CODING OF SPEECH AND IMAGE SIGNALS USING GABOR DECOMPOSITION

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Emre Gündüzhan

July 1994

CODING OF SPEECH AND IMAGE SIGNALS USING GABOR DECOMPOSITION

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Emre Gündüzhan

July 1994

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

A. E. Çetin

Assoc. Prof. Dr. A. Enis Çetin(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

O. Arıkan

Assist. Prof. Dr. Orhan Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Billur Barshan

Assist. Prof. Dr. Billur Barshan

Approved for the Institute of Engineering and Sciences:

M. Baray

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

CODING OF SPEECH AND IMAGE SIGNALS USING GABOR DECOMPOSITION

Emre Gündüzhan
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. A. Enis Çetin
July 1994

A new low bit rate speech coding method which uses Gabor time-frequency decomposition and the matching pursuit algorithm is developed. A new algorithm based on the projections onto convex sets method is used to smooth the discontinuities between speech frames. A two-dimensional extension of the Gabor time-frequency decomposition is also developed for image coding. Simulation examples are presented.

Keywords : Speech coding, image coding, time-frequency dictionaries, matching pursuit algorithm.

ÖZET

GABOR AÇILIMI KULLANARAK SÖZ VE İMGE SİNYALLERİNİN KODLANMASI

Emre Gündüzhan
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. A. Enis Çetin
Temmuz 1994

Gabor zaman-sıklık açılımı ve karşılaştırmalı takip algoritması kullanılarak yeni bir az bitle söz kodlama tekniği geliştirildi. Söz çerçeveleri arasındaki süreksizliğin azaltılması için içbükey kümeler üzerine izdüşüm tekniğine dayanan yeni bir algoritma kullanıldı. Gabor zaman-sıklık açılımının iki boyuta genellemesi yapıldı ve imge kodlama için kullanıldı. Benzetim çalışmaları yapıldı.

Anahtar kelimeler : Söz kodlama, imge kodlama, zaman-sıklık sözlükleri, karşılaştırmalı takip algoritması.

ACKNOWLEDGMENT

I would like to thank Assoc. Prof. Dr. A. Enis Çetin for his supervision, guidance, suggestions and encouragement throughout the development of this thesis.

Contents

1 Introduction 1

1.1 Notation 2

1.2 Time-Frequency Atomic Decompositions 3

1.3 Matching Pursuit in Finite Spaces 3

1.4 Matching Pursuit with Gabor Time-Frequency Dictionaries . . . 6

2 Speech Coding 9

2.1 Reducing Discontinuities Using the Method of Projections Onto Convex Sets 12

2.2 Simulation Examples 14

2.3 Computational Complexity of the New Speech Coding Method . 16

3 Image Coding 20

3.1 Simulation Examples 22

4 Conclusion 26

List of Figures

1.1 *(a) A discrete Gabor atom with parameters* $(s, p, k, \phi) = (50, 80, 25, \frac{\pi}{4})$ *and length* $N = 200$ *and (b) the magnitude of its Fourier transform.* 8

2.1 *The 4 bit uniform quantizer used to quantize the angles.* 14

2.2 *(a) An original speech signal, (b) the signal after coding/decoding using the Gabor time-frequency decomposition, and (c) the signal after coding/decoding using an LPC-10 vocoder.* 18

2.3 *(a) A voiced speech segment, (b) the coded/decoded segment having two discontinuities, and (c) the same segment after applying the POCS based algorithm twice.* 19

2.4 *(a) An unvoiced speech segment, (b) the coded/decoded segment having two discontinuities, and (c) the same segment after applying the POCS based algorithm twice.* 19

3.1 *The original (a) and the reconstructed (b) Barbara images.* 24

3.2 *The comparison of the matching pursuit and JPEG algorithms using the Barbara image.* 25

3.3 *The comparison of the matching pursuit and JPEG algorithms using the Lena image.* 25

List of Tables

2.1 *The 6 bit μ-law quantizer used to quantize the inner products.* . . 15

3.1 *The compression results for the Barbara image.* 23

3.2 *The compression results for the Lena image.* 23

Chapter 1

Introduction

Signal decompositions over a family of functions are widely used in signal processing. The family of functions used in such a decomposition is called a dictionary. The main advantage of signal decomposition methods is to represent a given signal by a countable set of coefficients. Most commonly used decompositions are the Short-time Fourier transform (STFT) [2], the Wigner-Ville transform [3], and the wavelet transforms [4], [5]. The family of functions in a dictionary are usually linearly independent and they form an orthonormal basis. Using a basis as a dictionary of a decomposition may result in the smallest possible dictionary for the exact reconstruction of the signal, however, it is difficult to extract some time-frequency properties of a signal from such an expansion. For example, the decomposition of signals well localized in time over a Fourier basis or the decomposition of signals having a narrow high-frequency support over a wavelet basis results in a poor representation. These facts motivate decompositions over large and redundant dictionaries. Instead of using a single basis in this thesis we select functions, which are most useful to represent our signal, from a redundant dictionary and decompose the signal over these functions.

Using a large dictionary is especially useful in the representation of a signal whose localization in time and frequency vary widely. In this case using a single basis will not usually be sufficient for a good representation and the dictionary should contain functions which are well-localized both in time and in frequency. The elements of such a dictionary are called the time-frequency atoms. The decomposition of a signal over a redundant dictionary is adaptive. A signal is represented using a chosen subset of the dictionary such that the atoms in the subset are best adapted to the signal properties. There is an algorithm, called the matching pursuit, which can perform such an adaptive decomposition over families of functions [1]. There is a similar algorithm developed by Qian and Chen to expand signals over a time-frequency dictionary [6].

In this thesis a time-frequency atom family, called the Gabor dictionary [1], is used in speech coding. A low bit rate speech vocoder which uses a time-frequency decomposition is proposed. The same method is also applied to image coding using two-dimensional atoms. In the next section we introduce the time-frequency atoms and the matching pursuit algorithm which are studied in detail in [1]. In Chapter 2, a speech coding method using Gabor time-frequency atoms is described and a new algorithm to smooth the discontinuities between speech frames is developed. Some simulation examples are also given in Chapter 2. In Chapter 3, two-dimensional Gabor atoms are developed and an image coding algorithm is given. Some simulation results for image coding are also shown in this chapter.

1.1 Notation

Consider the Hilbert space $\mathbf{L^2(C)}$ of complex valued functions, f, which satisfy

$$\int_{-\infty}^{\infty} | f(t) |^2 \, dt < \infty. \tag{1.1}$$

For any $f, g \in \mathbf{L^2(C)}$, the inner product is defined by

$$< f, g >= \int_{-\infty}^{\infty} f(t) g^*(t) dt \tag{1.2}$$

where $g^*(t)$ is the complex conjugate of $g(t)$. The continuous-time Fourier transform of $f \in \mathbf{L^2(C)}$ is defined by

$$\hat{f}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-jwt} dt. \tag{1.3}$$

Let $\mathbf{H}$ be the inner product space of complex valued discrete-time functions periodic with N. We define the inner product of $f, g \in \mathbf{H}$ by

$$< f, g >= \sum_{n=0}^{N-1} f[n] g^*[n]. \tag{1.4}$$

For any $a \to 0$ (or ∞), any quantity that is bounded by a constant times a is denoted by $O(a)$.

1.2 Time-Frequency Atomic Decompositions

In general, a family of time-frequency atoms can be generated by scaling, shifting, and modulating a window function $g(t) \in \mathbf{L}^2(\mathbf{R})$. We assume that $g(t)$ is real, continuously differentiable, has unity norm, and $\mid g(t) \mid$ is $O(\frac{1}{t^2+1})$. A time-frequency atom is defined by

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g(\frac{t-u}{s}) e^{j\varepsilon t} \tag{1.5}$$

where $\gamma = (s, u, \varepsilon)$ which are the parameters of γ. The parameter s is a scaling factor, u is a shift term and ε is the modulating index. Here γ is an element of the set $\Gamma = \mathbf{R}^+ \times \mathbf{R}^2$. The energy of $g_\gamma(t)$ is concentrated in a neighborhood of u, whose time window size is proportional to s. If we take the Fourier transform of both sides in (1.5) then we get

$$\hat{g}_\gamma(w) = \sqrt{s}\hat{g}(s(w-\varepsilon))e^{-j(w-\varepsilon)u}. \tag{1.6}$$

The energy of $\hat{g}_\gamma(w)$ is concentrated in a neighborhood of ε, whose frequency window size is proportional to s^{-1}.

The dictionaries in a window Fourier transform and a wavelet transform are special cases of (1.5). In a window Fourier transform the parameter s is fixed, so the time scale does not change [2] and in a wavelet transform s and ε are inversely proportional to each other [4].

In a time-frequency atomic decomposition the aim is to select a countable subset of atoms $(g_{\gamma_n}(t))_{n \in \mathrm{N}}$ from the dictionary and express the signal, $f(t)$, as a weighted sum of these atoms, i.e.,

$$f(t) = \sum_{n=0}^{\infty} a_n g_{\gamma_n}(t). \tag{1.7}$$

In the next section we describe an algorithm, called the matching pursuit, which can carry such a decomposition.

1.3 Matching Pursuit in Finite Spaces

The dictionary used in a time-frequency decomposition is redundant. In order to decompose a signal "optimally" we must choose atoms which best match the signal properties. The matching pursuit algorithm can be used for such an adaptive decomposition.

Let $\mathbf{H}$ be a signal space with a finite dimension N. We define a dictionary $\mathcal{D} = (g_\gamma)_{\gamma \in \Gamma}$ as a family of vectors in $\mathbf{H}$, such that $\| g_\gamma \| = 1$ for all $\gamma \in \Gamma$. Let $\mathbf{V}$ be the closed linear span of vectors in $\mathcal{D}$. We assume that the dictionary $\mathcal{D}$ is complete, i.e., $\mathbf{V} = \mathbf{H}$.

Let $f \in \mathbf{H}$. We want to get the closest approximation of f in norm by one of the vectors in $\mathcal{D}$. Clearly, we must choose $g_{\gamma_0} \in \mathcal{D}$ such that $|< f, g_{\gamma_0} >|$ is maximum. In some cases, it is only possible to find a vector g_{γ_0} satisfying

$$|< f, g_{\gamma_0} >| \geq \alpha \sup_{\gamma \in \Gamma} |< f, g_\gamma >| \quad , \quad 0 < \alpha \leq 1. \tag{1.8}$$

Next, f is decomposed as

$$f = < f, g_{\gamma_0} > g_{\gamma_0} + Rf, \tag{1.9}$$

where Rf is the residual vector. Since $g_{\gamma_0} \perp Rf$,

$$\| f \|^2 = |< f, g_{\gamma_0} >|^2 + \| Rf \|^2 . \tag{1.10}$$

The matching pursuit algorithm sub-decomposes the residual vector Rf on a vector in $\mathcal{D}$ and this procedure is repeated for each new residual vector. Let $R^0 f = f$. Suppose the n^{th} order residue, $R^n f$, was computed for some $n \geq 0$. We choose a vector $g_{\gamma_n} \in \mathcal{D}$ which satisfies

$$|< R^n f, g_{\gamma_n} >| \geq \alpha \sup_{\gamma \in \Gamma} |< R^n f, g_\gamma >| . \tag{1.11}$$

Then, we sub-decompose $R^n f$ into

$$R^n f = < R^n f, g_{\gamma_n} > g_{\gamma_n} + R^{n+1} f \tag{1.12}$$

to obtain the $(n+1)^{\text{st}}$ residue. If we carry this decomposition up to the order m, then we obtain an m^{th} order approximation of f, i.e.,

$$f = \sum_{n=0}^{m-1} < R^n f, g_{\gamma_n} > g_{\gamma_n} + R^m f. \tag{1.13}$$

Since $R^{n+1} f \perp g_{\gamma_n}$ for all $0 \leq n < m$, we get an energy conservation equation

$$\| f \|^2 = \sum_{n=0}^{m-1} |< R^n f, g_{\gamma_n} >|^2 + \| R^m f \|^2 . \tag{1.14}$$

The following theorem proved in [1] shows the convergence of this algorithm.

Theorem 1.1 *If the dictionary $\mathcal{D}$ is complete then $(g_{\gamma_n})_{n\geq 0}$ and $(R^n f)_{n\geq 0}$ defined inductively by (1.11) and (1.12) satisfy*

$$f = \sum_{n=0}^{\infty} < R^n f, g_{\gamma_n} > g_{\gamma_n} \tag{1.15}$$

and

$$\| f \|^2 = \sum_{n=0}^{\infty} |< R^n f, g_{\gamma_n} >|^2 \tag{1.16}$$

for any $f \in \mathbf{H}$.

If we stop the iterations at an order m then the decomposition is given by (1.13). Here, f is approximated with an error equal to $R^m f$. However, this approximation is not the best one can achieve using the vectors $(g_{\gamma_n})_{0\leq n<m}$. Let $\mathbf{V}_m$ be the span of $(g_{\gamma_n})_{0\leq n<m}$ and $\mathbf{P}_{\mathbf{V}_m}$ be the orthogonal projector on $\mathbf{V}_m$. The closest approximation of f using the vectors in $\mathbf{V}_m$ is given by

$$\mathbf{P}_{\mathbf{V}_m} f = \sum_{n=0}^{m-1} < R^n f, g_{\gamma_n} > g_{\gamma_n} + \mathbf{P}_{\mathbf{V}_m} R^m f. \tag{1.17}$$

In general, the family of vectors $(g_{\gamma_n})_{0\leq n<m}$ is not orthogonal. In this case $\mathbf{P}_{\mathbf{V}_m} R^m f \neq 0$. The computation of

$$\mathbf{P}_{\mathbf{V}_m} R^m f = \sum_{n=0}^{m-1} x_n g_{\gamma_n} \tag{1.18}$$

is called a backprojection. Using backprojection we can decompose f as

$$f = \sum_{n=0}^{m-1} (< R^n f, g_{\gamma_n} > + x_n) g_{\gamma_n} + \mathbf{P}_{\mathbf{W}_m} f \tag{1.19}$$

where $\mathbf{P}_{\mathbf{W}_m} f$ is the orthogonal projection of f onto the space $\mathbf{W}_m$, which is the orthogonal complement of $\mathbf{V}_m$ in $\mathbf{H}$. We have to solve the following linear system of equations to calculate the coefficients $(x_n)_{0\leq n<m}$. For any $0 \leq k < m$,

$$< R^m f, g_{\gamma_k} > = \sum_{n=0}^{m-1} x_n < g_{\gamma_n}, g_{\gamma_k} > . \tag{1.20}$$

If the vectors $(g_{\gamma_n})_{0\leq n<m}$ are linearly dependent then the solution is not unique, but we can always find a solution.

If the dictionary is very redundant the search for the vector g_{γ_n} in (1.11) can be limited to a sub-dictionary $\mathcal{D}_\alpha = (g_\gamma)_{\gamma\in\Gamma_\alpha} \subset \mathcal{D}$, where $\Gamma_\alpha \subset \Gamma$ is a finite index set such that for any $f \in \mathbf{H}$

$$\sup_{\gamma\in\Gamma_\alpha} |< f, g_\gamma >| \geq \alpha \sup_{\gamma\in\Gamma} |< f, g_\gamma >| . \tag{1.21}$$

Depending on the dictionary, Γ_α can be much smaller than Γ. At each iteration, instead of searching the whole dictionary, we search for a vector $g_{\gamma_n} \in \mathcal{D}_\alpha$ such that

$$|< R^n f, g_{\gamma_n} >| = \sup_{\gamma \in \Gamma_\alpha} |< R^n f, g_\gamma >| \,. \tag{1.22}$$

The rest of the algorithm remains the same.

1.4 Matching Pursuit with Gabor Time-Frequency Dictionaries

A time-frequency dictionary is a family of time-frequency atoms defined in (1.5). If a matching pursuit is used with a time-frequency dictionary, then any function $f \in \mathbf{L^2(C)}$ is decomposed into a sum of time-frequency atoms that best match its residues. It has been shown that a time-frequency dictionary is complete in $\mathbf{L^2(C)}$ [7]. Therefore, for any $f \in \mathbf{L^2(C)}$ a matching pursuit yields

$$f = \sum_{n=0}^{\infty} < R^n f, g_{\gamma_n} > g_{\gamma_n} \tag{1.23}$$

where $\gamma_n = (s_n, u_n, \varepsilon_n)$ and

$$g_{\gamma_n}(t) = \frac{1}{\sqrt{s_n}} g(\frac{t - u_n}{s_n}) e^{j\varepsilon_n t}. \tag{1.24}$$

If the window function, $g(t)$, is chosen as the normalized Gaussian function

$$g(t) = 2^{\frac{1}{4}} e^{-\pi t^2} \tag{1.25}$$

then the corresponding family of functions $(g_\gamma)_{\gamma \in \Gamma}$ is called the Gabor time-frequency dictionary.

In practice the matching pursuit algorithm has to be used in discrete-time. We suppose that we want to decompose a real discrete-time signal with N samples. Our vector space $\mathbf{H}$ is the set of all discrete signals periodic with N. The window function is again the normalized Gaussian given by (1.25). For each scale s we sample the window function and periodize it over N points to get

$$\tilde{g}_s[n] = \frac{K_s}{\sqrt{s}} \sum_{m=-\infty}^{\infty} g(\frac{n - mN}{s}) \tag{1.26}$$

where K_s normalizes the discrete norm of $\tilde{g}_s$. Then, for any integers $0 \leq p \leq N-1$ and $0 \leq k \leq N-1$ we denote $\gamma = (s, p, \frac{2\pi k}{N})$ and define the corresponding discrete complex Gabor atom as

$$g_\gamma[n] = \tilde{g}_s[n - p] e^{j\frac{2\pi k}{N} n}. \tag{1.27}$$

Since we want to decompose a real signal, we must use real time-frequency atoms to get real expansion coefficients. For any $\gamma = (s, p, \frac{2\pi k}{N})$ and a phase $\phi \in [0, 2\pi)$ the real discrete time-frequency atoms are given by

$$g_{(\gamma,\phi)}[n] = K_{(\gamma,\phi)}\tilde{g}_s[n-p]\cos(\frac{2\pi k}{N}n + \phi) \tag{1.28}$$

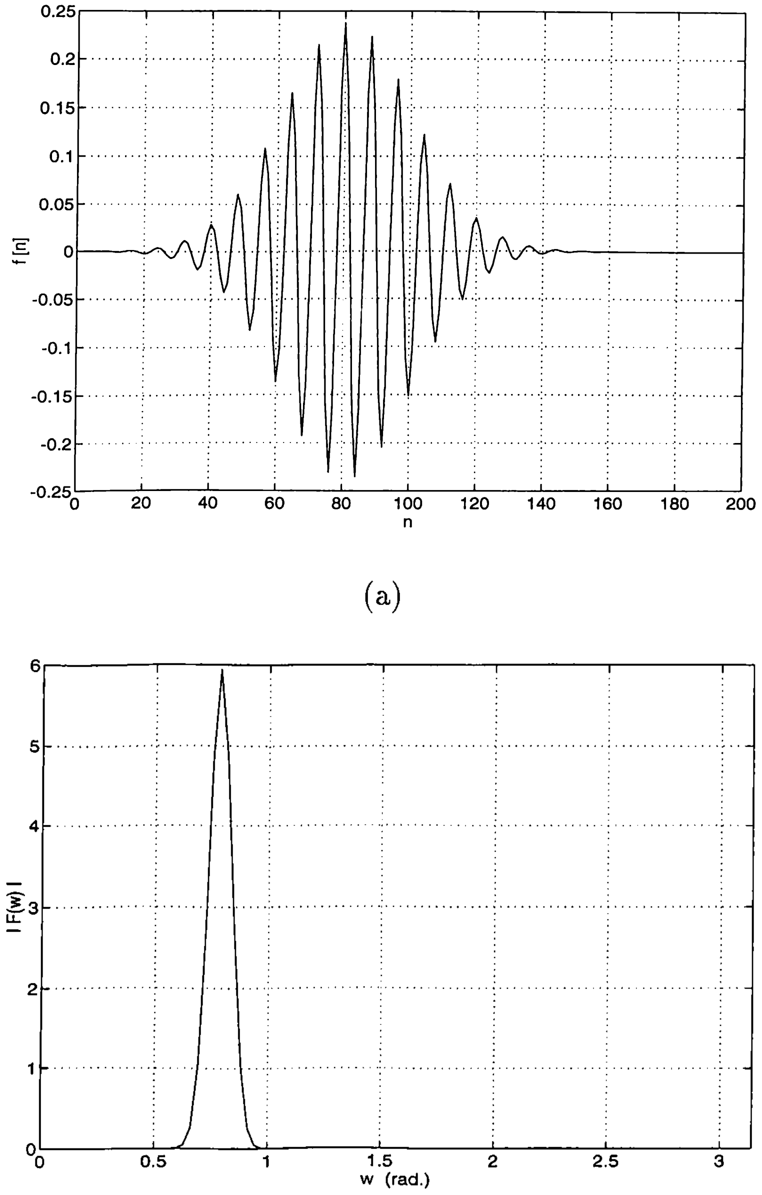
where $K_{(\gamma,\phi)}$ is the normalization constant to make $\| g_{(\gamma,\phi)} \| = 1$. The normalization constant is given by

$$K_{(\gamma,\phi)} = \frac{\sqrt{2}}{\sqrt{1 + Real\{e^{2j\phi} < g_\gamma, g_\gamma^* >\}}}. \tag{1.29}$$

For each $\gamma = (s, p, \frac{2\pi k}{N})$ and phase ϕ, the real and complex discrete atoms are related by

$$g_{(\gamma,\phi)}[n] = \frac{K_{(\gamma,\phi)}}{2}(e^{j\phi}g_\gamma[n] + e^{-j\phi}g_\gamma^*[n]). \tag{1.30}$$

A real discrete Gabor atom and its Fourier transform are shown in Figure 1.1 (a) and (b), respectively. The parameters of this atom are $(s, p, k, \phi) = (50, 80, 25, \frac{\pi}{4})$. It can be seen that its energy is concentrated around $n = p$ in the time domain and around $w = \frac{2\pi k}{N}$ in the frequency domain. In the next chapter we use the matching pursuit algorithm with discrete Gabor atoms in speech coding. In Chapter 3 a two-dimensional extension of the Gabor dictionary is used in image coding.



(a)

(b)

Figure 1.1: *(a) A discrete Gabor atom with parameters* $(s, p, k, \phi) = (50, 80, 25, \frac{\pi}{4})$ *and length* $N = 200$ *and (b) the magnitude of its Fourier transform.*

Chapter 2

Speech Coding

Speech coding is one of the main areas in speech processing. There are many algorithms which compress speech signals at bit rates ranging from 64 kbps to 2 kbps with varying degrees of speech quality [14]. Speech coders can be classified into two main groups: waveform coders and voice coders. Most commonly used waveform coders are pulse-code modulation (PCM), adaptive differential pulse-code modulation (ADPCM) and adaptive subband coding systems which can compress speech at typical bit rates of 64 kbps, 32 kbps and 16 kbps, respectively [8]. These coders produce an output speech of high quality, nearly indistinguishable from the original speech. Voice coders (vocoders) can compress speech signals at lower bit rates than waveform coders but they produce lower quality speech. Unlike waveform coders, vocoders extract some parameters of the speech signal and send these parameters instead of the signal itself. Some commonly used vocoders are the multipulse linear predictive coder [9], the code excited linear predictive coder [10], and the LPC vocoder [11]. These vocoders can compress speech signals at typical bit rates of 8 kbps, 4 kbps and 2 kbps, respectively. The output speech of multipulse and code excited linear predictive coders are of communication quality, i.e., there is some distortion but the intelligibility is very high. The output speech of the LPC vocoder is of synthetic quality, which means words are mostly intelligible but the speaker's identity can not always be distinguished and a metallic sounding speech is reconstructed at the decoder.

The speech coding algorithm that we developed is like a voice coder. It sends some parameters extracted from the speech signal. Like most vocoders the signal is divided into frames and each frame is processed separately. Each frame is approximated by Gabor time-frequency atoms whose parameters are transmitted to the decoder. An arbitrarily close approximation of the frame may require a large number of Gabor atoms. Depending on the bit rate finitely many atoms are selected in the approximation. As described in the previous chapter, each Gabor atom is characterized by three parameters and a phase. In our implementation we used a fourth order approximation and each speech frame is characterized by four Gabor atoms and the inner products of these atoms with the residual signals. These parameters are quantized and transmitted to the receiver. The receiver reconstructs the speech frame from these parameters.

Let $f[n]$ be a speech frame consisting of N samples. We start the algorithm by computing the inner product of f with all complex atoms in the dictionary. Then, at each step n, we find γ_n such that $|< R^n f, g_{\gamma_n} >|$ is maximum. It follows from (1.30) that for any real residue $R^n f$ and any phase ϕ

$$< R^n f, g_{(\gamma_n,\phi)} >= K_{(\gamma_n,\phi)} Real\{\mathrm{e}^{-j\phi} < R^n f, g_{\gamma_n} >\}. \qquad (2.1)$$

If we choose ϕ_n equal to the complex phase of $< R^n f, g_{\gamma_n} >$, then we obtain

$$< R^n f, g_{(\gamma_n,\phi_n)} >= K_{(\gamma_n,\phi_n)} \; |< R^n f, g_{\gamma_n} >| \, . \qquad (2.2)$$

Finally we compute the inner product of the new residue with the complex atoms using

$$< R^{n+1} f, g_\gamma >=< R^n f, g_\gamma > - < R^n f, g_{(\gamma_n,\phi_n)} >< g_{(\gamma_n,\phi_n)}, g_\gamma > . \qquad (2.3)$$

We already know $< R^n f, g_\gamma >$ and $< R^n f, g_{(\gamma_n,\phi_n)} >$ and we compute the last inner product using

$$< g_{(\gamma_n,\phi_n)}, g_\gamma >= \frac{K_{(\gamma_n,\phi_n)}}{2}(\mathrm{e}^{j\phi_n} < g_{\gamma_n}, g_\gamma > + \mathrm{e}^{-j\phi_n} < g^*_{\gamma_n}, g_\gamma >) \qquad (2.4)$$

which follows from (1.30). The inner product of two complex atoms is given by

$$\begin{aligned} < g_{\gamma_1}, g_{\gamma_2} > \;=\; & K_{s_1} K_{s_2} \sqrt{\tfrac{2 s_1 s_2}{s_1^2+s_2^2}} \exp(-j p_2 \tfrac{2\pi(k_2-k_1)}{N}) \times \\ & \textstyle\sum_{m=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} (\exp(-\pi \tfrac{(p_2-p_1+mN)^2}{s_1^2+s_2^2}) \exp(-\pi \tfrac{(k_2-k_1+qN)^2}{N^2(s_1^{-2}+s_2^{-2})}) \times \\ & \exp(j \tfrac{s_2^2}{s_1^2+s_2^2} \tfrac{2\pi}{N} (k_2 - k_1 + qN)(p_2 - p_1 + mN))). \end{aligned} \qquad (2.5)$$

If the values of the exponential functions are tabulated, the above formula yields an efficient computation of the inner products. Note that since the two exponentials decay very fast with m and q it is sufficient to use only a few terms of the double summation. The overall algorithm can be summarized as follows:

Initialization: Compute $< R^0 f, g_\gamma >=< f, g_\gamma >$ for all $\gamma \in \Gamma_\alpha$.

$(n+1)^{st}$ step: Assume $< R^n f, g_\gamma >$ is known for all $\gamma \in \Gamma_\alpha$.

i. Find $\gamma_n = \arg\max_{\gamma \in \Gamma_\alpha} |< R^n f, g_\gamma >|$.

ii. Set $\phi_n = angle\{< R^n f, g_{\gamma_n} >\}$.

iii. Compute $K_{(\gamma_n, \phi_n)}$ using (1.29).

iv. $< R^n f, g_{(\gamma_n,\phi_n)} >= K_{(\gamma_n,\phi_n)} \, |< R^n f, g_{\gamma_n} >|$.

v. Using (2.4) and (2.5) compute $< g_{(\gamma_n,\phi_n)}, g_\gamma >$ for all $\gamma \in \Gamma_\alpha$.

vi. Using (2.3) compute $< R^{n+1} f, g_\gamma >$ for all $\gamma \in \Gamma_\alpha$.

The residual signal remaining after the algorithm is terminated can have a component in the subspace spanned by the selected atoms (g_{γ_n}). Therefore the approximation obtained by the algorithm is not the best one can achieve with (g_{γ_n}). Using backprojection as explained in the previous chapter we improve the approximation and store the new parameters

$$a_n =< R^n f, g_{(\gamma_n,\phi_n)} > + x_n \tag{2.6}$$

where x_n are calculated using (1.20).

For each selected atom the three parameters (s_n, p_n, k_n) are indexed, the phase ϕ_n is uniformly quantized and the parameter a_n is μ-law coded. All of these parameters are sent to the receiver. The receiver reconstructs the frame by

$$\hat{f} = \sum_{n=0}^{M-1} a_n g_{(\gamma_n,\phi_n)} \tag{2.7}$$

where M is the approximation order.

Since we process the speech signal frame by frame, there exist discontinuities between frames in the reconstructed signal especially at low bit rates. This effect can be reduced using overlapping frames. Each frame is windowed before processing and the receiver overlaps and adds the successive reconstructed frames. However, this method uses frames of larger length, hence, the speech quality is decreased. Alternatively, we use a method based on the projections onto convex sets algorithm. A similar method is used to remove blocking effect in JPEG-coded images [12], [13].

2.1 Reducing Discontinuities Using the Method of Projections Onto Convex Sets

In the coding method described in the previous section each speech frame is coded separately. This results in large discontinuities in the output speech especially at low bit rates. These discontinuities can be removed by low-pass filtering the output speech, but then some high frequencies present in the original signal are lost. In this section we develop an iterative algorithm based on the method of projections onto convex sets (POCS) [15], [16] which reduces the discontinuities considerably without affecting the high frequency components of the original signal much.

We define two convex sets to use the POCS algorithm. Since the discontinuities correspond to high frequencies, we choose the first set, C_1, as the set of signals bandlimited to a frequency W. It is well-known that C_1 is a closed convex set [17].

Let f_s be the speech signal of length L and

$$L = KN \tag{2.8}$$

where N is the frame length and K is the number of frames. Let $f_s^{(i)}$ denote the i^{th} frame of f_s. Our coding algorithm decomposes each frame $f_s^{(i)}$ as

$$f_s^{(i)} = \mathbf{P}_{\mathbf{V}_M^{(i)}} f_s^{(i)} + \mathbf{P}_{\mathbf{W}_M^{(i)}} f_s^{(i)} \quad ; \quad i = 1, \ldots, K \tag{2.9}$$

where $\mathbf{V}_M^{(i)}$ is the subspace spanned by $(g^{(i)}_{(\gamma_n,\phi_n)})_{0 \le n < M}$, $\mathbf{W}_M^{(i)}$ is the orthogonal complement of $\mathbf{V}_M^{(i)}$, and $\mathbf{P}_{\mathbf{V}_M^{(i)}}$ and $\mathbf{P}_{\mathbf{W}_M^{(i)}}$ are, respectively, two projectors onto these subspaces. Using the transmitted parameters the receiver can reconstruct

$$\tilde{f}^{(i)} = \mathbf{P}_{\mathbf{V}_M^{(i)}} f_s^{(i)} \quad ; \quad i = 1, \ldots, K. \tag{2.10}$$

We define our second set as the set of real signals of length L whose frames have the same projections onto $\mathbf{V}_M^{(i)}$ with the frames of the original speech signal, i.e.,

$$C_2 = \{ f : \mathbf{P}_{\mathbf{V}_M^{(i)}} f^{(i)} = \tilde{f}^{(i)} \text{ for } i = 1, \ldots, K \text{ and } f[n] = 0 \text{ for } n < 0 \text{ and } n \ge L \}. \tag{2.11}$$

To show that C_2 is convex consider any two signals $f_1, f_2 \in C_2$. For all $i = 1, \ldots, K$ we have

$$\mathbf{P}_{\mathbf{V}_M^{(i)}} f_1^{(i)} = \mathbf{P}_{\mathbf{V}_M^{(i)}} f_2^{(i)} = \tilde{f}^{(i)}. \tag{2.12}$$

Let

$$f_\alpha = \alpha f_1 + (1 - \alpha) f_2 \tag{2.13}$$

for any $\alpha \in (0,1)$. Then, for all $i = 1, \ldots, K$

$$\mathbf{P}_{\mathbf{V}_M^{(i)}} f_\alpha^{(i)} = \alpha \mathbf{P}_{\mathbf{V}_M^{(i)}} f_1^{(i)} + (1-\alpha) \mathbf{P}_{\mathbf{V}_M^{(i)}} f_2^{(i)} = \tilde{f}^{(i)}. \tag{2.14}$$

Therefore, $f_\alpha \in C_2$ which shows that C_2 is convex. Next, we find a projection on C_2. For any $f \in \mathbf{l}^2(\mathbf{R})$ we want to find $f_p \in C_2$ such that $|| \; f - f_p \; ||^2$ is minimized. We can write this as

$$\begin{aligned}
|| \; f - f_p \; ||^2 &= \sum_{n=-\infty}^{\infty} | \; f[n] - f_p[n] \; |^2 \qquad (2.15) \\
&= \sum_{n=-\infty}^{-1} | \; f[n] \; |^2 + \sum_{n=L}^{\infty} | \; f[n] \; |^2 + \sum_{i=1}^{K} || \; f^{(i)} - f_p^{(i)} \; ||^2 \qquad (2.16) \\
&= || \; f \; ||^2 + \sum_{i=1}^{K} || \; f_p^{(i)} \; ||^2 - 2 \sum_{i=1}^{K} < f^{(i)}, f_p^{(i)} > \qquad (2.17) \\
&= || \; f \; ||^2 + \sum_{i=1}^{K} || \; \mathbf{P}_{\mathbf{V}_M^{(i)}} f_p^{(i)} \; ||^2 + \sum_{i=1}^{K} || \; \mathbf{P}_{\mathbf{W}_M^{(i)}} f_p^{(i)} \; ||^2 \\
&\quad - 2 \sum_{i=1}^{K} < f^{(i)}, \mathbf{P}_{\mathbf{V}_M^{(i)}} f_p^{(i)} > \\
&\quad - 2 \sum_{i=1}^{K} < f^{(i)}, \mathbf{P}_{\mathbf{W}_M^{(i)}} f_p^{(i)} > . \qquad (2.18)
\end{aligned}$$

Since $\mathbf{P}_{\mathbf{V}_M^{(i)}} f_p^{(i)} = \tilde{f}^{(i)}$ for all i, the above expression is minimized by minimizing

$$\sum_{i=1}^{K} || \; \mathbf{P}_{\mathbf{W}_M^{(i)}} f_p^{(i)} \; ||^2 - 2 \sum_{i=1}^{K} < f^{(i)}, \mathbf{P}_{\mathbf{W}_M^{(i)}} f_p^{(i)} > . \tag{2.19}$$

Therefore, we must choose

$$\mathbf{P}_{\mathbf{W}_M^{(i)}} f_p^{(i)} = \mathbf{P}_{\mathbf{W}_M^{(i)}} f^{(i)} \quad ; \quad i = 1, \ldots, K. \tag{2.20}$$

Then, the projection of f onto C_2 is equal to f_p where

$$f_p^{(i)} = \tilde{f}^{(i)} + \mathbf{P}_{\mathbf{W}_M^{(i)}} f^{(i)} \quad ; \quad i = 1, \ldots, K. \tag{2.21}$$

We initiate our algorithm with the reconstructed signal. Then, we iteratively make projections onto C_1 and C_2. The projection onto C_1 is just filtering the signal by an ideal low-pass filter with a cutoff frequency W. Since the two sets are convex, the convergence of this algorithm is guaranteed by the theorem of POCS. In practice, we use a non-ideal low-pass filter. In this case filtering is not a projection onto C_1 and the convergence is not guaranteed. However, the algorithm can still remove the discontinuities after a few iterations.

2.2 Simulation Examples

We implement the matching pursuit algorithm with Gabor time-frequency atoms using the C programming language. The dictionary used in our simulation studies consists of 256 Gabor atoms. To select these atoms we first ran the algorithm with a much larger dictionary using a long speech signal and we selected the most frequently used 256 atoms from this dictionary. We use speech frames of length 200. We obtain the fourth order approximation of each frame using the matching pursuit algorithm. For each selected atom $g_{(\gamma_n, \phi_n)}$, the inner product $< R^n f, g_{(\gamma_n, \phi_n)} >$ and the angle ϕ_n are quantized. The angles are uniformly quantized using 4 bits and the inner products are μ-law quantized using 6 bits. The uniform quantizer and the μ-law quantizer are shown in Figure 2.1 and Table 2.1, respectively. After the matching pursuit algorithm is terminated the inner products are modified using the backprojection method. The resultant coefficients are further quantized using the μ-law quantizer. These quantized coefficients and angles are sent to the receiver together with the index of the selected atom. Since there are 256 atoms, the index is sent using 8 bits. For each atom a total number of 18 bits are sent to the receiver. The speech signals we used in our simulations were sampled at 8000 Hz, so the method described above results in a bit rate of 2880 bits/second.

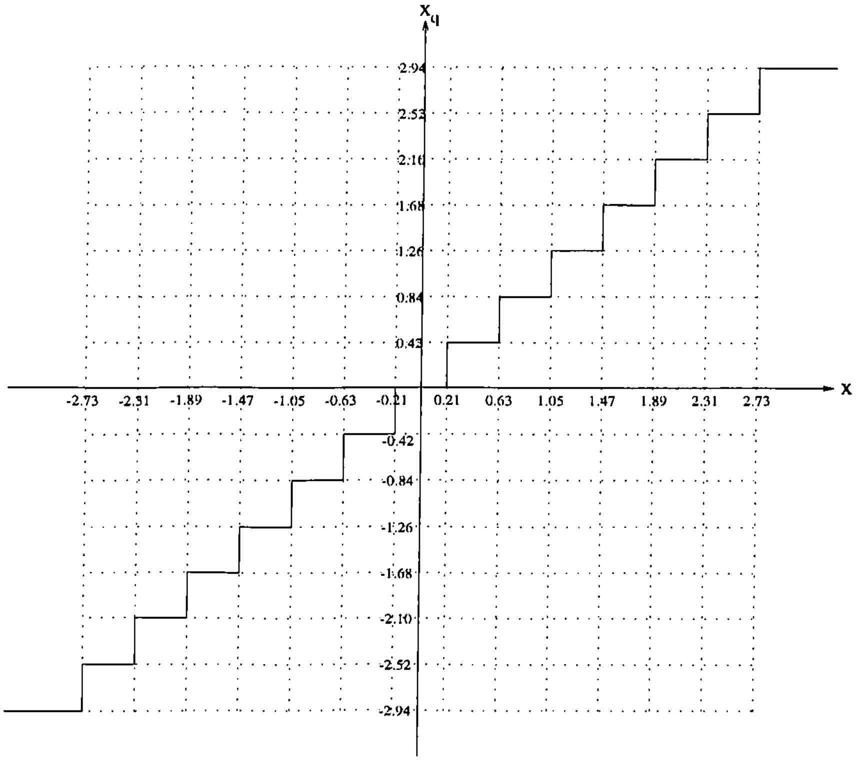


Figure 2.1: *The 4 bit uniform quantizer used to quantize the angles.*

Input Magnitude	Step Size	Segment Code	Level Code	Decoded Magnitude
0 - 10	20	000	000	0
10 - 30			001	20
⋮			⋮	⋮
130 - 150			111	140
150 - 190	40	001	000	170
⋮			⋮	⋮
430 - 470			111	450
470 - 550	80	010	000	510
⋮			⋮	⋮
1030 - 1110			111	1070
1110 - 1270	160	011	000	1190
⋮			⋮	⋮
2230 - 2390			111	2310
2390 - 2710	320	100	000	2550
⋮			⋮	⋮
4630 - 4950			111	4790
4950 - 5590	640	101	000	5270
⋮			⋮	⋮
9430 - 10070			111	9750
10070 - 11350	1280	110	000	10710
⋮			⋮	⋮
19030 - 20310			111	19670
20310 - 22870	2560	111	000	21590
⋮			⋮	⋮
38230 - ∞			111	39510

Table 2.1: *The 6 bit μ-law quantizer used to quantize the inner products.*

At the receiver each frame is reconstructed and the POCS based algorithm described in the previous section is used to remove the discontinuities between successive frames. A Hamming filter of length 7 is used in the algorithm. The projections are carried out only for one or two iterations. An original speech signal is shown in Figure 2.2 (a). The coded/decoded signals using our method and an LPC-10 vocoder are shown in Figure 2.2 (b) and Figure 2.2 (c), respectively. Although the bit rate is slightly higher than the bit rate of a typical LPC-10 vocoder (2400 bits/second), our method produces a better quality speech than an LPC-10 vocoder. However, a mean opinion score (MOS) study has not been done due to practical difficulties.

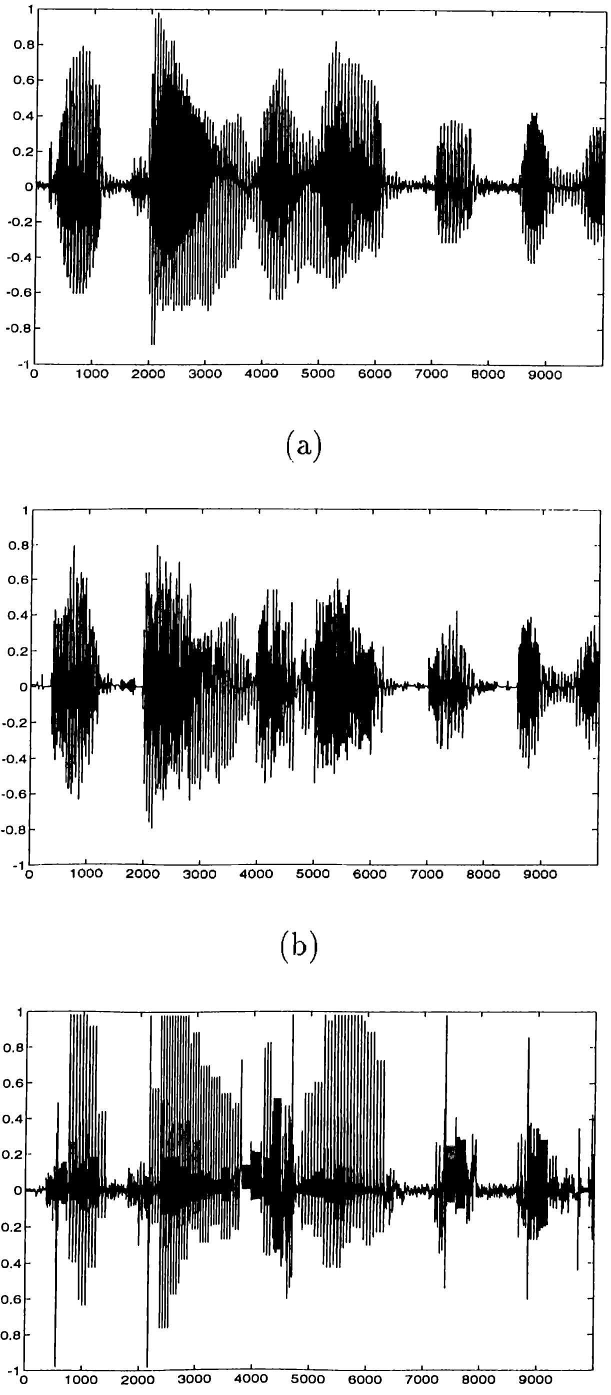
It can also be observed that the POCS based algorithm removes the discontinuities. A voiced speech segment and an unvoiced speech segment are shown in Figure 2.3 (a) and Figure 2.4 (a), respectively. There are two discontinuities between frames in the coded/decoded segments shown in Figure 2.3 (b) and Figure 2.4 (b). The result of applying our algorithm for 2 iterations is shown in Figure 2.3 (c) and Figure 2.4 (c). It is seen that the discontinuities have been removed completely.

2.3 Computational Complexity of the New Speech Coding Method

The computational complexity of our speech coding method depends on the efficient computation of the inner products. The inner product of an atom with all other atoms in the dictionary can be computed by $O(M)$ multiplications [1] using the algorithm described in Chapter 2, where M is the number of atoms in the dictionary. For each frame, this algorithm is initiated by computing the inner products of the frame with the atoms in the dictionary. This operation requires $O(MN)$ multiplications, where N is the frame length.

The computational complexity of a typical LPC-10 vocoder depends on the pitch prediction method. For each frame the number of multiplication operations can be $O(N)$ or $O(N \log N)$. However the computational complexity of an LPC-10 vocoder is smaller than the computational complexity of our method. We compared our method with an implementation of the LPC-10 algorithm using C programming language. Our method and the LPC-10 algorithm process each frame in 0.22 seconds and 0.02 seconds, respectively, in a SUN SPARC-10 workstation.

The methods we used in this chapter can also be extended to two-dimensions (2-D). In the next chapter we define 2-D time-frequency atoms and use them with the matching pursuit algorithm in image coding.



(a)

(b)

(c)

Figure 2.2: *(a) An original speech signal, (b) the signal after coding/decoding using the Gabor time-frequency decomposition, and (c) the signal after coding/decoding using an LPC-10 vocoder.*

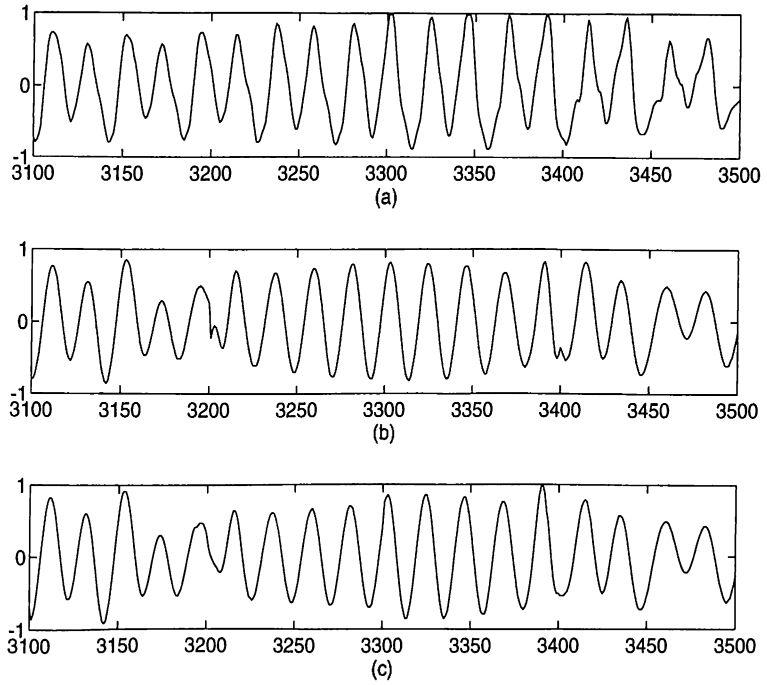


Figure 2.3: *(a) A voiced speech segment, (b) the coded/decoded segment having two discontinuities, and (c) the same segment after applying the POCS based algorithm twice.*

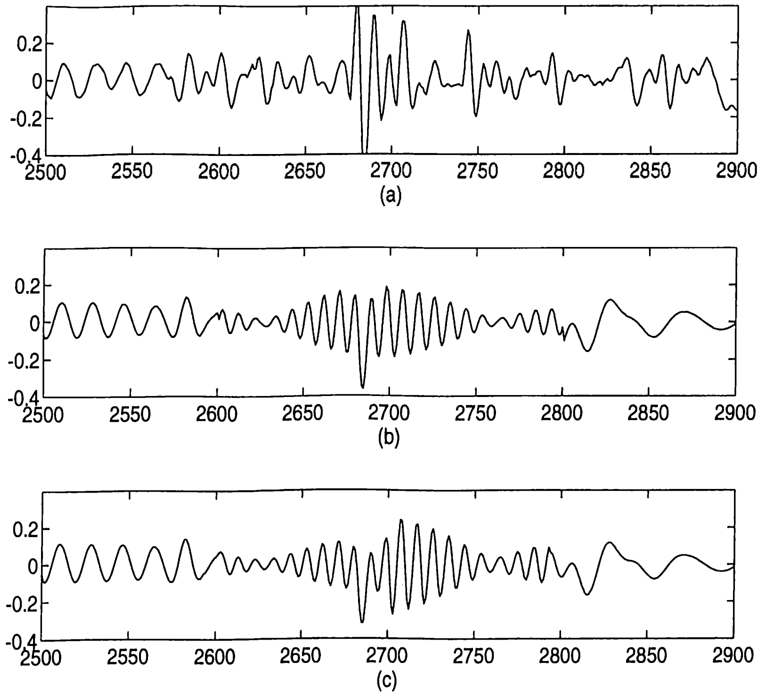


Figure 2.4: *(a) An unvoiced speech segment, (b) the coded/decoded segment having two discontinuities, and (c) the same segment after applying the POCS based algorithm twice.*

Chapter 3

Image Coding

The matching pursuit algorithm and Gabor time-frequency atoms described in the previous chapters can also be applied to image coding. For this purpose we must first define a new vector space and two-dimensional (2-D) Gabor atoms.

Our vector space will be the set of 2-D sequences periodic with N in both vertical and horizontal coordinates, i.e., for any f in this vector space,

$$f[m+kN, n+lN] = f[m,n] \quad \text{for all} \quad k,l,m,n \in \mathbf{Z}. \tag{3.1}$$

We define the inner product of any two sequences f and g in this space by

$$< f, g >_2 = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f[m,n] g^*[m,n] \tag{3.2}$$

and the norm of a sequence f by

$$\| f \|_2 = \sqrt{< f, f >_2}. \tag{3.3}$$

We will use a subscript 2 to distinguish this inner product and norm from 1-D inner product and norm. We define 2-D Gabor atoms as follows. Let $g_s[n]$ again be the scaled, sampled and shifted Gaussian periodized to N points, i.e.,

$$g_s[n] = \frac{K_s}{\sqrt{s}} \sum_{l=-\infty}^{\infty} g(\frac{n-lN}{s}) \tag{3.4}$$

where $g(t)$ is the normalized Gaussian defined in (1.25) and K_s normalizes the 1-D norm of g_s. For any $\gamma_1 = (s_1, p_1, \frac{2\pi k_1}{N})$ and $\gamma_2 = (s_2, p_2, \frac{2\pi k_2}{N})$, we define the complex 2-D Gabor atom by

$$g_{\gamma_1,\gamma_2}[m,n] = g_{s_1}[m-p_1] g_{s_2}[n-p_2] e^{j\frac{2\pi k_1}{N}m + j\frac{2\pi k_2}{N}n}. \tag{3.5}$$

Similarly, the real 2-D atoms are defined by

$$g_{(\gamma_1,\gamma_2,\phi)}[m,n] = K_{(\gamma_1,\gamma_2,\phi)} g_{s_1}[m-p_1] g_{s_2}[n-p_2] \cos(\frac{2\pi k_1}{N}m + \frac{2\pi k_2}{N}n + \phi) \tag{3.6}$$

where ϕ is the phase and $K_{(\gamma_1,\gamma_2,\phi)}$ is chosen such that

$$\| g_{(\gamma_1,\gamma_2,\phi)} \|_2 = 1. \tag{3.7}$$

The matching pursuit algorithm is the same for a 2-D decomposition. The inner products in this algorithm are now 2-D inner products, but they can easily be computed by successive 1-D inner products. Let f be a sequence in our vector space. Then,

$$\begin{aligned} < f, g_{\gamma_1,\gamma_2} >_2 &= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f[m,n] g^*_{\gamma_1,\gamma_2}[m,n] \\ &= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f[m,n] g_{s_1}[m-p_1] g_{s_2}[n-p_2] \mathrm{e}^{-j\frac{2\pi k_1}{N}m - j\frac{2\pi k_2}{N}n} \\ &= \sum_{m=0}^{N-1} (\sum_{n=0}^{N-1} f[m,n] g_{s_2}[n-p_2] \mathrm{e}^{-j\frac{2\pi k_2}{N}n}) g_{s_1}[m-p_1] \mathrm{e}^{-j\frac{2\pi k_1}{N}m} \\ &= << f[m,n], g_{\gamma_2}[n] >, g_{\gamma_1}[m] > . \end{aligned} \tag{3.8}$$

Hence, we can compute the 2-D inner product of f with a 2-D Gabor atom g_{γ_1,γ_2} by first computing the inner products of the columns of f with g_{γ_2} and then calculating the inner product of the result with g_{γ_1}. Although this method requires more multiplication operations than taking a single 2-D inner product, we have to store only 1-D atoms. This reduces the required memory considerably, because storing 2-D atoms instead of 1-D atoms squares the required memory. To compute the inner product of two 2-D Gabor atoms, the above equation is further simplified to

$$< g_{\gamma_1,\gamma_2}, g_{\gamma_1',\gamma_2'} >_2 = < g_{\gamma_1}, g_{\gamma_1'} >< g_{\gamma_2}, g_{\gamma_2'} > . \tag{3.9}$$

In our image coding method we process the images block by block. The input image is first divided into $N \times N$ blocks. Then, each block is made zero-mean by calculating the mean of the block and subtracting this from each pixel in the block. These means are quantized and sent to the receiver. Next, the matching pursuit algorithm is applied to each block. The inner products and the angles are quantized and sent to the receiver together with the index of the atoms selected. The receiver reconstructs an approximation of each block using these coefficients. We can apply backprojection at the coder to improve the approximation and a POCS based algorithm at the receiver to remove discontinuities between blocks.

3.1 Simulation Examples

We implement the image coding algorithm in C programming language. We used 8×8 blocks in our simulation studies. The dictionary we used is formed by all possible pairwise products of 32 1-D atoms, therefore it contains 1024 2-D atoms. The 1-D atoms were determined by running the algorithm on several images and selecting the most frequently used 32 atoms. The approximation order is adaptive in our simulations, i.e., the matching pursuit algorithm runs until the maximum inner product is below a predetermined threshold value. In this way, the blocks with a low local variance is approximated with a lower order than the blocks with a high local variance. It is possible to compress an image at different compression ratios by changing this threshold. The 1024 atoms are indexed using 10 bits and the inner products and the angles are quantized and transmitted using 6 bits and 4 bits, respectively.

We used the 512×512 Lena image and the 672×560 Barbara image in our simulation studies. The simulation results using the Barbara image and the Lena image are shown in Table 3.1 and Table 3.2, respectively. The original Barbara image and the compressed/decompressed Barbara image with a compression ratio of 6.0 are shown in Figure 3.1. We also coded these images using the Joint Photographic Experts Group (JPEG) compression standard [13]. The comparison of our method with JPEG for the Barbara image and the Lena image are shown in Figure 3.2 and Figure 3.3, respectively. It is seen that JPEG has a better performance than our method at all compression ratios.

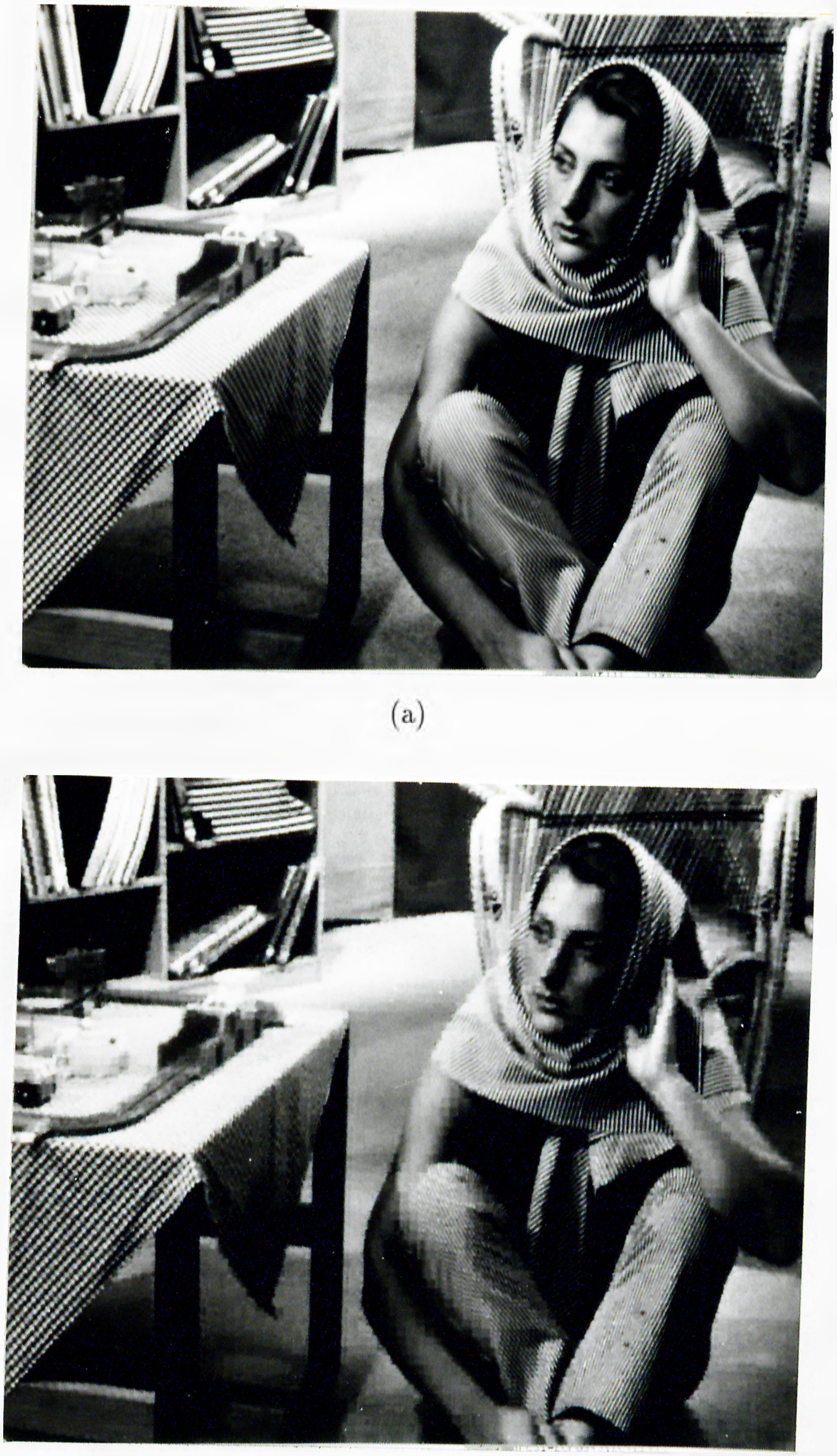
Gabor time-frequency atomic decomposition does not have a good performance in image coding, although it has a good performance in speech coding. The main reason for this fact is the different characteristics of speech and image signals. A speech signal is an "oscillatory" signal which looks like a typical Gabor atom, hence it can be represented efficiently by Gabor atoms. An image signal is non-oscillatory and has different characteristics than a Gabor atom. Therefore, the Gabor decomposition does not give good results for image signals.

	Compression Ratio	MSDE	SNR
0.62 bpp	12.99	13.32	39.48
0.66 bpp	12.20	12.91	40.02
0.71 bpp	11.23	12.40	40.72
0.78 bpp	10.20	11.85	41.51
0.88 bpp	9.05	11.18	42.52
1.03 bpp	7.74	10.36	43.84
1.08 bpp	7.38	10.12	44.25
1.20 bpp	6.67	9.63	45.11
1.33 bpp	6.00	9.14	46.02
1.41 bpp	5.66	8.88	46.52
1.64 bpp	4.88	8.25	47.80
1.78 bpp	4.49	7.90	48.55
2.17 bpp	3.68	7.10	50.42

Table 3.1: *The compression results for the Barbara image.*

	Compression Ratio	MSDE	SNR
0.59 bpp	13.59	8.96	46.70
0.62 bpp	12.81	8.58	47.45
0.68 bpp	11.79	8.09	48.47
0.76 bpp	10.57	7.56	49.65
0.78 bpp	10.21	7.40	50.02
0.84 bpp	9.51	7.10	50.73
0.92 bpp	8.71	6.75	51.61
0.96 bpp	8.29	6.56	52.11
1.09 bpp	7.34	6.12	53.31
1.17 bpp	6.82	5.88	54.00
1.39 bpp	5.77	5.36	55.61
1.61 bpp	4.96	4.95	57.01
1.97 bpp	4.06	4.45	58.86

Table 3.2: *The compression results for the Lena image.*



(a)

(b)

Figure 3.1: *The original (a) and the reconstructed (b) Barbara images.*

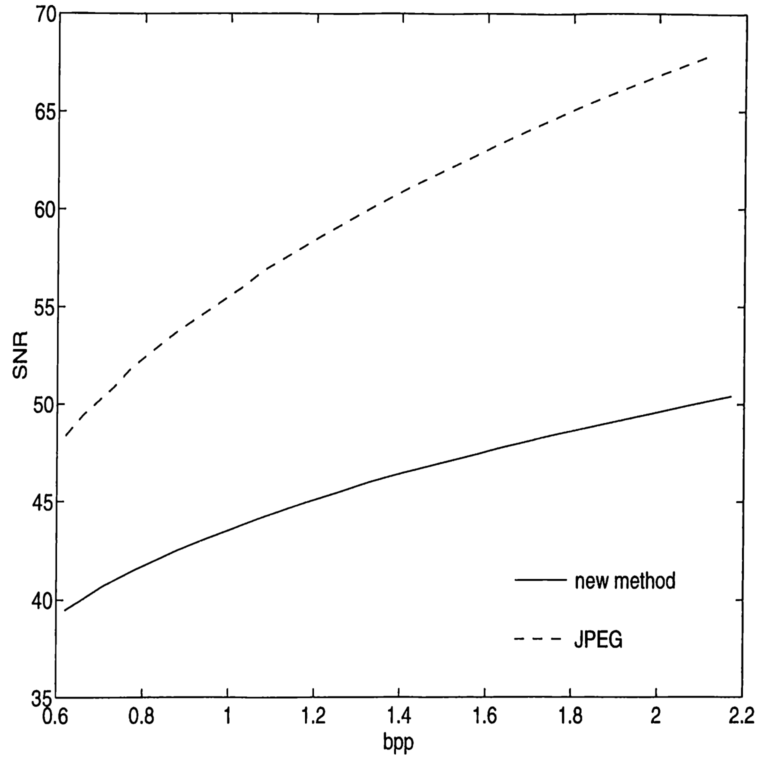


Figure 3.2: *The comparison of the matching pursuit and JPEG algorithms using the Barbara image.*

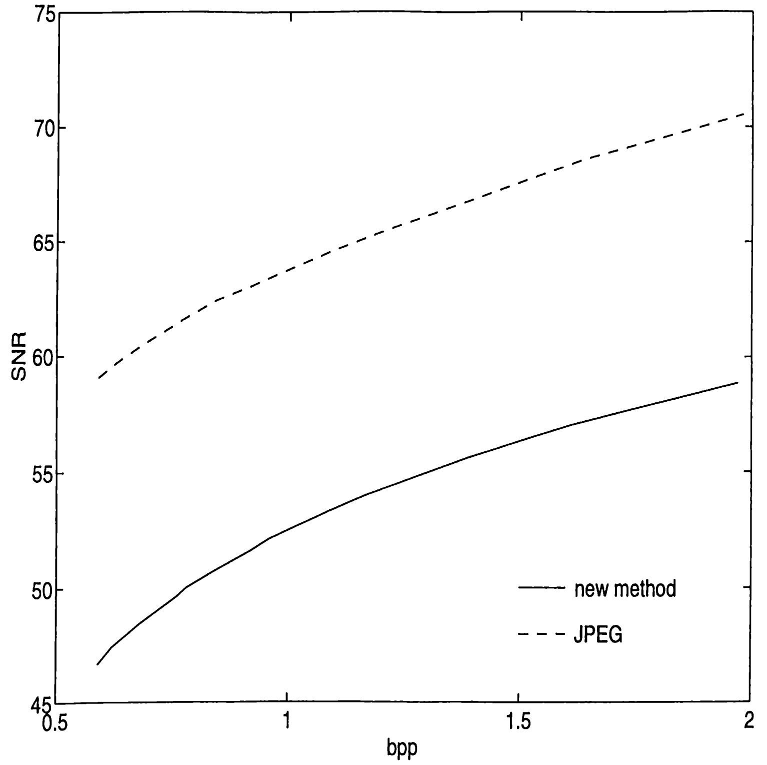


Figure 3.3: *The comparison of the matching pursuit and JPEG algorithms using the Lena image.*

Chapter 4

Conclusion

In this thesis a new low bit rate speech coding method was developed. The method is based on an adaptive decomposition of speech signals over Gabor time-frequency atoms. This decomposition is efficiently implemented using the matching pursuit algorithm. The new method can code speech signals at a slightly greater bit rate than a standard LPC-10 vocoder, but it produces better quality speech. However, the computational complexity of the new method is large compared to an LPC-10 vocoder. This is mainly due to the calculation of inner products with all atoms in the dictionary.

The new vocoder is a fixed bit rate vocoder. It can also be made a variable bit rate vocoder by making the approximation order of each frame adaptive or using a variable length source coding method such as Huffman coding. Some of the atoms in the dictionary are selected by the algorithm more frequently than others. The frequency of occurrence of each atom can be found by running the algorithm with long test speeches and then the method can be made variable bit rate using Huffman coding. This can reduce the average bit rate without affecting the speech quality.

Finally, a projection onto convex sets based method was developed to remove discontinuities between successive speech frames. The method can remove the discontinuities after only a few iterations.

The method for speech coding was also extended to 2-D. For this purpose, 2-D time-frequency atoms were developed. These atoms are used in the matching pursuit algorithm for image coding. Although this method works, its performance is far below a standard JPEG coder. This is because an image signal has different characteristics than a Gabor atom.

References

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

[2] J. B. Allen and L. R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis", *Proceedings of the IEEE*, vol. 65, pp. 1558-1564, 1977.

[3] L. Cohen, "Time-frequency distributions: a review", *Proceedings of the IEEE*, vol. 77, No. 7, pp. 941-981, July 1989.

[4] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Series in Applied Mathematics, SIAM, 1992.

[5] R. Coifman, Y. Meyer and V. Wickerhauser, "Size properties of wavelet packets", *Wavelets and Their Applications*, Mary Beth Ruskai, Ed., Jones and Bartlett Publishers, Boston, pp. 453-470, 1992.

[6] S. Qian and D. Chen, "Signal representation in adaptive normalized Gaussian functions", submitted to *IEEE Transactions on Signal Processing.*

[7] B. Torresani, "Wavelets associated with representations of the affine Weyl-Heisenberg group", *Journal of Mathematical Physics*, vol. 32, No. 5, pp. 1273-1279, May 1991.

[8] N. S. Jayant, *Digital Coding of Waveforms*, Prentice-Hall, 1984.

[9] S. Singhal and B. S. Atal, "Amplitude optimization and pitch prediction in multipulse coders", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, No. 3, pp. 317-327, March 1989.

[10] V. Cuperman, B. S. Atal and A. Gersho, *Advances in Speech Coding*, Kluwer Academic Publishers, 1991.

[11] Military Agency for Standardization, "NATO standardization agreement, Stanag 4196, parameters and coding characteristics that must be common to assure interoperability of 2400 bps linear predictive encoded digital speech".

[12] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 2, No. 1, pp. 91-95, March 1992.

[13] G. K. Wallace, "The JPEG still picture compression standard", *Communications of the ACM*, vol. 34, No. 4, pp. 30-44, April 1991.

[14] N. S. Jayant, "Coding speech at low bit rates", *IEEE Spectrum*, pp. 58-63, August 1986.

[15] L. M. Bregman, "The method of successive projection for finding a common point of convex sets", *Dokl. Akad. Nauk SSSR*, vol. 162, No. 3, 1965.

[16] D. C. Youla, "Generalized image restoration by the method of alternating orthogonal projections", *IEEE Transactions on Circuits and Systems*, vol. CAS-25, No. 9, pp. 694-702, September 1978.

[17] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation", *IEEE Transactions on Circuits and Systems*, vol. CAS-22, No. 9, September 1975.