Column by Column Image-Signal Enhancement

by

Eilat Vardi-Gonen

A dissertation submitted to the Graduate Faculty in Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2008

This manuscript has been read and accepted for the
Graduate Faculty in Computer Science in satisfaction of the
dissertation requirements for the degree of Doctor of Philosophy.

Date — Dr. Gabor T. Herman, Chair of Examining Committee

Date — Dr. Theodore Brown, Executive Officer

Dr. Michael Chan

Dr. Robert Haralick

Dr. Arlene Neuman
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Column by Column Image-Signal Enhancement

by

Eilat Vardi-Gonen

Advisor: Gabor T. Herman

The general problem that we consider is the real-time enhancement of a noisy image as it arrives column by column (time is associated with the horizontal axis). The enhancement is modeled by a column of real values between 0 and 1 (with as many pixels as in the noisy image column), which is used as a mask on the noisy image column. That is, the clean image column is estimated by multiplying the noisy image column by the mask column, pixel by pixel. In this work, the real-valued column is attained by "fuzzifying" a binary column, which is estimated from the noisy image column using a prior model and a noise model. The prior model is specified by a statistical distribution that assigns to binary columns a probability of occurrence. The noise model is specified by a statistical distribution that assigns the likelihood of occurrence of the noisy image column given the binary column. These statistics should either be known or estimated from a training set of typical images from the application area. Given models of this type, our task is to design an algorithm that will produce the sought-after binary and "fuzzified" columns in "real-time," that is as the noisy image arrives column by column.

In many applications signals are transformed into images by taking the squared norm of the Short-Time Fourier Transform of the signal, called the spectrogram. These spectrograms are extremely redundant; columns represent spectral information of overlapping time intervals. We, therefore, tested the possible applicability of

image processing algorithms to the problem of increasing recognition in the hearing aid application. It was our hypothesis that for every spectrogram image of noisy speech there exist a binary image and its "fuzzified" image that (along with the noisy spectrogram image) contain the information essential for estimating the clean speech signal. The proposed processing was tested on 24 normal hearing subjects using the Modified Rhyme Test. Unfortunately, the specific choices that we made in the development of our processing methodology decreased the recognition of noisy speech signals.

I dedicate this to my parents. You are the reason that I have chosen this path. May it be a good path!

Acknowledgments

I would like to thank many people and funding opportunities that made this Ph.D. a reality.

First, I must thank my advisor, Dr. Gabor Herman. He has guided me, supported me and advised me through the years. His accessibility and responsiveness has always been much appreciated! This project would not be possible without his time, attentiveness, care and patience.

I would like to thank the rest of my committee for dedicating their time throughout my Ph.D. years. Each member had an important role in making the dissertation the project it ended up being. In particular, I would like to thank Dr. Arlene Neuman. She was the other required ingredient to make this project possible. She helped Dr. Herman and myself to come up with this thesis topic, formulate it and understand the difficulties in this application area. She was responsible for keeping the project realistic in terms of the application we were considering. She made data available to us, listened to results with us, and gave her input on results and write-ups. I would particularly like to thank her for being available to chat online when the need arose. I would like to thank Dr. Robert Haralick, who listened to my semesterial seminar talks and provided invaluable ideas and suggestions. I thank Dr. Michael Chan for giving his time and ideas, especially as an outside member. I am grateful to the input and attention provided to me by the late Dr. Dan Butnariu and Dr. Ivan Kazantsev while working with me on this project during their long visits at the Discrete Imaging and Graphics (DIG) Group. I am so appreciative to the previous graduate students and post-docs - Dr. Bruno Carvalho, Dr. Sébastien Fourey, Dr. Edgar Gardũno, Dr. Miroslaw Kalinowski, Dr. Hstau Liao and Dr. Roberto Maribini. Thanks for being wonderful role models and for your assistance along my way. I want to thank the rest of the previous DIG members - Laslo Cernetic, Joel Dubowy, Arun Kulshreshth, Lajos Rodek, László Ruskó and Deniz Sarioz. I

am thankful to all of them for helping me with all computer related issues and for being wonderful friends! I am grateful to the current DIG students - Wei Chen, Ran Davidi, Joanna Klukowska, and Lucas de Melo Oliveira, for helping me with administrative issues, lyx issues, computer issues and for being great friends. I don't know where I would be without all of your collective help.

I would like to acknowledge the funding opportunities made available to me. Foremost, I would like to acknowledge my individual fellowship from the National Library of Medicine, National Institute of Health, grant F37 LM008611, which generously supported me financially from 2005-2007. I am also indebted to the Graduate Center, CUNY, for providing me with science fellowships, a tuition fellowship and a university fellowship. I thank Dr. Herman, who supported me through his NSF grant DMS0306215 and NIH grant HL70472. I thank Dr. Felix Wehrli for providing me with a NIH training grant, NIH T32 CA 74781, during my time at the University of Pennsylvania.

Finally, I have to thank my family and (non-school) friends, without whom this degree would not have started or have been completed. I thank my parents for starting me on this path. I thank my in-laws for their financial support as I started my commute as a Ph.D. student in New York. Thanks to my siblings for their support and friendship. Thanks to my good friends Lydia Musher, Carly Goldberg and Michael Goldberg, who gave me the strength to finish. They were there to talk to me, be with me and help me with all kinds of issues. I am a **lucky** person to have them as my friends! Thanks to my dear child, my **beloved** son Nathan - he puts a smile on my face and reminds me why life is precious and beautiful. Most of all, thanks to my husband Itamar, who has stood by me all along, through better and worse times, through "poorness", health and death. I love you. Thanks for allowing, pushing, ..., me to do this degree.

Contents

List of Tables

List of Figures

Nomenclature

α	a large integer, 100 in our case
β	the inverse temperature in the annealing schedule
μ	a binary value
θ	a grayscale image
τ	the sampling distance
ω	a binary image
d	the integer $\frac{2I-2}{s}$, 8 in our case
i	a row number
j	a column number
p	$\left\{ \frac{M\left(\varpi_2; \hat{j}\right)}{M\left(\varpi_1; \hat{j}\right)} \right\}^{\hat{\beta}}$
s	the time skip step
w	a window function
H_j	the set of pixels in column j, except the bottommost pixel
I	the number of rows in an image
J	the number of columns in an image

L the length of the time sequence

NR the number of times an annealing schedule is run per column

NSR the noise sampling rate, 22,500 Hz in our case

SSR the signal sampling rate, 10,000 Hz in our case

TSR the sampling rate of a time sequence

$\times$ times

$\langle x \rangle$ the nearest integer to the real number x

$\lfloor x \rfloor$ the nearest integer that is no greater than the real number x, the floor of x

$|x|$ the Euclidean norm of the complex number x

$(a,b]$ the interval between a and b, including b

$[a,b]$ the interval between a and b, including a and b

$\{x,y,z\}$ a set containing objects x, y and z

$x \in X$ x belongs to set X

$x \notin X$ x does not belong to set X

$x \simeq y$ x is approximated by y

$b[h]$ the band number of pixel h

$\mathrm{bin}[\theta[h], b[h]]$ the band-dependent bin number of gray value $\theta[h]$ at pixel h

$x[i]$ the value of the sequence x at index i

$N[h]$ the locations of the (at most) six pixels in the neighborhood of the pixel h

$N_\omega[h]$ the binary values in the neighborhood of the pixel h in the binary image ω

$x[i,j]$ the value of the two-dimensional image x at point $[i,j]$

$f(t)$ the value of the function f at point t

$p(X)$ the probability of event X occuring

$\exp(x) = e^x$ the base of the natural logarithm to the power of x

$\min(\delta,\eta)$ the minimum of the real values δ and η

$\max(\delta,\eta)$ the maximum of the real values δ and η

$\log_{10}(x)$ the logarithm based 10 of x

$\ln(x)$ the natural logarithm of x

$[\mathscr{F}f](r)$ the Fourier transform of function f at frequency r

$\left[\mathscr{F}^{-1}F\right](t)$ the inverse Fourier transform of function F at time t

$[\mathscr{D}_I x][i]$ the discrete Fourier transform of the sequence x at frequency index i

$\left[\mathscr{D}_I^{-1}X\right][m]$ the inverse discrete Fourier transform of the sequence X at time index m

$\left[\mathscr{D}^w_{L,I,n}x\right][i]$ the discrete Fourier transform of the windowed sequence x at frequency index i

$[\mathscr{E}_I X][m]$ the inverse discrete Fourier transform of the windowed sequence X at time index m

$\left[\mathscr{T}^w_{L,I,s}x\right][i,j]$ the Short-Time Fourier Transform of the sequence x at time index i and frequency index j

$\left[\mathscr{S}^w_{L,I,s}x\right][i,j]$ the spectrogram of the sequence x at time index i and frequency index j

$M\left(\varpi;\hat{j}\right)$ the pseudo-posterior likelihood of the binary image ϖ at column $\hat{j}$

$C\left(\varpi;\hat{j}\right)$ the column energy of the binary image ϖ at column $\hat{j}$

Chapter 1

Introduction

This chapter describes the background and significance of this work in Section 1.1, the different noise types that can arise in real world problems in Section 1.2, the existing methodologies for this general area of work in Section 1.3 and a brief description of our methodology in Section 1.4.

1.1 Background and Significance

There are many situations in which signal enhancement techniques are necessary [11, 57]. Some examples of such situations are communication in noisy environments (communication with pilots), communication in an environment with competing speakers (a party situation), communication in reverberant spaces (in an enclosed area), communication over noisy channels (over a cellular phone), removing scratches from old recordings, and differentiating between heart and lung sounds [30].

The motivation for the work described here is the desire to enhance noisy signals in "real-time." The applications for such work arise in real-time signal processing problems [11] such as design of hearing aids [57] and critical patient monitoring for atrial arrhythmias [69]. The real-time aspect of the signal processing is essential in

these applications. In the hearing aid case it is imperative that the estimated clean speech signal not be delayed so much that understanding decreases due to lack of simultaneity with visual clues. In the atrial arrhythmia application, the real-time enhancement is necessary in order to bring about a timely correcting action.

In many signal processing applications, the signals are viewed as two-dimensional gray images called spectrograms, which are the squared norm of the short-time Fourier transform (STFT) [11, 57, 69] of the signal. Furthermore, a linear transformation of the spectrogram is performed in order to create spectrogram images which are used for viewing the spectrograms. The STFT, the spectrogram and the spectrogram image are all time-frequency representations of the signal that provide us with a two-dimensional image whose columns are spectral representations of short time intervals of the windowed signal; time is on the horizontal axis and frequency is on the vertical axis. The STFT is a complex-valued image. The spectrogram is a real-valued grayscale image whose values represent the signal intensity in the time/frequency location. The spectrogram image is a non-negative integer-valued image whose values are a function of the intensity of the signal in that time/frequency location. Though the spectrogram is a commonly used signal to image transform, it is only one of many transforms that produce a sequence of real-valued columns from a signal. Other time-frequency representations include the positive time-frequency distributions [56], wavelets [11, 15, 19, 48, 57] and Gabor frames [15, 26, 27, 33].

In [57], Quatieri claims that experienced researchers can "read" the spectrograms of clean speech signals. By that it is meant that experienced users of speech spectrograms can determine what was said from the spectrograms of the clean speech signals. That would suggest that visually clean speech spectrograms contain enough information to recover the clean speech signal, as is noted in [79]. By "reading" the spectrograms, we theorized that given a clean speech spectrogram, a binary

indicator image describing the presence of speech in each time/frequency location (pixels are assigned 1 if clean speech is present and 0 otherwise) is of importance.

Let us define a $[0,1]$**-valued column** to be a column of real values between 0 and 1, and let $[0,1]$**-valued image** be a collection of $[0,1]$-valued columns. In the case of noisy speech spectrograms, we believed that much of the essential information in the spectrograms is capturable by $[0,1]$-valued images, which indicate the probability of presence of clean speech in each time/frequency location (pixel). It was, therefore, our hypothesis that the recognition performance of noisy signals can be improved using such $[0,1]$-valued images. In this work, we estimated these $[0,1]$-valued columns from binary columns. The binary columns were estimated from the noisy speech spectrogram using prior information associated with clean signals (the prior model) and noise characteristics (the noise model). In particular, using these models, the noisy speech spectrogram is segmented to produce a binary image, column by column. This binary image is "fuzzified" to create the $[0,1]$-valued image, column by column, that is then used to enhance the noisy speech signal. We estimate the clean speech spectrogram by using the $[0,1]$-valued image as a multiplicative mask (pixel by pixel) on the noisy spectrogram.

We investigated the use of image processing techniques for the hearing aid application for two main reason: 1) current signal processing techniques used for noise reduction in hearing aids do not increase recognition performance [38, 46], which is the main goal of signal processing in this application, and 2) image processing techniques have been shown to be useful in other signal processing applications [30], but to our knowledge have not been used in this way in this field.

Therefore, the work described here differs from existing signal processing techniques for the hearing aid application in that our approach 1) models the binary images corresponding to the unknown clean speech spectrograms and 2) models the correspondence between the noisy speech spectrogram and the corresponding

binary image. Furthermore, both a "fuzzy" version of the binary image, a $[0,1]$-valued image that arises from a binary image, and the noisy speech spectrogram are used to estimate the clean speech. The estimation of the binary image is performed by segmenting the noisy spectrogram using the Metropolis algorithm with an annealing schedule and prior information. Our experience with the Metropolis algorithm with an annealing schedule has been a positive one. We applied it successfully to two very different applications, which we now describe briefly.

The first application was that of reconstructing two-dimensional binary images from only three projection directions [13]. The implemented algorithm, which was later considerably sped up in [72], successfully reconstructed both semiconductor surface layer representations in [29] and mathematically defined cardiac cross-section images based on [59] Figure 2-23.

The second application was that of segmenting trabecular bone from *in-vivo* magnetic resonance (MR) images [32, 71]. Such segmentation algorithms are needed for the evaluation of the extent of osteoporosis in patients. The difficulty with this task is that *in-vivo* MR images are of low resolution as compared to the trabecular bone structures, which is the reason that other simple segmentation techniques had not proven useful.

In the last few years, computers have become faster and have more memory. This trend is expected is continue in the coming years. This has made it possible to use more advanced techniques to solve the general signal enhancement problem. In particular, it has made it possible to consider the use of the time-consuming iterative Metropolis algorithm with an annealing schedule for this type of work.

1.2 Noise Types

Listening to speech rarely occurs in perfectly quiet environments. Signal-to-noise ratio (SNR) in decibels (dB) is the metric used to describe the relationship between the level of the signal and the level of the noise. The SNR in decibels is 20 times the logarithm base-10 of the ratio between the root-mean-square (rms) amplitude of the signal and the rms amplitude of the noise [61], symbolically

$$\mathrm{SNR} = 20 \times \log_{10} \left(\frac{\mathrm{rms}\,(\mathrm{signal})}{\mathrm{rms}\,(\mathrm{noise})} \right). \tag{1.1}$$

In a survey of speech levels in various noise environments [55], noise levels and speech levels were measured in schools, homes, hospitals, department stores, trains, and aircrafts. The SNR values in these environments ranged from -2 dB to +14 dB. That means that in the poorest case, the speech signal was 2 dB below the noise level. In the best case measured, speech was 14 dB above the average noise level. For reasons further explained in Section 4.2, we decided to test our methodology at two low SNR levels in the normal range, 0 and 5 dB SNR. These are values for which signal processing is the most difficult and has the potential of being the most beneficial.

Noisy speech signals can arise in many ways. The noise can be correlated (arise from speech echos) or uncorrelated (arise from a separate signal source) with the speech signal. The noise can be stationary (fan, computer) or non-stationary (people talking in background). The noise can be multiplicative, additive or convolutional.

The specific problem that we addressed is that of increasing speech recognition in real-time of noisy speech signals that were degraded by uncorrelated, non-stationary, additive noise. We further assume that only one channel is available, meaning that only the noisy speech signal is available. The reason for choosing to look at speech degraded by uncorrelated, non-stationary noise is that it is the

common situation in which hearing aids are used and is the most difficult case. Using correlated or stationary noise would give more information about the noise and would simplify the problem. The additive noise used in this work is multi-talker babble noise, which is a recording of multiple people talking. We chose to add multi-talker babble noise due to its obvious relevance to the hearing aid application and the fact that it provides a difficult challenge. The reason that noisy signals degraded by additive babble noise are difficult to enhance lies in the fact that some characteristics of the multi-talker babble noise are similar to those of clean speech.

1.3 Existing Methodologies

Much work has been done in the last three decades to enhance signals degraded by noise (in the way described in Section 1.2) when only the noisy signal is available. A good review of the important works done in this field can be found in [43]. These works have been divided into three main classes by [46]: spectral subtraction algorithms, statistical model-based algorithms and subspace algorithms.

Spectral subtraction algorithms are by far the easiest algorithms to implement. These algorithms assume the existence of a voice activity detector (VAD), which decides whether speech is present in the noisy speech signal. The algorithms estimate the noise spectrum of the noisy signal segments when the VAD determines that speech is absent. The estimated noise spectrums are subtracted from the noisy signal spectrums. These algorithms were described by [76] in the correlation domain and later by [9] in the Fourier domain.

The statistics-based algorithms try to enhance the speech in a statistical framework. Such algorithms try to estimate the Fourier transform coefficients of the clean speech signal from those of the noisy speech signal. In [23] a unified statistical approach to solving the main speech enhancement problems is presented. In [49],

a maximum-likelihood approach was proposed. This work was followed by [24], which focused on estimating the Fourier components of the clean speech using the minimum mean square error criterion. Follow-ups to this work include: [34] that discusses a psycho-acoustic spectral weighting rule, [12] that uses a mixture model to better model the clean speech spectrum, and [28] that evaluates an equivalent rectangular bandwidth frequency scale noise reduction technique. Wiener algorithms, which are optimal for stationary noise, were described in [44, 45]. In [63] modeling the speech signal using a non-stationary-state hidden Markov model for speech enhancement purposes is described.

Subspace algorithms are based on linear algebra. In particular, they assume that the clean speech signal is contained in a subspace of the noisy speech signal space. These algorithms, therefore, decompose the noisy speech signal assuming that there exists a subspace that is mainly occupied by the clean speech signal and an orthogonal subspace that is mainly occupied by the noise signal. The clean speech signal is estimated by nulling the noise component of the signal. The work described in [21] uses singular value decomposition on the time-domain amplitude values to decompose the space into subspaces. In [25], the decomposition of noisy signal space is performed using an eigenvalue decomposition of the signal covariance matrix. It is shown that spectral subtraction is a special case of the proposed signal decomposition approach.

Algorithms not described in [46] include an adaptive signal filtering approach proposed in [77]. In [31] a Bayesian approach to locating bursts of additive noise in a degraded signal is presented. The method of [70] restores non-linearly distorted auto-regressive signals using a Markov chain Monte Carlo approach. In [39, 50] two iterative techniques in which the noisy speech is enhanced using sinusoidal modeling are described. [35] introduces an iterative speech enhancement procedure in which sequential maximization of different criteria are applied. Finally, some

newer and promising work in the separation of a mixture of sounds (several speakers for example) is introduced in [62].

In [38, 46], Hu et al and Loizou compared eight of the main algorithms mentioned above in terms of different criteria (quality and speech recognition) at two noise levels (5 and 0 dB SNR) and with four difference noise types. The experiment they ran was very similar to that used in this work, see Section 4.2. In terms of increasing speech recognition, the goal of our work, none of the algorithms increased speech recognition in the general case (over both SNR levels and for all noise types); algorithms either decreased speech recognition or maintained that of the noisy speech signal.

Related algorithms in automatic speech recognition (ASR) are described in [5, 17]. Speech and data recognition work using masks are described in [20, 64]. Both of these works perform the recognition once the whole signal is available. That differs from our work, in which we attempt to do "real-time" processing; in other words, we consider only *causal* algorithms that process the data as it arrives (column by column). A method assuming the binaural condition, hearing aids in both ears (as opposed to the monaural condition we are working with), is described in [60].

1.4 Outline of Our Methodology

In the hearing aid application, the goal of speech enhancement is to increase recognition. None of the techniques described in Section 1.3 increase recognition performance of the noisy speech signal for all the different noise types and noise levels [46, 57] in the setting we consider.

We describe an image processing based methodology aimed at increasing recognition of noisy speech signals. For the hearing aid application, it is necessary to es-

timate the unknown clean speech signal in real-time from the noisy speech signal. Our methodology consists of steps occurring in "real-time." We use "real-time" instead of just *real-time*, since we do not promise that our programs were written so as to perform fast enough or that we used fast enough computing power. Furthermore, we allowed a delay of approximately 25 milliseconds (ms) in estimating the clean speech signal based on the noisy speech signal input, which is considered acceptable by hearing aid users [36, 66, 68].

Four steps are performed as the noisy speech signal arrives (i.e. in "real-time"). Step 1 transforms a segment of noisy speech signal into a noisy grayscale column. The chosen transform is the squared norm of the Short-Time Fourier Transform; a collection of such columns is called a spectrogram. In Step 2, a binary column is estimated that represents the presence/absence of the unknown clean speech frequencies from the noisy grayscale column using prior information. In Step 3, the binary column is "fuzzified" using the noisy grayscale column. This "fuzzy" column is a $[0,1]$-valued column and is used to mask the noisy grayscale column. The noisy spectrogram column and the "fuzzy" column are multiplied pixel by pixel to estimate a clean grayscale column corresponding to the unknown clean speech segment. Step 4 estimates a segment of clean speech from several already estimated clean grayscale columns. Figure 1.1 displays a schematic of our methodology.

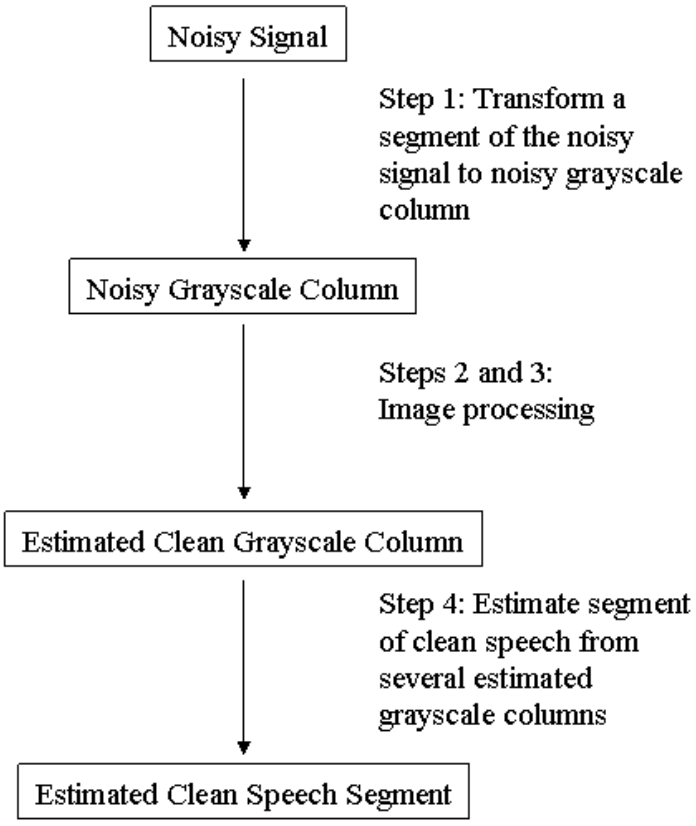


Figure 1.1: Schematic of our methodology.

Chapter 2

Preliminaries

This chapter describes the clean speech and noise signals provided to us, and how we created noisy speech signals and training sets from them. Section 2.1 describes the pre-processing that we performed on the clean speech signals. Section 2.2 describes a noise signal and how we produced noisy speech signals. Section 2.3 describes how we created the training sets required for our methodology. Section 2.4 contains a discussion of why our methodology has the potential to be efficacious.

2.1 Clean Signal Pre-Processing

Clean speech signals, recordings of the Modified Rhyme Test (MRT), which are described in detail in Section 4.2, were provided by the Speech Research Laboratory at the Department of Psychological and Brain Sciences of Indiana University. Ten data-sets were provided, each corresponding to one male talker. There were problems with some data-sets and so only five data-sets were used. One of the five data-sets was randomly chosen to be used for testing purposes. The remaining four data-sets were used for training our methodology. Each data-set contains six MRT lists (300 words total) recorded in a noiseless environment. These signals were saved as digital wave files with a sampling rate of SSR $= 10,000$ Hertz (Hz), where

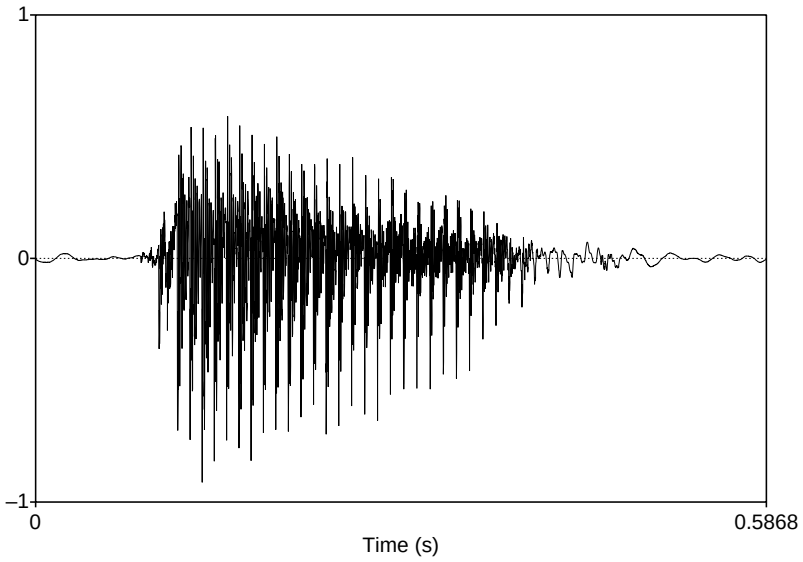


Figure 2.1: The waveform of the original signal of the word “bad” in a noiseless environment spoken by the talker from the testing data-set. The scaled signal values (signal value divided by 32768) are on the vertical axis.

Hz is samples per second, and 16 bit quantization. Each of these speech signals is on the order of 500 ms long. From here onwards, by **data-set** we mean a collection of such recordings of the 300 MRT words spoken by one talker.

We define the **waveform** of a signal to be a one-dimensional graph with time on the horizontal axis and signal value on the vertical axis. Waveforms are used to display signals as a function of time. The waveforms displayed in this document were created by [8] with time in seconds (s) on the horizontal axis and the scaled signal value (signal value divided by 32768) on the vertical axis. For consistency throughout this write up, the vertical axis will have the same minimum (-1.0) and maximum (1.0) values. Figure 2.1 displays the waveform of a clean speech signal spoken by the talker from the testing data-set. The signal is of the word “bad,” which will be used in all the examples in this text unless otherwise specified.

We pre-processed the clean speech signals. First, we normalized each signal to have a zero mean (by subtracting the mean value from each signal value). Next,

we define the average root-mean-square (rms) value of signals. We searched over non-overlapping windows of length 25 ms. The last window contains the remaining time points, and is at most 25 ms long. For each such window, we calculated the value of the square root of the average squared signal values. The highest such value over all the windows was recorded and is referred to as the **maximum rms value**. This value is converted into dB by taking

$$20 \times \log_{10}(\text{maximum rms value}) = 10 \times \log_{10}\left((\text{maximum rms value})^2\right) \quad (2.1)$$

and is referred to as the **maximum dB rms value**.

A second pass through the non-overlapping windows was performed. Let **dynamic range** be the audible sound range (in our work we assumed that the value is 50 dB). Therefore, the **minimum dB rms value** is defined as

$$\text{minimum dB rms=maximum dB rms-dynamic range.} \quad (2.2)$$

Any window whose dB rms value was below the minimum dB rms value was marked as a silent window. We calculated the square root of the average square signal values in the windows not marked as silent. This gives us the **average rms value** for one signal. We converted this value to dB as in Equation 2.1 and refer to it as the **average dB rms value** of the signal**.**

Next, we scaled each of the signals so that each signal's average dB rms value was 66 dB rms. This value was the highest for which no clipping occurred during subsequent processing. Clipping is a process in which signal amplitudes whose absolute value are too high are set to lower values (in the absolute value), which creates distortions in the signal. We therefore prevented clipping, by scaling all the signals to the same rms level. The way to do this scaling is to multiply each signal

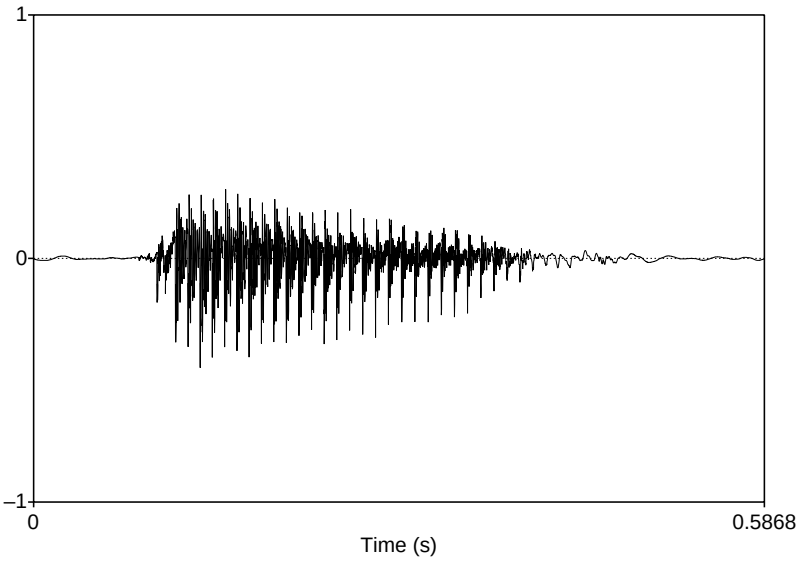


Figure 2.2: The waveform of the signal after normalization and scaling.

value by

$$10^{(66-\text{average dB rms value})/20}. \tag{2.3}$$

Figure 2.2 displays the waveform of the signal after the normalization and scaling.

Finally, a new signal is created by adding 500 ms of silence to the beginning and end of each speech signal; these new signals start at time 0. These padded, processed, noiseless speech signals are referred to as the **clean speech signals**, see Figure 2.3.

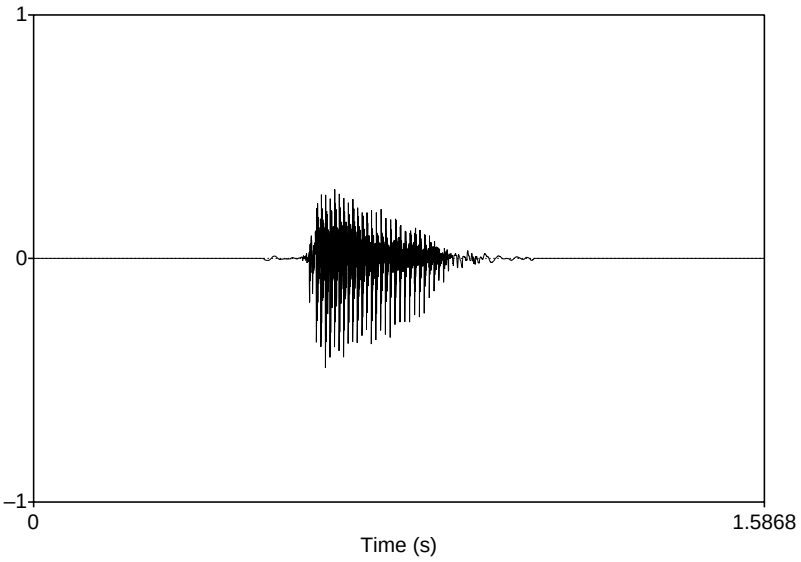


Figure 2.3: The waveform of the clean speech signal.

2.2 Noisy Speech Signals

As described in Section 1.2, noisy speech signals were created by adding multi-talker babble noise to the clean speech signals. One multi-talker babble noise signal was taken from a recording of the Speech Perception in Noise-Revised test [7], see Figure 2.4. The digital recording, about 5 minutes long, was saved with a sampling rate of NSR $= 22,500$ Hz. It was down-sampled to match the SSR $= 10,000$ Hz sampling rate of the speech before creating the noisy speech signals.

Let $n[0],\ldots,n[K-1]$ be the original babble noise signal. We define its sub-sampled version, *sn* by

$$sn[k] = \sum_{l=-3}^{3} \mathrm{N}(l;0,1) \times n\left[3 + \left\langle k \times \frac{\mathrm{NSR}}{\mathrm{SSR}} \right\rangle + l\right], \tag{2.4}$$

for $k = 0,1,\ldots,\left\lfloor (K-7) \times \dfrac{\mathrm{SSR}}{\mathrm{NSR}} \right\rfloor$, see Figure 2.5. Here, $\mathrm{N}(l;0,1)$ is the value of the normal distribution with mean 0 and variance 1 at l, $\langle x \rangle$ is the nearest integer

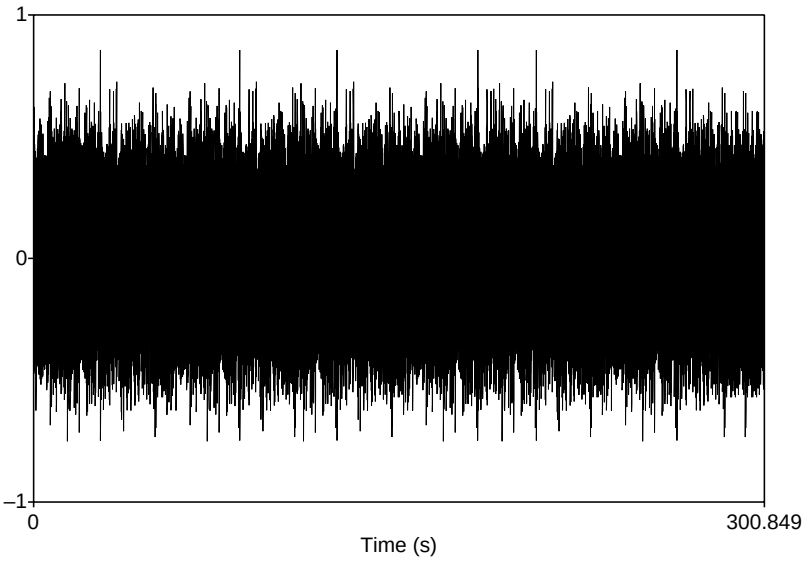


Figure 2.4: The waveform of the whole original multi-talker babble noise signal.

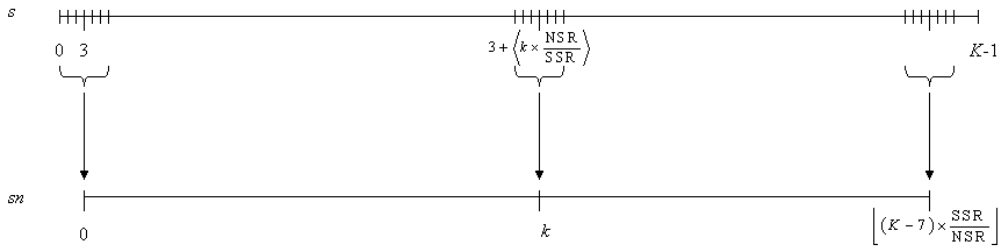
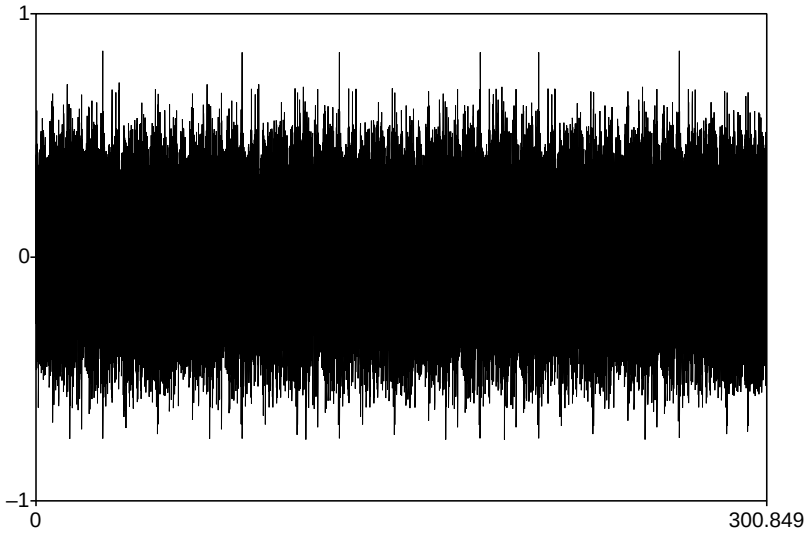


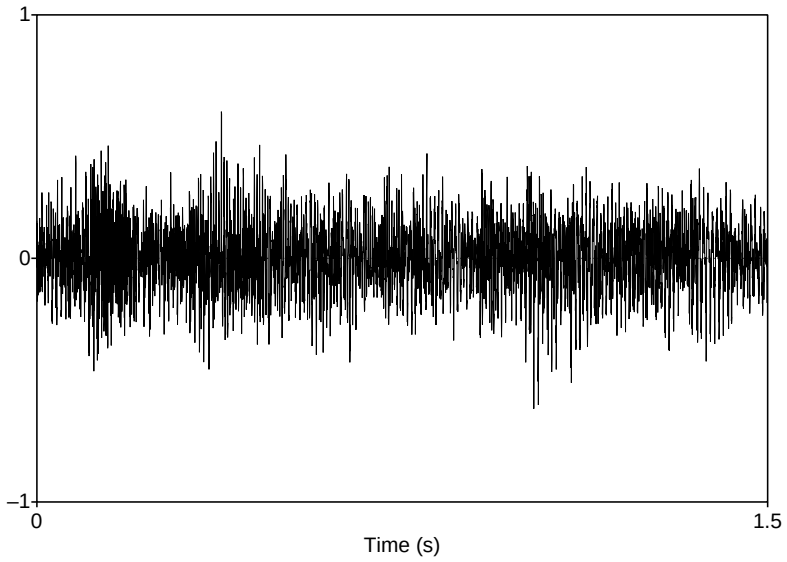
Figure 2.5: Sub-sampling the noise signal.

to the real value x, and $\lfloor x \rceil$ is the nearest integer to the real value x that is not greater than x. We normalized the sub-sampled noise signal to have a zero mean (by subtracting the mean noise value from each noise value). We call this zero-mean signal the **sub-sampled babble noise**, see Figure 2.6a. For comparison with the clean speech signal in Figure 2.3, Figure 2.6b displays the first 1500 ms of the sub-sampled babble noise.

In order to create the noisy speech signals for each SNR level, we had to determine the correct scaling of the sub-sampled babble noise. We calculated the average dB rms value of the sub-sampled babble noise as described in Section 2.1. For each



a)



b)

Figure 2.6: a) The waveform of the sub-sampled noise signal. b) The waveform of the first 1500 ms of the sub-sampled noise signal.

SNR level (5 and 0 dB), we scaled the sub-sampled babble noise so that its dB rms value will be

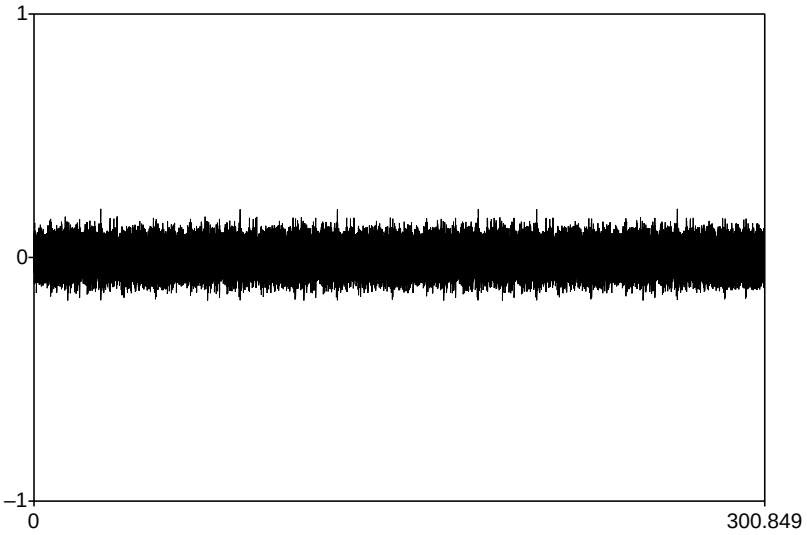
$$66 - \text{dB SNR value}. \tag{2.5}$$

The way to do this is to multiply each noise value by

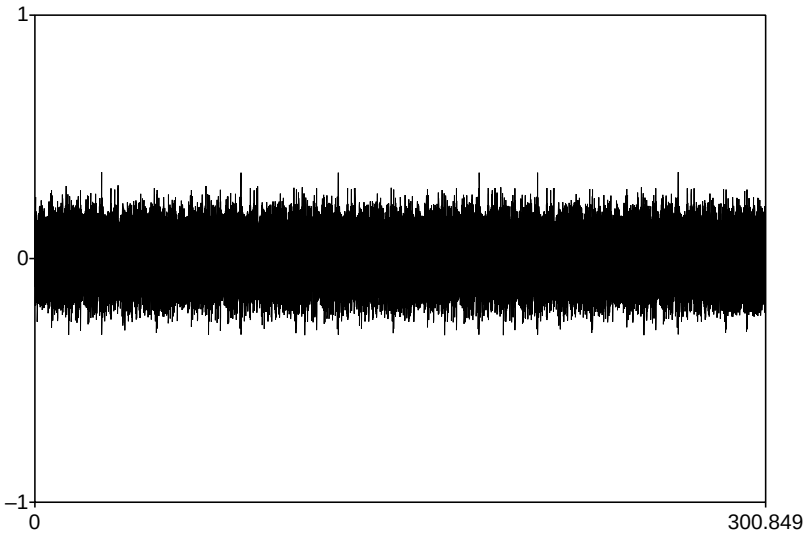
$$10^{((66-\text{dB SNR value})-\text{average noise dB rms value})/20}. \tag{2.6}$$

We, therefore, have two babble noise signals, each corresponding to one of the SNR levels. These are referred to as the **noise signals** for 5 and 0 dB SNR.

Throughout the rest of this write-up, we display figures for both of the SNR levels that we are testing. Furthermore, for consistency, the 5 dB SNR images are always displayed above the 0 dB SNR images. Figure 2.7 displays the waveforms of the noise signals for 5 and 0 dB SNR. For comparison to the clean speech signal in Figure 2.3, Figure 2.8 displays the waveforms of the first 1500 ms of the noise signals for 5 and 0 dB SNR.

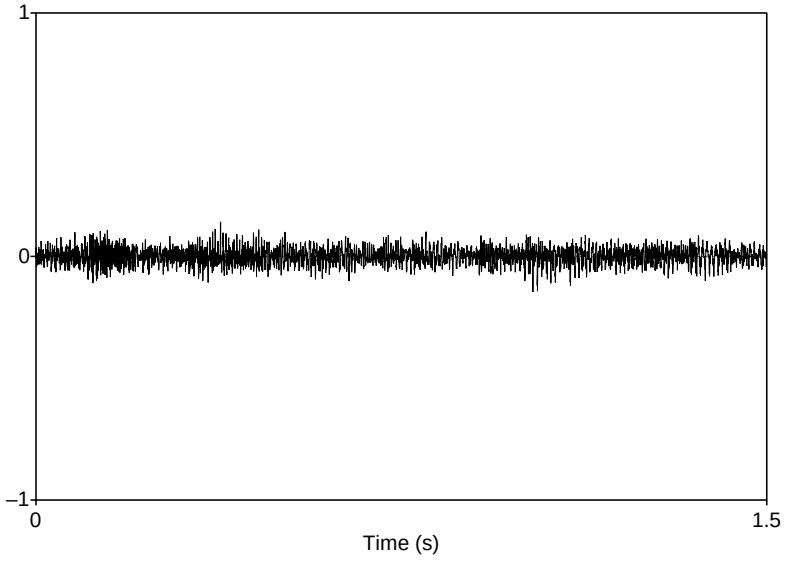


a)

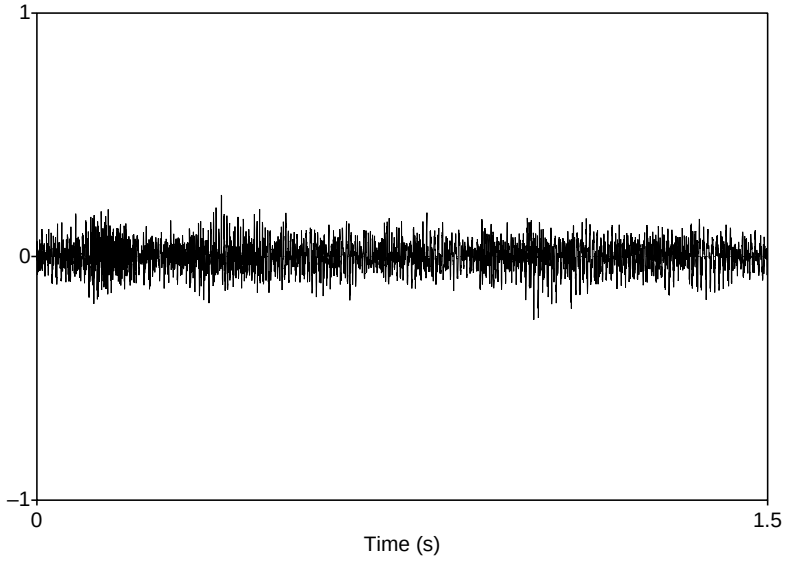


b)

Figure 2.7: The waveforms of noise signals with a) 5 dB SNR and b) 0 dB SNR.



a)



b)

Figure 2.8: The waveforms of the first 1500 ms of the noise signal for a) 5 dB SNR and b) 0 dB SNR.

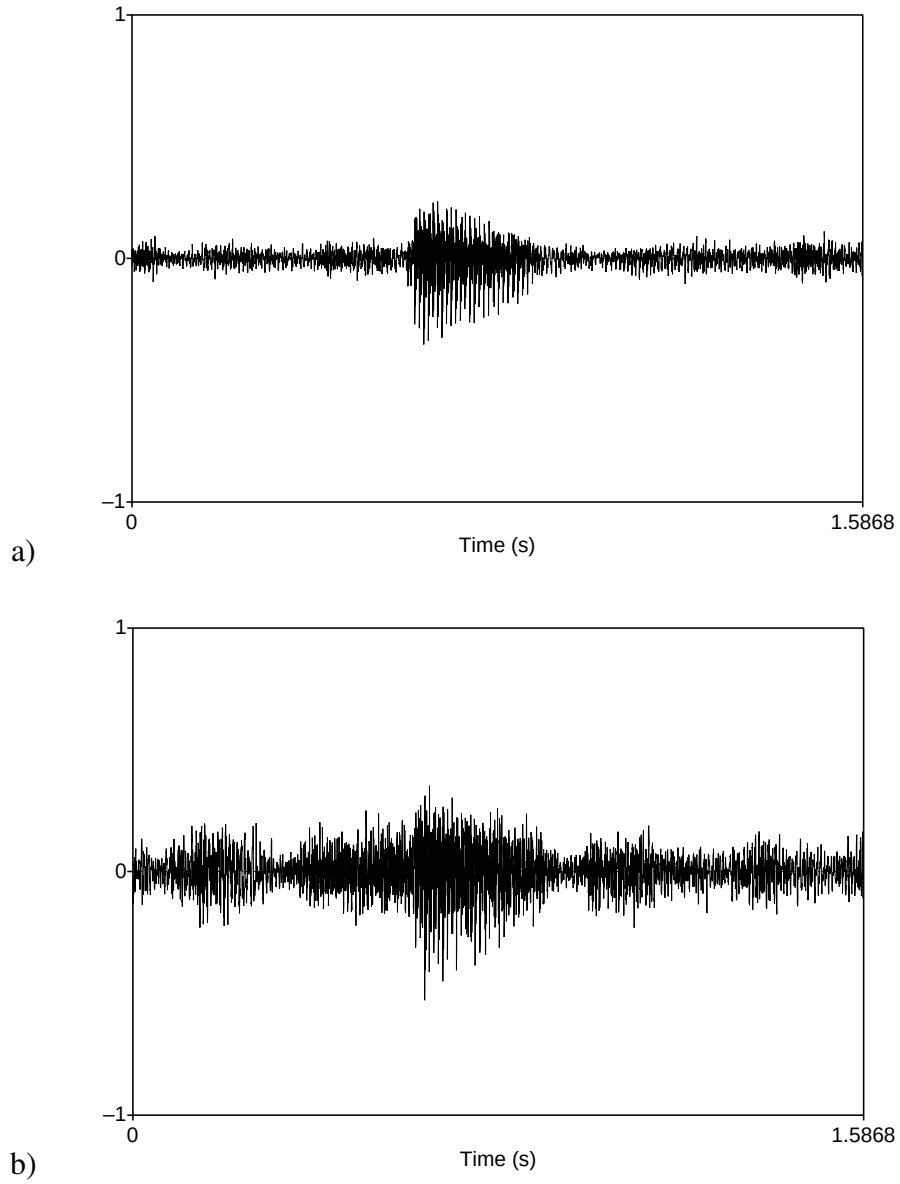


Figure 2.9: The waveforms of the noisy speech signals with a) 5 dB SNR, and b) 0 dB SNR.

For a given clean speech signal and SNR level, we created a **noisy speech signal** in the following way. We chose a random location in the noise signal for that SNR level and added the following consecutive noise signal values at that location to the clean speech signal values. Figure 2.9 shows the waveforms of noisy speech signals with 5 and 0 dB SNR. Compare these to the waveform of the clean speech signal in

Figure 2.3. For each SNR level and for each of the testing and training clean speech signals, we created noisy speech signals in this way.

2.3 Training Set

For our methodology, training was required. In this section we describe the images that make up the training set. Since we decided to test our methodology at two SNR levels (5 and 0 dB SNR), we created two training sets and trained separately for each SNR level. The training from these sets is described in Sections 3.2 and 3.3.

In Section 3.1, we detail how signals are transformed into images called spectrograms. For display purposes, spectrogram images are also be defined. The spectrogram and spectrogram image of each clean speech signal were calculated. These are referred to as the **clean spectrograms** and the **clean spectrogram images**, respectively. The spectrogram and spectrogram image of each noisy speech signal were calculated. These are referred to as the **noisy spectrograms** and the **noisy spectrogram images**, respectively.

For each SNR level, the training set consists of pairs of images representing the noisy speech signal and corresponding binary images describing the underlying clean speech signal. The "correspondence" is in terms of the images arising from the same clean speech signals. Now we describe how we acquired the necessary binary training images. We created binary training images by thresholding the clean spectrogram images at value 254. Specifically, for each pixel location, the binary image value is set to 0 if and only if the clean spectrogram image value at that pixel is 255, see Figure 2.10.

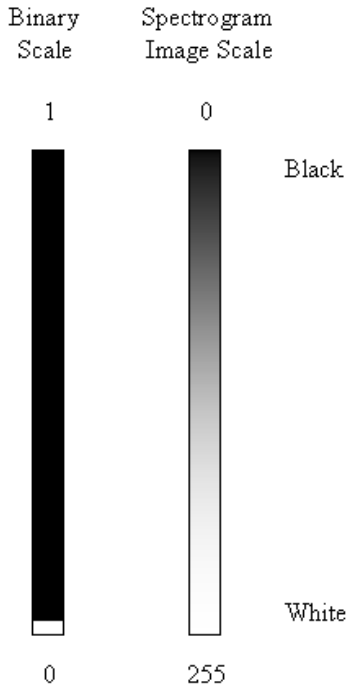
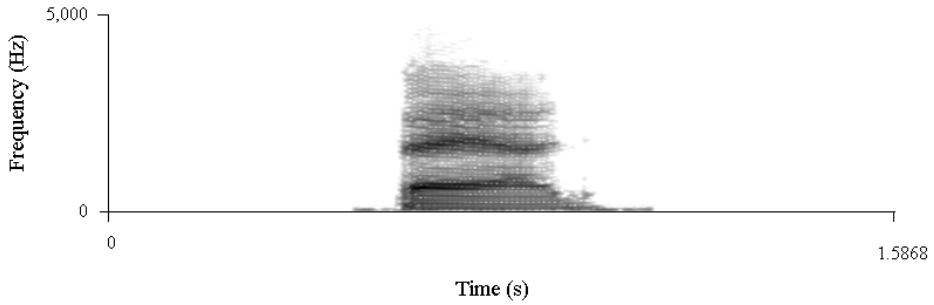
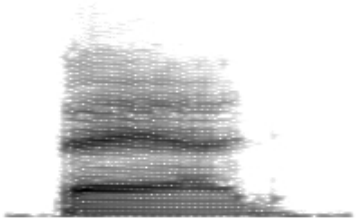


Figure 2.10: Scales and display colors of binary and spectrogram images.

These binary images are referred to as the **clean binary images**, since they arose from the clean spectrogram images. Figure 2.11a displays a clean spectrogram image with drawn axes. Henceforth, the spectrogram images (and images derived from it) will not have drawn axes, as in Figure 2.11b; the axes in Figure 2.11a should be super-imposed on all the images. Figure 2.12 displays the corresponding clean binary image. The binary training images are not dependent on the SNR level and are therefore the same for the different SNR levels. Only the noisy grayscale training image differ for different SNR levels. Figure 2.13 shows a schematic of how the training set was created from the clean speech signals.



a)



b)

Figure 2.11: a) A clean spectrogram image with drawn axes. b) A clean spectrogram image as displayed in the rest of the text.

Figure 2.12: The clean binary image.

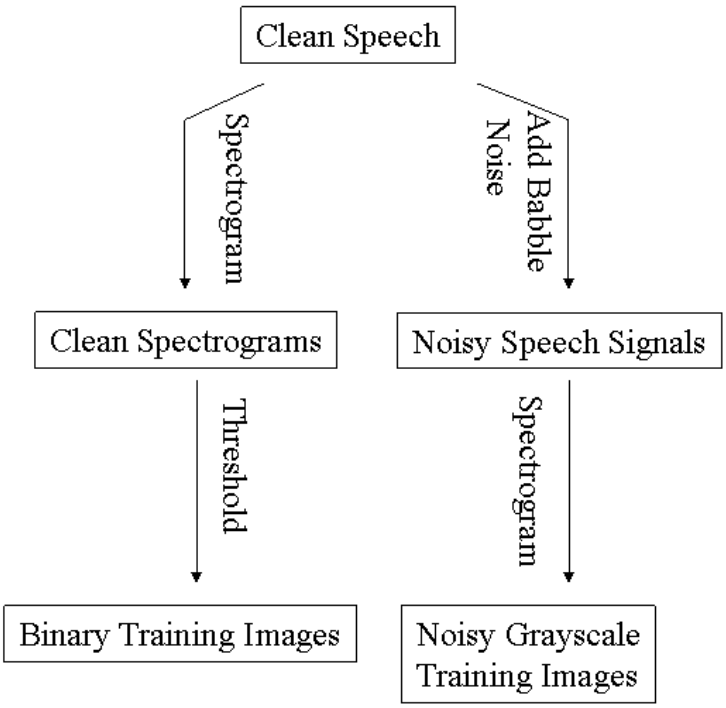


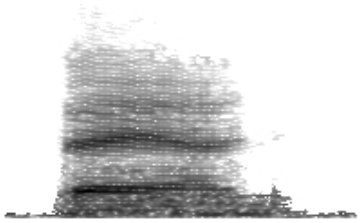
Figure 2.13: A schematic of how the training set was created from the clean speech signals.

2.4 Discussion

Our signal enhancement methodology involves transforming a signal into the image, performing image processing in order to estimate an image that corresponds to the unknown clean signal, and then transforming the image to get an estimate of the unknown clean signal. The specific type of image processing we are proposing is that of using a $[0,1]$-valued image as a multiplicative mask on the noisy spectrograms.

In Section 1.1, it was mentioned that we believed that the clean binary image of a clean spectrogram is of importance. If the clean binary image is used as a mask on a clean spectrogram, the resulting image is the clean spectrogram (by definition), see Figures 2.11 and 2.12. Furthermore, if we estimated the spectrogram of the unknown clean speech by multiplying the clean binary image by the noisy spectrogram, it would be close to the unknown clean spectrogram, see Figure 2.14.

a)



b)

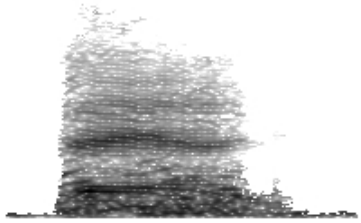


Figure 2.14: Estimated clean spectrogram using the clean binary image as a mask on the noisy spectrogram with a) 5 dB SNR, and b) 0 dB SNR.

Compare these estimated clean spectrograms to the clean spectrogram in Figure 2.11; the images appear quite similar. The only differences lie in the spectrogram values within the binary mask. Notice that this estimation requires access to the clean binary image, which is not available.

In preliminary work, we estimated the clean speech signals from such estimates of the unknown clean speech spectrograms. These estimated clean speech signals sounded like good estimates of the clean speech signals. This led us to conclude that indeed there is merit to using binary masks in this way to estimate the underlying clean image (and clean signal).

The difficulty, therefore, lies in the estimation of such a binary image when the clean speech signal and spectrogram are not available. We attempted to estimate such a binary image from the noisy spectrogram using prior information obtained about noisy spectrograms and corresponding clean binary images. We further search for a "fuzzy" version of the binary image, a $[0, 1]$-valued image, that is used to mask the noisy speech spectrogram. The reason for this "fuzzification" is that it smooths out the mask boundaries and dampens the signal within the binary mask. These "fuzzy" masks are multiplied by the noisy speech spectrograms in or-

der to estimate the clean speech spectrograms. In particular, that means that in this work we restricted our image enhancement to be performed using $[0, 1]$-valued images. The main part of this work will be detailing the estimation of an appropriate $[0, 1]$-valued image that will bring about the desired image enhancement.

Chapter 3

Methodology

This chapter describes our methodology for enhancing noisy speech signals degraded by uncorrelated, non-stationary, additive, multi-talker babble noise from only one channel. The chapter is divided in the following way. Section 3.1 contains definitions required to describe the transformation of a segment of noisy speech signal into a noisy grayscale column, Step 1 of our methodology. Sections 3.2 describes the noise model and its estimation from a training set. Section 3.3 describes the prior model and its estimation from a training set. Section 3.4 describes how the binary column corresponding to the unknown clean speech signal is estimated from the noisy grayscale column obtained in Step 1, Step 2 of our methodology. Section 3.5 describes how the binary column is "fuzzified" to create a $[0, 1]$-valued column of probabilities. The "fuzzy" column and the noisy grayscale column from Step 1 are used to estimate a grayscale column corresponding to the clean speech segment, Step 3 of our methodology. Section 3.6 describes how to estimate a segment of clean speech from several already estimated clean grayscale columns produced in Step 3, Step 4 of our methodology. Finally, Section 3.7 details the time each step takes.

3.1 Step 1: Signal to Column Transform

The first step in our methodology involves transforming the segments of the noisy speech signal into noisy grayscale columns, using the spectrogram. Spectrograms are two-dimensional, real-valued, non-negative images. Spectrogram images are transformations of spectrograms which produce two-dimensional, integer-valued, non-negative images, where the value 0 corresponds the maximum spectrogram amplitude and the value 255 corresponds to the lowest spectrogram amplitudes. Spectrogram images are graphical displays of time-frequency information of a signal with time on the horizontal axis and frequency on the vertical axis. The reason for this choice in transforms lies in the fact that the spectrogram seems to be the mostly widely used imaging technique of signals. The advantage is that much is known about the characteristics of spectrograms, in particular speech spectrograms [42, 46, 57]. This section defines the transforms necessary for this work in order of dependence.

Subsection 3.1.1 defines that Fourier Transform and its inverse. Subsection 3.1.2 defines the Discrete Fourier Transform and its inverse. Subsection 3.1.3 defines the Discrete Fourier Transform of windowed signals and its inverse. Subsection 3.1.4 defines the Short-Time Fourier Transform. Subsection 3.1.5 defines the spectrogram and Subsection 3.1.6 defines the spectrogram image.

3.1.1 Fourier Transform

First, we define an operator that transforms a continuous, complex-valued time signal into a continuous, complex-valued frequency signal. This operator, the **Fourier Transform** (FT) [11], is defined by

$$[\mathscr{F} f](r) = \int_{-\infty}^{\infty} f(t) \exp\left(-2\pi\sqrt{-1}rt\right) dt, \tag{3.1}$$

for $-\infty < r < \infty$, where $\exp(y) = e^y$. We also define the **Inverse Fourier Transform** (IFT) [11], which transforms a continuous, complex-valued frequency signal into a continuous, complex-valued time signal. The Inverse Fourier Transform of a frequency signal F is defined by

$$\left[\mathscr{F}^{-1}F\right](t) = \int_{-\infty}^{\infty} F(r) \exp\left(2\pi\sqrt{-1}tr\right) dr, \tag{3.2}$$

for $-\infty < t < \infty$. The FT and IFT are inverses of one another. In other words, under some mild conditions [54] it holds that for every time signal f, frequency signal F, $-\infty < t < \infty$, and $-\infty < r < \infty$,

$$\left[\mathscr{F}^{-1}\left[\mathscr{F}f\right]\right](t) = f(t) \tag{3.3}$$

and

$$\left[\mathscr{F}\left[\mathscr{F}^{-1}F\right]\right](r) = F(r). \tag{3.4}$$

It is easy to show that if f is a real-valued function, then its FT is *half-redundant* in the sense that, for all frequencies r, $[\mathscr{F}f](-r) = [\mathscr{F}f](r)$.

In applications, there is a necessity for a similar invertible operator that transforms finite-length, discrete time sequences into finite-length, discrete frequency sequences. The time sequences are sampled versions of continuous time signals. It is assumed that the time sequences are of finite-length. The frequency sequences are sampled version of the continuous frequency signal. In this work, we will only consider uniformly sampled time sequences and frequency sequences. The discretized version of the FT is defined next.

3.1.2 Discrete Fourier Transform

Now, we define an operator called the **Discrete Fourier Transform** (DFT) [46, 54, 57] that transforms a finite-length, complex-valued, discrete time sequence

into a finite-length, complex-valued, discrete frequency sequence. To be precise, each operator $\mathscr{D}_I$ in the family is determined by one positive integer parameter (I - determines the length of the input and output sequences). The domain of the operator $\mathscr{D}_I$ is the set of discrete sequences x of $2I-2$ complex numbers, $x[0], x[1], \ldots, x[2I-3]$. The range of $\mathscr{D}_I$ is the set of discrete sequences of $2I-2$ complex numbers. The Discrete Fourier Transform of a sequence x into a sequence $\mathscr{D}_I x$ is defined by

$$[\mathscr{D}_I x][i] = \frac{1}{\sqrt{2I-2}} \sum_{m=0}^{2I-3} x[m] \exp\left(-\frac{2\pi\sqrt{-1}im}{2I-2}\right), \tag{3.5}$$

for $0 \leq i < 2I-2$.

Note the property that the DFT of a real-valued sequence is a complex-valued sequence that is *half-redundant* in the sense that, for $I \leq i < 2I-2$, $[\mathscr{D}_I x][i] = [\mathscr{D}_I x][2I-2-i]$. For this reason, when dealing exclusively with the DFT of real-valued sequences, it is only necessary to save the first I values of $[\mathscr{D}_I x][i]$ (for $0 \leq i < I$).

We also define the **Inverse Discrete Fourier Transform** (IDFT) [46, 54, 57], which is an operator that transforms a finite-length, complex-valued, discrete frequency sequence into a finite-length, complex-valued, discrete time sequence. To be precise, each operator $\mathscr{D}_I^{-1}$ in the family is determined by one positive integer parameter (I - determines the length of the input and output sequences). The domain of the operator $\mathscr{D}_I^{-1}$ is the set of sequences X of $2I-2$ complex numbers, $X[0], X[1], \ldots, X[2I-3]$. The range of $\mathscr{D}_I^{-1}$ is the set of sequences of $2I-2$ complex numbers. The Inverse Discrete Fourier Transform of a sequence X, namely $\mathscr{D}_I^{-1} X$, is defined by

$$\left[\mathscr{D}_I^{-1} X\right][m] = \frac{1}{\sqrt{2I-2}} \sum_{i=0}^{2I-3} X[i] \exp\left(\frac{2\pi\sqrt{-1}mi}{2I-2}\right), \tag{3.6}$$

for $0 \leq m < 2I-2$. The DFT and IDFT are inverses of one another [11, 57]; it holds that for every time sequence x, frequency sequence X, $0 \leq m < 2I-2$, and $0 \leq i < 2I-2$,

$$\left[\mathscr{D}_I^{-1}\left[\mathscr{D}_I x\right]\right][m] = x[m] \tag{3.7}$$

and

$$\left[\mathscr{D}_I\left[\mathscr{D}_I^{-1} X\right]\right][i] = X[i]. \tag{3.8}$$

The justification of the claim that the DFT is the discrete version of the FT can be presented in a number of ways. For example, let x be a uniformly sampled, finite sequence of a time signal f with time sampling distance τ; symbolically,

$$x[m] = f(m \times \tau), \tag{3.9}$$

for $0 \leq m < 2I-2$. Furthermore, let $F(r) = [\mathscr{F} f](r)$ and $X[i] = [\mathscr{D}_I x][i]$. Then, under the assumption that $f(t) = 0$ for $t \leq -0.5\tau$ and $t \geq (2I-2.5)\tau$, the Riemann sum approximation of the Fourier integral (3.1) can be used to derive that

$$F\left(\frac{i}{(2I-2)\tau}\right) \approx \sqrt{2I-2}\tau X[i], \tag{3.10}$$

for $0 \leq i < I$. The values of F for negative arguments can then be derived using the fact that F is half-redundant; see Subsection 3.1.1. The **Nyquist frequency** [11, 46, 54] is the largest absolute frequency at which F can be estimated using Equation 3.10 and is equal to $1/2\tau$. The frequencies are estimated using the step size $1/(2I-2)\tau$.

3.1.3 Discrete Fourier Transform of Windowed Sequences

Next, we define a family of operators that transforms a finite-length, real-valued, discrete time sequence into a finite-length, complex-valued discrete frequency sequence. These operators are the **Discrete Fourier Transform of windowed sequences** [46, 57]. To be precise, each operator $\mathscr{D}^w_{L,I,n}$ in the family is determined by three positive integer parameters (L - the length of the time sequence, I - the length of the output , and n - a time point) and a window function w. The restrictions on the parameters are that $2I-3 \leq n < L$. The window function w is a sequence of $2I-2$ real numbers, $w[0], w[1], \ldots, w[2I-3]$. The domain of $\mathscr{D}^w_{L,I,n}$ is the set of sequences x of L real numbers, $x[0], x[1], \ldots, x[L-1]$. The range of $\mathscr{D}^w_{L,I,n}$ is the set of sequences of I complex numbers. Let $y_n[m] = x[n-m]\, w[m]$ for $0 \leq m \leq 2I-3$. Then the Discrete Fourier Transform of a windowed sequence x, namely $\mathscr{D}^w_{L,I,n}x$, is defined by

$$\left[\mathscr{D}^w_{L,I,n}x\right][i] = \left[\mathscr{D}_I y_n\right][i] \tag{3.11}$$

for $0 \leq i < I$. This can be understood as the DFT of sequence x weighted by window w. Notice that the DFT of windowed sequences only contains the first I non-redundant frequencies, see the comment in Subsection 3.1.2.

The value of L is the length of the finite-length time sequence x. Let TSR be the uniform sampling rate of the time sequence; i.e., TSR is $1/\tau$, where τ is the sampling distance. The Nyquist frequency is equal to $\mathrm{TSR}/2 = 1/(2\tau)$. The value of I, the length of the output signal, defines the frequency step size; it is $\mathrm{TSR}/(2I-2)$. The value of n is the location of the window.

Using the IDFT, we can define a family of operators that transforms a finite-length, complex-valued, discrete frequency sequence into a finite-length, real-valued, discrete time sequence. These operators are the **Inverse Discrete Fourier Trans-**

form of windowed sequences [46, 57]. To be precise, each operator $\mathscr{E}_I$ in the family is determined by one positive integer (I - the length of the input). The domain $\mathscr{E}_I$ of is the set of sequences of I complex numbers (which is different from the domain of $\mathscr{D}_I^{-1}$). The range of $\mathscr{E}_I$ is the set of sequences of $2I-2$ real numbers. For a sequence $X[i]$ defined for $0 \leq i < I$, let $Y[i] = X[i]$ for $0 \leq i < I$ and $Y[i] = X[2I-2-i]$ for $I \leq i < 2I-2$. Then $\mathscr{E}_I X$ is defined by

$$[\mathscr{E}_I X][m] = \left[\mathscr{D}_I^{-1} Y\right][m], \tag{3.12}$$

for $0 \leq m < 2I-2$. The important property of $\mathscr{E}_I$ is that if x, L, I, n, and w satisfy the conditions stated in the definition of the Discrete Fourier Transform of windowed sequences, then it follows from Equation 3.7 that

$$\left[\mathscr{E}_I\left[\mathscr{D}_{L,I,n}^{w} x\right]\right][m] \quad = \quad x[n-m]\, w[m], \tag{3.13}$$

for $0 \leq m \leq 2I-3$.

3.1.4 Short-Time Fourier Transform

Next, we define a family of operators that transforms a finite-length, real-valued, discrete time sequence into an image, called the **Short-Time Fourier Transform** (STFT) [46, 57]. Henceforth, by sequence we mean a finite-length, real-valued discrete time sequence. To be precise, each operator $\mathscr{T}_{L,I,s}^{w}$ in the family is determined by three positive integer parameters (L - the length of the sequence, I - the number of rows in the STFT, and s - the time skip step) and a window function w. The restrictions on the parameters are that $s \leq 2I-2 \leq L$. The window function w is a sequence of $2I-2$ real numbers, $w[0], w[1], \ldots, w[2I-3]$. The domain of $\mathscr{T}_{L,I,s}^{w}$ is the set of sequences x of L real numbers, $x[0], x[1], \ldots, x[L-1]$. The range of $\mathscr{T}_{L,I,s}^{w}$ is the set of $I \times J$-dimensional complex-valued images, where $J = \left\lfloor \dfrac{L-(2I-2)}{s} \right\rfloor + 1$, see Figure 3.1. We define the STFT of a sequence x to be

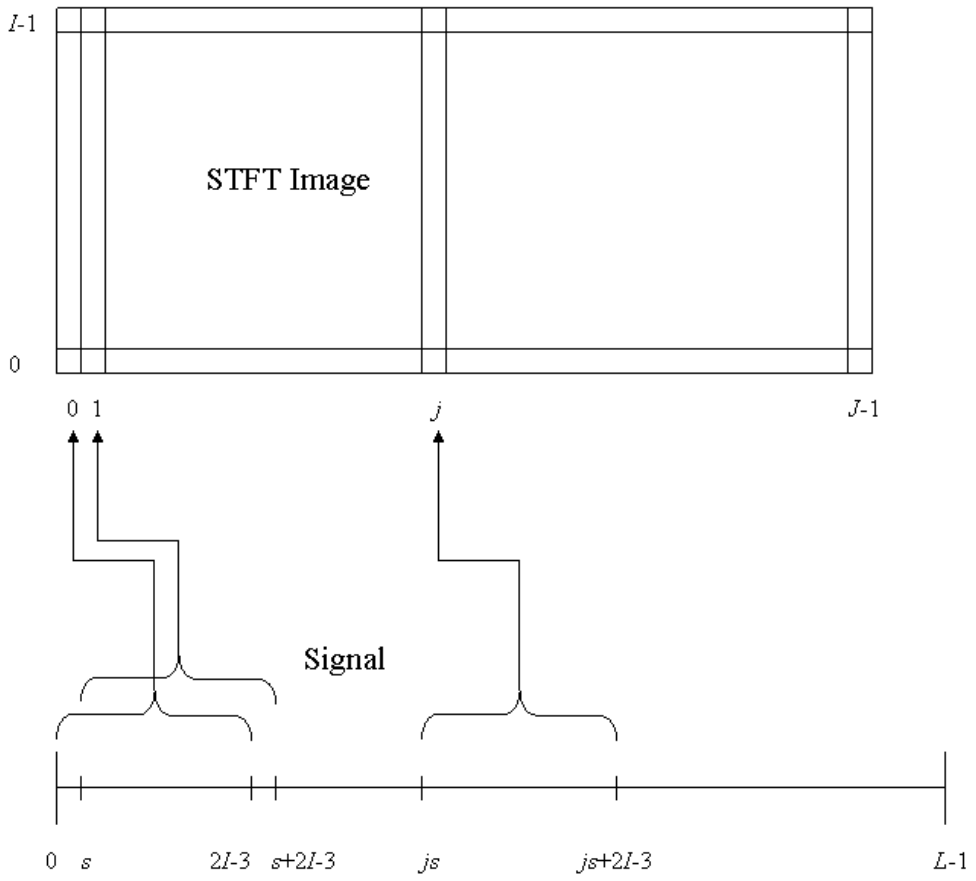


Figure 3.1: A display of how a STFT image is created from a time sequence.

$$\left[\mathscr{T}^{w}_{L,I,s}x\right][i,j] = \begin{cases} 0, & \text{if } i = 0, \\ \left[\mathscr{D}^{w}_{L,I,2I-3+js}x\right][i], & \text{otherwise,} \end{cases} \tag{3.14}$$

for $0 \leq i < I$ and $0 \leq j < J$. Figure 3.1 displays how the STFT image is created from a time sequence.

3.1.5 Spectrogram

Finally, we define another family of operators that transforms a time sequence into an image, called a **spectrogram** [46, 57]. To be precise, each operator $\mathscr{S}^{w}_{L,I,s}$ in the family is determined by three positive integer parameters (L - the length

of the sequence, I - the number of rows in the spectrogram, and s - the time skip step) and a window function w. The restrictions on the parameters are that $s \leq 2I-2 \leq L$. The window function w is a sequence of $2I-2$ real numbers, $w[0], w[1], \ldots, w[2I-3]$. The domain of $\mathscr{S}^w_{L,I,s}$ is the set of sequences x of L real numbers, $x[0], x[1], \ldots, x[L-1]$. The range of $\mathscr{S}^w_{L,I,s}$ is the set of $I \times J$-dimensional non-negative real-valued images, where $J = \left\lfloor \frac{L-(2I-2)}{s} \right\rfloor + 1$, just as in Figure 3.1. We define the spectrogram of a sequence x, namely $\mathscr{S}^w_{L,I,s}x$, to be

$$\left[\mathscr{S}^w_{L,I,s}x\right][i,j] = \begin{cases} 0, & \text{if } i = 0, \\ \left|\left[\mathscr{D}^w_{L,I,2I-3+js}x\right][i]\right|^2, & \text{otherwise,} \end{cases} \tag{3.15}$$

for $0 \leq i < I$ and $0 \leq j < J$, where $|y|$ is the norm of the complex value y.

We have defined the spectrogram to be an image containing I rows and J columns, where I is a parameter to the operator and the value of J is set based on the values of the parameters L, I, and s. The index j (and parameter I) is responsible for determining the value of n for the operator $\mathscr{D}^w_{L,I,n}$. The bottommost row of all spectrograms (as defined here), i.e. the row for which $i = 0$, is set to zeros. For $0 < i < I$ and $0 \leq j < J$, the (i,j)-pixel of image $\mathscr{S}^w_{L,I,s}x$ is set to the squared norm of the i^{th} frequency of the DFT of the windowed sequence x with parameters L, I, and $n = 2I-3+js$ and window w, symbolically $\left|\left[\mathscr{D}^w_{L,I,2I-3+js}x\right][i]\right|^2$. In other words, the spectrogram is the squared norm of the STFT. Notice that by taking the squared norm of the complex number $\left[\mathscr{D}^w_{L,I,2I-3+js}x\right][i]$, we have lost the phase/angle information. For this reason, the spectrogram is not an invertible operator.

The following are the parameter values we chose in the creation of spectrograms in this work and text. The parameter L is the number of samples in sequence x, which is variable. The processed speech sequences that we used were in the order of 1,500 ms and were sampled with sampling rate TSR = SSR = 10,000 Hz. Therefore L is on the order of 15,000 samples. We chose the number of rows

in the spectrograms to be $I = 125$. The reason for this particular choice in the number of rows has to do with frequency banding used in our processing method. This corresponds to a window size of $2I - 2 = 248$ samples, which corresponds to 24.8 ms. A window size of 25ms is commonly used [58] and is further reason for the choice of the value of I. Finally, this choice in window size led to spectrograms which visually looked as we wished; other values of I produced spectrogram images with visual characteristics different from our preferences. The highest frequency estimated was $\mathrm{TSR}/2 = 5{,}000$ Hz. Since $I = 125$, the frequency step size is $10{,}000/(250-2) \approx 40.32$ Hz. The time skip size was chosen to be $s = 31$ samples, which corresponds to 3.1 ms. Our training images with these parameters have approximately 400-500 columns.

To simplify this discussion, we have chosen I and s such that $\dfrac{2I-2}{s}$ is a positive integer that we denote by d. In our case $d = 8$. The consequence of this is the following. For a sequence value $x[\hat{n}]$ at $\hat{n} = 2I - 3 + \hat{j}s - a$ where $0 \leq \hat{j} \leq J - d$ and $0 \leq a < s$, there are exactly d columns of the spectrogram that are affected by the value of $x[\hat{n}]$ (and consequently contain information about that point). These columns are $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet, j]$ for $\hat{j} \leq j < \hat{j} + d$, see Appendix A for a proof.

The window function we used is the Hamming window [46, 57], defined as

$$w[m] = 0.54 - 0.46\cos\left(\frac{2\pi m}{2I-3}\right), \tag{3.16}$$

for $0 \leq m < 2I - 2$. We chose this window function since it is widely used in the signal processing field for spectrograms.

3.1.6 Spectrogram Image

Spectrograms, as described here, contain a wide range of real values. We therefore performed some post-processing of the spectrograms to get images that are useful.

First a new image was created that contains the spectrogram values in decibels (dB). The decibel values were calculated from the spectrogram values by

$$10 \times \log_{10}\left(\frac{\text{value}+\varepsilon}{\text{reference}}\right), \tag{3.17}$$

where "value" is the spectrogram value to be converted, $\varepsilon = 10^{-30}$ assures that the numerator is non zero, and "reference" is set 100. The value of "reference" can be set to anything. We chose the value 100 since it is a commonly used value. Next we set the value of **maximum amplitude** to be the largest dB value in the new image. **Dynamic range** is the range of viewable dB values and was set to 50 dB since that is the commonly used value. Therefore, the **minimum amplitude** is defined as

$$\text{minimum amplitude=maximum amplitude-dynamic range.} \tag{3.18}$$

In the new image, all dB values below the minimum amplitude were set to the minimum amplitude. Finally the modified dB values were converted into a 0-255 integer scale by

$$255 - \left\langle \frac{255 \times (\text{modified dB value-minimum amplitude})}{\text{dynamic range}} \right\rangle. \tag{3.19}$$

Henceforth, such a post-processed image of the spectrogram is referred to as the **spectrogram image**; these are the images used for display in this text.

Figure 3.2 displays a clean spectrogram image, as defined in Section 2.3. Figure 3.3 displays the spectrogram images of the first 1500 ms of the noise signal for 5 and 0 dB SNR. Since the two images are scaled independently, they are visually hard to differentiate. Figure 3.4 displays the noisy spectrogram images with 5 and 0 dB SNR. In Figure 3.4, the difference between the SNR levels is more visible.

To test the accuracy of our spectrogram software (which creates both the spec-

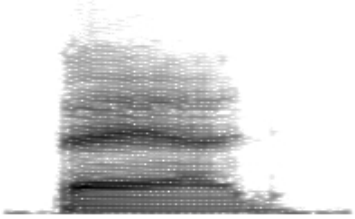


Figure 3.2: A clean spectrogram image.

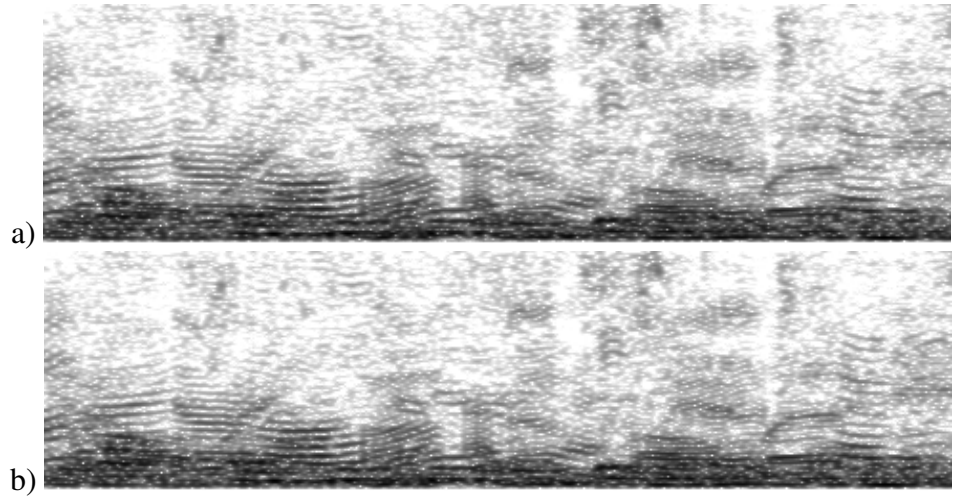


Figure 3.3: Spectrogram images of the first 1500 ms of the noise signal with a) 5 dB SNR, and b) 0 dB SNR.

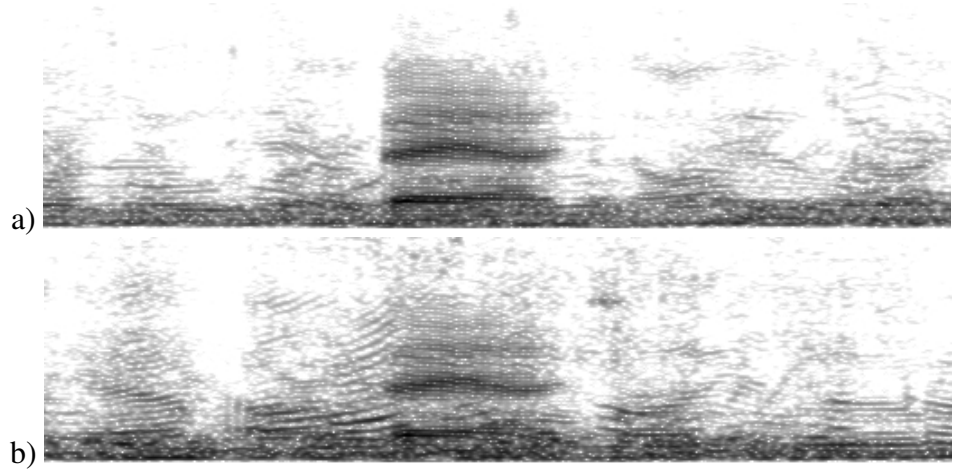


Figure 3.4: Noisy spectrogram image with a) 5 dB SNR, and b) 0 dB SNR.

trogram and the spectrogram image simultaneously), we visually compared the spectrogram images produced by our software to those produced with the same parameters using Praat [8].

3.2 Noise Model and Training

We now describe our way of estimating the statistics of the noisy grayscale columns as a function of the clean binary columns from our training set. This provides us with information about the relationship of noisy spectrograms and clean spectrograms. Since the training depends on the noisy spectrograms, the training is SNR dependent. The training was performed for each SNR level and used appropriately. We used histograms of likelihoods of noisy grayscale values as a function of each of the binary values. As described in Subsection 3.1.6, noisy grayscale values in spectrograms are real numbers and cover a wide range of values. For that reason, the histograms are functions of noisy grayscale value bins (each containing an interval of grayscale values), rather than the noisy grayscale values themselves.

We define hist0$[q]$ (respectively hist1$[q]$) to be the number of pixels in the training set whose noisy grayscale value is in bin q and its corresponding binary value is a 0 (respectively 1). We also define count0 (respectively count1) to be the number of pixels in the training set whose binary value is a 0 (respectively 1). If $\theta[h]$ is the grayscale value at pixel h in a noisy spectrogram θ, $\omega[h]$ is the binary value at pixel h in a binary image ω, and bin$[\theta[h]]$ is the bin number of $\theta[h]$, then the **noise information probability** is the conditional probability that bin$[\theta[h]] = q$ given the value $\omega[h]$. The noise information probabilities were estimated as

$$p(\text{bin}[\theta[h]] = q | \omega[h] = 0) \backsimeq \frac{\text{hist0}[q]}{\text{count0}} \tag{3.20}$$

and

$$p\left(\text{bin}\left[\theta\left[h\right]\right]=q|\omega\left[h\right]=1\right)\backsimeq\frac{\text{hist1}\left[q\right]}{\text{count1}}. \tag{3.21}$$

Notice that the estimation of the noise information probability assumes that the noisy spectrogram value $\theta[h]$ depends only on the binary value at h; it does not depend on other binary values, such as those in the vicinity of h.

Speech signals are known to have different characteristics at different frequencies. Consequently, clean and noisy speech spectrograms do not have the same characteristics at different frequency rows. For that reason the clean binary training images (defined in Section 2.3), which were created directly from the clean spectrogram images, also do not have the same characteristics at different rows. Therefore, we split the noisy spectrograms and binary images into five frequency bands in the following way.

Let **Band 0** contain the 4 rows above the bottommost row (for which $i=0$) of the spectrogram, which corresponds to a sampling of the frequencies 40.32-161.29 Hz; let **Band 1** contain the next 8 rows, which corresponds to 201.61-483.87 Hz; let **Band 2** contain the next 16 rows, which corresponds to 524.19-1129.03 Hz; let **Band 3** contain the next 32 rows, which corresponds to 1169.35-2419.35 Hz; and let **Band 4** contain the top 64 rows, which corresponds to 2459.68-5000 Hz. The reason for this choice in banding has to do with the desire for each band size to double (as the frequencies increase), which is often justified from a human auditory perception point of view [58]. Figure 3.5 displays these bands.

The histograms were calculated for each of the five bands separately; $\text{hist0}[q,b]$, $\text{hist1}[q,b]$, $\text{count0}[b]$ and $\text{count1}[b]$ are used as the band-dependent notations of $\text{hist0}[q]$, $\text{hist1}[q]$, count0 and count1 respectively, where $b\in\{0,1,2,3,4\}$ is the band number. If $\theta[h]$ is the grayscale value at pixel h in a noisy spectrogram θ, $\omega[h]$ is the binary value at pixel h in a binary image ω, $\text{bin}\left[\theta\left[h\right],b\left[h\right]\right]$ is the band-dependent bin number of $\theta[h]$, and $b[h]$ is the band number of pixel h, then the

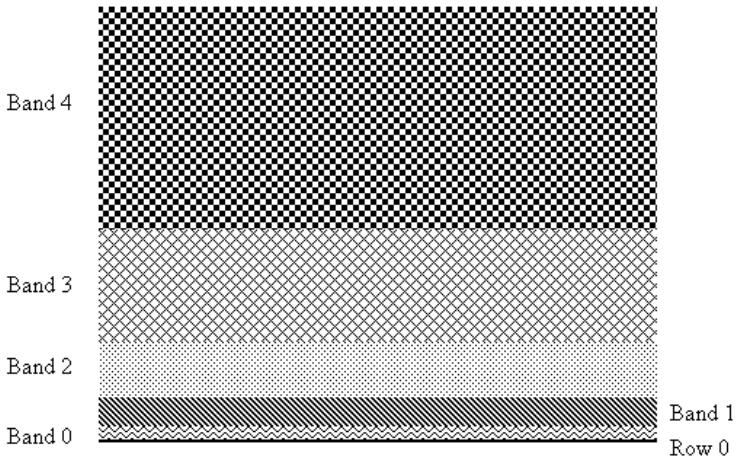


Figure 3.5: The five frequency bands.

band-dependent noise information probability is the band-dependent conditional probability that $\text{bin}[\theta[h], b[h]] = q$ given the value $\omega[h]$. We estimated the band-dependent noise information probabilities as,

$$p(\text{bin}[\theta[h], b[h]] = q | \omega[h] = 0; b[h]) \backsimeq \frac{\text{hist0}[q, b[h]]}{\text{count0}[b[h]]}, \tag{3.22}$$

and

$$p(\text{bin}[\theta[h], b[h]] = q | \omega[h] = 1; b[h]) \backsimeq \frac{\text{hist1}[q, b[h]]}{\text{count1}[b[h]]}. \tag{3.23}$$

Figure 3.6 displays the estimates of the noise information probability. The figure contains five rows corresponding to the five bands, and two columns corresponding to the two SNR levels. This figure helped evaluate the correctness of the binning and noise information probability estimation.

Our noisy grayscale bins are not uniform in length. Instead they are defined in the following way. For a particular band $b \in \{0,1,2,3,4\}$, $\text{count1}[b]$ is the number of ones in the training set in that band. For small noisy grayscale values, corresponding zeros (clean speech is absent) in the binary images are much more likely than ones. The reason for this is that small noisy grayscale values in the noisy spec-

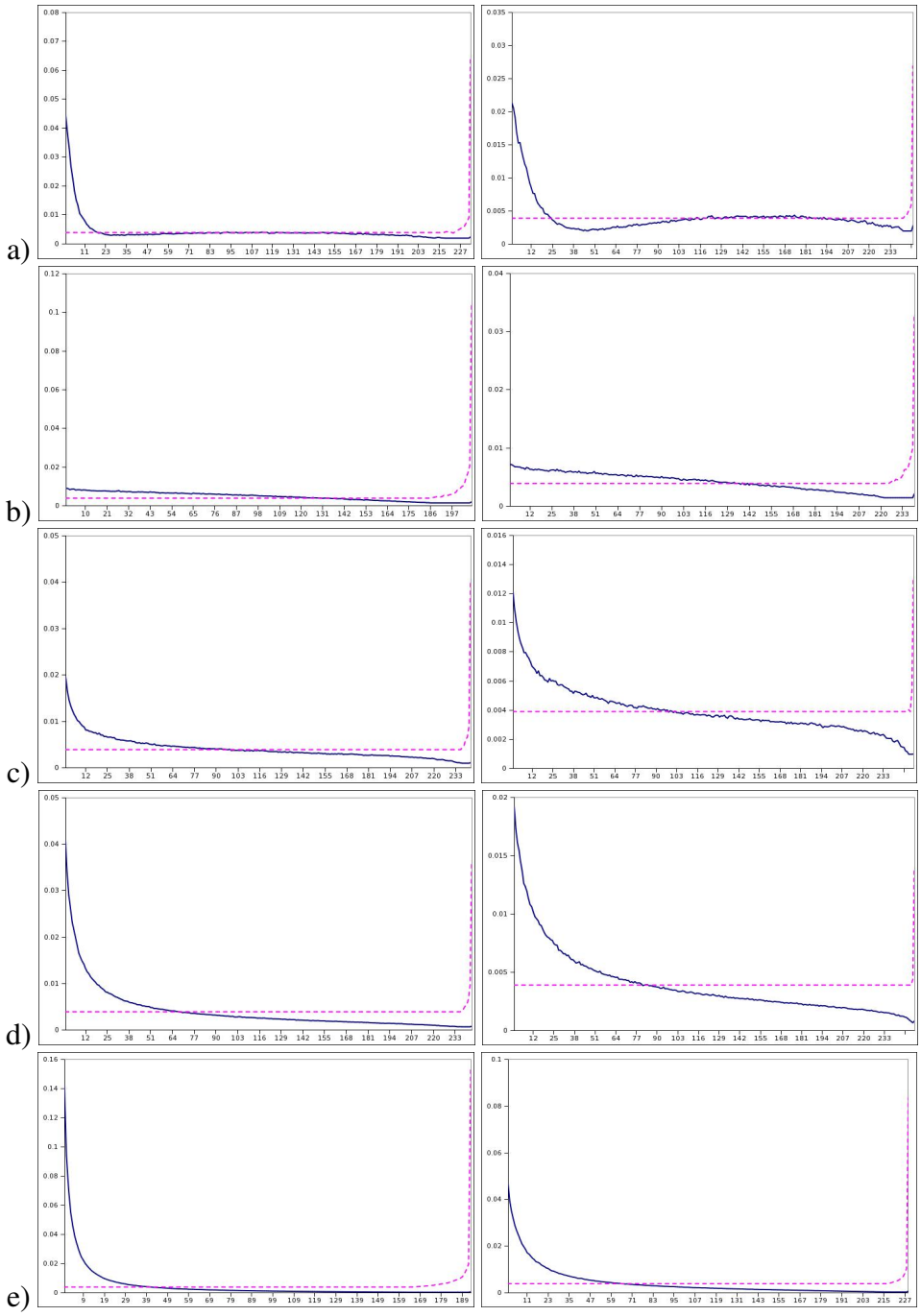


Figure 3.6: Estimated noise information probabilities for a) band 0, b) band 1, c) band 2, d) band 3, and e) band 4. The left column corresponds to the estimates for 5 dB SNR and the right column corresponds to the estimates for 0 dB SNR. The blue solid line displays the values of $p\left(\text{bin}\left[\theta\left[h\right],b\left[h\right]\right]=q|\omega\left[h\right]=0;b\left[h\right]\right)$ and the pink dashed line displays the values of $p\left(\text{bin}\left[\theta\left[h\right],b\left[h\right]\right]=q|\omega\left[h\right]=1;b\left[h\right]\right)$.

trograms are more likely to have arisen from signal segments containing noise only (rather than speech and noise). For this reason, we initially define the bin delimiters in such a way as to contain $\lfloor \text{count1}[b]/256 \rfloor$ ones in each bin. In other words, the band-dependent bin delimiters are set sequentially; the next band-dependent bin delimiter was set to the lowest gray scale value for which $\lfloor \text{count1}[b]/256 \rfloor$ ones with corresponding noisy grayscale values in that range appear in the training set. The denominator, 256, was chosen arbitrarily to be the maximum allowable number of bins. For some noisy grayscale value, corresponding ones in binary images become more probable than zeros. This means that for some noisy grayscale value, it is more likely that speech was present than not. From that point onwards, the bin delimiters are defined in such a way that they each contain $\lfloor \text{count1}[b]/256 \rfloor$ zeros in each bin. The last bin is defined to contain at least $\lfloor \text{count1}[b]/256 \rfloor$ zeros but strictly less than $2 \times \lfloor \text{count1}[b]/256 \rfloor$ zeros. For each band, the bin delimiters and the number of bins used were stored, which were not the same for the different bands and SNR values, see Figure 3.6.

The reason for such binning is to ensure that for each bin q, the counts in $\text{hist0}[q,b]$ and $\text{hist1}[q,b]$ are at least $\lfloor \text{count1}[b]/256 \rfloor$. That way, small bin counts that lead to poor estimation of the noise information probability are avoided. Appendix B displays the bin delimiters for both SNR levels.

Several other binning methods were tested, including uniform linear binning and uniform log binning. Both the uniform linear and uniform log binning contained bins whose histogram counts were small, less than 30 (generally at the extreme noisy grayscale values). That is the reason that we used the much more complex binning described above.

3.3 Prior Model and Training

We now describe our way of learning the appearance of typical clean binary images in the training set. Specifically, we estimated the likelihood of appearance of binary configurations on a particular neighborhood. Since the binary images are noise-independent, the training performed is SNR-independent and must only be performed once. Previous work on different neighborhoods and comparisons of ways to estimate them are described in [74, 75]. In those works, graphical model theory outperformed the other estimation methods, which is the reason it was chosen for this work.

Graphical model theory [18] describes a way to estimate distributions of configurations over large spaces (called "neighborhoods") from distributions of configurations over small spaces (called "cliques" and "separators"). This is useful when the training set is not large enough to accurately train the distribution over neighborhoods directly. Specifically, graphical model theory gives us that under certain conditions

$$p\,(\text{neighborhoood}) = \frac{\prod_{\text{cliques}} p\,(\text{clique})}{\prod_{\text{separators}} p\,(\text{separator})}, \tag{3.24}$$

where $p\,()$ denotes the probability of a particular binary configuration occurring over a certain space. This equality says that the probability of a binary configuration over a given neighborhood is equal to the product over all the cliques that cover the neighborhood of the probability of the binary clique configuration, divided by the product over all the separators between cliques of the probability of the binary separator configuration.

In our work, we estimated the probability distribution on the binary valued 7-pixel neighborhood drawn in Figure 3.7. We refer to the the rightmost pixel in the middle-row in the neighborhood as **pixel of interest**. The neighborhood is covered

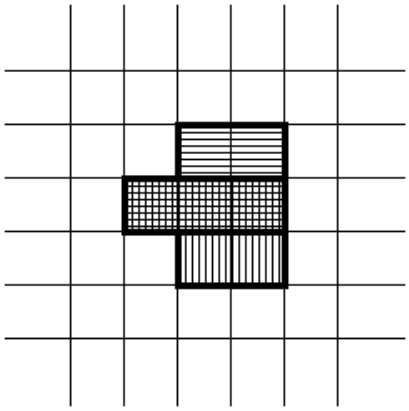


Figure 3.7: The bold black lines surround the 7-pixel neighborhood. The top 5 pixels in the neighborhood, marked with horizontal lines, make up the top clique. The bottom 5 pixels in the neighborhood, marked with vertical lines, make up the bottom clique. The separator is made up of the middle 3 pixels in the intersection of the two cliques, marked by both horizontal and vertical lines.

by two 5-pixel cliques: the **top clique** that is made up of the top five pixels in the neighborhood and the **bottom clique** that is made up of the bottom five pixels in the neighborhood, see Figure 3.7. The **separator** contains the pixels in the intersection of these two cliques, which in this particular case is the three pixels in the middle row of the neighborhood, see Figure 3.7. The neighborhood of a pixel of interest in the topmost row ($i = I - 1$) is defined differently: it contains only the pixels in the bottom clique. Similarly, the neighborhood of a pixel of interest in row $i = 1$ contains only the pixels in the top clique. For our work, there is no need to deal with the case when the pixel of interest is in the bottommost row ($i = 0$).

The reason that such an asymmetric neighborhood was chosen has to do with the estimation algorithm described in Section 3.4. For now recall that spectrograms have time on the horizontal axis. Therefore when estimating the clean spectrogram column corresponding to the unknown clean speech segments, decisions must be based only on heard noisy speech segments and decisions already made. That means that a pixel of interest can be estimated based only on pixels in the same column or columns to its left.

We now describe how to estimate clique and separator probability distributions. The binary training images were padded with two columns of 0s on the left. The clique probability distributions were estimated by counting directly from the binary training set. That means that the probability of a clique being a certain configuration was estimated by the number of times that clique was assigned that configuration in the training set divided by the number of such cliques in the training set. When counting the occurrences of the clique configurations in the training set images, we required that the clique be fully contained in the padded image excluding row 0. The separator distribution is estimated from the bottom clique probability distribution. Specifically, the separator distribution was estimated from the bottom clique distribution by summing over all the possible configurations of the bottom two pixels in the neighborhood. That means that for a particular separator configuration, its probability was estimated by summing over the probabilities of all the bottom clique configuration having the separator configuration in the top row. (The separator distribution could similarly have been estimated from the top clique distribution.) Once the clique and separator distributions were estimated, we estimated the neighborhood probability distribution using graphical model theory, Equation 3.24.

For the same reasons described in Section 3.2, we chose to estimate probability distributions over the binary neighborhood configurations for the frequency bands separately, which we refer to as the band-dependent neighborhood probability distributions. The band-dependent clique probability distributions were estimated by counting the cliques configuration occurrences in bands, similarly to the way described for the whole image. Specifically, for each band, clique configurations were estimated by counting only those fully contained in that band. The band-dependent separator probability distribution were estimated in the exact same way from the band-dependent bottom clique probability distribution. The band-dependent neigh-

borhood probability distribution was estimated using Equation 3.24 for each band separately.

If h is a pixel of interest and ω is a binary image, then let $N[h]$ contain the locations of the other (at most) six pixels in the neighborhood of h and let $N_\omega[h]$ be the binary values of the pixels in $N[h]$ in image ω. For $\mu \in \{0,1\}$,

$$p(\omega[h] = \mu, N_\omega[h]; b[h]) \tag{3.25}$$

is the **band-dependent prior joint probability** that $\omega[h] = \mu$ and $N[h]$ is assigned the configuration $N_\omega[h]$. For all pixels of interest not located in the bottommost two rows or the topmost row, the band-dependent prior joint probability was estimated by the band-dependent neighborhood probability. For pixels in row $i = 1$, the band-dependent prior joint probability was estimated by the band-dependent probability of the top clique configuration. For pixels in the topmost row ($i = I - 1$), the band-dependent prior joint probability was estimated by the probability of the bottom clique configuration. For pixels in row $i = 0$, the band-dependent prior joint probability need not be defined.

We can now define the **band-dependent prior conditional probability** that $\omega[h] = \mu$, given the neighborhood configuration $N_\omega[h]$, to be

$$p(\omega[h] = \mu | N_\omega[h]; b[h]) = \frac{p(\omega[h] = \mu, N_\omega[h]; b[h])}{p(N_\omega[h]; b[h])} \tag{3.26}$$

where

$$p(N_\omega[h]; b[h]) = p(\omega[h] = 0, N_\omega[h]; b[h]) + p(\omega[h] = 1, N_\omega[h]; b[h]). \tag{3.27}$$

Putting Equations 3.26 and 3.27 together, we see that the band-dependent prior conditional probability can be estimated directly from the band-dependent prior

joint probabilities.

To assess the correctness of the estimated prior model, the following preliminary experiment was conducted. The iterative Metropolis Algorithm can be used to sample distributions [51]; it is known that the probability of an image appearing in the Metropolis sequence is as likely as the probability of the image in the distribution. We, therefore, sampled the prior distribution using the Metropolis Algorithm. We then estimated the percent of black pixels in each band and compared it to the percent of black pixels in the training set.

Several other (larger) neighborhoods (with different cliques and separators) were considered for the binary prior. We found that with our banding we could not accurately estimate cliques and separator distributions containing more than five pixels from the given training set. By that we mean that for cliques containing more than five pixels, we found that there existed at least one binary clique configuration that appeared less than thirty times in the training set. Furthermore, not all collections of cliques formed neighborhoods for which Graphical Model theory could be used. These constraints were the reason for the chosen small neighborhood depicted in Figure 3.7.

3.4 Step 2: Binary Column Estimation Algorithm

Using the information learned during the training processes detailed in Sections 3.2 and 3.3, the second step in our methodology is to estimate a binary column from a noisy grayscale spectrogram column. The binary columns should ideally tell us in which time-frequency locations the speech in the unknown clean speech signal was contained. For that reason, the creation of binary images from noisy spectrograms takes into account both the characteristics of binary images corresponding to clean speech signals, as well as the information relating noisy speech spectrograms to the

binary training images.

Subsection 3.4.1 describes the function we wish to maximize based on the prior and noise models described earlier. Subsection 3.4.2 describes the way we have chosen to search for a global maximum of our defined maximization function. Subsection 3.4.3 describes the look-up tables we pre-calculated and saved in order to speed up the implementation of the algorithm that searches for a maximum. Subsection 3.4.4 describes the update of the maximization function based on the pre-calculated look-up tables.

3.4.1 Maximization Function

Let ω be a binary image and $\mu \in \{0,1\}$. The band-dependent prior conditional probability that $\omega[h] = \mu$ given its binary neighborhood configuration $N_\omega[h]$, namely

$$p(\omega[h] = \mu | N_\omega[h]; b[h]), \tag{3.28}$$

was estimated using graphical models as described in Section 3.3. Let θ be a grayscale image in general and a noisy spectrogram in particular in this work. The band-dependent noise information probability that $\text{bin}[\theta[h]; b[h]] = q$ given the binary value $\omega[h]$, namely

$$p(\text{bin}[\theta[h]; b[h]] = q | \omega[h]; b[h]), \tag{3.29}$$

was estimated using the histograms described in Section 3.2. By Bayes' Law, the **band-dependent posterior probability** (or posterior probability for short) that $\omega[h] = \mu$ given $\text{bin}[\theta[h]; b[h]] = q$ and the neighborhood configuration $N_\omega[h]$, namely

$$p(\omega[h] = \mu | \text{bin}[\theta[h]; b[h]] = q, N_\omega[h]; b[h]) \tag{3.30}$$

is the normalized product of the band-dependent prior conditional probability and

the band-dependent noise information probability. Symbolically, it is equal to

$$\begin{gathered} p\left(\omega\left[h\right]=\mu|\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right) \\ = \\ \frac{p\left(\omega\left[h\right]=\mu|N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu;b\left[h\right]\right)}{p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|N_{\omega}\left[h\right];b\left[h\right]\right)}. \end{gathered} \tag{3.31}$$

The posterior probability is also equal to

$$\begin{gathered} p\left(\omega\left[h\right]=\mu|\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right) \\ = \\ \frac{p\left(\omega\left[h\right]=\mu,N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu;b\left[h\right]\right)}{p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right)}. \end{gathered} \tag{3.32}$$

The difference between these two equations is the following. In Equation 3.31, the prior probability and the normalizer are conditional probabilities. In Equation 3.32, the prior probability and the normalizer are joint probabilities. See Appendix C for a proof of Equations 3.31 and 3.32. Equations 3.31 and 3.32 hold due to the assumption that a noisy grayscale value $\theta\left[h\right]$ depends only on the binary value $\omega\left[h\right]$ and not on the binary configuration $N_{\omega}\left[h\right]$. This means that

$$\begin{aligned} &p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu,N_{\omega}\left[h\right];b\left[h\right]\right) \\ &=p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu;b\left[h\right]\right). \end{aligned} \tag{3.33}$$

The denominator of Equation 3.31, the normalizer

$$p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|N_{\omega}\left[h\right];b\left[h\right]\right), \tag{3.34}$$

can be rewritten as the weighted average of

$$p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu;b\left[h\right]\right) \tag{3.35}$$

where the weights are

$$p\left(\omega\left[h\right]=\mu|N_{\omega}\left[h\right];b\left[h\right]\right). \tag{3.36}$$

Symbolically, the normalizer is equal to

$$\begin{gathered} p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|N_{\omega}\left[h\right];b\left[h\right]\right) \\ = \\ p\left(\omega\left[h\right]=0|N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=0;b\left[h\right]\right) \\ +p\left(\omega\left[h\right]=1|N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=1;b\left[h\right]\right). \end{gathered} \tag{3.37}$$

See Appendix D for a proof of Equation 3.37. Similarly, the denominator of Equation 3.32, the normalizer

$$p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right), \tag{3.38}$$

can be written as the weighted average of

$$p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu;b\left[h\right]\right) \tag{3.39}$$

where the weights are

$$p\left(\omega\left[h\right]=\mu,N_{\omega}\left[h\right];b\left[h\right]\right). \tag{3.40}$$

Symbolically the normalizer is equal to

$$\begin{gathered} p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right) \\ = \\ p\left(\omega\left[h\right]=0,N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=0;b\left[h\right]\right) \\ +p\left(\omega\left[h\right]=1,N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=1;b\left[h\right]\right). \end{gathered} \tag{3.41}$$

See Appendix D for a proof of Equation 3.41.

Let us now denote by $\hat{\theta}$ the specific noisy grayscale spectrogram that we are processing with the aim of obtaining the binary image $\hat{\omega}$, which is our estimate of

the unavailable clean binary image. We do this column by column. We note that in the application we are considering, by Equation 3.15, the bottommost row of any spectrogram is zero-valued and, therefore, the bottommost row of $\hat{\omega}$ need not be estimated; it can be set directly to zeros. For any column index j, we define the set $H_j = \{h | h \text{ is in the } j\text{th column but not in the bottommost row}\}$.

As the noisy speech signal arrives, noisy grayscale spectrogram columns are revealed. At time $\hat{j}s + 2I - 3$, column $\hat{j}$ is revealed. At this time, we know $\hat{\theta}[h]$ for $h \in H_{\hat{j}}$, and we have already estimated $\hat{\omega}[h]$ for all $h \in \bigcup_{0 \leq j < \hat{j}} H_j$. We estimate $\hat{\omega}[h]$ for $h \in H_{\hat{j}}$ as follows. Let ϖ be any binary image such that $\varpi[h] = \hat{\omega}[h]$, for all $h \in \bigcup_{0 \leq j < \hat{j}} H_j$. For any pixel $h \in H_{\hat{j}}$, let $\hat{q}[h] = \text{bin}\left[\hat{\theta}[h]; b[h]\right]$. We now define the **pseudo-posterior likelihood** [6] $M\left(\varpi; \hat{j}\right)$ of the $\hat{j}$th column of ϖ to be

$$M\left(\varpi; \hat{j}\right) = \prod_{h \in H_{\hat{j}}} p\left(\omega[h] = \varpi[h] \,|\text{bin}\left[\theta[h]; b[h]\right] = \hat{q}[h], N_{\varpi}[h]; b[h]\right), \qquad (3.42)$$

which is the product of the band-dependent pixel posterior probabilities, see Equation 3.30, over the $I - 1$ pixels in $H_{\hat{j}}$.

We now aim at finding a binary image ϖ that maximizes $M\left(\varpi; \hat{j}\right)$. Since $M\left(\varpi; \hat{j}\right)$ depends only those values of $\varpi[h]$ for which $h \in H_{\hat{j}}$ (values of $\varpi[h]$ in earlier columns are fixed and those in later columns do not enter into Equation 3.42), this is equivalent to finding a single column of binary values. Once this has been done, we can extend the definition of $\hat{\omega}$ to the $\hat{j}$th column by defining $\hat{\omega}[h]$ to be $\varpi[h]$, for $h \in H_{\hat{j}}$, for an ϖ that maximizes $M\left(\varpi; \hat{j}\right)$. In what follows, we refer to the $\hat{\omega}$ obtained in this fashion as the **hard segmentation image** (or simply as the hard segmentation), and to its active (i.e., $\hat{j}$th) column as the **hard segmentation column**.

In preliminary work, we attempted to maximize the binary column probability,

the product over all the pixels in $H_{\hat{j}}$ of the band-dependent prior conditional probability and the band-dependent noise information probability. This function implicitly assumed pixel value independence, which is not a valid assumption. Thereafter, an alternative method for estimating potentials for Gibbs distributions that provide for a good fit (in the minimum square error sense) was discussed. After some consideration, it was agreed that there was no good reason to believe that it would perform better than pseudo-likelihood. Since using pseudo-likelihood did not require much change to the existing programs, and since pseudo-likelihood has been used by others in similar situations [6], it is the maximization function we chose to work with.

3.4.2 Maximizing the Function

We would like to search for a global maximum of the function $M\left(\varpi;\hat{j}\right)$ defined in Equation 3.42. The Metropolis Algorithm [51] with simulated annealing has been shown to be a useful tool for searching for global maximum of functions [78]. As the signal arrives, in other words column by column, we use the Metropolis Algorithm with an annealing schedule to search for the binary column that maximizes the function $\left\{M\left(\varpi;\hat{j}\right)\right\}^{\beta}$. The parameter β is called the inverse temperature; it controls an annealing schedule.

The Metropolis Algorithm can be initialized arbitrarily. We initialize the column $\hat{j}$ with zeros. Since we know that all the clean speech signals (from which the noisy speech signals arise) start and end in silence, this helps speed up the algorithm by being a close starting point. We pad two columns on the left of the binary image ϖ with zeros so as to assure that neighborhoods are defined even for columns for which $\hat{j} \in \{0,1\}$. We set the bottommost pixel of the hard segmentation to zero; its value remains unchanged during the iterative algorithm.

For each column $\hat{j}$ and inverse temperature $\hat{\beta}$ in turn, an iterative step of the

Metropolis Algorithm is as follows. Select a random pixel location in $H_{\hat{j}}$, namely $\hat{h}$. Change the binary color of the current binary image $\varpi_1\left[\hat{h}\right]$ to create a new binary image ϖ_2. Let

$$p = \left\{ \frac{M\left(\varpi_2;\hat{j}\right)}{M\left(\varpi_1;\hat{j}\right)} \right\}^{\hat{\beta}} \tag{3.43}$$

That is, p is the quotient of the value of the maximization function after the pixel color change divided by the value of the maximization function before the pixel color change, raised to the power of $\hat{\beta}$. The value of p tells us how much more likely the binary image ϖ_2 is as compared to the binary image ϖ_1. Binary image ϖ_2 replaces image ϖ_1 with probability $\min(p,1)$, where $\min(\delta,\eta)$ returns the minimum of the real numbers δ and η. Therefore, the change in the color of pixel $\hat{h}$ is accepted with probability $\min(p,1)$.

The consequence of such an iterative step is that the color change is definitely accepted if $M\left(\varpi_2;\hat{j}\right) \geq M\left(\varpi_1;\hat{j}\right)$, i.e. if the value of the maximization function is not less after the color change. The color change may be accepted even if $M\left(\varpi_2;\hat{j}\right) < M\left(\varpi_1;\hat{j}\right)$. In this case, the less desirable the color change is, in terms of the value of $M\left(\varpi;\hat{j}\right)$, the less likely it is that the color change will be accepted. These potential downward steps, decreases in the value of $M\left(\varpi;\hat{j}\right)$ during the iterative process, are what allows the algorithm to get out of a local maximum.

The Metropolis Algorithm without annealing (in other words with $\hat{\beta} = 1$) was proved to sample from the distribution $M\left(\varpi;\hat{j}\right)$ [51]. That means that images that are likely in the distribution $M\left(\varpi;\hat{j}\right)$ will appear often in the sampling during the iterative process. In particular, each image will appear proportionally to its value in $M\left(\varpi;\hat{j}\right)$. Furthermore, the Metropolis Algorithm with slow simulated annealing was proved to converge to a global maximum in the limit [10]. In [78] discrete annealing schedules used with the Metropolis Algorithm were proven to converge to a global maximum in the limit.

The reason for this is that the annealing schedule has the effect of making the distribution $\left\{M\left(\varpi;\hat{j}\right)\right\}^{\beta}$ more peaked as the value of β increases and more flat as the value of β decreases. The value of $\beta > 0$ does not affect the location of the maximal binary image ω in $\left\{M\left(\varpi;\hat{j}\right)\right\}^{\beta}$. This explains why using a slow enough annealing schedule with the Metropolis Algorithm is a useful tool for searching for global maximum.

Theorem 5.2.1 in [78] describes an annealing schedule (also called a cooling schedule). The theorem states that for any schedule $\beta[v]$, $v = 0, 1, \ldots$, such that

$$\beta[v] \leq \frac{1}{\sigma\Delta}\ln(v), \tag{3.44}$$

and for an arbitrary initialization, the Metropolis algorithm will converge to a global maximum. In our work the constants σ and Δ are 123 and 11.33 respectively. The quickest schedule that this theorem tells us is guaranteed to converge is

$$\beta[v] = \frac{1}{\sigma\Delta}\ln(v). \tag{3.45}$$

In order for $\beta[v]$ to reach the value of 0.1, v would have to equal 3.333×10^{60}. From experience, we know that we need a schedule that continues until $\beta = 1.50$. Such a schedule would be prohibitively long.

We therefore use the following much faster annealing schedule. We define a **cycle** to have $I-1$ iterations. Our annealing schedule per column starts at $\beta = 0.50$, and increases the value of β by 0.01 at every 200 cycles until $\beta = 1.50$. In this work, the initial column values at the following β in the annealing schedule are the values of the sampled column for which the value of $\left\{M\left(\varpi;\hat{j}\right)\right\}^{\beta}$ was the highest for the previous value of β.

We ran the annealing schedule NR times per column. The column for which the value of $M\left(\varpi;\hat{j}\right)$ is highest over the NR runs of the stochastic algorithm is the

a)

b)

Figure 3.8: The hard segmentation of the noisy spectrogram with a) 5 dB SNR, and b) 0 dB SNR.

one chosen to be the estimated binary column for the noisy spectrogram column. In this work we used NR $= 5$. Figure 3.8 shows a hard segmentation produced by our methodology with the above annealing schedule. The reason that the annealing schedule was run more than once per column had to do with the short annealing schedule used. We found that infrequently, a column was estimated poorly. Unfortunately, it strongly affected the estimation of the following columns for the worse. Instead of using a longer annealing schedule (more cycles before increasing β, smaller increments of β, a smaller start value of β, a larger end value of β), we found that running a short annealing schedule several times per column assured "good" column estimations in terms of the maximization function.

To test the accuracy of the binary image estimation, manual calculations were performed. In particular, the posterior probabilities were manually calculated for pixels in the first white row of Figure 3.8b. Similarly, the posterior probabilities were manually calculated for pixels in a column located after (to the right of) clean speech had ended.

3.4.3 Look-Up Tables

During each Metropolis iteration, the value of p must be calculated in order to decide whether a certain pixel value should be changed. This is the reason that running such an iterative algorithm is so time consuming - the calculation of p is expensive and must be preformed frequently. For that reason, we attempted to pre-calculate and store in advance look-up tables that facilitate the calculation of p as much as possible. These look- up tables are described in this section.

Recall from Subsection 3.4.2 that p is defined to be the quotient $p = \left\{ \frac{M\left(\varpi_2; \hat{j}\right)}{M\left(\varpi_1; \hat{j}\right)} \right\}^{\hat{\beta}}$, where binary images ϖ_1 and ϖ_2 differ only at pixel $\hat{h}$ in column $\hat{j}$. Also recall that $M\left(\varpi; \hat{j}\right)$ is the product of the band-dependent posterior probabilities over the $I-1$ pixels in $H_{\hat{j}}$. That means that the band-dependent posterior probabilities

$$p\left(\omega\left[h\right] = \varpi_1\left[h\right] | \text{bin}\left[\theta\left[h\right]; b\left[h\right]\right] = q, N_{\varpi_1}\left[h\right]; b\left[h\right]\right) \tag{3.46}$$

and

$$p\left(\omega\left[h\right] = \varpi_2\left[h\right] | \text{bin}\left[\theta\left[h\right]; b\left[h\right]\right] = q, N_{\varpi_2}\left[h\right]; b\left[h\right]\right) \tag{3.47}$$

are equal for all pixels, except for pixels $\hat{h}$, h^+ (which we define to be the pixel above $\hat{h}$) and h^- (which we define to be the pixel below $\hat{h}$). Let $\hat{\theta}$ be the noisy grayscale image whose clean version we are trying to estimate. Let $\hat{q}\left[h^+\right] = \text{bin}\left[\hat{\theta}\left[h^+\right]; b\left[h^+\right]\right]$, $\hat{q}\left[\hat{h}\right] = \text{bin}\left[\hat{\theta}\left[\hat{h}\right]; b\left[\hat{h}\right]\right]$, and $\hat{q}\left[h^-\right] = \text{bin}\left[\hat{\theta}\left[h^-\right]; b\left[h^-\right]\right]$. The value of p can be re-written as the product of the following three ratios of posterior probabilities,

$$
\begin{aligned}
p = & \\
& \left\{ \frac{p\left(\omega\left[h^{+}\right]=\varpi_2\left[h^{+}\right]\left|\operatorname{bin}\left[\theta\left[h^{+}\right];b\left[h^{+}\right]\right]=\hat{q}\left[h^{+}\right],N_{\varpi_2}\left[h^{+}\right];b\left[h^{+}\right]\right.\right)}{p\left(\omega\left[h^{+}\right]=\varpi_1\left[h^{+}\right]\left|\operatorname{bin}\left[\theta\left[h^{+}\right];b\left[h^{+}\right]\right]=\hat{q}\left[h^{+}\right],N_{\varpi_1}\left[h^{+}\right];b\left[h^{+}\right]\right.\right)} \right\}^{\hat{\beta}} \\
& \times \left\{ \frac{p\left(\omega\left[\hat{h}\right]=\varpi_2\left[\hat{h}\right]\left|\operatorname{bin}\left[\theta\left[\hat{h}\right];b\left[\hat{h}\right]\right]=\hat{q}\left[\hat{h}\right],N_{\varpi_2}\left[\hat{h}\right];b\left[\hat{h}\right]\right.\right)}{p\left(\omega\left[\hat{h}\right]=\varpi_1\left[\hat{h}\right]\left|\operatorname{bin}\left[\theta\left[\hat{h}\right];b\left[\hat{h}\right]\right]=\hat{q}\left[\hat{h}\right],N_{\varpi_1}\left[\hat{h}\right];b\left[\hat{h}\right]\right.\right)} \right\}^{\hat{\beta}} \\
& \times \left\{ \frac{p\left(\omega\left[h^{-}\right]=\varpi_2\left[h^{-}\right]\left|\operatorname{bin}\left[\theta\left[h^{-}\right];b\left[h^{-}\right]\right]=\hat{q}\left[h^{-}\right],N_{\varpi_2}\left[h^{-}\right];b\left[h^{-}\right]\right.\right)}{p\left(\omega\left[h^{-}\right]=\varpi_1\left[h^{-}\right]\left|\operatorname{bin}\left[\theta\left[h^{-}\right];b\left[h^{-}\right]\right]=\hat{q}\left[h^{-}\right],N_{\varpi_1}\left[h^{-}\right];b\left[h^{-}\right]\right.\right)} \right\}^{\hat{\beta}}.
\end{aligned}
\tag{3.48}
$$

We refer to these three ratios, without the power of $\hat{\beta}$, as **Term 1**, **Term 2**, and **Term 3**, respectively.

Notice that the three terms in the calculation of p are similar but not equivalent. Term 1 describes the change in the posterior probability of pixel h^{+} when the value of $\hat{h}$ is changed. This is clearly different from Term 2, which describes the change in the posterior probability of pixel $\hat{h}$ when its value is changed. Term 3 is similar to Term 1 but describes the change of the posterior probability of pixel h^{-} when the color of $\hat{h}$ is changed.

Notice that if $\hat{h}$ is in row $i = 1$, only two of the three terms (Term 1 and Term 2) exist in the calculation of p and that in Term 2 the prior joint probabilities of $\hat{h}$ in images ϖ_1 and ϖ_2 are estimated from only one clique (the top clique). Similarly for $\hat{h}$ in row $i = I - 1$, only two terms (Term 2 and Term 3) exist in the calculation of p and in Term 2 the prior join probabilities of $\hat{h}$ in images ϖ_1 and ϖ_2 are estimated from only one clique (the bottom clique). Also notice that if $\hat{h}$ is in row $i = 2$, three terms exist in the calculation of p but in Term 3 the prior joint probabilities of h^{-} in images ϖ_1 and ϖ_2 are estimated from only one clique (the top clique). Similarly for $\hat{h}$ in row $i = I - 2$, three terms exist in the calculation of p but in Term 1 the prior joint probabilities of h^{+} in images ϖ_1 and ϖ_2 are estimated from only one clique (the bottom clique).

To simplify the calculation of the value of p performed in the Metropolis Algorithm, several look-up tables were created and stored in advance. The following is a description of these look-up tables and one more look-up table created for later needs, which is described first. One table is of type float (4 bytes) and contains the estimated band-dependent posterior probability that $\omega[h] = 1$ given $\text{bin}[\theta[h]; b[h]] = q$ and the neighborhood configuration $N_\omega[h]$ (Equations 3.31 and 3.37), namely

$$\begin{gathered} \text{posterior}\left(b[h], \text{bin}[\theta[h]; b[h]], N_\omega[h]\right) \\ = \\ p\left(\omega[h] = 1 | \text{bin}[\theta[h]; b[h]] = q, N_\omega[h]; b[h]\right). \end{gathered} \tag{3.49}$$

This table is an 8-dimensional table whose indices are: one index for the band number $b[h]$, one index for the noisy grayscale bin number $\text{bin}[\theta[h]; b[h]]$, and six indices for the binary neighborhood configuration $N_\omega[h]$ (that excludes the value of h). The use of this table will be described in Subsection 3.4.4.

Three look-up tables were created, each one relating to one of the three terms in the calculation of p. These three look-up tables, which we refer to as the **regular look up tables**, are of type short (2 bytes). One of table is created for Term 2 and contains the value

$$\begin{gathered} \left\langle \alpha \times \ln\left(p\left(\omega\left[\hat{h}\right] = 1 | \text{bin}\left[\theta\left[\hat{h}\right]; b\left[\hat{h}\right]\right] = \hat{q}\left[\hat{h}\right], N_{\varpi_2}\left[\hat{h}\right]; b\left[\hat{h}\right]\right)\right)\right\rangle \\ - \left\langle \alpha \times \ln\left(p\left(\omega\left[\hat{h}\right] = 0 | \text{bin}\left[\theta\left[\hat{h}\right]; b\left[\hat{h}\right]\right] = \hat{q}\left[\hat{h}\right], N_{\varpi_1}\left[\hat{h}\right]; b\left[\hat{h}\right]\right)\right)\right\rangle, \end{gathered} \tag{3.50}$$

which approximates the value $\langle \alpha \times \ln(\text{Term 2})\rangle$ when $\varpi_2\left[\hat{h}\right] = 1$. The value of α is chosen to be a large positive integer for which the value of Equation 3.50 can be stored as shorts. In our case, we set $\alpha = 100$. This table is 8-dimensional whose indices are: one index for the band number $b\left[\hat{h}\right]$, one index for the noisy

grayscale bin number $\hat{q}\left[\hat{h}\right]$, and six indices for the binary neighborhood configuration $N_{\varpi_1}\left[\hat{h}\right] = N_{\varpi_2}\left[\hat{h}\right]$ (that excludes the value of $\hat{h}$).

The other two regular look-up tables were pre-calculated in the same way for Terms 1 and 3. For Term 1, let $h = h^+$ and $\hat{q}[h] = \hat{q}[h^+]$; for Term 3, let $h = h^-$ and $\hat{q}[h] = \hat{q}[h^-]$. The tables contain the values

$$\begin{aligned} &\langle \alpha \times \ln(p(\omega[h] = 1 | \text{bin}[\theta[h]; b[h]] = \hat{q}[h], N_{\varpi_2}[h]; b[h]))\rangle \\ &- \langle \alpha \times \ln(p(\omega[h] = 0 | \text{bin}[\theta[h]; b[h]] = \hat{q}[h], N_{\varpi_1}[h]; b[h]))\rangle, \end{aligned} \tag{3.51}$$

which approximates the values of $\langle \alpha \times \ln(\text{Term 1})\rangle$ and $\langle \alpha \times \ln(\text{Term 3})\rangle$ when $\varpi_2[h] = 1$. These two look-up tables are both 8-dimensional tables whose indices are: one index for the band number $b[h]$, one index for the noisy grayscale bin number $\hat{q}[h]$, and six indices for the binary neighborhood configuration $N_{\varpi_1}[h]$ including the value of h and excluding the value of $\hat{h}$ (which is the same as the binary neighborhood configuration $N_{\varpi_2}[h]$ including the value of h and excluding the value of $\hat{h}$). For pixels $\hat{h}$ in rows $3 \leq i \leq I-3$, the value of $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ (whose need will be described in Subsection 3.4.4) was estimated by adding the values of the regular look-up table for Term 1, the regular look-up table for Term 2 and the regular look-up table for Term 3, see Subsection 3.4.4 for more detail.

Four additional tables were created for the special cases in which $\hat{h}$ is in rows $i \in \{1, 2, I-2, I-1\}$, as described earlier in this section. We refer to these as the **special look-up tables**. For the case that $\hat{h}$ is in row $i = 1$, a special table was created similar to the regular look-up table for Term 2, with the exception that the prior joint probabilities are estimated from only one clique (the top clique). In this case, the value of $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ was estimated from the values of the regular look-up table for Term 1 and the special look-up table for Term 2. Similarly, a special table is created for Term 2 for the case that $\hat{h}$ is in row $i = I-1$, in which the prior

joint probabilities were estimated from only one clique (the bottom clique). In this case, the value of $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ was estimated from the values of the the special look-up table for Term 2 and regular look-up table for Term 3. For the case that $\hat{h}$ is in row $i = 2$, a special table is created similar to the regular look-up table for Term 3, with the exception that the prior joint probabilities are estimated from only one clique (the top clique). In this case, the value of $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ was estimated from the values of the regular look-up table for Term 1, the regular look-up table for Term 2 and the special look-up table for Term 3. Similarly, a special table is created for Term 3 for the case that $\hat{h}$ is in row $i = I-2$, in which the prior joint probabilities were estimated from only one clique (the bottom clique). In this case, the value of $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ was estimated from the values of the special look-up table for Term 1, the regular look-up table for Term 2 and the regular look-up table for Term 3.

These four special tables are all 5-dimensional tables of type short. Since these cases determine the band number ($b\left[\hat{h}\right] = 0$ if $\hat{h}$ is in rows $i \in \{1,2\}$ and $b\left[\hat{h}\right] = 4$ if $\hat{h}$ is in rows $i \in \{I-2, I-1\}$), the band number is not an index to these tables. The indices are: one index for the noisy grayscale bin number $\hat{q}\left[\hat{h}\right]$, and four indices for the binary configuration of the clique of interest, excluding the value of $\hat{h}$.

3.4.4 Column Energy

We now describe how the look-up tables defined in Subsection 3.4.3 are used during a run of the Metropolis Algorithm. Let the **column energy** of a column $\hat{j}$ in a binary image ϖ, $C\left(\varpi; \hat{j}\right)$, be defined as

$$C\left(\varpi; \hat{j}\right) = \ln\left(M\left(\varpi; \hat{j}\right)\right). \tag{3.52}$$

Notice that maximizing $M\left(\varpi; \hat{j}\right)$ is equivalent to maximizing $C\left(\varpi; \hat{j}\right)$ or $\alpha C\left(\varpi; \hat{j}\right)$

(since α is a positive number). Since before starting the annealing schedule, the elements of column $\hat{j}$ are all zero, we estimate the initial value of $\alpha C\left(\varpi;\hat{j}\right)$ in the following way. Let

$$\text{posterior}\left(b\left[h\right],\text{bin}\left[\theta\left[h\right];b\left[h\right]\right],N_{\varpi}\left[h\right]\right) \tag{3.53}$$

be the posterior probability float value saved in the look-up table defined in Equation 3.49 for pixel h in column $\hat{j}$ and band $b\left[h\right]$ whose grayscale value bin is $\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]$ and whose binary neighborhood is $N_{\varpi}\left[h\right]$. We estimate the value of $\alpha C\left(\varpi;\hat{j}\right)$ by

$$\alpha C\left(\varpi;\hat{j}\right)\simeq\sum_{h\in H_{\hat{j}}}\left\langle\alpha\times\ln\left(1\text{-posterior}\left(b\left[h\right],\text{bin}\left[\theta\left[h\right];b\left[h\right]\right],N_{\varpi}\left[h\right]\right)\right)\right\rangle. \tag{3.54}$$

This sum is the estimated value of $\alpha C\left(\varpi;\hat{j}\right)$ before the annealing schedule starts.

During the Metropolis step, rather than estimating the value of p, we estimate the value of $\left\langle\frac{\alpha}{\hat{\beta}}\times\ln\left(p\right)\right\rangle=\left\langle\alpha\times\ln\left(p^{1/\hat{\beta}}\right)\right\rangle$. This value is estimated by

$$\left\langle\alpha\times\ln\left(M\left(\varpi_2;\hat{j}\right)\right)\right\rangle-\left\langle\alpha\times\ln\left(M\left(\varpi_1;\hat{j}\right)\right)\right\rangle, \tag{3.55}$$

where ϖ_1 and ϖ_2 are binary images that differ only at pixel $\hat{h}$. This is further estimated (for pixels $\hat{h}$ not in rows $i\in\{1,2,I-2,I-1\}$) from three regular look-up table values and two additions. If $\varpi_2\left[\hat{h}\right]=1$, then the value $\left\langle\frac{\alpha}{\hat{\beta}}\times\ln\left(p\right)\right\rangle$ is estimated from the sum of the appropriate three regular look-up table values as described in Subsection 3.4.3; otherwise the value of $\left\langle\frac{\alpha}{\hat{\beta}}\times\ln\left(p\right)\right\rangle$ is estimated from the negative sum of the appropriate three regular look-up table values. For pixels $\hat{h}$ in rows $i\in\{1,2,I-2,I-1\}$, the value of $\left\langle\frac{\alpha}{\hat{\beta}}\times\ln\left(p\right)\right\rangle$ is estimated as described in Subsection 3.4.3. If $\varpi_2\left[\hat{h}\right]=1$, then the value $\left\langle\frac{\alpha}{\hat{\beta}}\times\ln\left(p\right)\right\rangle$ is estimated from the sum of the appropriate look-up table values; otherwise the value of $\left\langle\frac{\alpha}{\hat{\beta}}\times\ln\left(p\right)\right\rangle$ is estimated from the negative sum of the appropriate look-up table values.

In order to replace ϖ_1 by ϖ_2 with probability $\min(p,1)$, the value of p needs to be compared to a random number of the form $r \in (0,1]$. In practice, the value of $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ was compared to a random number of the form $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(r) \right\rangle$, where r is a random number such that $r \in (0,1]$. A table of such random numbers was created and stored ahead of time for each value of β in the annealing schedule.

The value of $\alpha C\left(\varpi;\hat{j}\right)$ is increased by $\left\langle \frac{\alpha}{\hat{\beta}} \times \ln(p) \right\rangle$ when a color change at pixel $\hat{h}$ is accepted, in other words whenever ϖ_1 is replaced by ϖ_2. The value of $\alpha C\left(\varpi;\hat{j}\right)$ was tracked; the image $\hat{\omega}$ for which the value of $\alpha C\left(\varpi;\hat{j}\right)$ was highest was saved along with the value of $\alpha C\left(\varpi;\hat{j}\right)$. At the end of each $\hat{\beta}$ in the annealing schedule, the column for which the value of $\alpha C\left(\varpi;\hat{j}\right)$ was highest is used as an initialization for the next $\hat{\beta}$ in the annealing schedule.

3.5 Step 3: Clean Grayscale Column Estimation

In the previous section, we described how to estimate the binary column from the noisy grayscale spectrogram column. We now describe Step 3 of our methodology, in which we take the estimated binary column and the noisy grayscale spectrogram column to estimate the clean grayscale spectrogram column that corresponds to the unknown clean speech signal.

We first describe how to "fuzzify" the estimated binary columns to create a $[0,1]$-valued column. We refer to this $[0,1]$-valued column as the **soft segmentation column** of the noisy spectrogram column. Similarly to the terminology of the previous section, we refer to the collection of soft segmentation columns as the **soft segmentation image**, or simply as the soft segmentation.

We "fuzzify" the hard segmentation column $\varpi[h]$ for $h \in H_{\hat{j}}$, given the noisy spectrogram column $\hat{\theta}[h]$ for $h \in H_{\hat{j}}$, in the following way. For $h \in H_{\hat{j}}$, let $\hat{q}[h] = \text{bin}\left(\hat{\theta}[h]; b[h]\right)$. We set the value of the soft segmentation column at pixel h to be

a)

b)

Figure 3.9: Soft segmentation images of the noisy spectrogram with a) 5 dB SNR, and b) 0 dB SNR.

the band-dependent posterior probability that the hard segmentation column (binary column) value at pixel h is 1, given that $\text{bin}(\theta[h];b[h]) = \hat{q}[h]$ and the binary values in the neighborhood configuration of the pixel h are as in $N_{\varpi}[h]$. Symbolically, each pixel of the soft segmentation column is set to

$$p(\omega[h] = 1 | \text{bin}[\theta[h];b[h]] = \hat{q}[h], N_{\varpi}[h];b[h]). \tag{3.56}$$

The soft segmentation column value of pixel h in row $i = 0$ was set to zero. The meaning of such a column is that it locally estimates the conditional probability that a time-frequency location in the spectrogram (a pixel) contains clean speech.

In the implementation, the posterior column values are set using one look-up table value from the float table described in Equation 3.49. Figure 3.9 displays soft segmentation images for 5 and 0 dB SNR. Notice that these images are indeed smoother than the hard segmentations displayed in Figure 3.8.

Many other more simplistic "fuzzification" methods were considered but not tested. The "fuzzification" described here was more specific to the problem at hand and seemed to provide results that were better than expected with the simple "fuzzification" techniques.

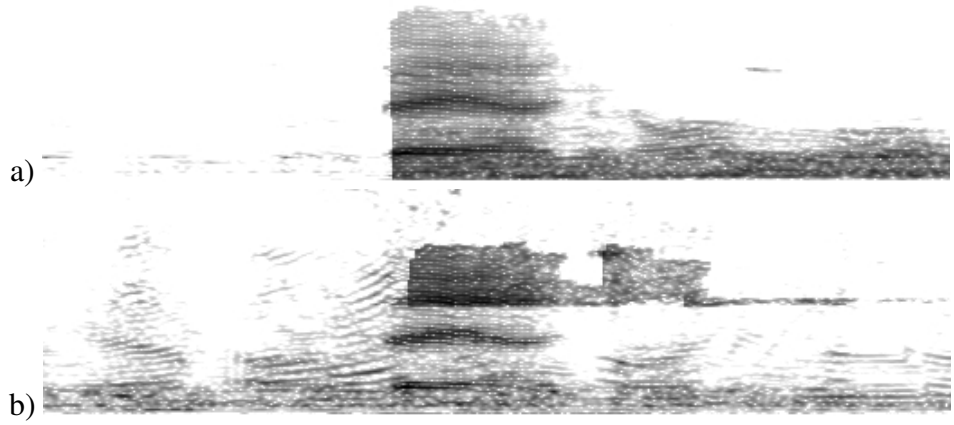


Figure 3.10: Estimated clean spectrogram of the noisy spectrogram with a) 5 dB SNR, and b) 0 dB SNR.

Next, we estimated the clean grayscale column that represents the unknown clean speech segment by multiplying the noisy grayscale spectrogram column by the soft segmentation column, pixel by pixel. This can be considered as using the soft segmentation column to mask the noisy grayscale spectrogram column. Figure 3.10 shows the estimated clean spectrograms with 5 and 0 dB SNR. Since the soft segmentation column is a $[0, 1]$-valued column, the effect of the multiplication of the soft segmentation column by the noisy spectrogram column is that the spectrogram values are dampened.

Notice that areas that were set to white in the hard segmentations (speech is absent), see Figure 3.8, did not necessarily remain white in the soft segmentation (zero probability of clean speech). That way areas in which the hard segmentation may have made the "wrong" decision, the noisy speech signal is not completely eliminated in Figure 3.10. Furthermore, areas which contained high values in the noisy spectrograms, see Figure 3.4, were preserved with a lower value. This makes sense and is a desired behavior.

3.6 Step 4: Image to Signal Transform

The fourth and final step of our methodology is to estimate a segment of the unknown clean speech signal from the already estimated clean spectrogram columns described in Section 3.5. This necessitates an attempt to inverse-transform the spectrogram, the chosen signal to image transform in Subsection 3.1.5. The STFT is an invertible transform; its inverse can be calculated based on the description in Subsection 3.1.4. The spectrogram is not invertible because the speech phases/angles were lost when the norm of the DFT value was squared. Furthermore, the estimated clean speech spectrogram is a modified spectrogram and may not be a spectrogram of any real signal. We must therefore estimate the speech signal that gives rise to the "closest" spectrogram to the given estimated clean speech spectrogram.

We define a distance measure on the norms of the square roots of the spectrograms, or in other words on the the norms of STFT images, rather than on the spectrograms directly [57]. If $A[i,j]$ and $B[i,j]$ are two $I \times J$-dimensional complex-valued images, then the distance between them is defined to be

$$\text{Dist}[A,B] = \sum_{i=1}^{I-1} \sum_{j=1}^{J-1} \left[|A[i,j]| - |B[i,j]| \right]^2. \tag{3.57}$$

Let $Y[i,j]$ denote the positive square root (pixel-wise) of the estimated clean spectrogram, in other words the estimated norm of the clean STFT. Let $x_e[n]$ be the estimated clean speech signal we are searching for and let $X_e[i,j]$ be the STFT of $x_e[n]$. Given $Y[i,j]$, we are searching for the estimated clean speech signal $x_e[n]$ that would minimize the distance $\text{Dist}[Y, X_e]$, see Figure 3.11.

In [57], an iterative algorithm for searching for such a signal is described. That algorithm assumes that the modified STFT image of the whole speech signal is available at the time of the signal estimation. In our case, we wish to estimate clean speech segments in "real-time," as the columns are revealed; we cannot assume

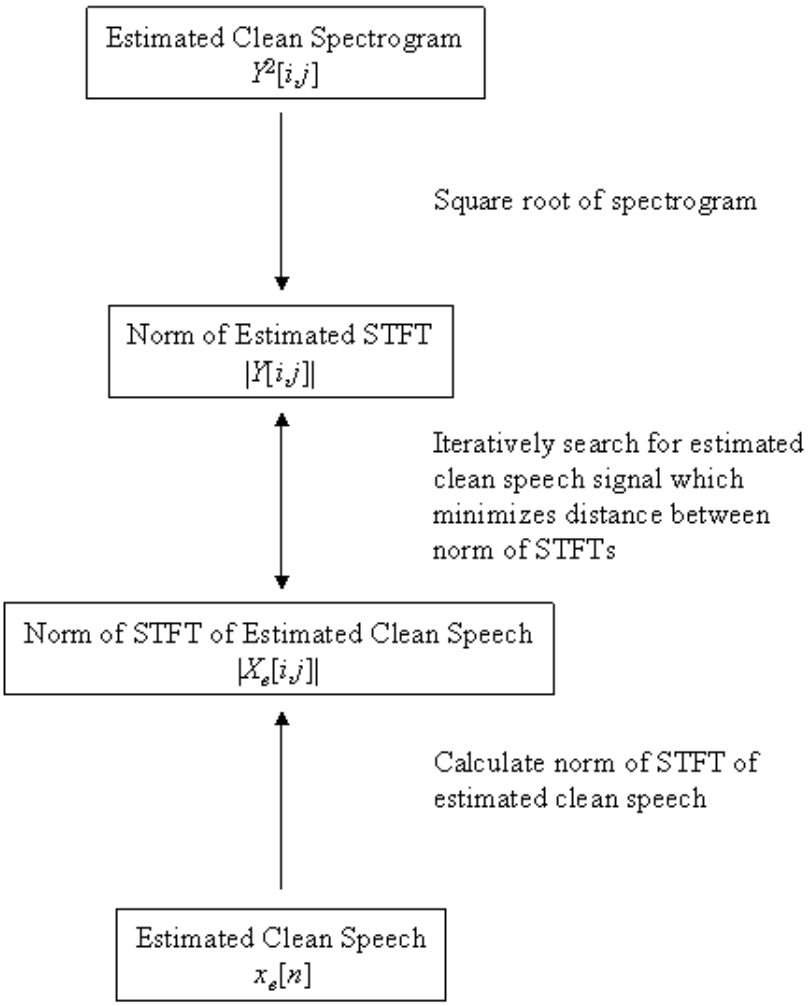


Figure 3.11: Schematic of search for the estimated clean speech signal from the estimated clean speech spectrogram.

that the whole image is available during the signal estimation. Next, we detail our modification to the iterative algorithm described in [57]. This "real-time" version of the signal estimation algorithm from a modified STFT still necessitates a short delay in signal estimation. Our modified algorithm contains an inner iterative loop (corresponding to Quatieri's iterative loop in [57]) within an outer iterative loop (a loop over the columns).

As a column of the clean spectrogram is estimated, we must estimate the clean speech signal that corresponds to that column. We describe this process as the column number $\hat{j}$ increases from 0 to $J-d$.

We initialized the estimated clean speech signal by

$$x^{(0)}[n] = \begin{cases} 0, & \text{for } 0 \leq n \leq 2I-3-s, \\ \text{noisy signal value}, & \text{for } 2I-3-s < n < L, \end{cases} \tag{3.58}$$

where s is the time skip step as defined in Subsections 3.1.4 and 3.1.5. The initial segment of $2I-2-s$ values was initialized to contain zeros and remains fixed during the updating process. This means, in our case, that there is an assumed 21.7 ms of silence at the beginning of our estimated clean speech signal. This is acceptable since the actual clean speech signal, from which the noisy speech signal arises, is padded with 500 ms of silence at its beginning. The remaining values were initialized to the noisy speech signal values, which is the best initial estimate of the clean speech signal.

In each outer iterative step (loop running over columns), a segment of $2I-2$ consecutive estimated clean speech values were updated. To be precise, for column $0 \leq \hat{j} \leq J-d$, we updated the values of $x_e^{(\hat{j})}[n]$ at $n = 2I-3+js-a$ where $\hat{j} \leq j < \hat{j}+d$ and $0 \leq a < s$. (Notice that this corresponds to updating exactly $ds = 2I-2$ values.) The details of this updating process are given below. After all this is done, we set $x_e^{(J-d+1)}[n]$ to 0 for $2I-3+(J-d)s \leq n < L$. This means in our case that we

assumed a 21.7-24.8 ms of silence at the end of our estimated clean speech signal. This is acceptable since the actual clean speech signal, from which the noisy speech signal arises, is padded with a 500 ms of silence at the end.

A single updating step of the signal $x_e^{(\hat{j})}[n]$ at $n = 2I - 3 + js - a$ where $\hat{j} \leq j < \hat{j} + d$ and $0 \leq a < s$ was done by performing an inner iterative loop. The result of the inner loop was a sequence $x_e^{(\hat{j},0)}, x_e^{(\hat{j},1)}, \ldots, x_e^{(\hat{j},K)}$ of $K+1$ signals of length L. The signal $x_e^{(\hat{j},0)}$ was initialized to be the signal $x_e^{(\hat{j})}$. In the iterative step from $x_e^{(\hat{j},k)}$ to $x_e^{(\hat{j},k+1)}$ exactly $ds = 2I - 2$ values of $x_e^{(\hat{j},k)}$ were changed. The values of $x_e^{(\hat{j},k)}$ that change are those for which the indices are $n = 2I - 3 + js - a$ where $\hat{j} \leq j < \hat{j} + d$ and $0 \leq a < s$; the indices that change in each inner loop are iteration independent. In other words, exactly the same ds values are updated during each inner iteration and these are the values that are updated in the outer loop when the inner loop completes. The changes are done according to a modification of the iterative algorithm outlined in [57] for finding a minimizing signal.

The estimation of the speech signal in this inner iterative step makes use of the d columns of $|Y[\bullet, j]|$ where $\hat{j} \leq j < \hat{j} + d$, see Figure 3.12. This need for an additional (future) $d-1$ columns of $|Y[\bullet, j]|$ where $\hat{j} < j < \hat{j} + d$ means that a delay of size $(d-1)s$, which corresponds to 21.7 ms in our case, is required. A delay of less than 20-30 ms is considered acceptable by hearing aid users [36, 66, 68], for example in terms of synchronization with lip reading. This has not been agreed upon in the literature [3, 65, 67], for example in terms of effect on speech production.

To be exact, for iteration $0 \leq k < K$, we define $X_e^{(\hat{j},k)}$ to be an image containing d columns. Symbolically for rows $0 \leq i < I$, columns $\hat{j} \leq j < \hat{j} + d$ and iteration $0 \leq k < K$,

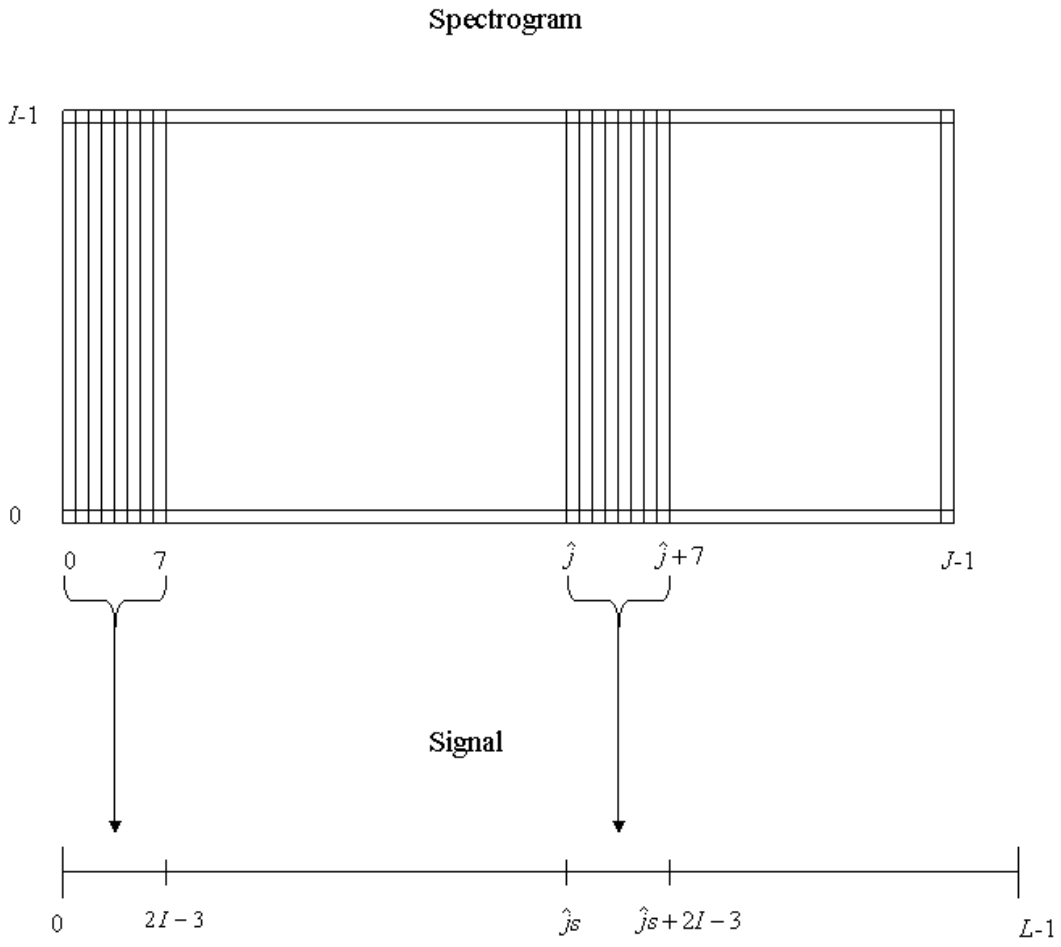


Figure 3.12: This figure displays how a segment of clean speech is estimated from d (in our case 8) columns, the current one and additional $d-1$ future columns.

$$X_e^{(\hat{j},k)}[i,j] = \begin{cases} \left[\mathscr{D}^w_{L,I,2I-3+js} x_e^{(0)}\right][0], & \text{if } i = 0, \\ \left[\mathscr{D}^w_{L,I,2I-3+js} x_e^{(\hat{j},k)}\right][i], & \text{otherwise.} \end{cases} \tag{3.59}$$

We refer to $X_e^{(\hat{j},k)}$ as the **modified STFT image** of signal $x_e^{(\hat{j},k)}$. Next for rows $0 \leq i < I$ and columns $\hat{j} \leq j < \hat{j}+d$, we calculated

$$Y_e^{(\hat{j},k)}[i,j] = \begin{cases} 0, & \text{if } \left|X^{(\hat{j},k)}[i,j]\right| = 0, \\ |Y[i,j]| \times \dfrac{X_e^{(\hat{j},k)}[i,j]}{\left|X_e^{(\hat{j},k)}[i,j]\right|}, & \text{otherwise.} \end{cases} \tag{3.60}$$

This is an image whose pixel norms are the same as $|Y[i,j]|$ and whose phases/angles were estimated from the current modified STFT image. Since the values of $Y[0,j]$ were zeros, the values of $Y_e^{(\hat{j},k)}[0,j]$ were all set to zeros.

Let $E_j^{(\hat{j},k)} = \left[\mathscr{E}_I Y_e^{(\hat{j},k)}\right][\bullet, j]$ (i.e., the transform $\mathscr{E}_I$ applied to the sequence $Y_e^{(\hat{j},k)}[0,j], Y_e^{(\hat{j},k)}[1,j], \ldots, Y_e^{(\hat{j},k)}[I-1,j]$). We calculated the d vectors $E_j^{(\hat{j},k)}$ where $\hat{j} \leq j < \hat{j}+d$. Let

$$B = \left\lfloor \frac{n - \left(\hat{j}-1\right)s - (2I-2)}{s} \right\rfloor. \tag{3.61}$$

Then for $n = 2I-3+js-a$ where $\hat{j} \leq j < \hat{j}+d$ and $0 \leq a < s$, we estimated $x_e^{(\hat{j},k+1)}[n]$ from the d vectors $E_j^{(\hat{j},k)}$ using the following formula:

$$x_e^{(\hat{j},k+1)}[n] = \frac{\sum_{b=B}^{d-1} w\left[2I-3-n+\left(\hat{j}+b\right)s\right] \times E_{\hat{j}+b}^{(\hat{j},k)}\left[2I-3-n+\left(\hat{j}+b\right)s\right]}{\sum_{b=B}^{d-1}\left(w\left[2I-3-n+\left(\hat{j}+b\right)s\right]\right)^2}. \tag{3.62}$$

For all other n, we set the value of $x_e^{(\hat{j},k+1)}[n]$ to $x_e^{(\hat{j},k)}[n]$. Notice that only $ds = 2I-2$ values of $x_e^{(\hat{j},k)}[n]$ were updated and all other values remained the same. At the end of this inner loop, the signal $x_e^{(\hat{j}+1)}$ is set to signal $x_e^{(\hat{j},K)}$.

In the calculation of $x^{(\hat{j},k+1)}[n]$ for $n = 2I-3+js-a$ where $\hat{j} \leq j < \hat{j}+d$

and $0 \leq a < s$, the numerator and denominator both sum from $b = B$ to $b = d-1$. This complex indexing insures that the values of $x_e^{\left(\hat{j},k+1\right)}[n]$ are estimated from the correct number of vectors of $E_j^{\left(\hat{j},k\right)}$. For example the values of $x_e^{\left(\hat{j},k+1\right)}[n]$ for $n = 2I-3+\hat{j}s-a$ (when $j = \hat{j}$) and $0 \leq a < s$ were estimated from all d columns of $E_j^{\left(\hat{j},k\right)}$ where $\hat{j} \leq j < \hat{j}+d$ (this corresponds to $b = 0$ to $b = d-1$), while the values of $x^{\left(\hat{j},k+1\right)}[n]$ for $n = 2I-3+\left(\hat{j}+1\right)s-a$ (when $j = \hat{j}+1$) and $0 \leq a < s$ were estimated only from the $d-1$ columns of $E_j^{\left(\hat{j},k\right)}$ where $\hat{j}+1 \leq j < \hat{j}+d$ (this corresponds to $b = 1$ to $b = d-1$), and so forth. The reason for this indexing is that for example, the values of $x^{\left(\hat{j},k+1\right)}[n]$ for $n = 2I-3+\left(\hat{j}+1\right)s-a$ and $0 \leq a < s$ do not depend on the vector $E_{\hat{j}}^{\left(\hat{j},k\right)}$ (that corresponds to $b = 0$).

We did a preliminary experiment to search for the number of iterations K to be used in the inner iterative process. The experiment consisted of two parts. First, we estimated the clean speech signals from the estimated clean spectrograms with several values of K for one word. We then looked at the spectrograms of the estimated clean speech signals. Ideally, these two spectrograms (the estimated clean speech spectrogram and the spectrogram of the estimated clean speech) should be the same. We found that the spectrogram of the estimated clean speech signal with $K = 20$ iterations looked closest to the estimated clean spectrogram. Figure 3.13 illustrates the spectrograms of the estimated clean speech with $K = 5, 10, 15, 20$ from an estimated clean spectrogram (which was estimated from the noisy speech spectrogram with 5 dB SNR).

Next, we chose a column for which the estimated clean spectrogram and the spectrogram of the estimated clean speech signal visually differed. The column we chose to look at was column 203 for the word "bat" (not the word displayed elsewhere in this text) for the 5 dB SNR noise level. We looked at the sub-sequence of $2I-2 = 248$ signal values from which column 203 of the spectrogram was calculated. We tracked the changes to that sub-sequence during the signal estimation

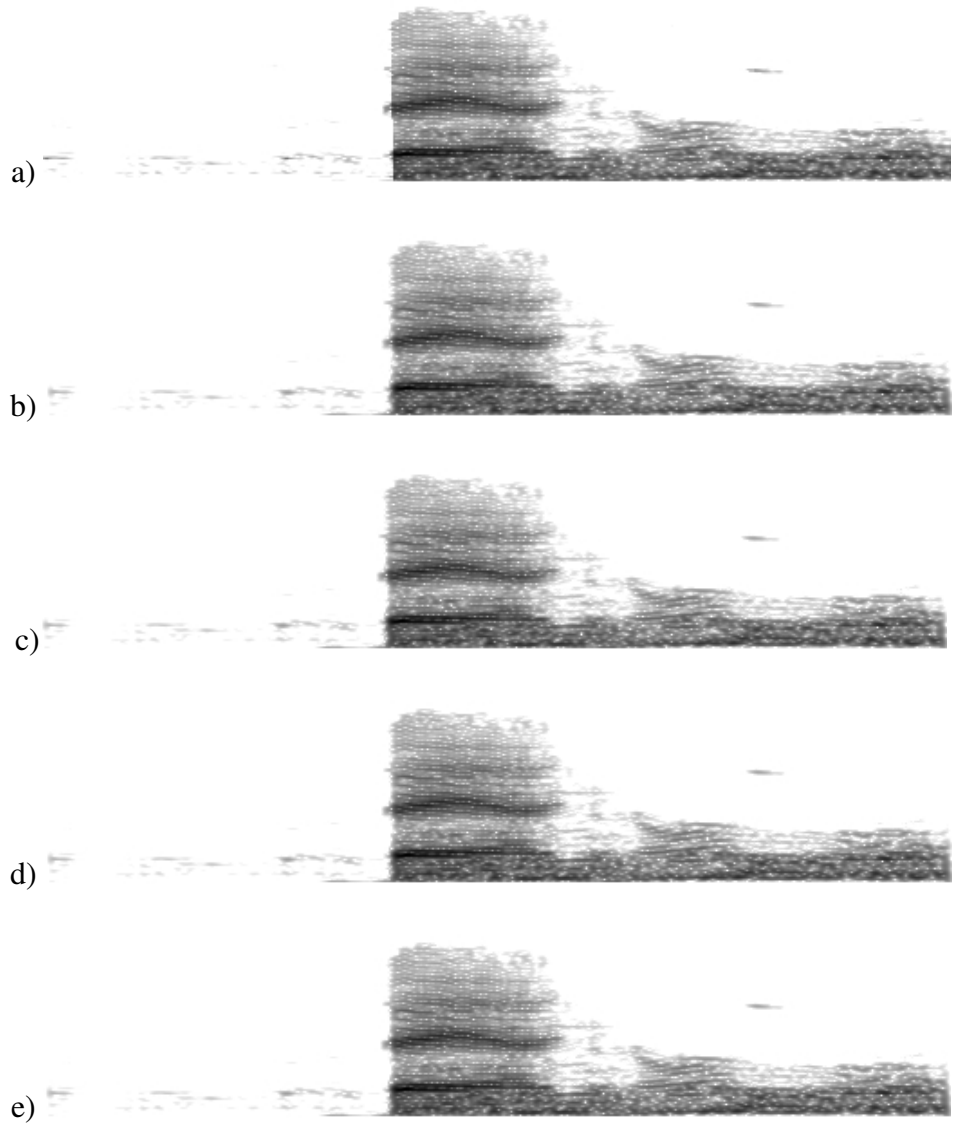


Figure 3.13: a) Estimated clean spectrogram from the noisy spectrogram with 5 dB SNR. b) Spectrogram of estimated clean speech with $K = 5$ iterations. c) Spectrogram of estimated clean speech with $K = 10$ iterations. d) Spectrogram of estimated clean speech with $K = 15$ iterations. e) Spectrogram of estimated clean speech with $K = 20$ iterations.

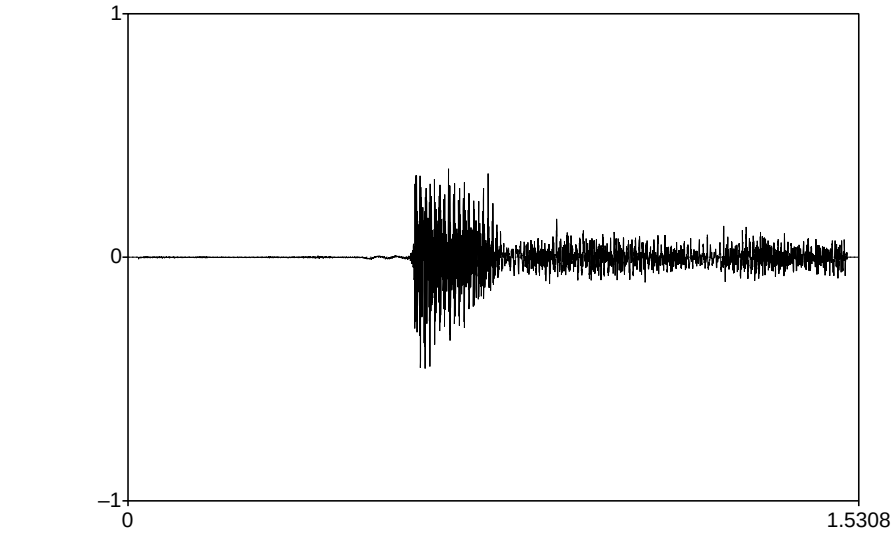
process, which changes a total of $(2d-1)K$ times. We calculated the absolute value of the differences in this sub-sequence value changes. We further calculated the average of these differences. We would expect the average of the differences to be monotonically decreasing as the signal changes during the estimation process. For $K = 20$ iterations, this average of differences decreased monotonically in all iterations but for column $j = \hat{j} + d - 1$, the last outer loop iteration. We therefore chose to use $K = 20$. Figure 3.14 displays the waveforms of the estimated clean speech from the noisy speech signals with 5 and 0 dB SNR.

To test the accuracy of this software, we performed the following preliminary test. We estimated the speech from a clean spectrogram (in contrast to the modified spectrograms available as an input to the software in real testing). We then created the spectrogram of the estimate speech. The original clean spectrogram and the spectrogram of the estimated speech were indeed very close. In fact, the spectrogram images were identical.

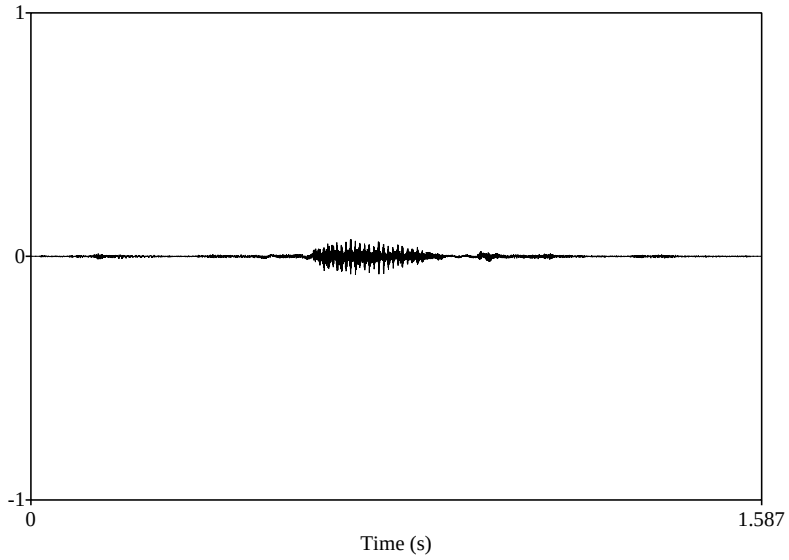
3.7 Timing

The timings reported in this section were performed on a Lenovo ThinkPad x60s with Intel Centrino Duo, 1.66GHz each, 1GB of RAM and running on Linux. The programs were compiled with GNU gcc (version 4.1.2), with the highest level of optimization ("-O3"). We report on the user CPU time as reported by the Linux "time" command. Table 3.1 displays a summary of the timings described in more detail below.

The time to create our training sets from the clean signals, which needs to be only performed once, is as follows. It took 4.440 s to pre-process the signals in all 5 data-sets, as described in Section 2.1. It took an additional 2.459 s to add the 500 ms of silence to the beginning and end of each of the pre-processed signals in all



a)



b)

Figure 3.14: The waveforms of the estimated clean speech signals from the noisy speech signals with a) 5 dB SNR, and b) 0 dB SNR.

Event	Time (s)	Note
Create Training Set:		performed once
Pre-processed signals	4.440	for all 5 data-sets
Create noise signal	0.289	for each SNR level
Create noisy speech signals	10.990	for all 5 data-sets and both SNR levels
Create clean spectrogram	0.101	for each of the 5 data-sets
Create clean binary image	2.051	for all 4 training data-sets
Estimate Models:		performed once
Estimate noise model	0.933	for each SNR level
Estimate binary model and posterior probability	0.165	for each SNR level
Estimate clean speech:		performed for each testing signal
Create noisy spectrogram	0.101	
Estimate hard and soft segmentations	236.555	
Estimate clean spectrogram	0.004	
Estimate clean speech	0.225	

Table 3.1: A table summarizing the timings of the different steps.

5 data-sets, as further described in Section 2.1. It took 0.289 s to create each of the two noise signals (one for each SNR level), as described in Section 2.2. It took 10.990 s to create the noisy speech signals by adding the clean speech signals to the noise signal segments for both SNR levels and all 5 data-sets, as further described in Section 2.2. It took 0.101 s to create the spectrogram (and simultaneously the spectrogram image) of each clean speech signal, as described in Section 3.1. The clean binary images were created by thresholding the clean spectrogram images in the four training data-sets, as described in Section 2.3, which took 2.051 s. This concludes the steps required to creating a training set.

The time to estimate the prior and noise models for each SNR level, which needs to be only performed once, is as followed. It took 0.933 s to estimate the noisy information probability for each SNR level from the training set, as described in Section 3.2. It took 0.165 s to estimate the posterior probability for each SNR level from the training set, as described in Section 3.3.

The time to estimate the clean speech signal from each noisy speech signal is as follows. It took 0.101 s to create the spectrogram of each noisy speech signal, as described in Section 3.1. It took 236.555 s (almost 4 minutes) to estimate simultaneously the hard segmentation and soft segmentation of each noisy speech spectrogram, as describe in Sections 3.4 and 3.5. This is clearly the bottle-neck of our methodology; it takes 3 orders of magnitude longer than any other step in our methodology. It took 0.004 s to estimate the clean spectrogram using the soft segmentation and the noisy spectrogram, as further described in Section 3.5. It took 0.225 s to estimate the clean speech signal from the estimated clean spectrogram, as described in Section 3.6. Notice that these are timings for processing one whole signal on the order of about 1.5 s.

For testing purposes, the estimated clean speech signals were post-processed. It took 0.944 s to post-process all the testing signals estimated by our methodology, an equivalent amount of time to post-process all the testing signals estimated by the gold standard methodology, and an equivalent amount of time to post-process all the noisy testing signals. For display purposes, the spectrogram images of the post-processed estimated clean speech signals by our methodology and the gold standard methodology were created, as well as the spectrogram images of the post-processed noisy testing signals. It took 0.101 s to create the spectrogram and spectrogram image of each such signal, as described in Section 3.1.

Chapter 4

Experiment and Results

In this chapter, we describe our experiment and results. Section 4.1 describes the purpose of the experiment we carried out. Section 4.2 describes the experimental design. Section 4.3 describes the instrumentation and Section 4.4 describes the subjects used in the experiment. Section 4.5 describes the statistical analysis that was carried out and Section 4.6 describes the results.

4.1 Experimental Purpose

This work describes a way of processing noisy images in the attempt to recover the unknown clean image in "real-time." We chose to test the use of such "real-time" image processing for estimating the unknown clean speech signals from noisy speech signals in "real-time" for the hearing aid application, where the motivation is to increase speech recognition performance. We chose to test the success of our processed signals as compared to signals processed by a gold standard methodology as well as the unprocessed noisy speech signals.

Multi-Band Spectral Subtraction (MBSS) [40, 41] was the agreed upon gold standard methodology for signal processing against which we compared our methodology. MBSS is a frequency-dependent speech enhancement methodology based on

spectral subtraction. In MBSS, speech is processed into frequency bands and spectral subtraction is performed independently on each band using band-specific over-subtraction factors. This method provides a great degree of flexibility and control on the noise subtraction levels that reduces artifacts in the enhanced speech, resulting in improved speech quality over conventional spectral subtraction. Results showed that the MBSS method with four linear-spaced frequency bands outperformed the conventional spectral subtraction method with respect to speech quality and reduced musical noise. In [38, 46] eight algorithms were tested for the improvement of recognition performance achieved by the estimated clean speech signals from noisy speech signals with 5 and 0 dB SNR. For noisy signals degraded by additive babble noise at the 5 dB SNR, MBSS performed equally well as several other algorithms and outperformed the rest of the methodologies. MBSS performed equally well as the noisy speech signal for both 5 and 0 dB SNR additive babble noise. That means that the best algorithms currently known do not increase recognition performance in the 5 and 0 dB SNR levels, and MBSS does not decrease recognition. An additional reason for choosing MBSS as the gold standard was that we were able to obtain the code for this methodology and would not be responsible for its implementation (and its success or lack of success).

The code for MBSS was made available in Matlab on the DVD attached to [46], which is what we used to create the estimated clean speech signals using the MBSS methodology. There were several parameters which had to be set when running the MBSS Matlab code. Below is a description and brief reasoning for the chosen values. The value of "AVRGING" was set to "yes," which means that pre-processing is performed; this is what is suggested in [40, 46]. The value of "FRMSZ", the frame length in ms (window size in our terminology), was set to 24.8 as in our methodology; in [40, 46] the value of 20 was used instead. The value of "OVLP," which controls the percent of overlap of adjacent frames, was set to

12.5, which is the value that corresponds to the choices made in our methodology; in [40, 46] the value used was 50. The value of "Noisefr", the number of noise frames at the beginning of the file to be used for noise spectrum estimation, was set to 20. This value corresponds to the assumption of 496 ms of noise at the beginning of the signal, which is reasonable since in our work the noisy speech signals indeed begin with 500 ms of noise. In [40, 46] the value 6 was used, which with FRMSZ set to 20 corresponds to the assumption that there exist only 120 ms of noise at the beginning of the signal. The value of "FLOOR," the spectral floor, was set to 0.002 as recommended in [40, 46]. The value of "VAD," which controls the use of a voice activity detector, was set to "yes" as suggested in [40, 46]. The banding that was chosen was "linear," since it was the only one tested in [40, 46] for recognition. We used 5 bands as in our work, instead of the 4 bands used in [40, 46].

4.2 Experimental Design

We decided to test our methodology at 5 and 0 dB SNR. These values give a good indication of how our methodology performs at the lower end of the normal SNR level range, see Section 1.2. These are also values for which there is a great need for successful signal processing since differentiation between speech and noise for a person with hearing loss becomes more difficult [43] and since current algorithms do not perform well in this range [38, 46]. These SNR values were also chosen in order to avoid having recognition performance values that were close to 100 percent, and thus have no room for improvement. In preliminary work, we found that for 10 dB SNR case, subjects were able to recognize most of the unprocessed noisy speech signals. For that reason, we chose to work with smaller SNR values for which it was no longer the case that all the unprocessed noisy speech signals were recognizable.

The Modified Rhyme Test (MRT) [37, 47] was used as the test material. The test consists of six equivalent lists of 50 "rhyming" monosyllabic words (consonant-vowel-consonant). The 300 words in the test are made up of 50 **foil sets**. Each set contains 6 words differing by one consonant, either the initial or final position. For example, the foil set {vest, test, rest, best, west, nest} differ in the initial position and the foil set {pat, pad, pan, path, pack, pass} differ in the final position. Of the 50 foil sets, 25 of the foil sets contain words that are confusable in the initial location and the other 25 foil sets contain words that are confusable in the final location. Appendix E contains a table of the MRT words. In Table E.1, the six list are placed in the six columns and the 50 foil sets are placed in the 50 rows.

Since the MRT consists of six equivalent word lists, only six conditions can be compared on one group of subjects without repeating lists. We decided to compare three **processing levels**: the unprocessed noisy speech signals, the estimated clean speech signals using MBSS and the estimated clean speech signals using our methodology. We refer to these as the Unprocessed, the MBSS Processed and Our Processed signals, respectively, for short. Due to the limitation imposed by the six conditions to be tested and the decision to run the experiment on only one group of subjects, these three processing levels were tested at only two **SNR levels**: 5 and 0 dB SNR. This is the reason for the restriction of using only two SNR levels.

There are six **processing-SNR conditions** (processing-SNR pairs): Unprocessed-5 dB SNR, MBSS Processed-5 dB SNR, Our Processed-5 dB SNR, Unprocessed-0 dB SNR, MBSS Processed-0 dB SNR and Our Processed-0 dB SNR. For the remainder part of this section, we will simply refer to the processing-SNR conditions as conditions. For example, the condition "Our Processed-5 dB SNR" corresponds to the estimated clean speech signal with our methodology from the noisy speech signal with 5 dB SNR.

The six lists were paired with the six conditions; this pairing was used for all

subjects. Each subject heard all 300 words in the following way. Each of the six conditions was chosen exactly once in random order. The subject was prompted that a new condition was being tested (without a description of the condition). The subject heard the 50 words for that condition in random order. After hearing each word, a visual appeared on the screen. The visual displayed in random order the six words in the foil set of the word that was played. The subject was asked to use the mouse to chose the word he/she heard. Once a condition/list was completed, the subject was prompted that a new condition was being tested and then heard the words in the corresponding list.

The first 5 randomly selected words from each list were used to familiarize the subject with the condition and the task. Feedback was given to the subjects during the familiarization process, but was not provided for the remaining 45 words in the list. As a result of this familiarization process, performance was measured using only 270 of the 300 MRT words. The test took no more than one hour. The subjects were allowed to take breaks during the experiment in order to reduce fatigue.

All subjects heard the same condition-list pairing. In this way, each of the subjects listened to and evaluated both SNR levels in very similar circumstances - no talker influence, which also means no gender influence, and the same condition-list pairing. Other things were kept random from subject to subject, so that they did not play a role - the conditions were presented in random order, the order of the words played within the condition was random, the words used for familiarization were chosen randomly from the condition list, and the text displayed the foil set words in randomized order. This experiment is clearly a preliminary one, in that it tested the success of our algorithm based on only one gender, one talker, 2 SNR levels, one condition-list pairing, and on one estimated signal using our methodology for each noisy speech signal.

The experiment was approved by the Institutional Review Board (IRB) of The

Graduate Center, City University of New York.

4.3 Instrumentation

The experiment was performed in a double-walled sound treated booth (Industrial Acoustics 120A-1), which offers a "high" level of external sound attenuation [1]. The signal files were stored on a Dell Precision 530 desktop computer and played through a SB Live! Wave Device. The output of the computer was sent to TDH-50P audiometric headphones. The "Telephonics' TDH-series earphones are the acoustic reference standard for the audiometric industry and are widely used in hearing tests to deliver precision sound replication" [2]. The output of the headphones was set to 70 dB sound pressure level (SPL), as measured in a 6 cm^3 coupler.

4.4 Subjects

Twenty-four normal hearing native English speakers between the ages of 21 and 50 years served as subjects. To test for normal hearing, potential subjects had their hearing screened using a screening level of 20 dB Hearing Level (HL) [4] at octave frequencies between 250 and 4000 Hz. The dB HL scale is used on audiometers; the reference level for this dB scale is the softest sound audible by people with normal hearing at each test frequency. The subjects were reimbursed for their participation in the experiment.

4.5 Method of Analysis

For each subject, the **figure of merit** (FOM) of a given condition is the proportion (a real number between 0 and 1, inclusive) of correctly identified MRT words. The arcsine transform [52] of the FOM was used in order to stabilize the error variance

before performing a three-way repeated measures analysis of variance (ANOVA) [16]. The three ANOVA factors are processing (Unprocessed, MBSS Processed and Our Processed), SNR (5 and 0 dB), and consonant test location (Initial and Final). In the remaining part of this section, we will simply refer to these processing-SNR-location conditions as simply conditions.

Let μ_x be the mean of the arcsine transform of the FOM for condition x over all 24 subjects. Our null hypotheses were: 1) $\mu_{\text{Unprocessed}} = \mu_{\text{MBSS Processed}} = \mu_{\text{Our Processed}}$, 2) $\mu_{\text{5 dB SNR}} = \mu_{\text{0 dB SNR}}$, 3) $\mu_{\text{Intial}} = \mu_{\text{Final}}$, and 4) the effects of one factor do not interact in any way with the the other factors. The alternative hypotheses are that the null hypotheses are not true. We used a three-way repeated measures ANOVA with an alpha level of 0.05 to test all hypotheses.

Tukey's Honestly Significant Difference (HSD) procedure [16] at the 0.05 level was used for post-hoc testing. Tukey's HSD procedure produces the critical value at which all pairs of groups must be tested in order to maintain the overall 0.05 (in our case) level. We carried out these post-hoc tests on all significant effects that related to the processing level. The post-hoc tests were performed on the mean of the arcsine transform values of the FOM over the 24 subjects.

4.6 Results

In this section, we describe the results. Subsection 4.6.1 shows the signal (waveforms and spectrograms) of the Unprocessed, the MBSS Processed and Our Processed signals. Subsection 4.6.2 describes the statistical results of the experiment we conducted. Subsection 4.6.3 details the words for which our processing failed (problematic words) and succeeded.

4.6.1 Signal Results

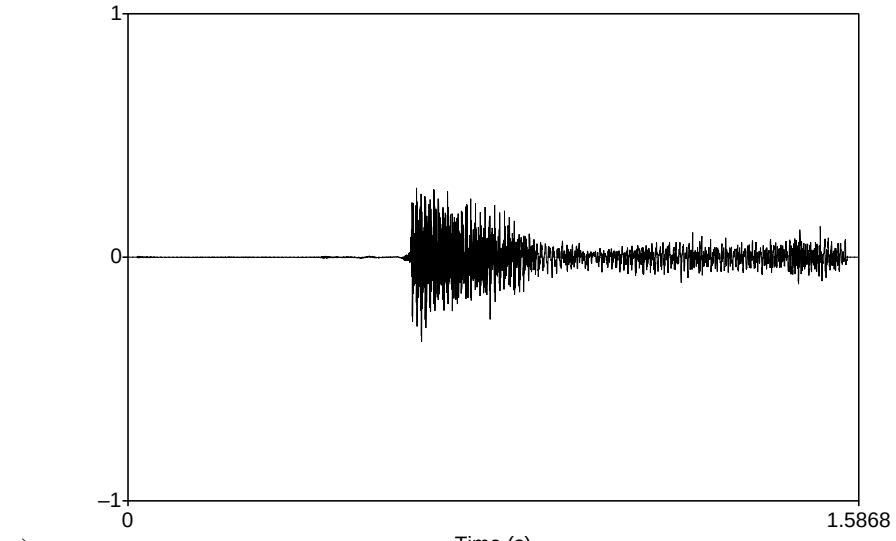
For testing purposes, all signals - Unprocessed, MBSS Processed and Our Processed (as defined in Section 4.2) at both SNR levels - were post-processed. Specifically, we scaled each signal so that the average dB rms value of each signal was the same. We scaled all signals to an average dB rms value of 62 since this was the highest value for which no clipping occurred during this post-processing step. The scaling was achieved by multiplying the signal values by

$$10^{(62-\text{average signal dB rms value})/20}. \tag{4.1}$$

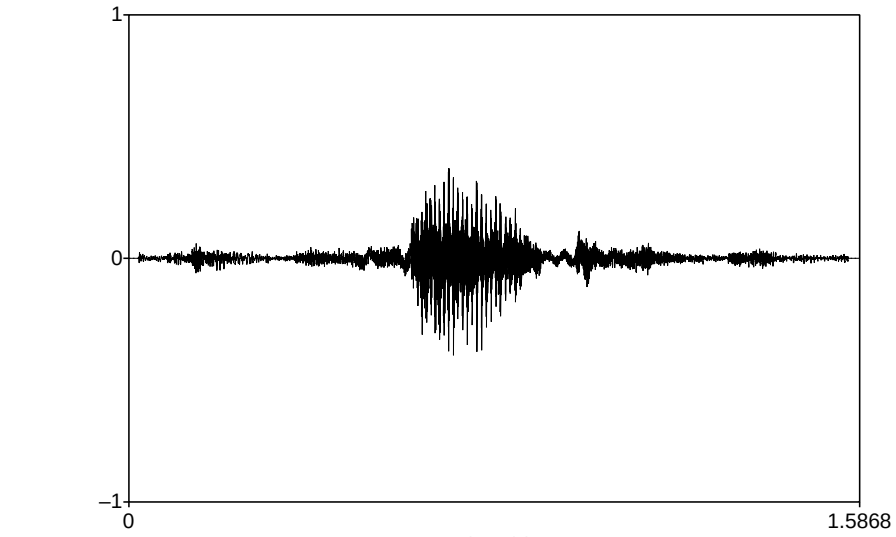
This meant that during experimentation, all signals were presented at the same volume.

Figure 4.1 shows the waveforms of the scaled Our Processed signals for 5 and 0 dB SNR. Figure 4.2 shows the waveforms of the scaled MBSS Processed signals for 5 and 0 dB SNR. For comparison, Figure 4.3 shows the waveforms of the scaled Unprocessed signals for 5 and 0 dB SNR. Figure 4.4 displays the spectrogram images of the scaled Our Processed signals for 5 and 0 dB SNR. Figure 4.5 displays the spectrogram images of the scaled MBSS Processed signals for 5 and 0 dB SNR. For comparison, Figure 4.6 displays the spectrogram images of the scaled Unprocessed signals for 5 and 0 dB SNR.

As can be seen by Figures 4.1 and 4.4, Our Processed signals have different behavior in the first 500 ms, in the middle section in which the noisy signal contained both speech and noise, and in the last 500 ms. In the first 500 ms, Our Processed signals seem to contain substantially less noise than the MBSS Processed signals (see Figures 4.2 and 4.5) that contain less noise than in the Unprocessed signals (see Figures 4.3 and 4.6). In the middle section, Our Processed signals still seem to contain less noise than the MBSS Processed signals. In the last 500 ms, Our Processed



a)



b)

Figure 4.1: The waveforms of the scaled Our Processed signals from the noisy speech signals with a) 5 dB SNR, and b) 0 dB SNR.

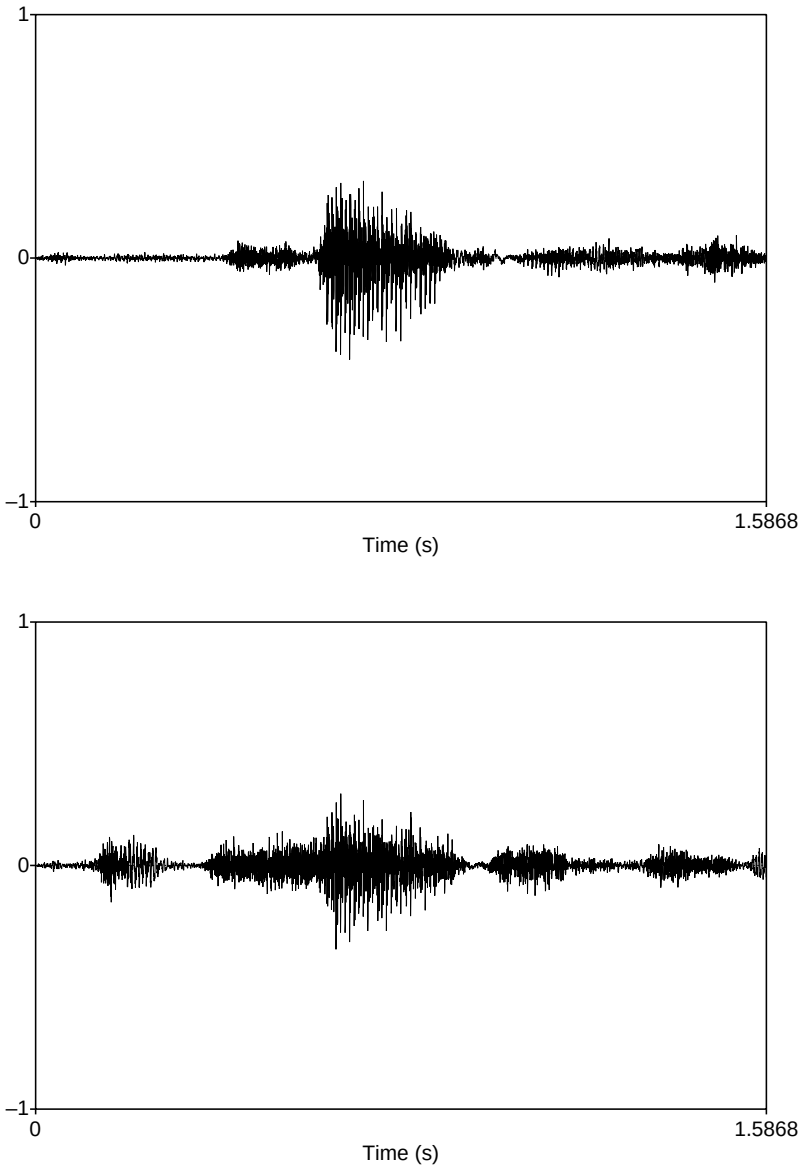
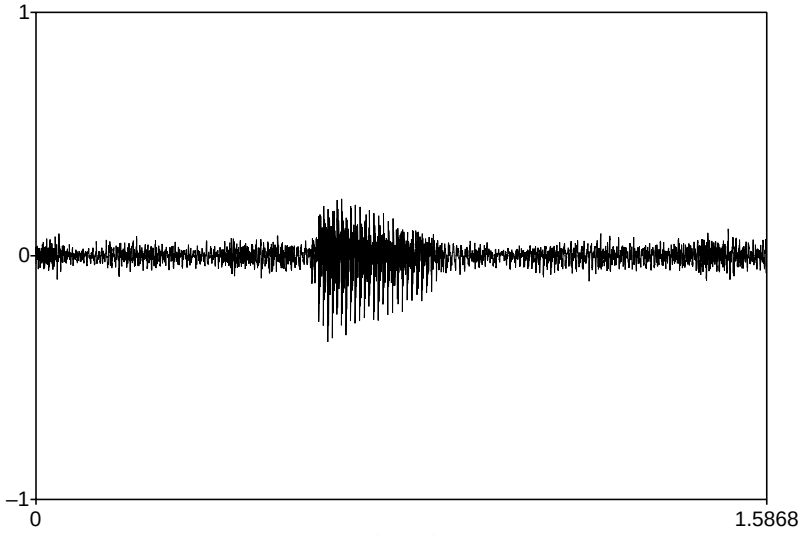
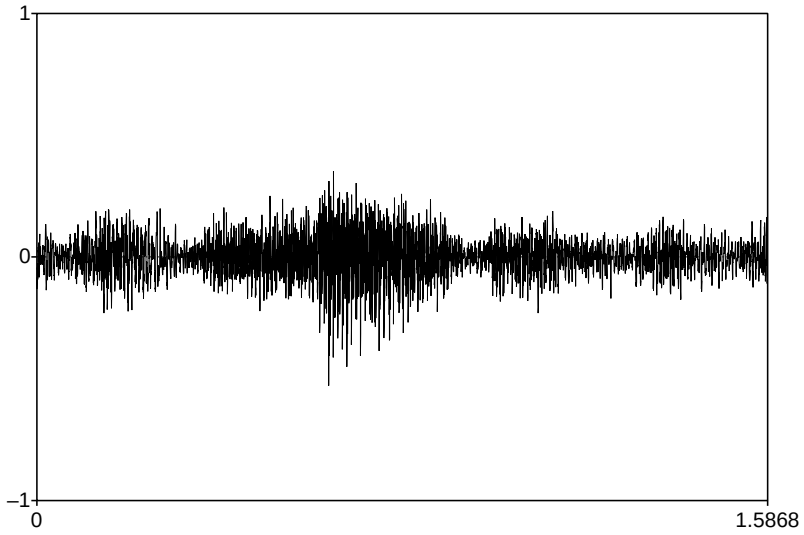


Figure 4.2: The waveforms of the scaled MBSS Processed signals from the noisy speech signals with a) 5 dB SNR and b) 0 dB SNR.



a)



b)

Figure 4.3: The waveforms of the scaled Unprocessed signals with a) 5 dB SNR and b) 0 dB SNR.

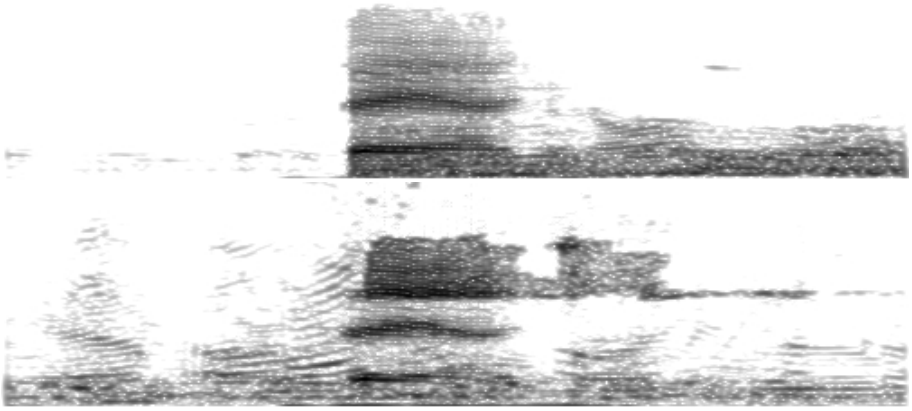


Figure 4.4: The spectrogram images of the scaled Our Processed signals from the noisy speech signals with a) 5 dB SNR, and b) 0 dB SNR.

a)

b)

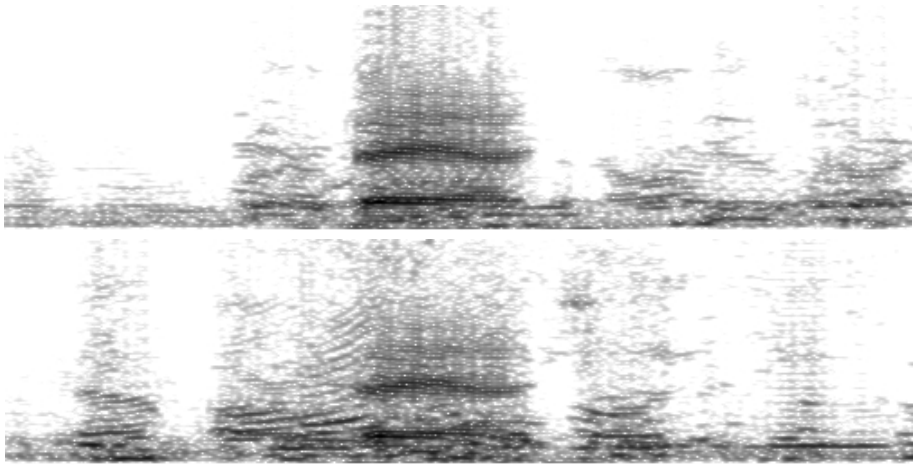


Figure 4.5: The spectrogram images of the scaled MBSS Processed signals from the noisy speech signals with a) 5 dB SNR and b) 0 dB SNR.

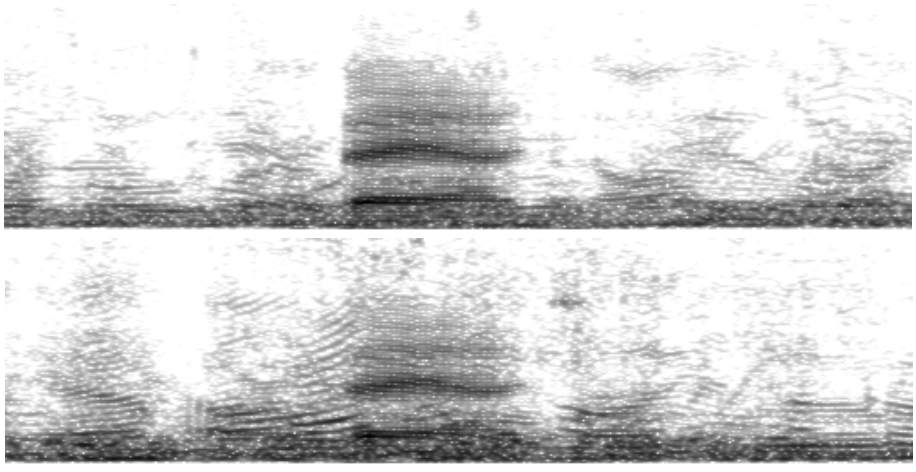


Figure 4.6: The spectrogram images of the scaled Unprocessed signals with a) 5 dB SNR and b) 0 dB SNR.

signals maintain much of the noise. Also, as can be seen in the hard segmentations in Figure 3.8, Our Processed signals are very much band-dependent. This is clearer in the 0 dB SNR level in which the pixels in the bottom four bands of the hard segmentation were set to all white (speech is absent). This band-dependence is not obvious in the MBSS Processed signals, see Figure 4.5. In terms of sound quality, Our Processed signals overall sounded cleaner but more mechanical than the MBSS Processed signals.

4.6.2 Statistical Results

Table 4.1 displays the mean and standard deviation, in percentages, of the FOM over the 24 subjects for each of the twelve conditions. The means ($\mu_{\text{Unprocessed}}$, $\mu_{\text{MBSS Processed}}$ and $\mu_{\text{Our Processed}}$) and standard deviations of the FOM over the processing levels, the means ($\mu_{5\text{ dB SNR}}$ and $\mu_{0\text{ dB SNR}}$) and standard deviations of the FOM over the SNR levels, and the means (μ_{Initial} and μ_{Final}) and standard deviations of the FOM over the location levels are also displayed in the table.

Table 4.2 is the summary table for the three-way ANOVA. Effects marked with

Table 4.1: Mean ± the standard deviation, in percentages, of the FOM over the 24 subjects for a) the Initial location level, b) the Final location level, and c) the means over the locations.

a)

Initial	Unprocessed	MBSS Processed	Our Processed	Mean
5 dB SNR	96%±3.3%	84%±5.2%	70%±8.5%	83%±12.3%
0 dB SNR	81%±6.7%	76%±6.7%	70%±7.9%	76%±8.5%
Mean	89%±9.3%	80%±7.0%	70%±8.2%	79%±11.2%

b)

Final	Unprocessed	MBSS Processed	Our Processed	Mean
5 dB SNR	90%±4.6%	81%±5.9%	81%±6.0%	84%±7.1%
0 dB SNR	84%±5.7%	77%±8.7%	61%±11.9%	74%±13.2%
Mean	87%±6.1%	79%±7.7%	71%±13.9%	79%±11.8%

c)

MEAN	Unprocessed	MBSS Processed	Our Processed	Mean
5 dB SNR	93%±5.0%	82%±5.7%	76%±9.1%	84%±12.1%
0 dB SNR	82%±6.3%	76%±7.7%	65%±10.9%	75%±11.1%
Mean	88%±7.9%	79%±7.3%	70%±11.3%	80%±11.5%

a * were significant at the chosen 0.05 alpha level. The null hypothesis that the mean arcsine transform of the FOM was independent of SNR can be rejected; the mean performance at the higher SNR was significantly better than the performance at the lower SNR. The null hypothesis that the mean arcsine transform of the FOM is independent of Processing can be rejected. The interaction between Processing and SNR was found to be significant. Finally, the interaction between Processing, SNR and consonant Location was found to be significant. Post-hoc testing was performed on the three significant sources which take the processing levels into account.

The post-hoc tests revealed that mean arcsine transform of the FOM for all three processing levels were significantly different from one another. That means that: 1) speech recognition performance for Our Processed signals was significantly poorer than performance for MBSS Processed signals and performance for the Unprocessed signals, and 2) speech recognition performance for MBSS Processed signals was significantly poorer than the performance for the Unprocessed signals. Sub-

Table 4.2: The summary table of the three-way repeated measures ANOVA applied to the arcsine transform of the FOM. The first column contains the sources of variation; the second column contains the sum of squares (SS); the third column contains the degrees of freedom (DF); the fourth column contains the mean of the sum of squares (MS); the fifth column contains the calculated F value; and the last column contains the probability that the sources are from the same population. Sources marked with a * were significant at the chosen 0.05 alpha level.

Source of Variation	SS	DF	MS	F	Significance
Processing	10.3	2	5.14	180	0.0010 *
SNR	4.28	1	4.28	123	0.0010 *
Location	0.0425	1	0.0425	2.07	0.161
Subjects	1.48	23	0.0641		
Processing×SNR	0.544	2	0.272	9.34	0.0010 *
Processing×Location	0.168	2	0.0840	3.16	0.0502
Processing×Subjects	1.32	46	0.0286		
SNR×Location	0.00421	1	0.00421	0.143	0.710
SNR×Subjects	0.798	23	0.0347		
Location×Subjects	0.473	23	0.0206		
Processing×SNR ×Location	1.79	2	0.896	30.1	0.0010 *
Processing×SNR ×Subjects	1.34	46	0.0292		
Processing×Location ×Subjects	1.22	46	0.0266		
SNR×Location ×Subjects	0.678	23	0.0295		
Processing×SNR ×Location×Subjects	1.37	46	0.0298		
Total	25.8	287			

section 5.2.1 discusses the reason why we believe our methodology has failed and Section 5.3 describes some future work to potentially correct for this failure. It is not clear why in this experiment the MBSS processing performed worse than no processing. The only reasonable explanation lies in our choice of parameters for running the MBSS. In particular, it seems that the low value for OVLP, which controls the percent of overlap between frames, must be the explanation for the poor MBSS results.

From the post-hoc testing on the interaction of Processing and SNR, see Table 4.1c, we learned that the Unprocessed-0 dB SNR condition is equivalent to the MBSS Processed-5 dB SNR condition, and that the MBSS-Processed-0 dB SNR condition is equivalent to the Our Processed-5 dB SNR condition. This gives us a sense of how much worse our methodology performed as compared to the MBSS methodology and no signal processing.

From the post-hoc testing on the interaction of Processing, SNR and Location, we learned the following, see Tables 4.1a and 4.1b. Let the **value** of a condition be the mean arcsine transform of the FOM of that condition. The condition with statistically the lowest value was Our Processed-0 dB SNR-Final, which means that our processing particularly failed in the 0 dB SNR case for foil sets that were confusable in the final location. The conditions Our Processed-0 dB SNR-Initial and Our Processed-5 dB SNR-Initial had a statistically greater value than the condition Our Processed-0 dB SNR-Final, but had a value statistically less than or equal to all other conditions. Conditions MBSS Processed-5 dB SNR-Initial, MBSS Processed-5 dB SNR-Final, Unprocessed-0 dB SNR-Initial and Unprocessed-0 dB SNR-Final all had statistically equal values. This agrees with the information attained from the post-hoc testing on the interaction of Processing and SNR that conditions MBSS Processed-5 dB SNR and Unprocessed-0 dB SNR were equivalent. Finally, the condition Unprocessed-5 dB SNR-Initial statistically had the highest value of all

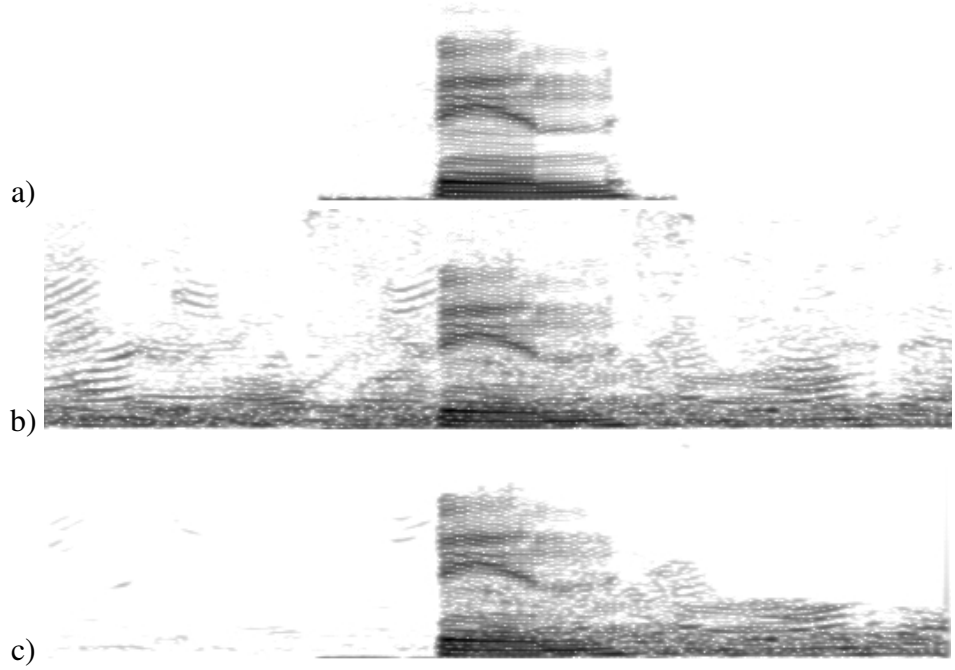


Figure 4.7: Spectrogram images for the word “fin”. a) Clean spectrogram image. b) Noisy spectrogram image with 5 dB SNR. c) Spectrogram of the scaled Our Processed signal from the noisy spectrogram with 5 dB SNR.

other conditions.

4.6.3 Problematic Words

There were a total of three words processed by our methodology (over the two SNR levels) which none of the subjects identified correctly. For the 5 dB SNR level, the words were: “fin” and “hook.” Figures 4.7 and 4.8 display the clean spectrogram images, the noisy spectrogram images and the spectrogram images of the scaled Our Processed signals for these two words, respectively. Both of these words are in foil sets whose words are confusable in the initial location. That would indicate that the algorithm loses some speech information at the onset of speech. This can be seen in Figure 4.8, the small burst in the lower frequencies in the clean speech signal is mostly gone in Our Processed signal.

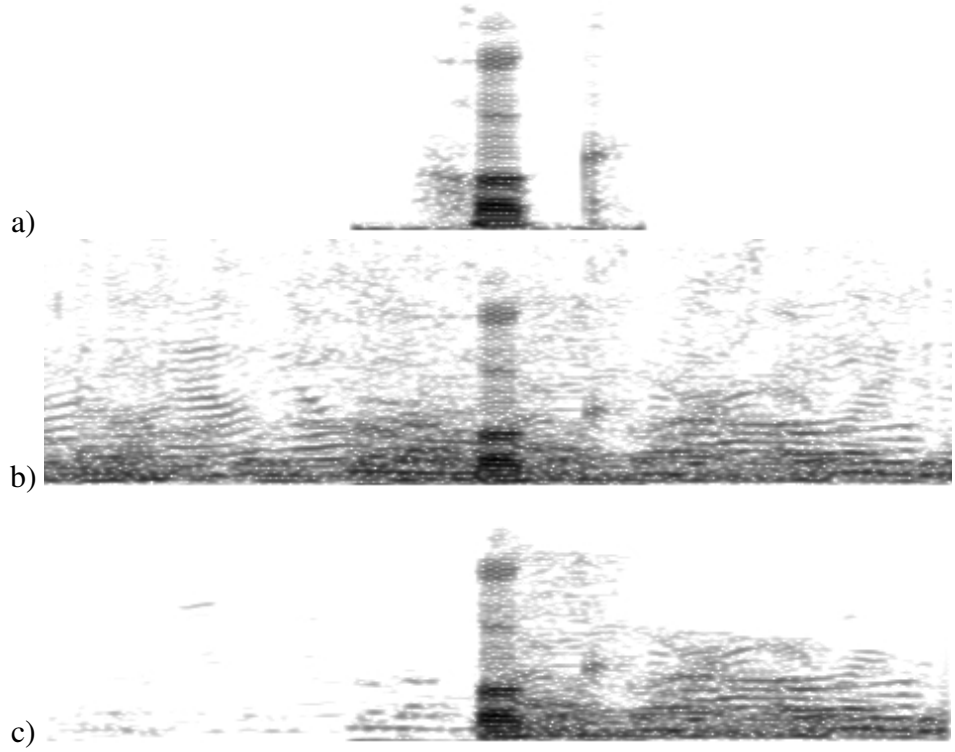


Figure 4.8: Spectrogram images for the word “hook”. a) Clean spectrogram image. b) Noisy spectrogram image with 5 dB SNR. c) Spectrogram of the scale Our Processed signal from the noisy spectrogram with 5 dB SNR.

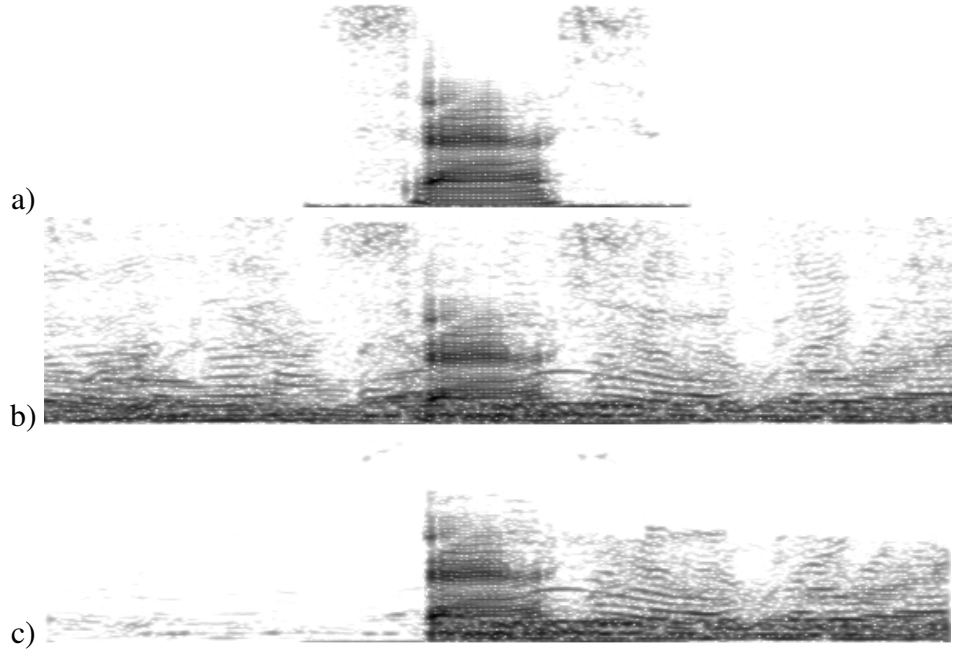


Figure 4.9: Spectrogram images for the word “sass”. a) Clean spectrogram image. b) Noisy spectrogram image with 0 dB SNR. c) Spectrogram of the scaled Our Processed signal from the noisy spectrogram with 0 dB SNR.

For the 0 dB SNR level, the word which none of the subjects identified correctly was “sass.” Figure 4.9 displays the clean spectrogram image, the noisy spectrogram image and the spectrogram image of the scaled Our Processed signal for this word. For this word, the foil set contains words confusable in the final location. Indeed in Figure 4.9, the bursts in the higher frequencies in the clean speech signal are mostly gone in Our Processed signal.

There were many words which all the subjects understood correctly. For the 5 dB SNR level, the words were: “bug,” “dud,” “fill,” “heal,” “kick,” “lake,” “lick,” “pane,” “peach,” “rave,” “red,” “rust,” “seat,” “seek,” “sing,” “tale,” “tame,” “teach,” “tip.” For the 0 dB SNR level, the words were: “cop,” “got,” “kit,” “look,” “mark,” “raw,” “rest,” “sale,” “sale,” “told,” “will.” The word “sale” appears twice in the list used for the “Our Processed - 0 dB SNR” condition, and in both cases all subjects identified it correctly.

This suggests there is no clear classification of problematic words to differen-

tiate from the other words. For example, it is not the case that all words with a particular consonant, such as “s,” are a problem.

Chapter 5

Conclusions and Future Work

Section 5.1 describes the summary and contributions of our work. Section 5.2 contains some final notes and discussions. Section 5.3 describes some possible improvements and interesting future work. Section 5.4 describes some steps taken towards starting some of the suggested future work.

5.1 Summary and Contributions

A novel way to enhance noisy images in "real-time," by which we mean column by column, was described. In particular, the enhancement was restricted to the "real-time" search for $[0,1]$-valued columns that were used as multiplicative masks on the noisy image columns. In this work the $[0,1]$-valued columns arose by "fuzzifying" binary columns. These binary columns were estimated directly from the noisy image columns using prior information. The prior information took into account the characteristics of binary images that correspond to clean images (prior model) and the correspondence between such binary images and the noisy images (noise model).

One application of such work is to the "real-time" estimation of clean speech signals from noisy speech signals for hearing aid users. In this work, the image pro-

cessing described above was performed on the spectrograms of the signals. The image processing performed attempted to estimate the spectrogram that corresponded to the unknown clean speech signal from the noisy speech spectrogram. Finally, a clean speech signal was estimated from the estimated clean speech spectrogram.

To quantify the success of our algorithm, speech recognition performance was measured on normal hearing listeners. Three processing levels included the noisy speech signals (Unprocessed) and two noise reduction processes: Multi-band spectral subtraction (MBSS Processed) was used as the gold standard, and our methodology (Our Processed). The Modified Rhyme Test (MRT) was used as the test material. The MRT contains 300 words, which are organized into sets of 6 confusable words, either in the initial or final location. Such an experiment enabled us to statistically test any improvement attained by our methodology. A three-way repeated measures analysis of variance (ANOVA) was used as the statistical test, where the three factors were Processing (Unprocessed, MBSS Processed and Our Processed), signal-to-noise ratio (SNR) (5 and 0 dB) and the location of the confusable consonant (Initial and Final).

The results of our experimental evaluation of our methodology are as follows. For each subject, the figure of merit (FOM) of a condition was the portion of correctly identified words for that condition. The arcsine transform of the FOM was used in order to stabilize the error variance before performing the statistical analysis. The most interesting result was that the null hypothesis that the mean arcsine transform of the FOM is independent of the processing level can be rejected with 5% confidence. Post-hoc tests showed that the three processing levels each differ statistically from one another. It was concluded that our methodology decreases recognition.

As can be seen by the resulting waveforms and spectrograms in Subsection 4.6.1, both algorithms perform worse in the 0 dB SNR level than the 5 dB SNR

level; i.e. the estimated clean speech signal is visibly noisier for the 0 dB SNR level. Our methodology produces waveforms and spectrograms that are visibly much better in the first 500 ms of the signal (noise-only portion of the noisy speech signal) in that it got rid of almost all the noise. Our method performs poorly in the last 500 ms of the signal (noise-only portion of the noisy speech signal). The noise is barely removed in this region. In comparison, the MBSS visually removed less noise overall. Our Processed signals sounded more mechanical but cleaner than the MBSS Processed signals.

5.2 Discussion

This section contains some final discussion. Our research and results have raised many additional questions, most of which could not be adequately looked at during this time frame and should be considered in future work. We describe some specific areas of interest in Section 5.3. This section is split into three subsections, each dealing with a different discussion topic regarding the current work. Subsection 5.2.1 explains why our hard segmentation algorithm fails. Subsection 5.2.2 discusses the value of NR. Subsection 5.2.3 explains the assumption of an SNR level knob on the hearing aid.

5.2.1 Hard Segmentations

For ease of discussion, Figures 3.8 and 3.10 are reproduced in this section as Figures 5.1 and 5.2.

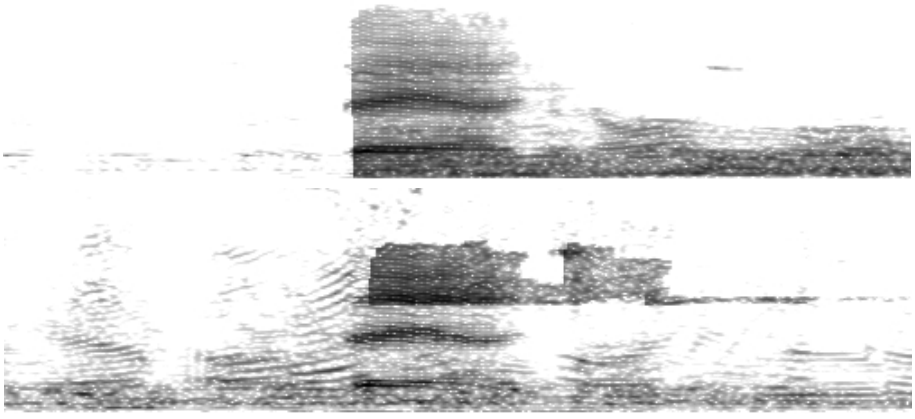
Notice the asymmetric behavior of the hard segmentations in Figure 5.1 in terms of the way the algorithm behaves at first (first 500 ms, after two columns of assumed silence) and after clean speech was present (last 500 ms). In the first 500 ms, the hard segmentation determined that clean speech is absent in the estimated binary

a)

b)

Figure 5.1: The hard segmentation of the noisy spectrogram with a) 5 dB SNR, and b) 0 dB SNR.

a)



b)

Figure 5.2: Estimated clean spectrogram of the noisy spectrogram with a) 5 dB SNR, and b) 0 dB SNR.

images. The consequence of this is that in the estimation of the clean spectrogram, see Figure 5.2, one can see that most of the noise was eliminated in the first 500 ms of the signal. (This can also be seen in the estimation of the clean speech, see Figure 3.14.)

In the last 500 ms, the hard segmentation determines that speech is present in the estimated binary image in much of the lower frequencies, see Figure 5.1. The consequence of this is that in the estimation of the clean spectrogram, see Figure 5.2, one can see that most of the noise remains in the last 500 ms of the signal. (This

can also be seen in the estimation of the clean speech, see Figure 3.14.)

The reason for this asymmetry has to do with the asymmetry of the chosen binary pixel neighborhood we chose, see Figure 3.7. The chosen neighborhood is such that the binary color at the pixel of interest can be estimated from several binary values in the current column and columns further to the left. This lop-sidedness of the neighborhood had to do with the "real-time" aspect of our work, in which only the current column (being estimated) and previous columns to the left are available for the estimation of the color of the pixel of interest.

During the training process, the binary neighborhood distribution is estimated. It is the case in the training set that when pixels to the left are white (speech is absent) in the lower bands, as in the first 500 ms of all signals, it was more likely to be followed by more white pixels (speech is absent). Similarly, when pixels to the left are black (speech is present) in the lower bands, such as in the middle section of the each signal, it was more likely to be followed by more black pixels (speech is present). This dependence on past information is the reason for the asymmetry in our hard segmentations.

In the 0 dB SNR level in Figure 5.1, notice that our algorithm is very much band-dependent. In this estimation of the binary image corresponding to the unknown clean speech spectrogram, the bottom four bands were all set to white pixels (speech is absent), despite the existence of black pixels (speech is present) in the highest band. The reason for this behavior has to do with the fact that at the 0 dB SNR level, the clean speech and noise levels are equivalent. It is therefore a difficult task for the algorithm to recognize the existence of clean speech. Despite that, as can be seen by the estimated clean speech spectrogram in Figure 5.2, the frequencies were not completely removed in the lower four bands.

We further note that since our algorithm is stochastic, the hard segmentations may differ from run to run. We hope that the hard segmentations do not differ much

between one run and another. In the case of the particular word chosen, we have seen extremely different hard segmentations for the 0 dB SNR level; some hard segmentations contained mostly black pixels (speech is present) in the lower four bands in the middle section. This, clearly, very much affects the estimation of the clean speech signal.

5.2.2 Choice of NR

In Subsection 3.4.2, in the description of the estimation of the binary column, an annealing schedule was described. The annealing schedule we used was run NR times per column and the column with highest value of the maximization function was chosen. In this work, the value of NR was set to 5. The annealing schedule we used started β at 0.5 and increased the value of β by 0.01 every 200 cycles, until it reached the value of 1.5.

Alternatively, we could have used NR $= 1$ with the value of β increasing every 1,000 cycles in the annealing schedule, which would have required exactly the same number of iterations. In preliminary work, we found that in the later case, often a column got stuck at a local maximum and affected the estimation of the subsequent columns for the worse. Certainly a longer schedule (more cycles or a smaller increase in the value of β) could have been used, but would have slowed down the algorithm. Since columns with higher values of $M\left(\omega;\hat{j}\right)$ are more likely, the higher the value of NR, the less likely we are to get stuck in a local maximum.

5.2.3 SNR Knob

In this work, we have assumed that the training sets exist or have been created for each SNR level. Furthermore, noise information probability estimation is SNR dependent. The reason behind this dependence on the SNR level is due to the fact that the correspondence between the noisy speech image and the binary image differ

drastically for different SNR levels. The need for the noise information probability arose in the second step of our methodology, that of estimating a binary column from a noisy image column. In order to control for this SNR dependence in a hearing aid, we assume that a knob (similar to a volume control) could be present, which the user uses to adjust the enhancement.

5.3 Future Work

We believe that despite the decreased recognition results, our methodology still has potential - both in the image processing and signal processing domains. In the following subsections are ideas that can possibly improve the real-time enhancement of noisy images as they arrive column by column. It is further possible that such improvements would lead to an improvement in the recognition of estimated clean speech signals from noisy speech signals using our methodology.

Subsection 5.3.1 describes possible general improvements. The remaining sections describe more specific possibilities. In particular, Subsection 5.3.2 describes the potential benefit of using a voice activity detector. Subsection 5.3.3 describes the reason to consider a larger time skip step. Subsection 5.3.4 explains the limitation on the size of the prior model neighborhood and what needs to be done to allow for larger neighborhoods. Subsection 5.3.5 describes the possibility of using a symmetric neighborhood for the binary column estimation in Step 2. Subsection 5.3.6 describes more complex noise models that could be attempted. 5.3.7 details other potential frequency banding. Subsection 5.3.8 describes a different way of estimating the soft segmentation column directly from the noisy spectrogram. Subsection 5.3.9 describes alternate ways of estimating the clean speech from columns of the estimated clean grayscale. Finally, Subsection 5.3.10 includes future capabilities of speeding up the hard segmentation algorithm in Step 2.

5.3.1 General

Each of the steps in our methodology required deciding on transforms or parameters. It is likely that a different combination of decisions would generate better results. We have chosen the spectrogram, rather than the STFT or other signal to image transforms, because the spectrogram seems to be the mostly widely used imaging technique of signals, the advantage of which is that much is known about the characteristics of speech spectrograms. Other transforms, such as those suggested in 1.1, could be tested. Several simple ways of "fuzzifying" the binary columns and using these $[0, 1]$ real-valued masks were briefly considered and could be further investigated.

5.3.2 Voice Activity Detector

Our methodology failed in the last 500 ms of the noisy speech signals. One potential way to improve these results is to introduce a voice activator detector (VAD), which estimates when clean speech is present in the noisy signal. (VAD was used by the MBSS gold standard; possibly, it is the major reason for its superior performance as compared with our methodology.) When the VAD detects only noise, one of two simple things could occur: either our algorithm is used to estimate the clean speech signal (that leaves a bit of background noise) or simply the output could be silence. As a note, once a binary column is set to all white (clean speech is absent), our methodology should behave well again, that is behave as it does in the first 500 ms of the signal.

The VAD could be used in a more sophisticated manner as well. During the noise training, training could be conducted as a function of the existence/absence of clean speech. In this way, not only would the training be (frequency) band-dependent but also dependent on the existence of clean speech in the noisy signal. During the estimation of the hard segmentation, this information would be used

to better estimate the binary image. In particular, the VAD would be active and the appropriate training information would be used to estimate each binary column based on the existence/absence of clean speech in the noisy speech signal.

5.3.3 Time Skip Step

We believe that the poor MBSS results in our experiment arose from our parameter choices. In particular, since the major difference between our parameter choices and those used in [38, 46] lies in the value of the frame overlap size, it seems that the small frame overlap size was to blame for the poor MBSS results. We can, therefore, hypothesize that our methodology would also have benefited from a larger frame overlap. In the language used here, that would correspond to using a larger time skip step: instead of the 3.1 ms used here, the value of 12.4 ms (corresponds to the 50% overlap used in [38, 46]) should be tested.

5.3.4 Bigger Neighborhoods

We hypothesize that larger neighborhoods in the prior model would lead to better binary column estimations. The limitation on the size of the neighborhoods in this work arose from two separate issues. One limiting factor has to do with the chosen bands and the size of the given training set. Since Band 0 contains only 4 rows, we were limited to accurately estimating from our training set binary clique distributions with at most 5 pixels. The other limiting factor has to do with Graphical Models Theory, which cannot be used for neighborhoods arising from complex combinations of cliques and separators. For that reason, with our banding and training set, the neighborhood used here was as big as we could come up with. If different banding was used or more training data was available, bigger cliques could be accurately estimated directly from the training set. That would allow for larger neighborhoods to be used in the prior model.

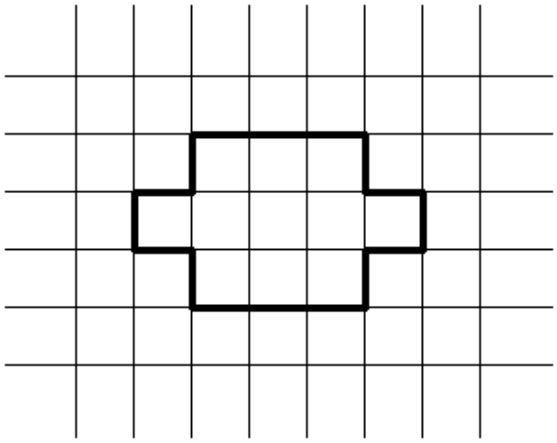


Figure 5.3: A symmetric neighborhood which would "look into the future."

5.3.5 Symmetric Neighborhood

An alternate improvement to Step 2 of our algorithm involves introducing a short delay in that step. Such a delay would allow for a symmetric neighborhood that would "look into the future." By that, it is meant that the neighborhood would contain pixels in columns further to the right. Our prior experience in this field [73] has been that such a neighborhood would perform much better (in terms of estimating the binary column) than the neighborhood we currently use. For this work, one suggested neighborhood is depicted in Figure 5.3, in which two binary columns to the right of the current one would be estimated to aid in the estimation of the current binary column. Since the time skip step between columns is 3.1 ms, see Figure 3.1, this would require incorporating only a 6.2 ms delay. In order to implement this suggestion, it remains to be decided how the "future" columns are estimated. In particular, these questions remain to be answered: how are the neighborhoods of pixels in these two future columns defined; how are these three columns (current and two future) initialized; and is an annealing schedule used for all three columns simultaneously or separately. It must further be evaluated whether such a delay would create an additional delay in our overall methodology.

It is possible this delay can be incorporated into the delay introduced already in Step 4.

5.3.6 More Complex Noise Model

If more RAM were available, more complex noise models could be considered. The current noise model assumes that the noisy grayscale value $\theta[h]$ depends only the binary value $\omega[h]$ (and the band number $b[h]$). It is possible to imagine noise models which would also take into account neighboring noisy grayscale values. Such a noise models would lead to much more complex posterior probability function, which would make the calculation of p more cumbersome and consequently would require larger look-up tables. Despite the binning used in this work to limit the size of the look-up tables' dependence on the noisy grayscale values, such noise models were currently not feasible.

5.3.7 Banding

As was mentioned in Subsection 5.2.1, our hard segmentation was overly band-dependent. Preliminary experiments in this application led us to believe that banding is necessary since the characteristics of speech are not the same at different frequencies. The banding we chose was such that the bands doubled in size as the frequencies increased. Several other banding choices could be tested for better performance, in terms of estimating horizontally smoother binary images. One frequency banding that might be worth testing is that of using only two bands - one for lower frequencies and one for higher frequencies. This would be the simplest banding possible. On the other extreme, another frequency banding worth considering is one in which each row is taken to be a separate band. The advantage of such banding is that no assumption is made on the behavior of adjacent frequencies. The disadvantage of such banding is the necessity to maintain much larger

look-up tables and the necessity for larger training sets in order to continue to accurately estimate the chosen neighborhood. Currently such banding would have been infeasible, but is something worth investigating in the future.

5.3.8 Different Soft Segmentation

The following potential future work could be performed to estimate a soft segmentation column directly from the noisy spectrogram, rather than first estimating a hard segmentation column and only then "fuzzifying" it to create the soft segmentation column. The idea is that instead of using maximum a posteriori estimation to estimate the column that maximizes the pseudo-posterior likelihood $M\left(\varpi;\hat{j}\right)$ (the mode of the frequency distribution $M\left(\varpi;\hat{j}\right)$), defined in Equation 3.42, it is possible to use the mininum mean square error estimation to estimate the column that is the mean of the frequency distribution $M\left(\varpi;\hat{j}\right)$ [14, 22]. That column is calculated by summing over all possible binary columns multiplied by their value of $M\left(\varpi;\hat{j}\right)$, which is no longer a binary column. This column is the new soft segmentation column.

Since there are $2^{I-1} = 2^{124}$ possible binary columns, calculating such a column is practically impossible. Instead, it is possible to sample the distribution of $M\left(\varpi;\hat{j}\right)$ in order to estimate the mean of its frequency. One way to do this would be to use the Metropolis Algorithm without an annealing schedule (set $\hat{\beta} = 1$), as described in Subsection 3.4.2. In this work, we were running the Metropolis Algorithm for 200 cycles for each of the 101 values of β, five times ($NR = 5$), for a total of 101,000 cycles per column. It is possible to sample the distribution every cycle after the first 5,000 cycles (after which the samples are unaffected by initial column), for a total of 96,000 samples. These 96,000 binary column samples would be averaged (summed and divided by 96,000) in order to estimate the mean frequency of $M\left(\varpi;\hat{j}\right)$. Such a column would be the new estimated soft segmentation column.

One complication is that in the calculation of p in each Metropolis iteration, we no longer know the previous two binary columns of ϖ. Only 4 binary values of ϖ from the previous two columns are required for this calculation. It is, therefore, possible to estimate the value of p by averaging the values of $M\left(\varpi_1; \hat{j}\right)$ and $M\left(\varpi_2; \hat{j}\right)$ over the 2^4 possible binary configurations of those 4 pixels. If all else remains the same, it would take $2^4 = 16$ times longer to produce this new soft segmentation column than it took to create the hard segmentation column.

5.3.9 Image to Signal Transform

Other methods of estimating the clean speech from estimated clean grayscale columns should be considered. In particular, it should be tested whether the estimation of the phases in Step 4 of our methodology affects speech recognition. It is possible that with the current way of doing things, the phases are randomly assigned and speech recognition is deteriorated. An alternative approach to that described in Section 3.6 it to simply estimate the clean speech signal by combining the estimated norm from the square root of the estimated clean grayscale columns and the phases/angles of the noisy speech signal. That is, the speech estimated by the given image-to-signal algorithm after the first iteration should be compared to that of the following iterations (or final iteration) by speech recognition or some other objective measure.

5.3.10 Speed Up Step 2

Other future work includes attempting to speed up the overall methodology. As can be seen in Section 3.7, Step 2 is by far the slowest step. The reason Step 2 is so slow has to do with the fact that the maximization algorithm is iterative, and that each iteration requires several look-ups in tables and several additions. Furthermore, each column is estimated several times using an annealing schedule. There are several ways in which this step could potentially be sped up. The easiest

way is to reduce the value of NR, which would produce a substantial reduction in run time. Alternatively, the number of cycles per β could be decreased, the step size in β could be increased, or the overall annealing schedule could be shortened. Preliminary experiments were carried out to determine the schedule used here, but extensive experiments could determine a shorter schedule that is just as affective.

The current implementation of Step 2 uses three regular look-up tables of size $163,840$ bytes each. If more RAM was available, these three tables could be replaced by one large regular table of size approximately 7 gigabytes. That would mean that in each iteration there would be only one look-up instead of three look-ups and two additions. This would also result in a substantial speed up. Similarly, the special tables could be combined into larger special tables.

5.4 Preliminary Future Work

Based on some final suggestions on this work, the following preliminary investigations were conducted. It was suggested that one way of optimizing all the parameters and transforms was to search for a transformation of the soft segmentation image to be used as a mask on the noisy spectrogram in the estimation of the clean spectrogram in Step 3 of our methodology, described in Section 3.5. It was further suggested that an objective measure be used to evaluate results prior to testing results using human subjects (which is time-consuming and expensive).

We first searched for a transformation of the soft segmentation image of the form

$$aSS + (1-a)\sqrt{SS} \tag{5.1}$$

where SS refers to a pixel value in a soft segmentation image. After some work, it was agreed that there was no good reason to use that form; the following more

Table 5.1: The band-dependent values of a, b, and c for the 0 dB SNR case.

	a	b	c
band 0	-5.757	5.827	0.0278
band 1	-6.444	6.970	-0.014
band 2	-7.907	8.702	-0.2034
band 3	-2.874	3.333	0.4665
band 4	-1.815	2.772	0.0279

general form was used instead

$$\max\left(aSS + b\sqrt{SS} + c, 0\right), \tag{5.2}$$

where $\max(\delta, \eta)$ is the maximum of the real numbers δ and η. The need for the maximum is due to the fact that spectrograms are non-negative. For that reason, the mask used to create the estimated clean spectrogram also need be non-negative. Notice that such a mask is no longer necessarily a $[0,1]$-valued image.

The values of a, b, and c were acquired in the following way. It was decided to minimize the distance between the estimated clean spectrograms (created by multiplying the transformed soft segmentations by the noisy spectrogram) and the clean spectrograms. To that end, the least squares error estimation was used. It was further decided to search for the values of a, b, and c band-dependently. Since the soft segmentations were available only for the testing set, the values of a, b, and c were trained on the testing set. This is clearly not ideal; in future work, these values should be trained from the training set. In this preliminary work, the values of a, b, and c were only band-dependently estimated for the 0 dB SNR case; their values are displayed in Table 5.1. Figure 5.4 shows the band-dependent transformations of the soft segmentation values into the new mask values for the 0 dB SNR case.

Once the values of a, b, and c were estimated, the new masks were created. Figure 5.5 shows the new mask created by band-dependently transformation of the soft segmentation in Figure 3.9b; compare this to the soft segmentation mask in

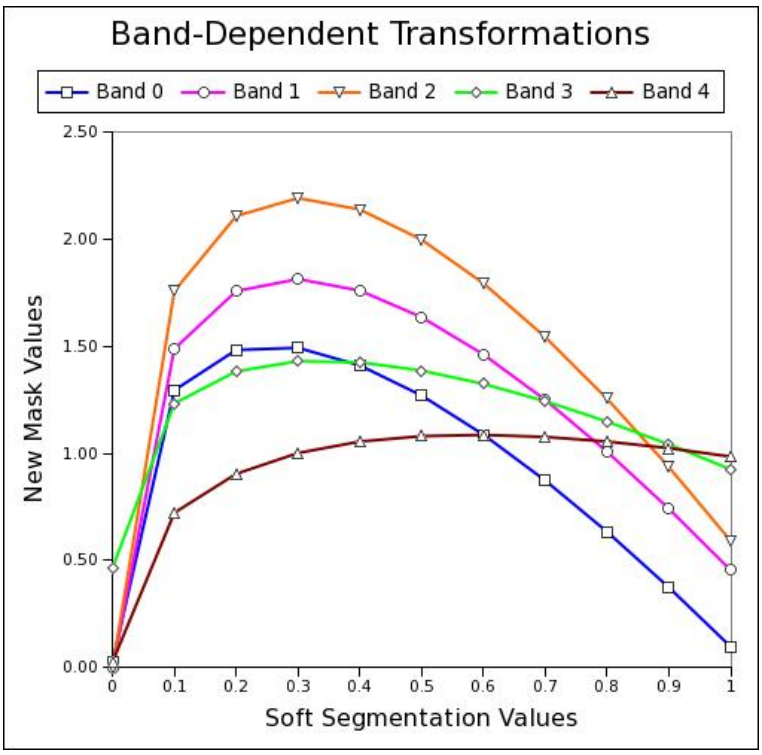


Figure 5.4: Band-dependent transformations of the soft segmentations for the 0 dB SNR case.

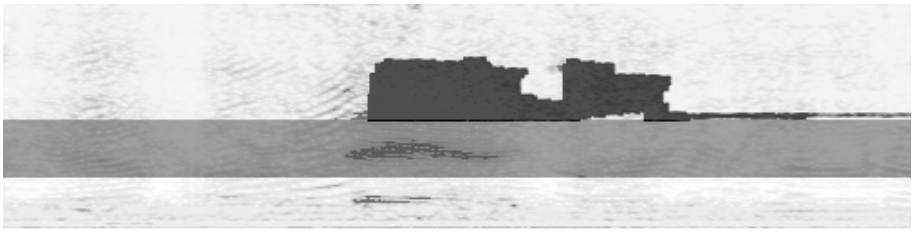


Figure 5.5: The new mask for the 0 dB SNR case.

Figure 3.9b. Notice that this mask is quite constant on a band by band basis and different from band to band. This is due to the fact that the soft segmentation are quite bimodal, with the two modes being at 0 and at 1, and that the values of the transformations are close at those two extremes.

Figure 5.6 shows the new estimated clean spectrogram created by using this new mask on the noisy spectrogram in Figure 3.4b. Compare this to the estimated clean spectrogram in Figure 3.10b. The estimated clean spectrogram in Figure 5.6 appears similar to the noisy spectrogram in Figure 3.4b, but that is not the case. In

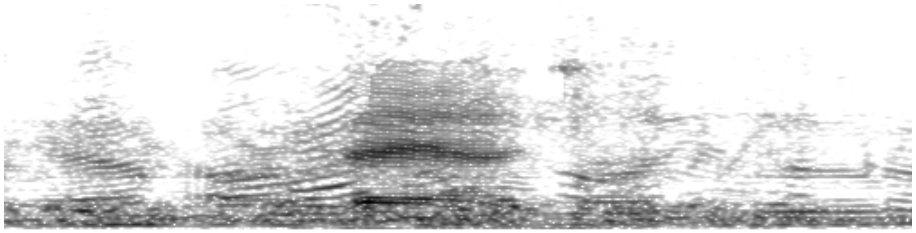
a) 

Figure 5.6: The newly estimated clean speech spectrogram for the 0 dB SNR case.

Table 5.2: The mean and standard deviation of the IS distances of the four processing levels and the clean signals over the testing images.

	Unprocessed and Clean	MBSS and Clean	Our Processed and Clean	Our Processed V2 and Clean
Mean	19.96	18.80	28.27	18.77
Standard Deviation	1.09	1.08	5.93	1.06

actuality, the values in the estimated clean spectrogram are much lower than those in the noisy spectrogram. The images only look similar due to the independent scaling in the images.

Next, the clean speech signals were estimated as in Step 4 of our methodology, see Section 3.6. We refer to these signals as the Our Processed V2 signals, see Figure 5.7. Compare this result to that Our Processed signal in Figure 3.14b. The objective measurement agreed upon to evaluate the Our Processed V2 signals was the Itakura-Saito (IS) distance measure [46]. The IS distance measure was one of several tested by [38, 46] in their comparison of different estimations of clean speech signals. The code for the IS distance measure was made available in Matlab on the DVD attached to [46], which is what we used.

We calculated the mean and standard deviation of the following four distances over all signals in the testing set: 1) between the Unprocessed and the Clean, 2) between the MBSS Processed and the Clean, 3) between Our Processed and the Clean, and 4) between the Our Processed V2 and the Clean. Table 5.2 displays these means and standard deviations.

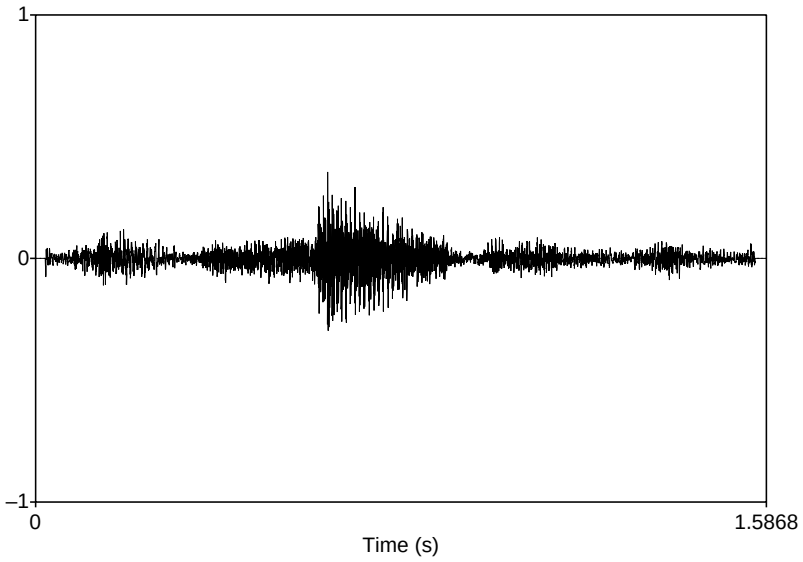


Figure 5.7: New Our Processed signal in the 0 dB SNR case.

Next we tested for significant differences between these distance values. The null hypothesis is that all 4 differences arise from the same distribution. Under this hypothesis, their distances are samples of a zero mean random variable. Since the sample size (300 in our case) is large, it can be assumed that the samples are normally distributed with zero mean. In order to test for significance of these IS distances, 6 t-tests [16, 53] were performed on the 4 differences with a 5% confidence interval. It was found that the null hypothesis that this differences arise from the same distribution can be rejected. In particular, all distances were significantly different except for the distance between the MBSS Processed and the Clean and the distance between Our Processed V2 and the Clean.

This does not follow the human subject results in our experiment. In our human subject experiment, speech recognition performance for MBSS Processed signals was significantly poorer than the performance for the Unprocessed signal. To the contrary, in terms of the IS distance the MBSS Processed signals are significantly better than the Unprocessed signals.

On the other hand, the results of the IS distance and human subject evalua-

tions agree that Our Processed signals performed poorer than the MBSS Processed signals and the Unprocessed signals. To that effect, Our Processed V2 signals performed significantly better than the Unprocessed signals and slightly better than the MBSS Processed signals. This is encouraging, but it must be noted that the New Our Processed Our Processed V2 signals were trained and tested on the same data-set.

In conclusion, to further this line of work, the following should be performed. Soft segmentations (either as described in Sections 3.4 and 3.5 or as described in Subsection 5.3.8) must be created for the training images. The values of a, b, and c need to be trained band-dependently on the training images for both SNR values. The new masks and estimated clean spectrograms must be created. The clean signals need to be estimated. The distance between these new estimated clean speech signals and the clean speech signals needs to be compared to the other 3 distances. It is necessary to assess the validity of using the IS distance as a preliminary evaluation of the quality of the results. In particular, a more thorough comparison of the IS distance and other perceptual distances from recent literature to the human subject recognition scores must be performed.

Appendix A

Spectrogram Columns

We would like to prove that for a given signal value $x\left[\hat{n} = 2I - 3 + \hat{j}s - a\right]$ where $0 \leq \hat{j} \leq J - d$ and $0 \leq a < s$, there are exactly d columns of the spectrogram that are affected by the value of $x[\hat{n}]$. We would like to further prove that these column are exactly the columns of $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet, j]$ for which $\hat{j} \leq j < \hat{j} + d$.

The spectrogram is defined as

$$\left[\mathscr{S}^{w}_{L,I,s}x\right][i,j] = \begin{cases} 0, & \text{if } i = 0, \\ \left|\left[\mathscr{D}^{w}_{L,I,2I-3+js}x\right][i]\right|^2, & \text{otherwise,} \end{cases} \tag{A.1}$$

in terms of the Discrete Fourier Transform of windowed signals. As in Subsection 3.1.3, let $y_n[m] = x[n-m]\,w[m]$, for $0 \leq m \leq 2I-3$. Then, the Discrete Fourier Transform of a windowed signal x is defined as

$$\left[\mathscr{D}^{w}_{L,I,n}x\right][i] = \left[\mathscr{D}_I y_n\right][i], \tag{A.2}$$

in terms of the DFT of signal y_n, which is defined as

$$\left[\mathscr{D}_I y_n\right][i] = \frac{1}{\sqrt{2I-2}} \sum_{m=0}^{2I-3} y_n[m] \exp\left(-\frac{2\pi\sqrt{-1}mi}{2I-2}\right). \tag{A.3}$$

Putting this together, we see that a column $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,\hat{j}]$ is affected by the values of $x[n]$ such that $\hat{j}s \le n \le 2I-3+\hat{j}s$, see Figure 3.1. Now, let us fix $0 \le \hat{j} \le J-d$ and $\hat{n} = 2I-3+\hat{j}s-\hat{a}$ where $0 \le \hat{a} < s$.

For $j < \hat{j}$, a column of $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$ is affected by the values of $x[n]$ for which

$$js \le n \le 2I-3+js < 2I-3+\hat{j}s-\hat{a} = \hat{n}. \tag{A.4}$$

Since $\hat{n}$ is not in the range of the ns that affect the column $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$, the column is not affected by the value of $x[\hat{n}]$.

For $\hat{j}+d \le j$, a column of $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$ is affected by the values of $x[n]$ for which

$$2I-3+js \ge n \ge js > \left(\hat{j}+d\right)s-1-\hat{a} = ds-1+\hat{j}s-\hat{a} = 2I-3+\hat{j}s-\hat{a} = \hat{n}. \tag{A.5}$$

Since $\hat{n} = 2I-3+\hat{j}s-\hat{a}$ is not in the range of the ns that affect the column $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$, the column is not affected by the value of $x[\hat{n}]$.

Finally for $\hat{j} \le j < \hat{j}+d$, a column of $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$ is affected by the values of $x[n]$ for which

$$js \le n \le 2I-3+js. \tag{A.6}$$

Since $\hat{n}$ is in the range of the ns that affect the column $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$,

$$js \le \left(\hat{j}+d\right)s-1-\hat{a} = ds-1+\hat{j}s-\hat{a} = 2I-3+\hat{j}s-\hat{a} = \hat{n} \le 2I-3+js, \tag{A.7}$$

these d columns $\left[\mathscr{S}^{w}_{L,I,s}x\right][\bullet,j]$ are affected by the value of $x[\hat{n}]$.

Appendix B

Histogram Delimiters

Figure B.1 displays range graphs for the 5 dB SNR level, one for each of the five bands. Figure B.2 displays range graphs for the 0 dB SNR level, one for each of the five bands. Each range graph displays the noisy spectrograms values on the horizontal axis and the bin numbers on the vertical axis. For each bin number on the vertical axis, a horizontal line indicates the range of grayscale values in that bin. The range of the last bin is so much larger than the other bins that if displayed, the remaining bins are not viewable. For that reason, all but the last bin are displayed in these range graphs. Notice that the ranges generally increase with the bin number.

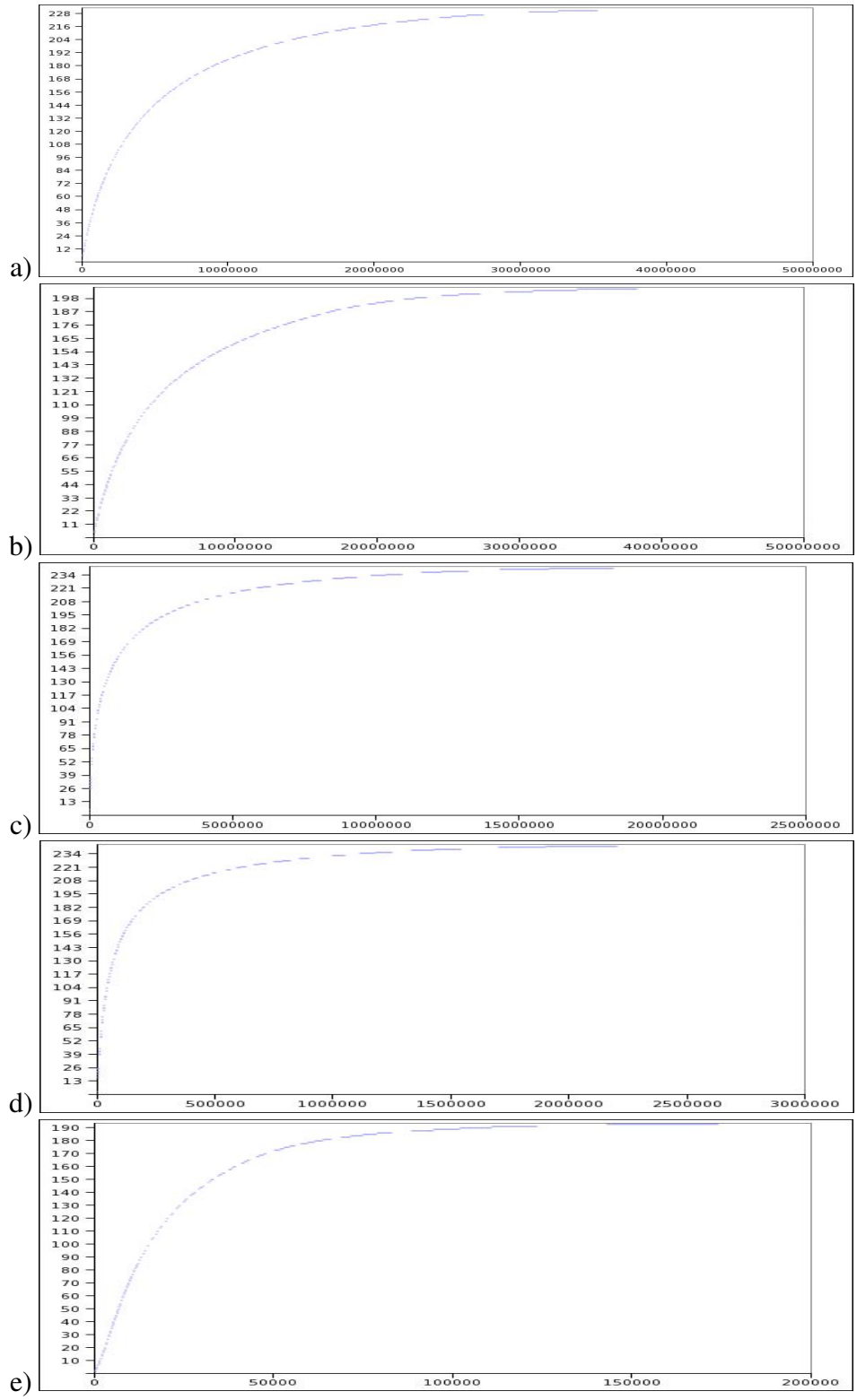


Figure B.1: The range graphs of the histogram bin delimiters for the 5 dB SNR level in a) band 0, b) band 1, c) band 2, d) band 3, and e) band 4.

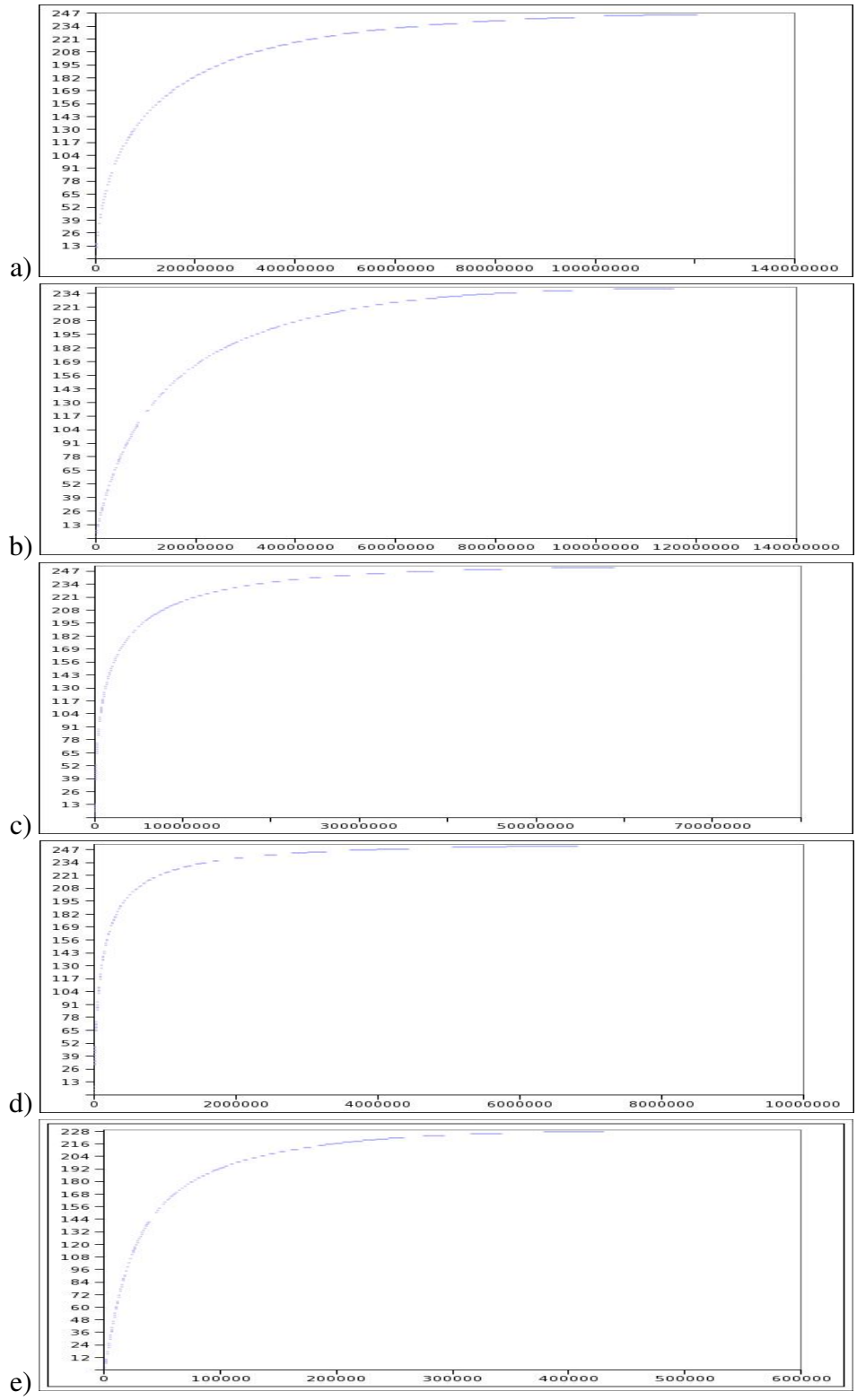


Figure B.2: The range graphs of the histogram bin delimiters for the 0 dB SNR level in a) band 0, b) band 1, c) band 2, d) band 3, and e) band 4.

Appendix C

Proof of Equations 3.31 and 3.32

Let us first prove Equation 3.32,

$$\begin{gathered} p\left(\omega\left[h\right]=\mu|\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right) \\ = \\ \frac{p\left(\omega\left[h\right]=\mu,N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=\mu;b\left[h\right]\right)}{p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right)}, \end{gathered} \tag{C.1}$$

For convenience, let us first change notations. Let $\omega\left[h\right]=\mu$ be event A, $\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q$ be event B, and $N_{\omega}\left[h\right]$ be event C. Therefore, we need to prove that

$$p\left(A|B,C;b\left[h\right]\right)=\frac{p\left(A,C;b\left[h\right]\right)p\left(B|A;b\left[h\right]\right)}{p\left(B,C;b\left[h\right]\right)}. \tag{C.2}$$

Notice that $b\left[h\right]$ is a "dummy" variable. Therefore, we actually need to prove that

$$p\left(A|B,C\right)=\frac{p\left(A,C\right)p\left(B|A\right)}{p\left(B,C\right)}. \tag{C.3}$$

By definition,

$$p\left(A|B,C\right)=\frac{p\left(A,B,C\right)}{p\left(B,C\right)}. \tag{C.4}$$

Since by definition it is also the case that $p(A,B,C) = p(A,C)\,p(B|A,C)$, we have that

$$p(A|B,C) = \frac{p(A,C)\,p(B|A,C)}{p(B,C)}. \tag{C.5}$$

In our new notation, Equation 3.33 tells us that $p(B|A,C) = p(B|A)$. Therefore,

$$p(A|B,C) = \frac{p(A,C)\,p(B|A)}{p(B,C)}, \tag{C.6}$$

which is what we wanted to prove.

Next, in order to prove Equation 3.31,

$$\begin{gathered} p(\omega[h] = \mu | \text{bin}[\theta[h]; b[h]] = q, N_\omega[h]; b[h]) \\ = \\ \frac{p(\omega[h] = \mu | N_\omega[h]; b[h])\, p(\text{bin}[\theta[h]; b[h]] = q | \omega[h] = \mu; b[h])}{p(\text{bin}[\theta[h]; b[h]] = q | N_\omega[h]; b[h])}, \end{gathered} \tag{C.7}$$

we need to prove that

$$p(A|B,C;b[h]) = \frac{p(A|C;b[h])\,p(B|A;b[h])}{p(B|C;b[h])}. \tag{C.8}$$

Notice that $b[h]$ is a "dummy" variable. Therefore, we actually need to prove that

$$p(A|B,C) = \frac{p(A|C)\,p(B|A)}{p(B|C)}. \tag{C.9}$$

We proved that

$$p(A|B,C) = \frac{p(A,C)\,p(B|A)}{p(B,C)}. \tag{C.10}$$

By definition, $p(A,C) = p(A|C)\,p(C)$ and $p(B,C) = p(B|C)\,p(C)$. Therefore,

$$p(A|B,C) = \frac{p(A|C)\,p(C)\,p(B|A)}{p(B|C)\,p(C)} = \frac{p(A|C)\,p(B|A)}{p(B|C)}, \tag{C.11}$$

which is what we wanted to prove.

Appendix D

Proof of Equations 3.37 and 3.41

Let us first prove Equation 3.37,

$$\begin{aligned} & p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|N_{\omega}\left[h\right];b\left[h\right]\right) \\ & = \\ & p\left(\omega\left[h\right]=0|N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=0;b\left[h\right]\right) \\ & +p\left(\omega\left[h\right]=1|N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=1;b\left[h\right]\right). \end{aligned} \tag{D.1}$$

For convenience, let us first change notations. Let $\omega\left[h\right]=0$ be event A_0, $\omega\left[h\right]=1$ be event A_1, $\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q$ be event B, and $N_{\omega}\left[h\right]$ be event C. Therefore, we need to prove that

$$p\left(B|C;b\left[h\right]\right)=p\left(A_0|C;b\left[h\right]\right)p\left(B|A_0;b\left[h\right]\right)+p\left(A_1|C;b\left[h\right]\right)p\left(B|A_1;b\left[h\right]\right). \tag{D.2}$$

Since $\omega\left[h\right]$ can only equal 0 or 1, this can be restated as

$$p\left(B|C;b\left[h\right]\right)=\sum_{A}p\left(A|C;b\left[h\right]\right)p\left(B|A;b\left[h\right]\right), \tag{D.3}$$

which is clearly true.

Equation 3.41 states that

$$\begin{aligned} & p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q,N_{\omega}\left[h\right];b\left[h\right]\right) \\ & = \\ & p\left(\omega\left[h\right]=0,N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=0;b\left[h\right]\right) \\ & +p\left(\omega\left[h\right]=1,N_{\omega}\left[h\right];b\left[h\right]\right)p\left(\text{bin}\left[\theta\left[h\right];b\left[h\right]\right]=q|\omega\left[h\right]=1;b\left[h\right]\right). \end{aligned} \tag{D.4}$$

As above, we must prove that

$$p\left(B,C;b\left[h\right]\right)=\sum_{A}p\left(A,C;b\left[h\right]\right)p\left(B|A;b\left[h\right]\right). \tag{D.5}$$

Notice that $b\left[h\right]$ is a "dummy" variable. Therefore, what needs to be proved is

$$p\left(B,C\right)=\sum_{A}p\left(A,C\right)p\left(B|A\right). \tag{D.6}$$

It is clearly the case that

$$p\left(B,C\right)=\sum_{A}p\left(A,B,C\right). \tag{D.7}$$

Since by definition $p\left(A,B,C\right)=p\left(A,C\right)p\left(B|A,C\right)$,

$$p\left(B,C\right)=\sum_{A}p\left(A,C\right)p\left(B|A,C\right). \tag{D.8}$$

Equation 3.33 tells us that $p\left(B|A,C\right)=p\left(B|A\right)$. Therefore,

$$p\left(B,C\right)=\sum_{A}p\left(A,C\right)p\left(B|A\right), \tag{D.9}$$

which is what we wanted to prove.

Appendix E

Modified Rhyme Test Words

Table E.1 contains the 300 MRT words. The six columns correspond to the six lists. The 50 rows correspond to the 50 foil sets.

Table E.1: Modified Rhyme Test Word Lists

went	sent	bent	dent	tent	rent
hold	cold	told	fold	sold	gold
pat	pad	pan	path	pack	pass
lane	lay	late	lake	lace	lame
kit	bit	fit	hit	wit	sit
must	bust	gust	rust	dust	just
teak	team	teal	teach	tear	tease
din	dill	dim	dig	dip	did
bed	led	fed	red	wed	shed
pin	sin	tin	fin	din	win
dug	dung	duck	dud	dub	dun
sum	sun	sung	sup	sub	sud
seep	seen	seethe	seek	seem	seed

not	tot	got	pot	hot	lot
vest	test	rest	best	west	nest
pig	pill	pin	pip	pit	pick
back	bath	bad	bass	bat	ban
way	may	say	pay	day	gay
pig	big	dig	wig	rig	fig
pale	pace	page	pane	pay	pave
cane	case	cape	cake	came	cave
shop	mop	cop	top	hop	pop
coil	oil	soil	toil	boil	foil
tan	tang	tap	tack	tam	tab
fit	fib	fizz	fill	fig	fin
same	name	game	tame	came	fame
peel	reel	feel	eel	keel	heel
hark	dark	mark	bark	park	lark
heave	hear	heat	heal	heap	heath
cup	cut	cub	cuff	cuss	cud
thaw	law	raw	paw	jaw	saw
pen	hen	men	then	den	ten
puff	puck	pub	pus	pup	pun
bean	beach	beat	beak	bead	beam
heat	neat	feat	seat	meat	beat
dip	sip	hip	tip	lip	rip
kill	kin	kit	kick	king	kid
hang	sang	bang	rang	fang	gang
took	cook	look	hook	shook	book

mass	math	map	mat	man	mad
ray	raze	rate	rave	rake	race
save	same	sale	sane	sake	safe
fill	kill	will	hill	till	bill
sill	sick	sip	sing	sit	sin
bale	gale	sale	tale	pale	male
wick	sick	kick	lick	pick	tick
peace	peas	peak	peach	peat	peal
bun	bus	but	bug	buck	buff
sag	sat	sass	sack	sad	sap
fun	sun	bun	gun	run	nun

Bibliography

[1] http://www.industrialacoustics.com/usa/people_envrionments/index.asp (Accessed on 9/4/08).

[2] http://www.telephonics.com/products/audio.asp (Accessed on 9/4/08).

[3] J. Agnew and J. M. Thornton. Just noticeable and objectionable group delays in digital hearing aids. *Journal of the American Academy of Audiology*, 11:330–336, 2000.

[4] American National Standards Institute (ANSI). American national standard specification for audiometers, ANSI 3.6-2004, 2004.

[5] J. P. Barker, M. P. Cooke, and D. P. W. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45:5–25, 2005.

[6] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.

[7] R. C. Bilger, J. M. Nuetzel, W. M. Rabinowitz, and C. Rzeczkowski. Standardization of a test of speech perception in noise. *Journal of Speech and Hearing Research*, 27:32–48, 1984.

[8] P. Boersma and D. Weenink. Praat: Doing phonetics by computer (version 4.6.06) [computer program]. http://www.praat.org (Accessed on 9/4/08).

[9] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27:113–120, 1979.

[10] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*. Springer, 1999.

[11] P. Brémaud. *Mathematical Principles of Signal Processing: Fourier and Wavelet Analysis*. Springer-Verlag, 2001.

[12] D. Burshtein and S. Ganot. Speech enhancement using a mixture-maximum model. *IEEE Transactions on Speech and Audio Processing*, 10:341–351, 2002.

[13] B. M. Carvalho, G. T. Herman, S. Matej, C. Salzberg, and E. Vardi. Binary tomography for triplane cardiography. In A. Kuba, M. Samal, and A. Todd-Pokropek, editors, *Information Processing in Medical Imaging*, pages 29–41. Springer-Verlag, 2003.

[14] M. Chan, G. T. Herman, and E. Levitan. Bayesian image reconstruction using a high-order interacting MRF model. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Image Analysis and Processing*.

[15] O. Christensen. Frames, Riesz bases, and discrete Gabor/wavelet expansions. *Bulletin of the American Mathematical Society*, 38:273–291, 2001.

[16] B. H. Cohen. *Explaining Psychological Statistics*. Wiley, third edition, 2007.

[17] M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, 34:267–285, 2001.

[18] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, 1999.

[19] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[20] S. Demange, C. Cerisara, and J.P. Haton. Mask estimation from missing data recognition using background noise sniffing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing*, May 2006.

[21] M. Dendrinos, S. Bakamides, and G. Carayannis. Speech enhancement from noise: A regenerative approach. *Speech Communication*, 10:45–57, 1991.

[22] Y. Ephraim. A Bayesian estimation approach for speech enhancement using hidden Markov models. *IEEE Transactions on Signal Processing*, 40:725–725, 1992.

[23] Y. Ephraim. Statistical-model-based speech enhancement systems. *Proceedings of the IEEE*, 80:1526–1555, 1992.

[24] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics Speech and Signal Processing*, 32:1109–1121, 1984.

[25] Y. Ephraim and H. L. Van Trees. A signal subspace approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, 3:251–266, 1995.

[26] H. G. Feichtinger and T. Strohmer. *Gabor Analysis and Algorithms: Theory and Applications*. Birkhäuser, 1998.

[27] H. G. Feichtinger and T. Strohmer. *Advances in Gabor Analysis*. Birkhäuser, 2002.

[28] T. Fillon and J. Prado. Evaluation of an ERB frequency scale noise reduction for hearing aids: A comparative study. *Speech Communication*, 39:23–32, 2003.

[29] P. Fishburn, P. Schwander, L. Shepp, and R. J. Vanderbei. The discrete Radon transform and its approximate inversion via linear programming. *Discrete Applied Mathematics*, 75:39–61, 1997.

[30] J. Gnitecki and Z. M. K. Moussavi. Separating heart sounds from lung sounds. *IEEE Engineering in Medicine and Biology Magazine*, 26(1):20–29, 2007.

[31] S. J. Godsill and P. J. W. Rayner. A Bayesian approach to the restoration of degraded audio signal. *IEEE Transactions on Speech and Audio Processing*, 3:267–278, 1995.

[32] B. R. Gomberg, M. Fernandez-Sear, B. S. Zemel, P. K. Saha, E. Vardi, L. Hilaire, and F. W. Wehrli. Measurement of trabecular bone volume fraction in the proximal femur. *Proceedings of the International Society for Magnetic Resonance in Medicine*, 2002.

[33] K. H. Gröchenig. Acceleration of the frame algorithm. *IEEE Transactions on Signal Processing*, 41:3331–3340, 1993.

[34] S. Gustaffson, R. Martin, P. Jax, and P. Vary. A psychoacoustic approach to combined acoustic echo cancellation and noise reduction. *IEEE Transactions on Speech and Audio Processing*, 10:245–256, 2002.

[35] J. H. L. Hansen and M. A. Clements. Constrained iterative speech enhancement with application to speech recognition. *IEEE Transactions on Signal Processing*, 39:795–805, 1991.

[36] M. Hashimoto and H. Seki. Limitations of lip-reading advantage by desynchronizing visual and auditory information in speech. In *Third International Conference on Spoken Language Processing*, pages 1155–1158, September 1994.

[37] A. S. House, C. E. Williams, M. H. L. Hecker, and K. D. Kryter. Articulation-testing methods: Consonantal differentiation with a closed response set. *Journal of Acoustical Society of America*, 37:158–166, 1965.

[38] Y. Hu and P. C. Loizou. A comparative intelligibility study of single-microphone noise reduction algorithms. *Journal of the Acoustical Society of America*, 122:1777–1786, 2007.

[39] J. Jensen and J. H. L. Hansen. Speech enhancement using a constrained iterative sinusoidal model. *IEEE Transactions on Speech and Audio Processing*, 9:731–740, 2001.

[40] S. Kamath. A multi-band spectral subtraction method for speech enhancement. Master's thesis, University of Texas-Dallas, 2001.

[41] S. Kamath and P. Loizou. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In *Proceedings of the IEEE International Conference on Acoustics, Speech, Signal Processing*, May 2002.

[42] R. D. Kent and C. Read. *The Acoustic Analysis of Speech*. Singular, second edition, 2002.

[43] H. Levitt. Noise reduction in hearing aids: A review. *Journal of Rehabilitation Research and Development*, 38(1):111–121, 2001.

[44] J. S. Lim and A. V. Oppenheim. All-pole modeling of degraded speech. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26:197–210, 1978.

[45] J. S. Lim and A. V. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67:1586–1604, 1979.

[46] P. C. Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[47] C. Mackersie, A. C. Neuman, and H. Levitt. Response time and word recognition using a modified-rhyme monitoring task: List equivalency and time-order effects. *Ear & Hearing*, 20:515–520, 1999.

[48] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[49] R. J. McAulay and M. L. Malpass. Speech enhancement using a soft-decision noise suppression filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28:137–145, 1980.

[50] R. J. McAulay and T. F. Quatieri. Speech analysis synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34:744–754, 1986.

[51] N. A. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[52] F. Mosteller and C. Youtz. Tables of the Freeman-Tukey transformations for the binomial and Poisson distributions. *Biometrika*, 48:433–440, 1961.

[53] R. F. Mould. *Introductory Medical Statistics*. Institute of Physics Publishing, third edition, 1998.

[54] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, second edition, 1999.

[55] K. S. Pearsons, R. L. Bennet, and S. Fidell. Speech levels in various noise environments. Technical Report EPA-600/1-77-025, U.S. Environmental Protection Agency, 1977.

[56] J. W. Pitton, L. E. Atlas, and P. J. Loughlin. Applications of positive-time frequency distributions to speech processing. *IEEE Transactions on Speech and Audio Processing*, 2:554–566, 1994.

[57] T. F. Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. Prentice Hall, 2002.

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

[59] E. L. Ritman, R. A. Robb, and L. D. Harris. *Imaging Physiological Functioning: Experience with the Dynamic Spatial Reconstructor*. Praeger Publishers, 1985.

[60] N. Roman, D. Wang, and G. J. Brown. Speech segregation based on sound localization. *Journal of the Acoustical Society of America*, 114:2236–2252, 2003.

[61] S. Rosen and P. Howell. *Signals and Systems for Speech and Hearing*. Academic Press, 1991.

[62] S. T. Roweis. One microphone source separation. In *Neural Information Processing Systems*, pages 793–799, November 2000.

[63] H. Sameti and L. Deng. Nonstationary-state hidden Markov model representation of speech signals for speech enhancement. *Signal Processing*, 82:205–227, 2002.

[64] M. L. Seltzer, B. Raj, and R. M. Stern. A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition. *Speech Communication*, 43:379–393, 2004.

[65] M. A. Stone and B. C. J. Moore. Tolerable hearing aid delays. I. Estimation of limits imposed by auditory path alone using simulated hearing losses. *Ear & Hearing*, 20:182–192, 1999.

[66] M. A. Stone and B. C. J. Moore. Tolerable hearing aid delays. II. Estimation of limits imposed during speech production. *Ear & Hearing*, 23:325–338, 2002.

[67] M. A. Stone and B. C. J. Moore. Tolerable hearing aid delays. III. Effect on speech production and perception of across-frequency variation in delay. *Ear & Hearing*, 24:175–183, 2003.

[68] M. A. Stone and B. C. J. Moore. Tolerable hearing aid delays. IV. Effect on subjective disturbance during speech production by hearing-impaired subjects. *Ear & Hearing*, 26:225–235, 2005.

[69] M. Stridh, L. Sörnmo, C.J. Meurling, and B. Olsson. Sequential characterization of atrial tachyarrythmias based on ECG time-frequency analysis. *IEEE Transactions on Biomedical Engineering*, 51:100–114, 2004.

[70] P. T. Troughton and S. J. Godsill. MCMC methods for restoration of nonlinearly distorted autoregressive signals. *Signal Processing*, 81:83–97, 2001.

[71] E. Vardi and G. T. Herman. Stochastic segmentation using Gibbs priors. *Electronic Notes in Theoretical Computer Science*, 46:381–392, 2001.

[72] E. Vardi, G. T. Herman, and T. Y. Kong. Speeding up stochastic reconstructions of binary images from limited projection directions. *Linear Algebra and Its Applications*, 339:75–89, 2001.

[73] E. Vardi-Gonen and G. T. Herman. Sequential vs. simultaneous stochastic segmentation. *IEEE International Symposium on Biomedical Imaging*, pages 1327–1330, 2004. IEEE Catalog Number 04EX821C, ISBN 0-7803-8389-3.

[74] E. Vardi-Gonen and G. T. Herman. Sequential binary image estimation. *IEEE 31st Annual Northeast Biomedical Conference*, pages 127–128, 2005. IEEE Catalog Number 05CH3764CC, ISBN 0-7803-9106-3.

[75] E. Vardi-Gonen and G. T. Herman. Sequential binary image estimation algorithms. *IEEE Signal Processing Society - 12th Digital Signal Processing Workshop; 4th Signal Processing Education Workshop*, pages 494–499, 2006. IEEE Catalog Number 06EX1488C, ISBN 1-4244-0535-1.

[76] M. Weiss, E. Aschkenasy, and T. Parsons. Study and the development of the intel technique for improving speech intelligibility. Technical Report NSC-FR/4023, Nicolet Scientific Corporation, 1974.

[77] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. Zeidler, E. Dong, and R. C. Goodlin. Adaptive noise canceling: Principles and applications. *Proceedings of the IEEE*, 63:1692–1716, 1975.

[78] G. Winkler. *Image Analysis, Random Field and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Springer, second edition, 2006.

[79] V. W. Zue. The use of speech knowledge in automatic speech recognition. *Proceedings of the IEEE*, 73:1602–1615, 1985.

Index

Autobiographical Statement

I have been a student all my life so far. After high school, I enjoyed one wonderful year at Bryn Mawr College, Bryn Mawr, PA, USA. Despite loving it, I decided to move to Israel. I graduated in 1999 with an B.A. in Applied Mathematics from the Technion - Israel Institute of Technology, Haifa, Israel. During my stay in Israel, I met and married my husband Itamar Gonen. We moved back to the US together at the beginning of 2000. I became a student with Dr. Gabor Herman in the Bioengineering Department at the University of Pennsylvania, Philadelphia, PA, USA, while my husband went on to complete his B.Sc. in Computer Engineering from Drexel University, Philadelphia, PA, USA. In 2002, I completed my M.S.E. in Bioengineering from the University of Pennsylvania. Dr. Herman became a Distinguished Professor at The Graduate Center, CUNY, New York, NY, USA. I joined him at the GC and have been working to complete my Ph.D. there since. In 2004, we gave birth to a wonderful son named Nathan. In 2007, we gave birth to a girl named Nadine who lived for only 12 days. 2008 will serve as the year "I finally finished my Ph.D."