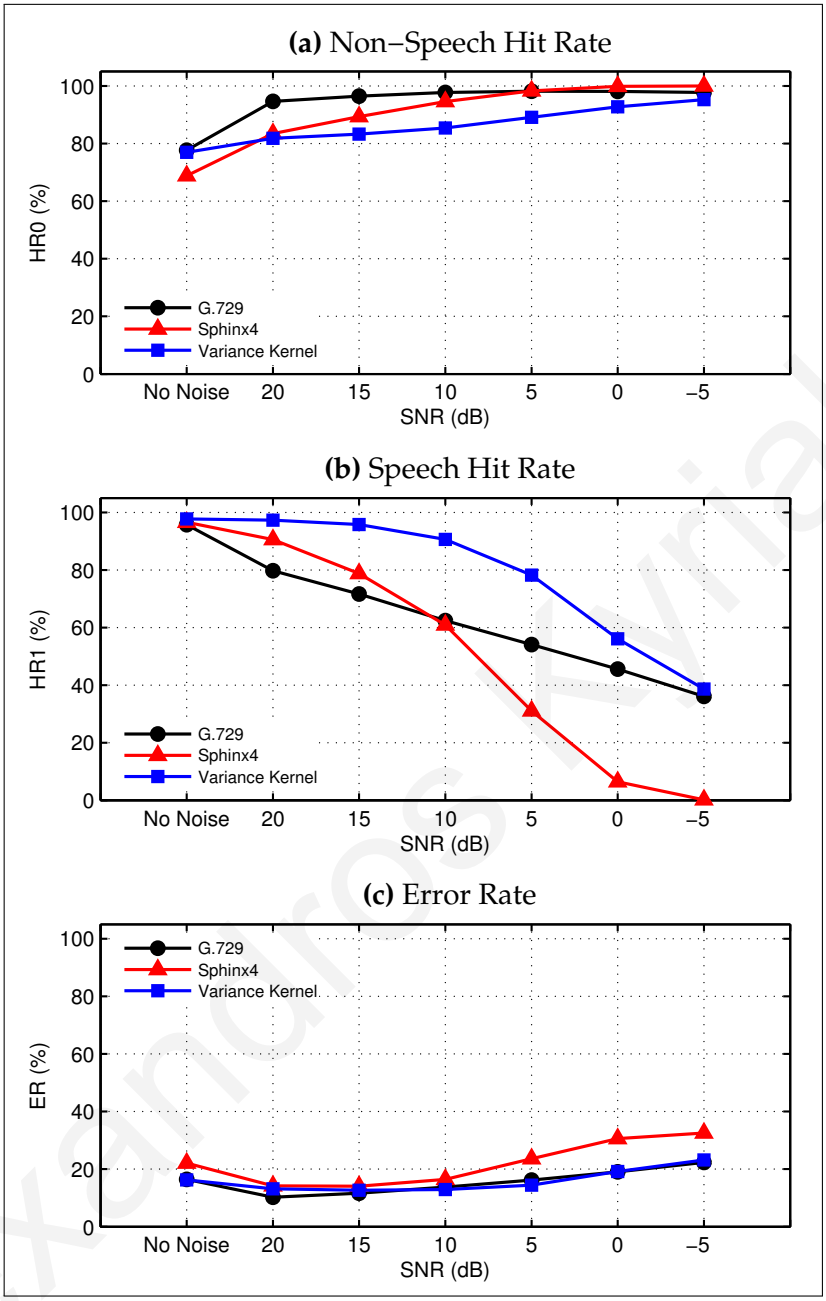

Figure A.62: A comparison of the voice activity detection performance of three different methods, using added noise of type "Street traffic" at various SNR's. The results are for 265 "non-clean" recordings, which contain sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

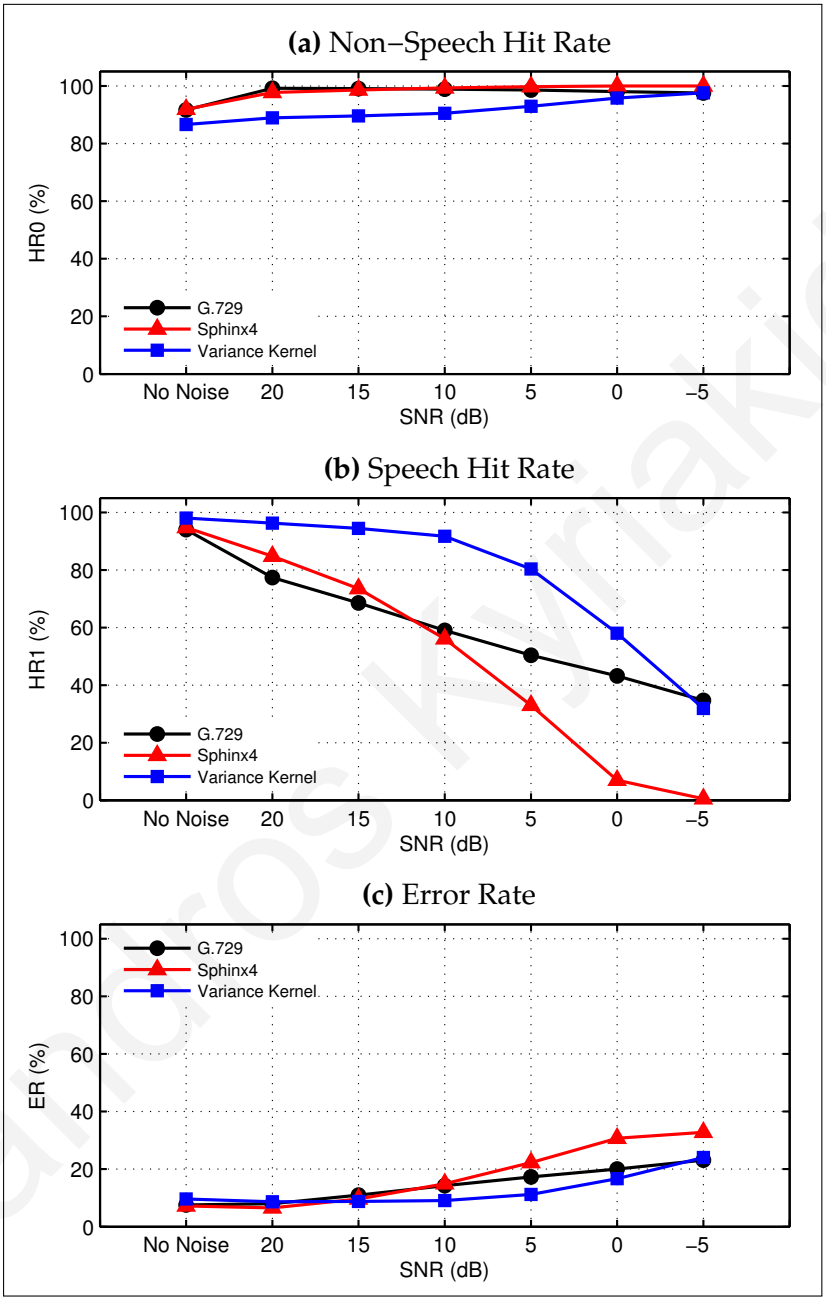

Figure A.63: A comparison of the voice activity detection performance of three different methods, using added noise of type "Street traffic" at various SNR's. The results are for 185 "clean" recordings, which do not contain any sound artifacts. The evaluation measures used are non-speech hit rate (**HR0**), speech hit rate (**HR1**), and error rate (**ER**).

Table A.42: Voice activity detection performance for three different methods using *added noise of type "Street traffic"* at various SNR's. The percentages are calculated from a total of 265 "non-clean" recordings, which contain sound artifacts, and 185 "clean" recordings, which do not contain any sound artifacts. The speech recordings are from 15 different words, each spoken by 15 male and 15 female speakers. The evaluation measures used are *non-speech hit rate* (**HR0**), *speech hit rate* (**HR1**), and *error rate* (**ER**).

		"non-clean" recordings			**"clean" recordings**		
	SNR	**G.729**	**Sphinx4**	**Var. Kernel**	**G.729**	**Sphinx4**	**Var. Kernel**
HR0 (%)	**no noise**	77.70	68.83	76.94	91.66	91.88	86.61
	20 dB	94.62	83.48	81.83	99.20	97.73	88.90
	15 dB	96.46	89.36	83.26	99.08	98.60	89.61
	10 dB	97.74	94.56	85.40	98.88	99.31	90.52
	5 dB	98.17	98.32	89.09	98.58	99.75	92.92
	0 dB	98.08	99.86	92.76	98.05	99.97	95.78
	-5 dB	97.77	99.99	95.24	97.57	100.00	97.61
HR1 (%)	**no noise**	95.74	96.59	97.73	94.01	94.73	98.04
	20 dB	79.77	90.55	97.28	77.34	84.79	96.26
	15 dB	71.66	78.76	95.77	68.58	73.56	94.47
	10 dB	62.39	60.90	90.63	58.99	56.11	91.69
	5 dB	54.10	31.05	78.20	50.37	32.95	80.35
	0 dB	45.56	6.40	56.11	43.21	6.95	58.02
	-5 dB	36.10	0.17	38.65	34.71	0.51	31.89
ER (%)	**no noise**	16.42	22.11	16.28	7.56	7.18	9.62
	20 dB	10.22	14.21	13.13	8.01	6.53	8.67
	15 dB	11.63	14.10	12.66	10.97	9.65	8.79
	10 dB	13.79	16.42	12.90	14.27	14.93	9.09
	5 dB	16.20	23.62	14.46	17.31	22.27	11.22
	0 dB	19.04	30.61	19.19	20.02	30.69	16.67
	-5 dB	22.34	32.55	23.21	23.15	32.79	24.05

Appendix B

Examples of Rank Order Kernel Weights

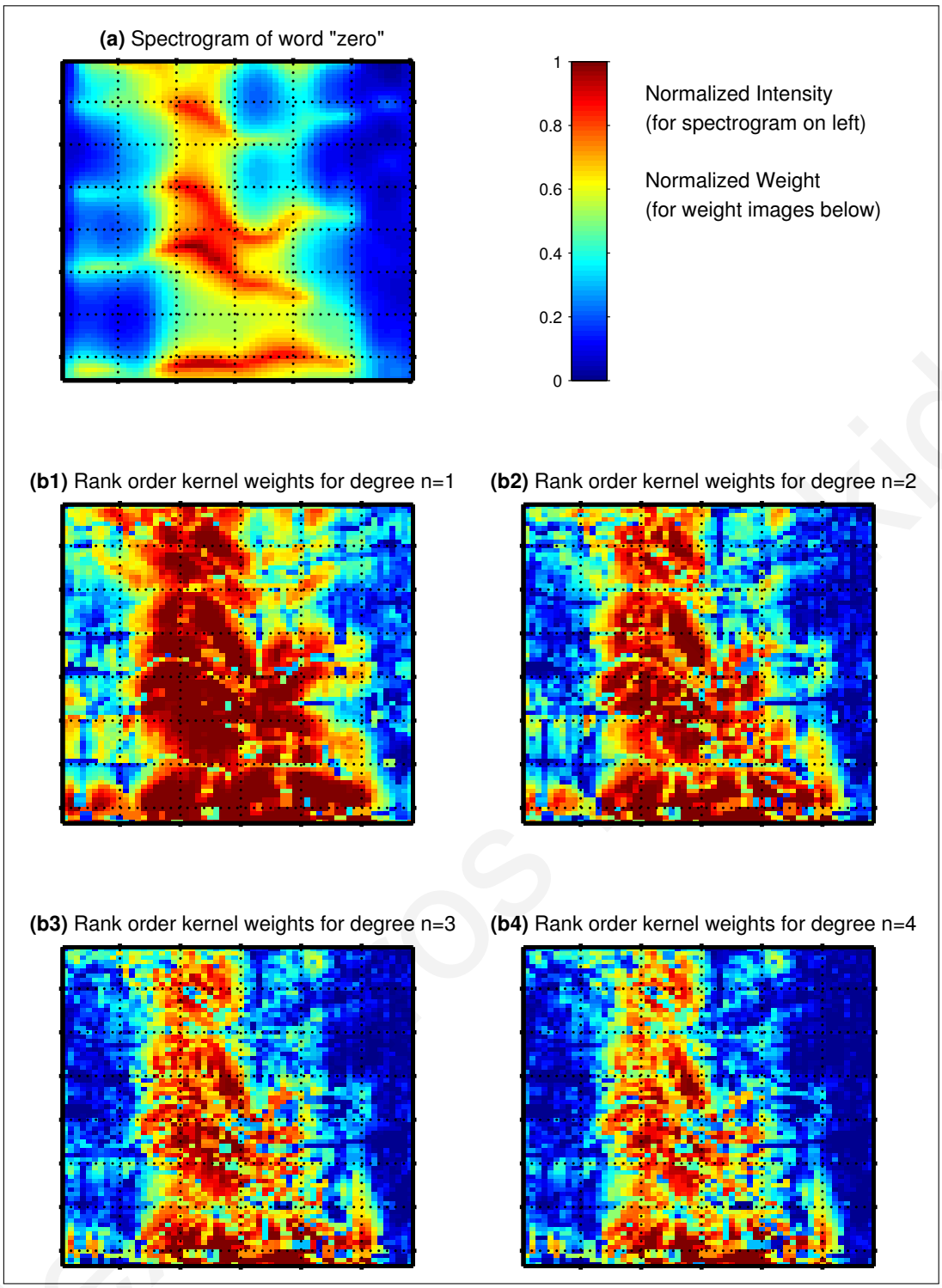

Figure B.1: Weights calculated for a training instance of the word "zero". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

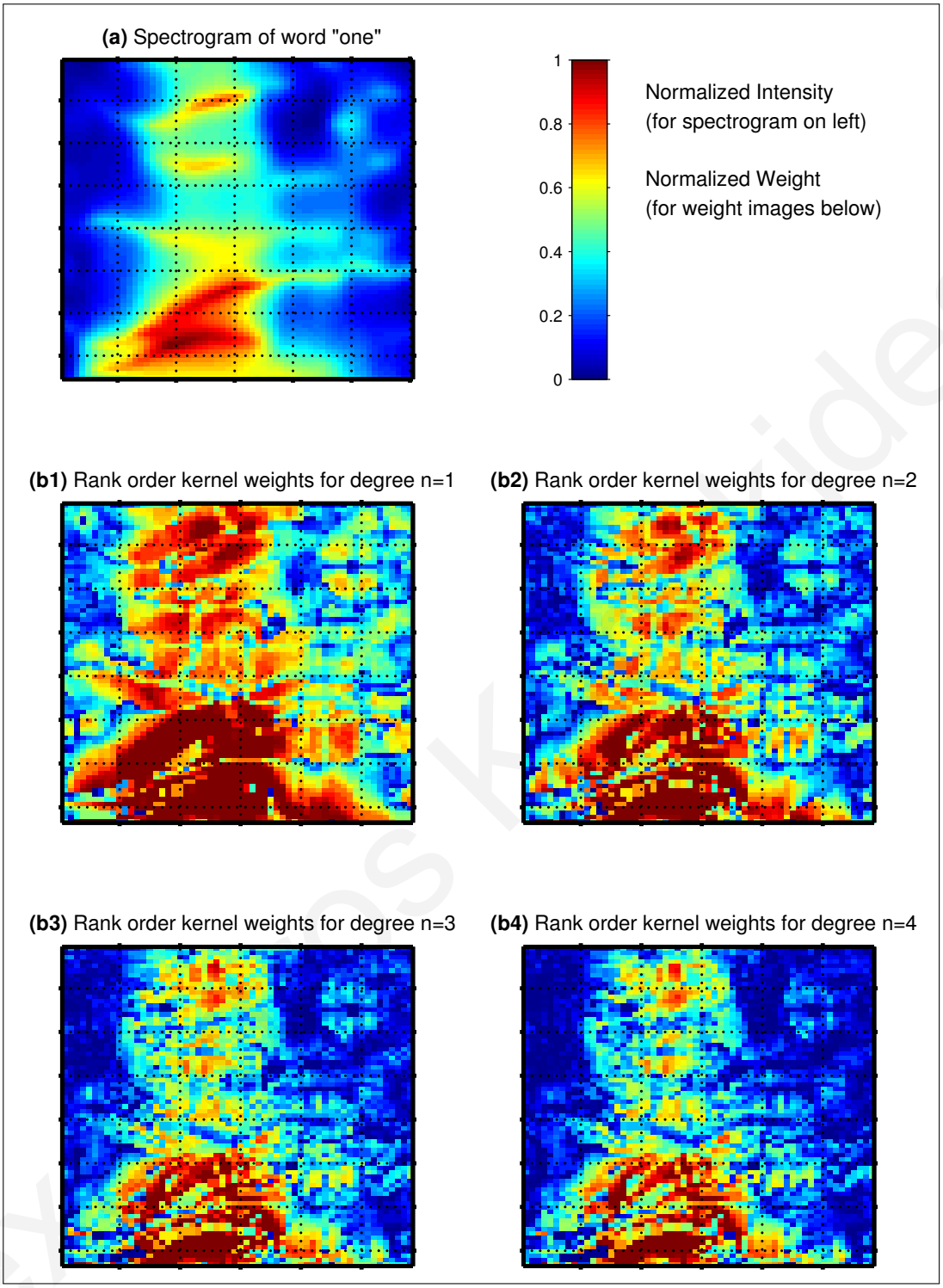

Figure B.2: Weights calculated for a training instance of the word "one". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

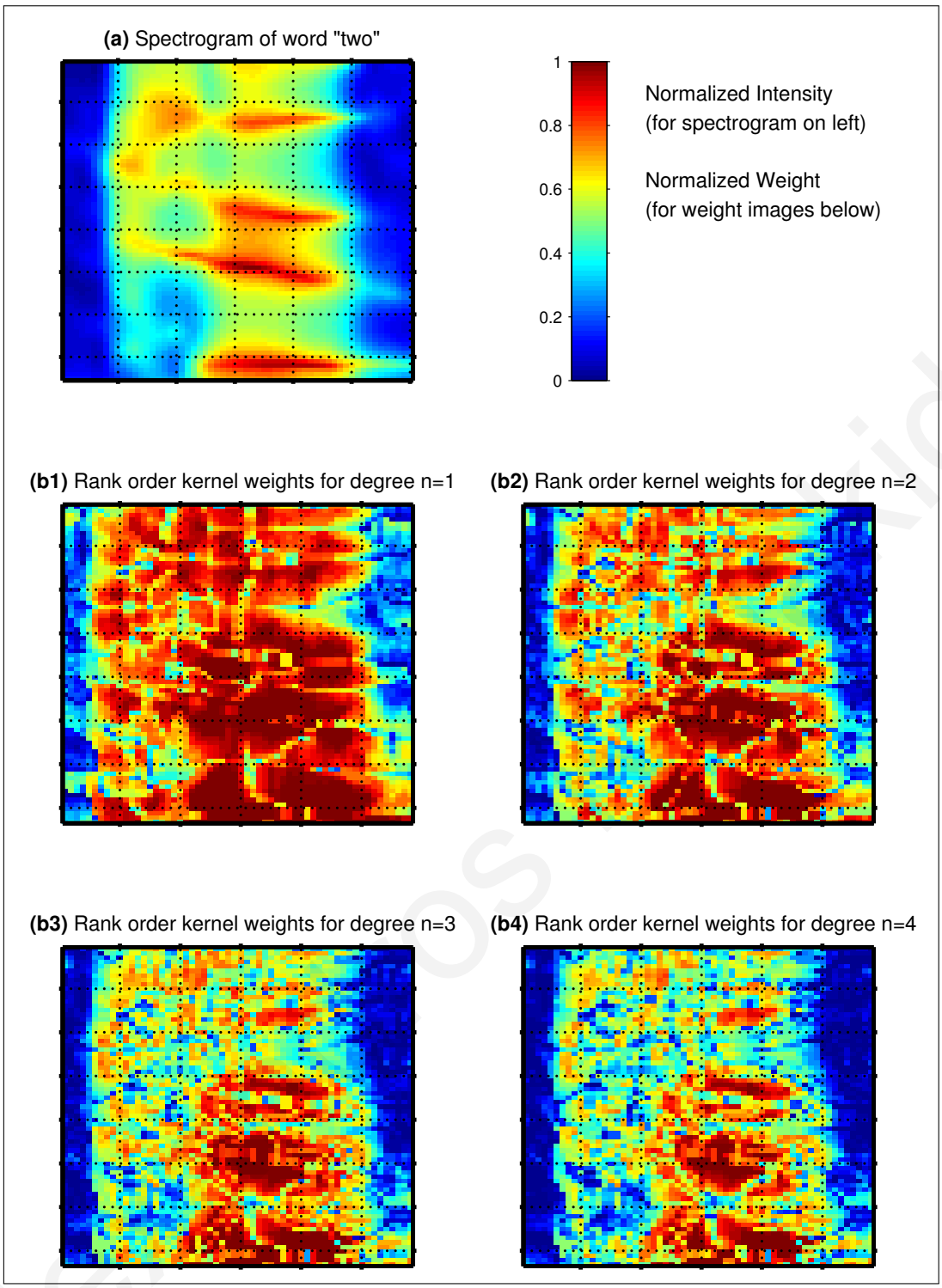

Figure B.3: Weights calculated for a training instance of the word "two". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

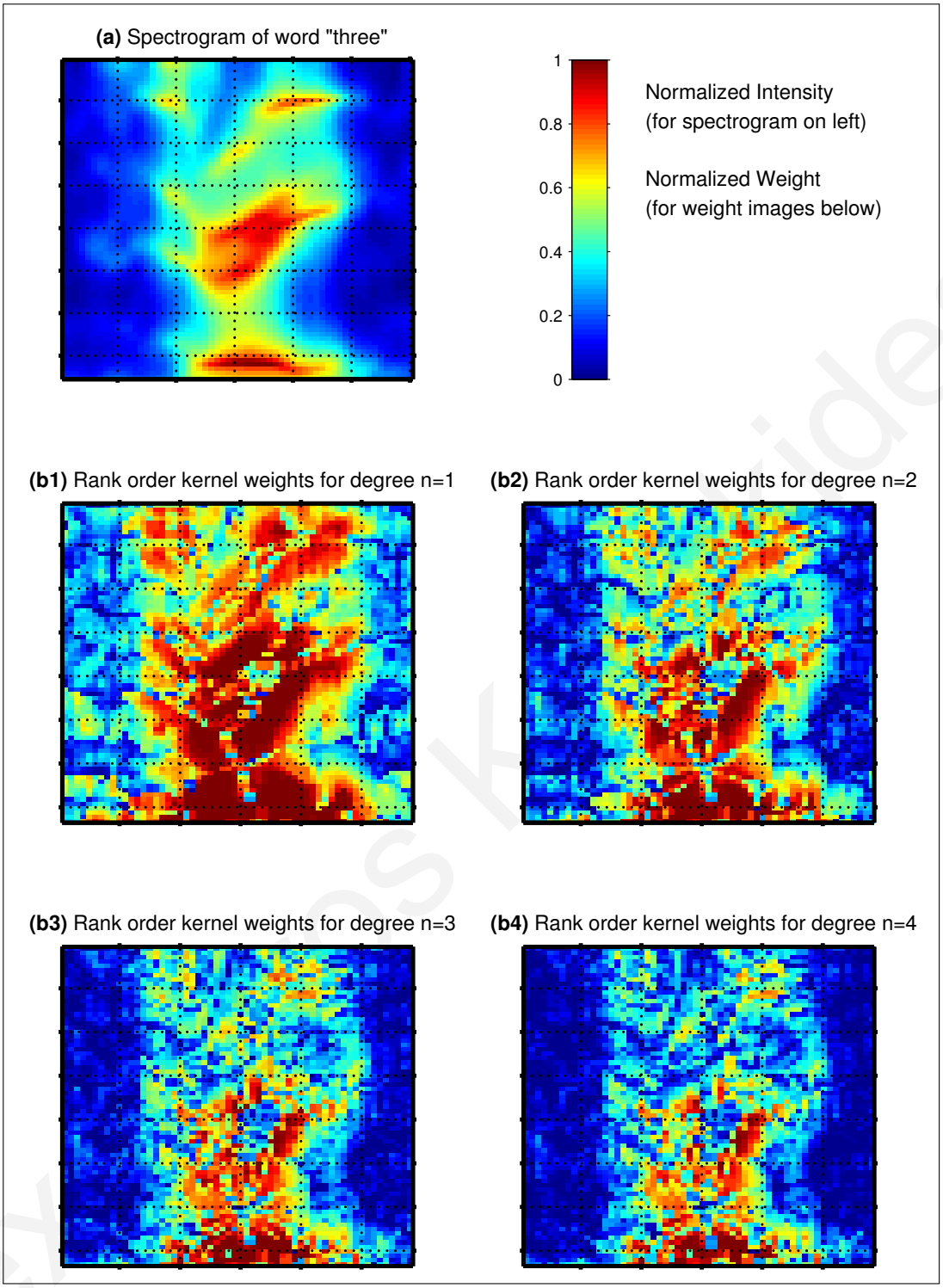

Figure B.4: Weights calculated for a training instance of the word "three". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

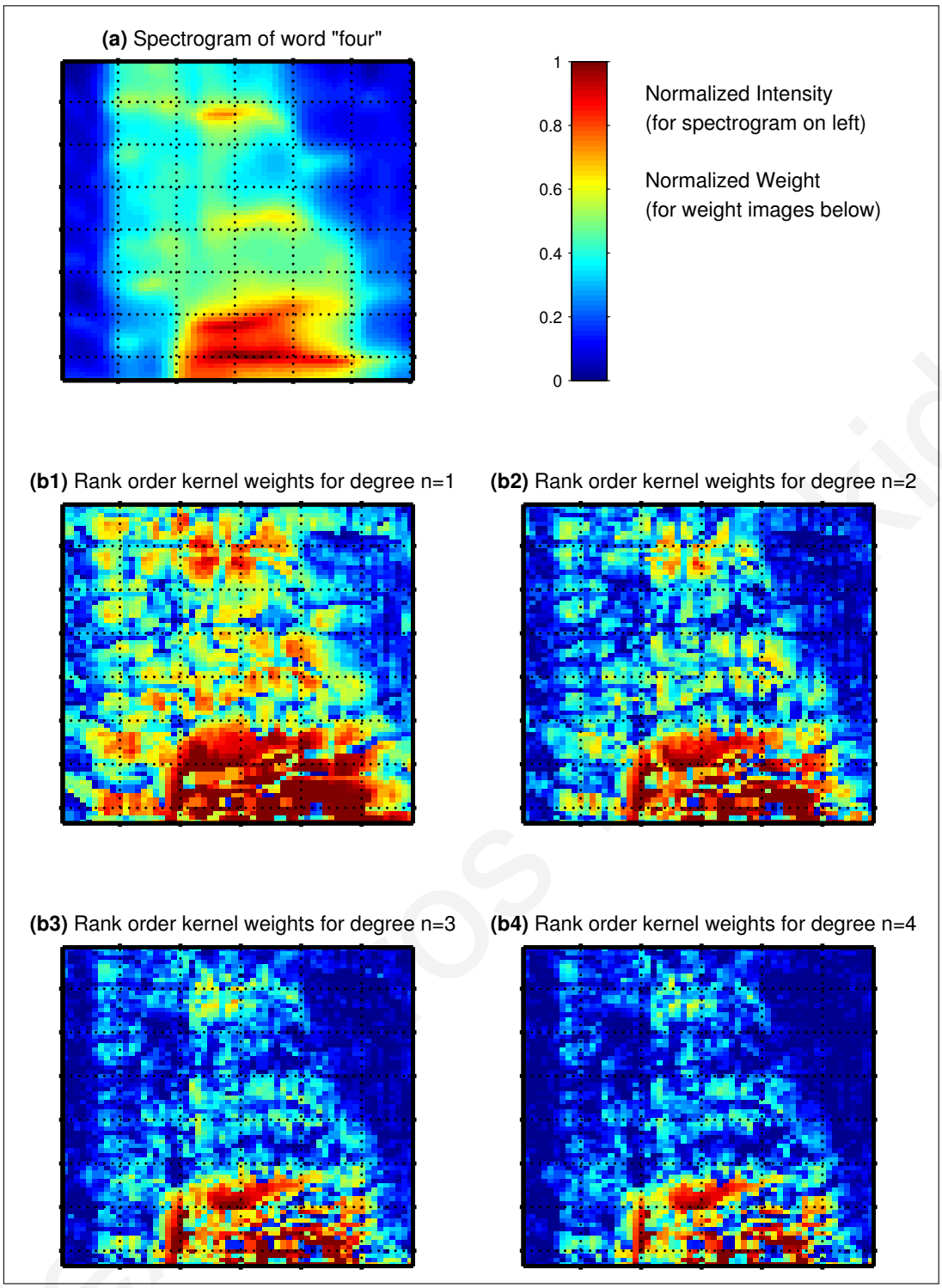

Figure B.5: Weights calculated for a training instance of the word "four". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

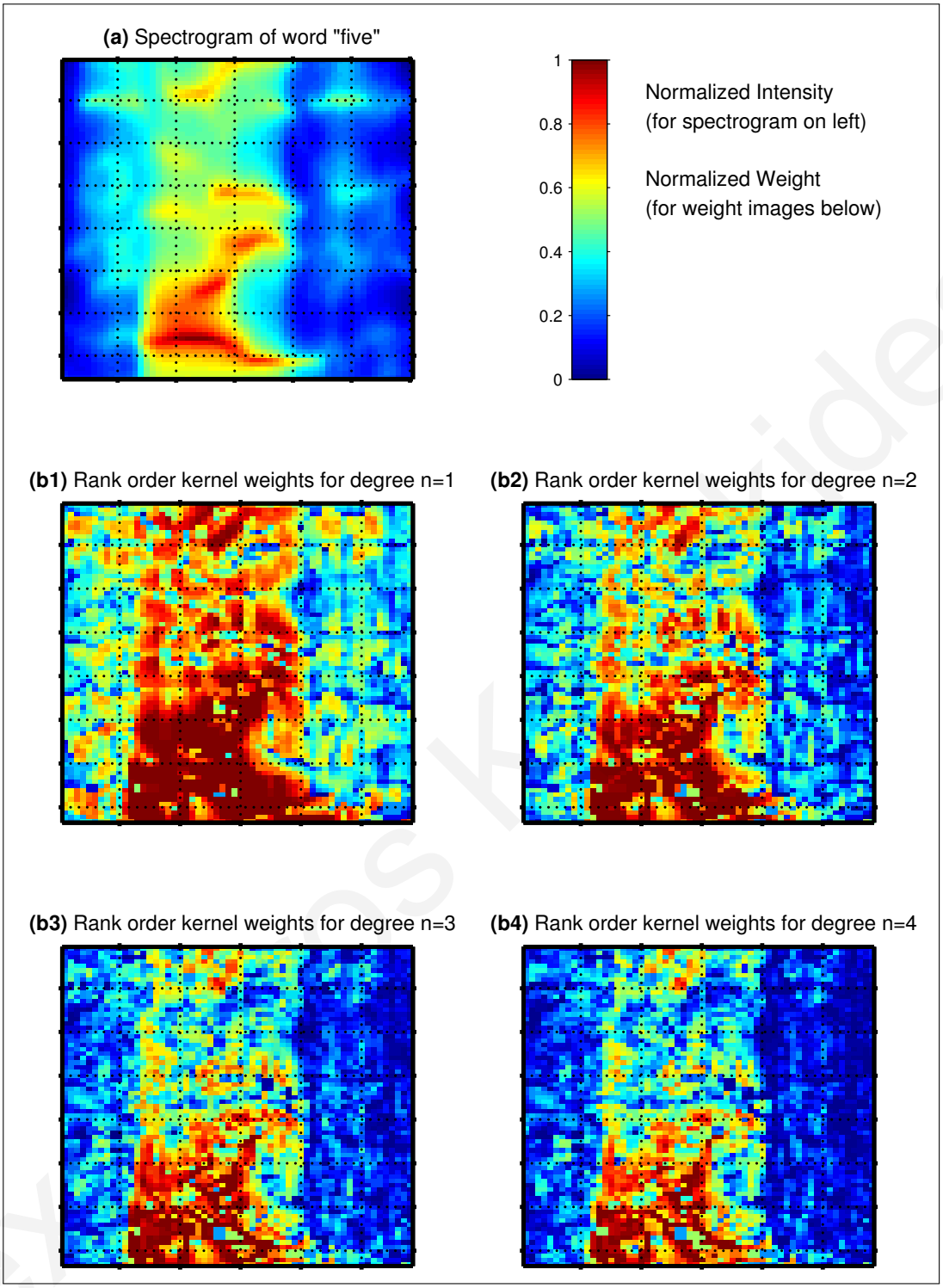

Figure B.6: Weights calculated for a training instance of the word "five". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

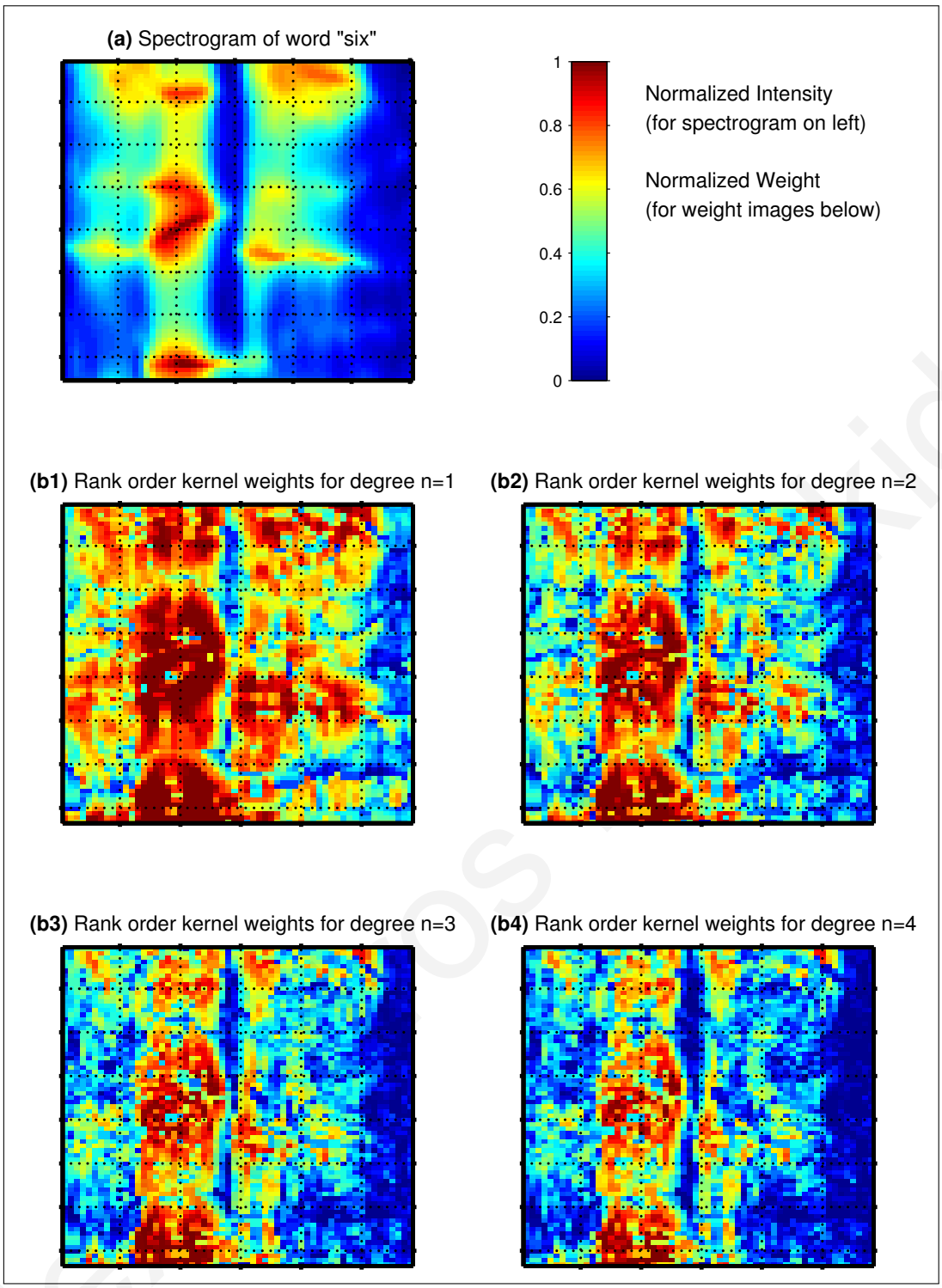

Figure B.7: Weights calculated for a training instance of the word "six". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

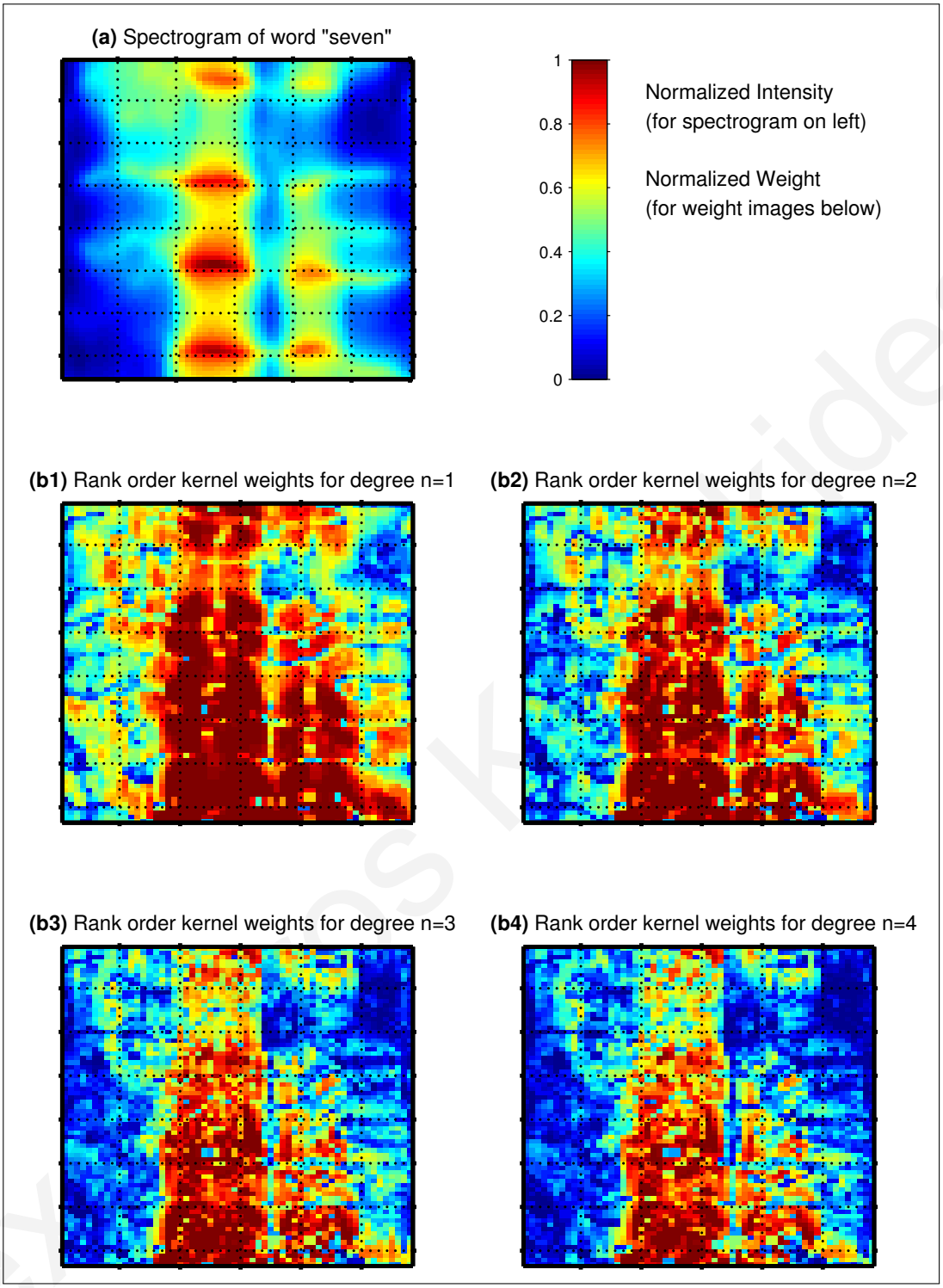

Figure B.8: Weights calculated for a training instance of the word "seven". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

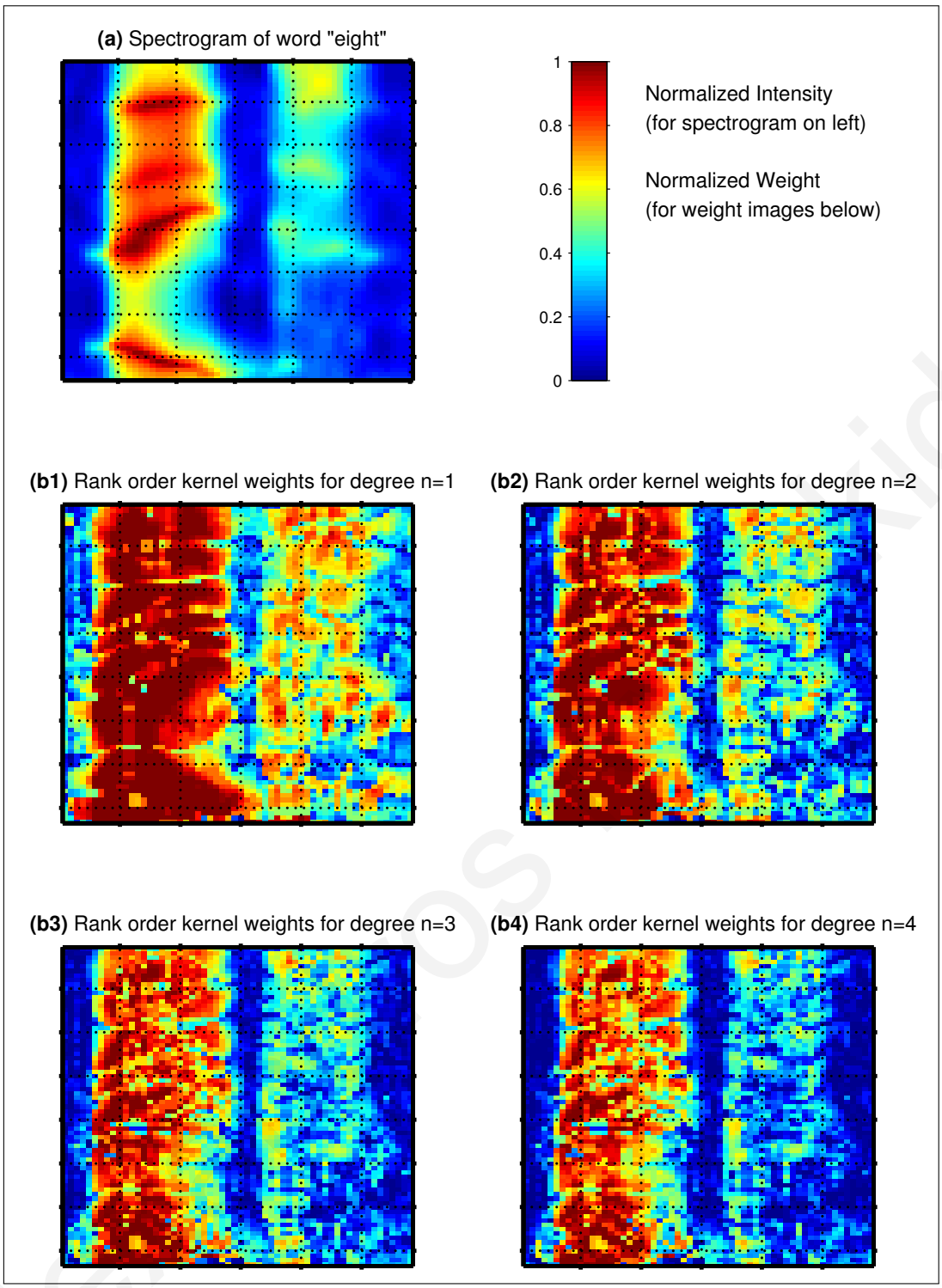

Figure B.9: Weights calculated for a training instance of the word “eight”. Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

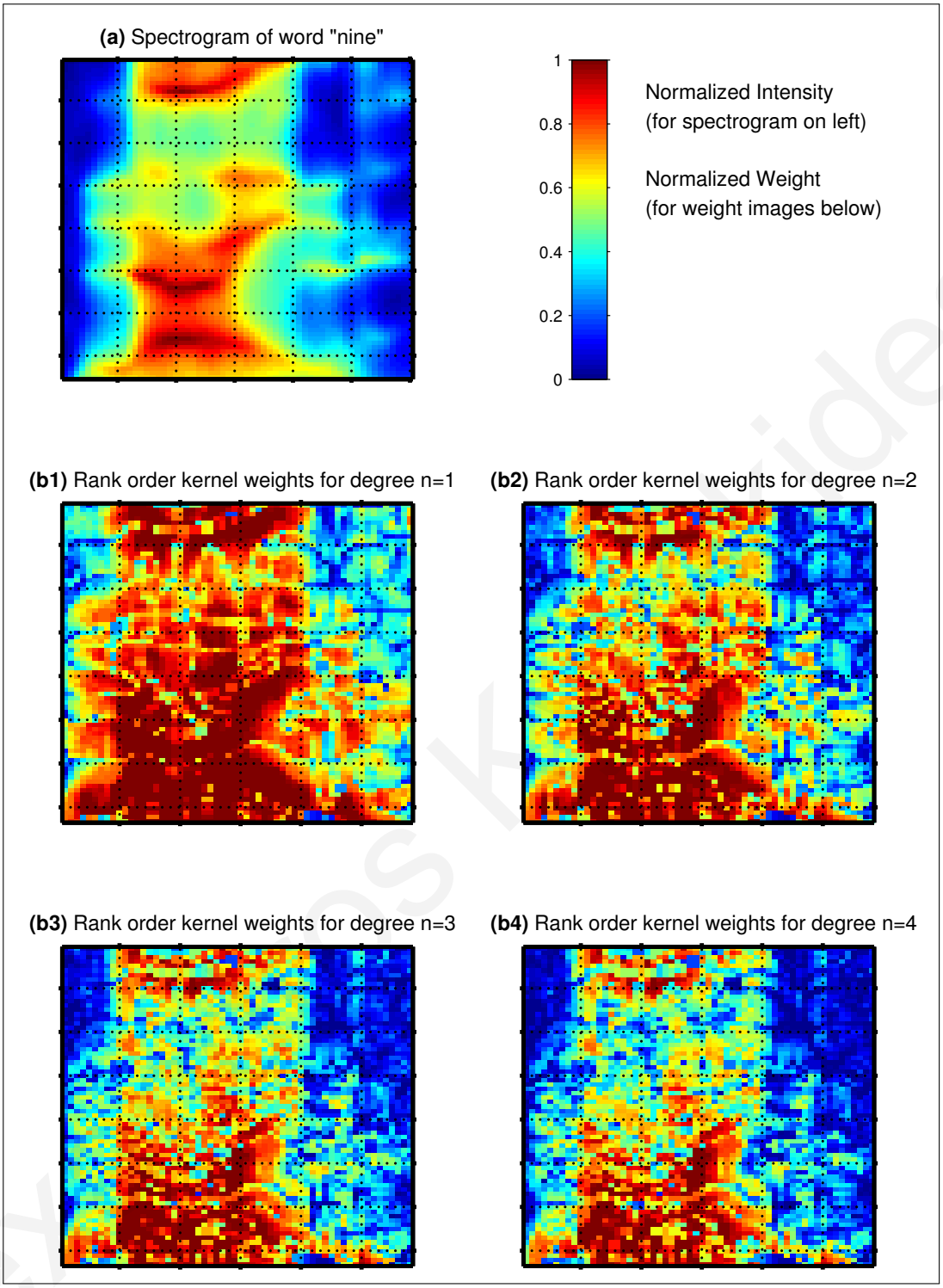

Figure B.10: Weights calculated for a training instance of the word "nine". Higher weights indicate rank order kernel locations which are more robust to white noise. (a) The spectrogram of the word. (b1) The weights calculated for kernels of degree n=1. High weights indicate locations where the rank order of the highest-valued pixel does not change easily with noise. (b2) The weights calculated for kernels of degree n=2. High weights indicate locations where the rank order of the two highest-valued pixels does not change easily with noise. (b3) The weights calculated for kernels of degree n=3. High weights indicate locations where the rank order of the three highest-valued pixels does not change easily with noise. (b4) The weights calculated for kernels of degree n=4. High weights indicate locations where the rank order of the four highest-valued pixels does not change easily with noise.

Appendix C

Significance Tests

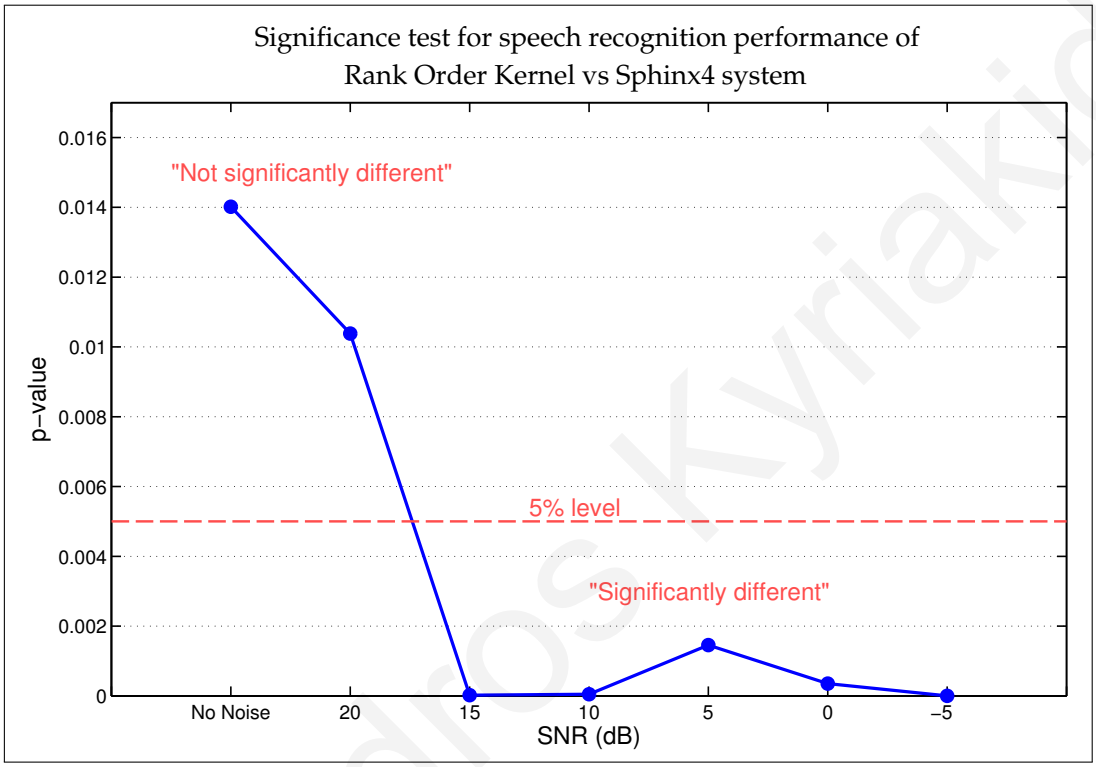

Figure C.1: Significance tests which show that at SNRs of 15dB and below, the speech recognition performance of the Rank Order Kernel method is significantly different from that of the Sphinx-4 system. The p-values were calculated using Fisher's exact test by comparing the number of "correct", "wrong", and "miss" counts between the two methods at each SNR. The performance measures used for these significance tests are the ones obtained using added white noise, as shown in Figure 5.4 on page 154.

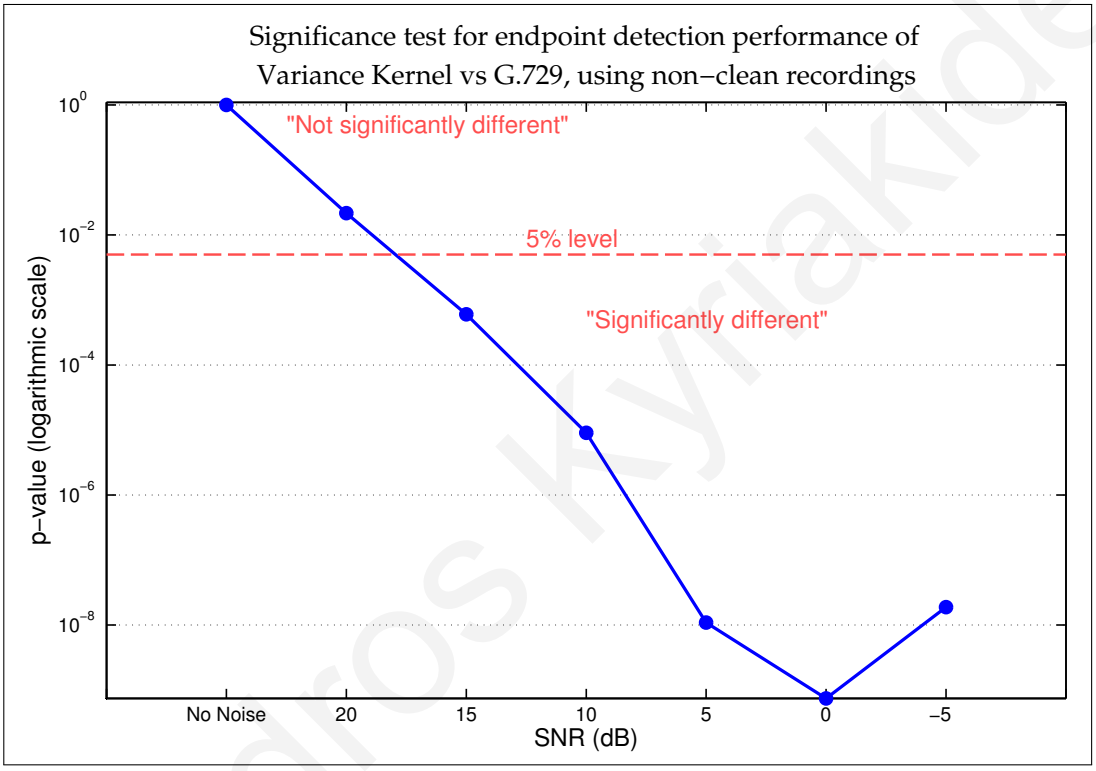

Figure C.2: Significance tests which show that at SNRs of 15dB and below, the endpoint detection performance of the Variance Kernel method is significantly different from that of the G.729 algorithm. The p-values were calculated using Fisher's exact test by comparing the number of "correct", "wrong", and "miss" counts between the two methods at each SNR. The performance measures used for these significance tests are the ones obtained using non-clean recordings and twenty noise types, as shown in Figure A.1(a) on page 178.

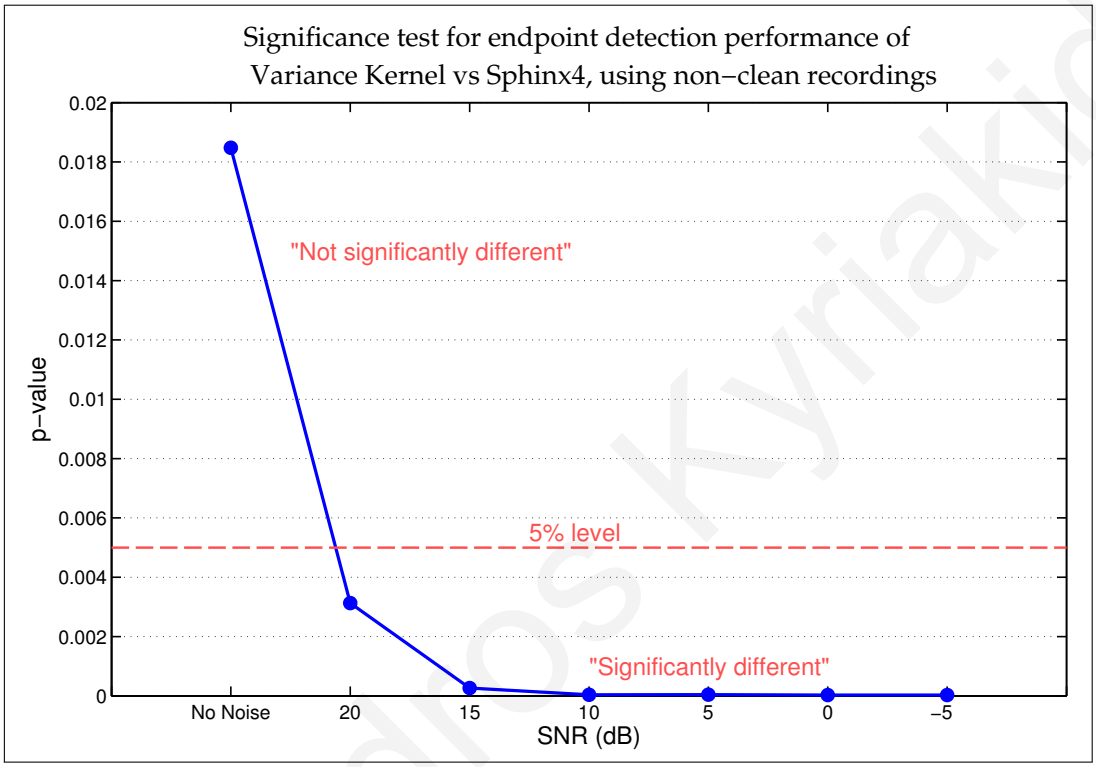

Figure C.3: Significance tests which show that at SNRs of 20dB and below, the endpoint detection performance of the Variance Kernel method is significantly different from that of the Sphinx-4 system. The p-values were calculated using Fisher's exact test by comparing the number of "correct", "wrong", and "miss" counts between the two methods at each SNR. The performance measures used for these significance tests are the ones obtained using non-clean recordings and twenty noise types, as shown in Figure A.1(a) on page 178.

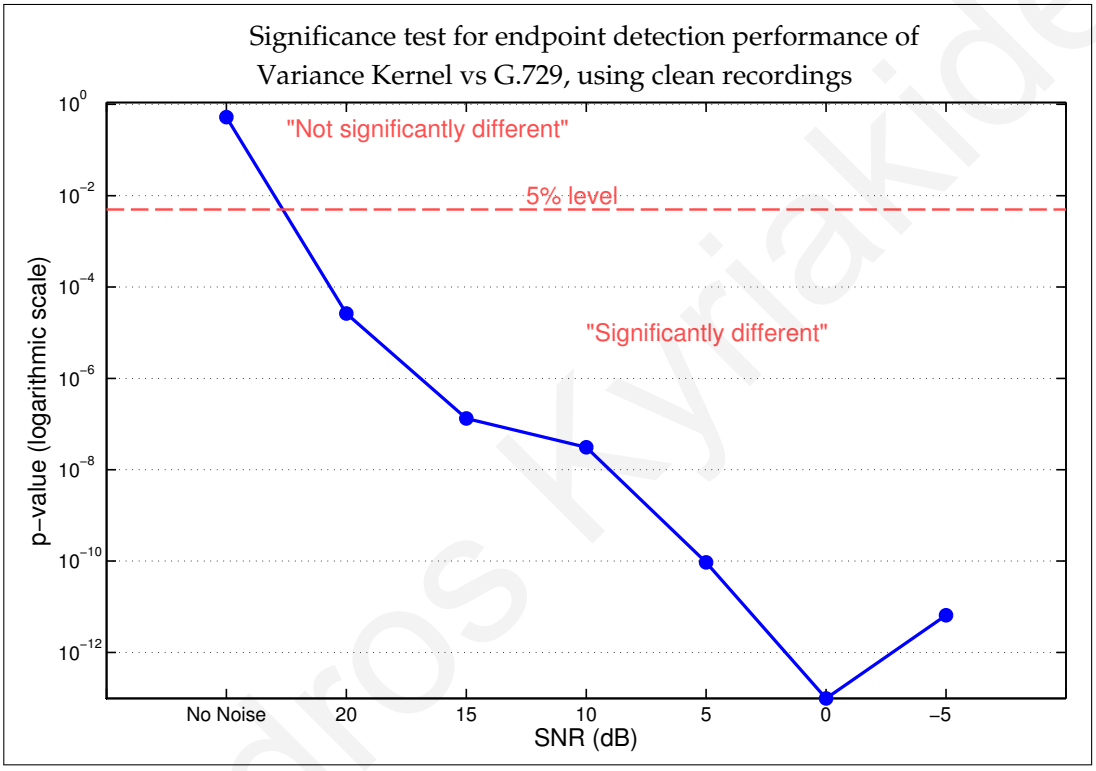

Figure C.4: Significance tests which show that at SNRs of 20dB and below, the endpoint detection performance of the Variance Kernel method is significantly different from that of the G.729 algorithm. The p-values were calculated using Fisher's exact test by comparing the number of "correct", "wrong", and "miss" counts between the two methods at each SNR. The performance measures used for these significance tests are the ones obtained using clean recordings and twenty noise types, as shown in Figure A.1(b) on page 178.

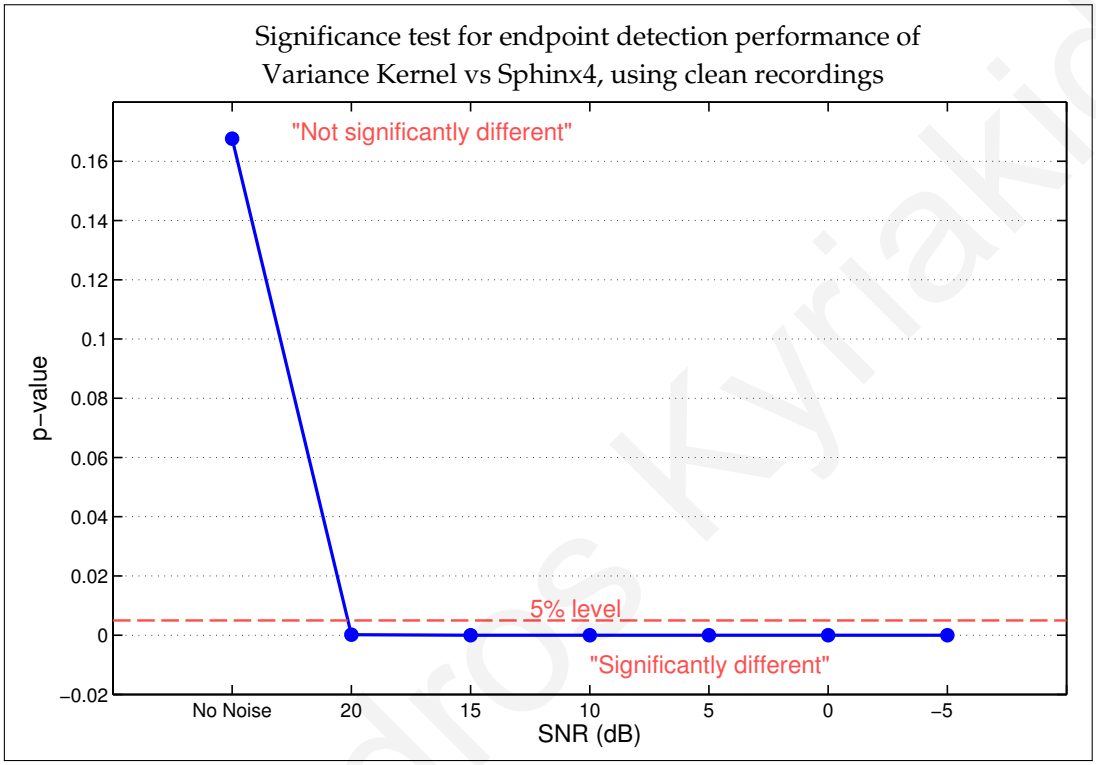

Figure C.5: Significance tests which show that at SNRs of 20dB and below, the endpoint detection performance of the Variance Kernel method is significantly different from that of the Sphinx-4 system. The p-values were calculated using Fisher's exact test by comparing the number of "correct", "wrong", and "miss" counts between the two methods at each SNR. The performance measures used for these significance tests are the ones obtained using clean recordings and twenty noise types, as shown in Figure A.1(b) on page 178.