Models of Visual Feature Detection and Spike Coding in the Nervous System

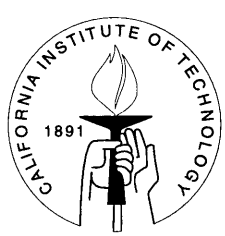
Thesis by

Thomas M. Annau

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

1996

(Submitted May 20, 1996)

Acknowledgements

A very significant part of my scientific education at Caltech was the result of discussions with graduate students and postdocs from various research groups, including the Hopfield group, the Allman lab, the Konishi lab and the Bower lab. While by no means an exhaustive list, I would especially like to thank the following people (in alphabetical order): Ron Benson, Buster Boahen, Carlos Brody, Allan Dobbins, Andreas Herz, Rich Jeo, Dave Kewley, Mike Lewicki, Sanjoy Mahajan, Jamie Mazer, Marcus Mitchell, Dave Perkel, Bill Press, Alex Protopapas, Sam Roweis, and Erik Winfree.

I would like to express my appreciation to the Computation and Neural Systems program for providing an interdisciplinary environment within which to conduct research on topics which require the synthesis of a broad range of ideas. I would also like to thank my advisor John Hopfield for his support and comments, and I never would have finished my thesis without the logistical and moral support of Laura Rodriguez and Janette Aguado.

Abstract

We propose mathematical models to analyze two nervous system phenomena. The first is a model of the development and function of simple cell receptive fields in mammalian primary visual cortex. The model assumes that images are composed of combinations of a limited set of specific visual features and that the goal of simple cells is to detect the presence or absence of these features. Based on a presumed statistical character of images and their visual features, the model uses a constrained Hebbian learning rule to discover the structure of the features, and thus the appropriate response properties of simple cells, by training on a database of photographs. The response properties of the model simple cells agree qualitatively with neurophysiological observation.

The second is a model of the coding of information in the nervous system by the rate of axonal voltage spikes. Assuming an integrate-and-fire mechanism for spike generation, we develop a quantization-based model of rate coding and use it to derive the mathematical relationship between the amplitude and temporal resolution of a rate encoded signal. We elaborate the model to include integrator leak in the spike generation mechanism and show that it compactly combines coding and the computation of a threshold function.

Contents

List of Figures

Chapter 1 Introduction

This introductory chapter begins with definitions of the mathematical notation to be used in the course of the thesis. We then present the threshold function, a unifying theme of this thesis, and an example of its use in the detection of digital signals in the presence of noise. This example introduces some of the issues and ideas used in the remaining chapters, whose contents are briefly summarized below:

- Chapter 2 describes a model of the development and function of simple cells receptive fields in mammalian primary visual cortex in which a threshold function is used to detect the presence of specific visual features.
- Chapter 3 describes a model of neuronal spike rate coding in which a threshold nonlinearity is used to encode somatic current into a series of axonally transmitted voltage spikes.
- Chapter 4 concludes the thesis with a discussion of the common themes of the two models.

1.1 Notation

This section introduces the notational conventions used throughout the remainder of the thesis. Since the primary mathematical concern is the characterization and manipulation of *signals*, we first define a signal, then present basic analyses to be performed on signals, including Fourier and statistical techniques, and finally introduce linear filtering and vector representations of signals. Futher development of these topics can be found in [Oppenheim et al., 1983], [Papoulis, 1991], [Oppenheim and Schafer, 1989].

1.1.1 Signals defined

A *signal* is a function of time and/or N-dimensional space. The value of the function is called the *amplitude* of the signal. If s is a strictly temporal signal, then $s(t)$ is a function representing the signal's amplitude at time t. Similarly, if s is a two-dimensional spatial

signal, then $s(x, y)$ is the signal's amplitude at spatial coordinates (x, y). We use one spatial dimensional signals in derivations for the sake of clarity, but all results can be easily extended to the two-dimensional case.

We will analyze two types of signals, *analog* and *digital*, which are classified by their characteristic amplitude values. The amplitude of an *analog* signal can assume a continuum of values, whereas that of a *digital* signal can only assume one of a discrete set of values. The most common digital signal is the *binary* signal, which can only assume one of two amplitude values; unless otherwise specified, all digital signals discussed are assumed to be binary. One of the themes of this thesis is the interconversion between digital and analog signals.

A type of analog signal which arise frequently is the *Gaussian process*, a signal whose amplitudes are distributed as a Gaussian probability distribution. Another signal which arises in different contexts in the next two chapters is the *impulse process*, a sum of shifted, non-overlapping Dirac delta functions:

$$\sum_i \delta(t - t_i), \quad \text{where } t_i \neq t_j \text{ if } i \neq j \tag{1.1}$$

1.1.2 Fourier analysis

A temporal signal x represented as a function of time t can also be represented as a function of frequency in radians per unit of time ω by use of the Fourier transform:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \tag{1.2}$$

where $j = \sqrt{-1}$. We follow the convention of using capital letters to designate the Fourier transform of a signal. The frequency representation of a signal can be converted back to the time representation with the inverse Fourier transform:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{-j\omega t} d\omega \tag{1.3}$$

Similarly, a one-dimensional spatial signal s represented as a function of spatial position x can also be represented as a function of spatial frequency ω_x in radians per unit of space.

A signal s is *bandlimited* if its Fourier transform is zero for all frequencies ω of magnitude greater than the *bandwidth* frequency, Ω_s. By the Nyquist sampling theorem, a bandlimited

signal can be completely represented by amplitude samples spaced T units apart, where $T \leq \frac{\pi}{\Omega}$. We use the notational convention $x[k]$ to denote these samples, where sample index k is an integer:

$$x[k] = x(kT) \tag{1.4}$$

A signal of particular interest is the *Kronecker delta function*, $\delta[k]$, which is the bandlimited analog of the Dirac delta function:

$$\delta[k] = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases} \tag{1.5}$$

Note that by the definition above, the Kronecker delta function is a binary signal.

1.1.3 Statistical analysis

We will use statistics as a powerful method for characterizing signals and their interrelationships. An important statistic is the *cross correlation* between two signals x and y, denoted $R_{xy}(\tau)$:

$$R_{xy}(\tau) = \Big\langle x(t+\tau)y(t) \Big\rangle \tag{1.6}$$

where $\langle \cdot \rangle$ denotes statistical expectation. The function $R_{xx}(t)$ is known as the *autocorrelation* of signal x.

If the two signals are bandlimited such that $\Omega_x \geq \Omega_y$, the cross correlation can be represented by samples at intervals of $T \leq \frac{\pi}{\Omega_x}$:

$$R_{xy}[n] = \Big\langle x[k+n]y[k] \Big\rangle \tag{1.7}$$

The Fourier transform of cross correlation is called the *cross-power spectrum* $S_{xy}(\omega)$.

$$S_{xy}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} R_{xy}(\tau) e^{j\omega\tau} d\tau \tag{1.8}$$

The Fourier transform of the autocorrelation is the signal's *power spectrum*. Since the autocorrelation is an even, real function, the power spectrum can be computed as follows:

$$S_{xx}(\omega) = \frac{1}{\pi}\int_0^{\infty} R_{xx}(\tau)\cos\omega\tau d\tau \tag{1.9}$$

A *white* signal has autocorrelation $R_{xx}(t) = \delta(t)$ and power spectrum $S_{xx}(\omega) = 1$. A *white bandlimited* signal x has an autocorrelation $R_{xx}[k] = \delta[k]$ and a power spectrum which is constant for frequencies under its bandlimit and zero elsewhere:

$$S_{xx}(\omega) = \begin{cases} \frac{\pi}{\Omega} & \text{if } -\Omega_x \leq \omega \leq +\Omega_x \\ 0 & \text{otherwise} \end{cases} \tag{1.10}$$

1.1.4 Linear filtering

A *linear filter* performs a particular linear operation which maps signal x to signal y. A linear filter is completely represented by its *impulse response*, which is itself a signal. If h is the impulse response of a given filter, the result of applying the filter to the signal x to produce the signal y is computed by an operation known as *convolution*:

$$y(t) = h(t) * x(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau \tag{1.11}$$

where "$*$" denotes the convolution operator. Convolution of two signals is equivalent to the multiplication of their Fourier transforms:

$$Y(\omega) = H(\omega)X(\omega) \tag{1.12}$$

For sampled signals, convolution is computed by a discrete sum:

$$y[k] = h[k] * x[k] = \sum_{i=-\infty}^{+\infty} h[i]x[k-i] \tag{1.13}$$

A filter of particular interest is the *finite impulse response* or *FIR* filter. The impulse response of an FIR filter is nonzero only in a specific time interval.

1.1.5 Vector representation

A bandlimited signal x can be usefully represented as the set of N-dimensional vectors $\mathbf{x}[k]$, where the individual elements of each vector $\mathbf{x}[k]$ are composed of $x[k]$ and the previous $N-1$ amplitude samples:

$$\mathbf{x}[k] = \begin{bmatrix} x[k] \\ x[k-1] \\ \vdots \\ x[k-N+1] \end{bmatrix} \tag{1.14}$$

This technique allows the application the concepts and methods of linear algebra to signal analysis. We will use boldface type to designate the vector representation of signals.

A bandlimited FIR filter whose impulse response h is nonzero in the range $a \leq k \leq b$ may be entirely represented by the N-dimensional vector $\mathbf{h}$, where $N = b - a + 1$. The elements of the vector $\mathbf{h}$ are composed of the amplitude samples of the impulse response:

$$\mathbf{h} = \begin{bmatrix} h[a] \\ h[a+1] \\ \vdots \\ h[b] \end{bmatrix} \tag{1.15}$$

These definitions allow convolution with a bandlimited FIR filter to be expressed as a dot product:

$$y[k] = h[k] * x[k] = \mathbf{h} \cdot \mathbf{x}[k+a] \tag{1.16}$$

1.2 The threshold function

As mentioned in the beginning of this chapter, the threshold function is a common theme in the models of the next two chapters. In this section, we begin with its definition and a discussion of two of its most important properties, then illustrate its use in the detection of digital signals in noise.

The threshold function $\Theta(x;\theta)$ outputs a one if its real valued input x is at least as large as the threshold value θ, zero otherwise.

$$\Theta(x;\theta) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases} \tag{1.17}$$

The graph of this function is shown in Figure 1.1a; it is obviously nonlinear. From the

definitions of the previous section, this function takes an analog amplitude as input and returns a binary amplitude.

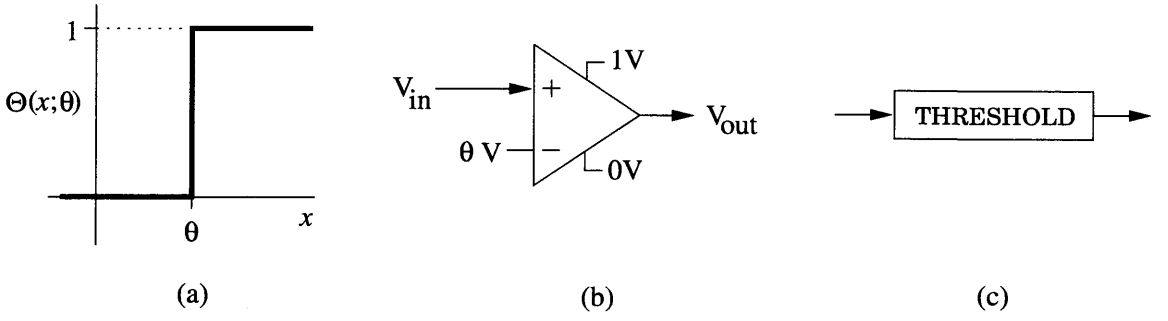


Figure 1.1: The threshold function. (a) A graph of the threshold function. (b) The threshold function implemented with a high gain amplifier configured as a comparator. (c) The function block with which the threshold function will be represented in block diagrams.

Before discussing its application, we highlight its two most important properties:

- **The threshold function has only two possible values.** The threshold function is non-invertible transformation; in information theoretic terms, $\Theta(x;\theta)$ conveys at most one bit, regardless of how much information was conveyed by x. The threshold function can be used to eliminate information which is irrelevant to a particular task, and for this reason, it is a computationally powerful tool.

- **The threshold function has a simple electronic implementation.** The threshold function can be implemented with a high gain amplifier configured as a *comparator*, as illustrated in Figure 1.1b. The high gain amplifying the differential input signal drives the amplifier's output to one of its two saturating voltages. Although the amplifier gain would have to be infinite to realize the threshold function exactly, it can be made large enough in practical situations to be considered effectively infinite. In the nervous system, the threshold function can be implemented by a particular neuronal ion channel, the voltage sensitive sodium channel. Either implementation is physically compact, making the threshold function an attractive building block when designing computing hardware, whether silicon- or carbon-based.

The threshold function will be represented in block diagrams with the symbol shown in Figure 1.1c.

1.2.1 The detection of digital signals in the presence of noise

One of the primary uses of the threshold function is for the detection of digital signals in the presence of noise, which has application in insuring the accuracy of digital signal transmission along a noisy, attenuating electrical transmission line. This is not only a standard situation in modern electronic communications, but also in the nervous system, as is discussed in Chapter 3. In Chapter 2, we show that it also has application to the detection of visual features in images.

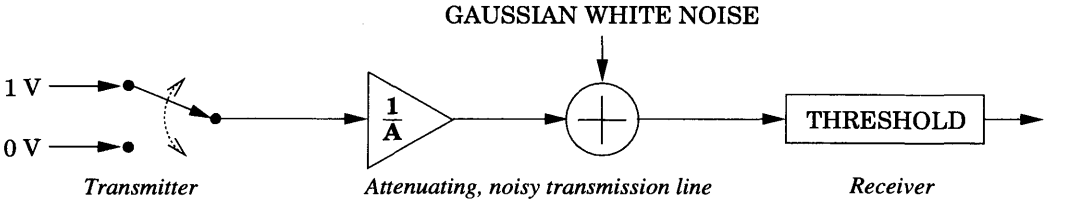


Figure 1.2: A schematic diagram illustrating the use of a threshold function to suppress noise introduced by transmitting a signal down an attenuating, noisy transmission line.

A schematic diagram of the transmission scenario is shown in Figure 1.2. At any particular time, the amplitude of the transmitter's signal is either one volt or zero volts. As the signal is travels along the transmission line, its power is attenuated and it is corrupted by the electrical noise, resulting in uncertainty at the receiver as to which amplitude was actually transmitted. The goal of the receiver is to estimate, with as little error as possible, which voltage was sent by the transmitter. As will be shown, this can be accomplished with the threshold function.

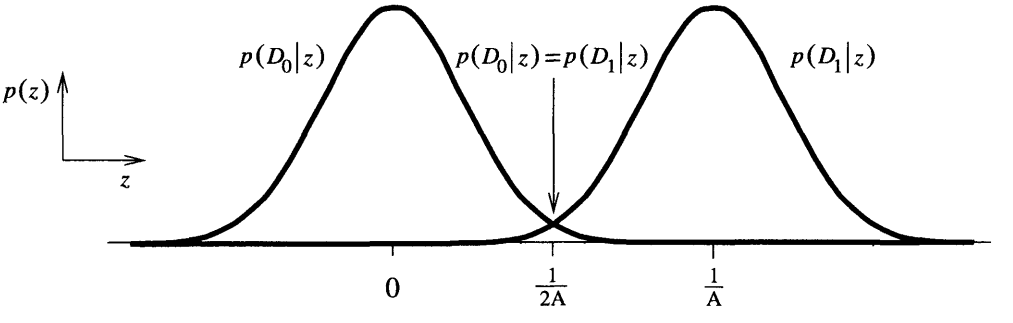


Figure 1.3: A graph of the probability distributions $p(D_0|z)$ and $p(D_1|z)$.

Let s denote the transmitted signal, D_0 denote that a zero was actually transmitted, and

D_1 denote that a one was transmitted:

$$D_1\text{: } s = 1 \qquad D_0\text{: } s = 0 \tag{1.18}$$

We model the attenuation with a gain of $\frac{1}{A}$, where A is the level of attenuation, and the effects of noise are represented by the addition of a zero mean, Gaussian random variable of variance σ^2. The probability distributions of the received voltage amplitude z given the two different possibilities, D_0 and D_1, are shown in Figure 1.3:

$$p(z|D_0) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-z^2}{2\sigma^2} \tag{1.19}$$

$$p(z|D_1) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(z - \frac{1}{A})^2}{2\sigma^2} \tag{1.20}$$

In order to minimize the error of estimation, the receiver must choose the more likely scenario, D_0 or D_1, given z. If D_1 is more likely given z, the receiver should assume that a one was transmitted:

$$\text{Assume } s = 1 \text{ if } p(D_1|z) > p(D_0|z) \tag{1.21}$$

By Bayes' theorem, this is equivalent to

$$p(z|D_1)p(D_1) > p(z|D_0)p(D_0) \tag{1.22}$$

If we assume that one and zero voltage transmissions are equally likely, then $p(D_0) = p(D_1) = \frac{A}{2}$. Since the additive Gaussian noise has the same variance for both distributions $p(z|D_0)$ and $p(z|D_1)$, (1.22) is equivalent to

$$z > \frac{A}{2} \tag{1.23}$$

So when $z > \frac{A}{2}$, the receiver should assume that a one was sent, otherwise, a zero. This comparison is naturally expressed with the threshold function. The receiver may simply pass its received voltage z through the threshold function and use the result as the estimate of the transmitted voltage $\hat{s}$:

$$\hat{s} = \Theta(z; \frac{A}{2}) \tag{1.24}$$

Note that the receiver will inevitably make some errors in its estimation of the transmitted signal. Due to the overlap of the two probability distributions $p(z|D_0)$ and $p(z|D_1)$, there are two types of errors. An error of *type I* occurs when values of z due to D_0 are misclassified as signifying a one was sent (this is also known as a *false alarm*). Similarly, when values of z due to D_1 are misclassified as signifying a zero was sent, this is a *type II* error. The total error rate is their sum, and for our classification scheme this is

$$\int_{-\infty}^{\frac{A}{2}} p(z|D_1)dz + \int_{\frac{A}{2}}^{+\infty} p(z|D_0)dz \tag{1.25}$$

By performing the classification as described above, this probability was minimized. However, type I and type II errors can be weighted asymmetrically if their consequences are not equivalent. In this case, the threshold level may shift from one half to minimize a weighted error probability. This point is discussed in Chapter 2 in the context of feature detection.

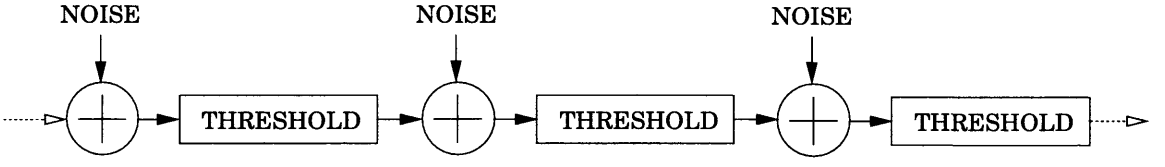


Figure 1.4: The use of threshold functions as repeaters. Attenuation reduces the signal voltage as it travels down the noisy transmission line, but the repeaters periodically reset the voltage to their best estimate of the originally transmitted value. This can be used to limit the error introduced by noise regardless of what distance the signal travels.

The greater the attenuation (*i.e.*, the larger A), the greater the overlap of the two probability distributions $p(z|D_0)$ and $p(z|D_1)$ and thus the higher the error rate. Since attenuation is typically a monotonically increasing function of distance traveled along the conductor, the farther the signal travels, the worse the error rate. If the transmitter and receiver are far apart, the error level may rise to an unacceptable level. To counter this, electronic implementations of threshold functions called *repeaters* can be placed at intervals along the conductor to reset the attenuated voltages to their estimated original values; this strategy is shown schematically in Figure 1.4. This can limit the error rate to an arbitrarily small value

given enough repeaters regardless of how far the signal must travel to reach its destination. Repeaters are commonly used in electrical communication, and as is pointed out in Chapter 3, in the axonal transmission lines of the nervous system.

If the signal to be transmitted down the conductor is not in digital form, it is possible to take advantage of the noise suppression mechanisms of digital signal communication by converting the signal to digital form. The conversion process is equivalent to the discrete approximation of a continuous value. As we show in Chapter 3, this involves use of a threshold nonlinearity.

Chapter 2 A Model of Visual Feature Detection

2.1 Introduction

This chapter describes a model of the development and function of simple cell receptive fields in mammalian primary visual cortex. The model assumes that *images are composed of combinations of a limited set of specific visual features* and that the goal of simple cells is to detect the presence or absence of these features. Based on a presumed statistical character of images and their visual features developed below, the model uses a constrained Hebbian learning rule to discover the structure of the features, and thus the appropriate response properties of simple cells, by training on a database of photographs. The response properties of the model simple cells agree qualitatively with neurophysiological observation.

2.1.1 The primary stages of mammalian visual processing

We begin with a brief review of the focus of our modeling efforts: the structure and function of the primary stages of mammalian visual processing (for a complete discussion see [Wandell, 1995]). The luminance intensities of an incoming visual image are transduced by the photoreceptor array of the retina. Since the dynamic range of their output is limited, the receptors automatically adjust the scale of their photosensitivity to match the ambient light level. Retinal circuitry then enhances image contrast by computing the differential between each photoreceptor's output and the outputs of its neighbors; this is known as the *center-surround* organization of the retina. The resulting image is relayed through the thalamus to the visual area of cortex. In the first area of cortical visual processing, labeled *V1* in primates, there are a class of cells known as *simple cells* which respond to certain properties of the image in a restricted spatial location known as the cell's *receptive field.* Two properties of particular interest are spatial frequency and orientation. A given simple cell responds maximally to the presence of a certain spatial frequency and orientation at a particular location. Typical simple cell receptive fields are illustrated in Figure 2.1. Note that we are purposely excluding from consideration other stimulus properties such as color and motion.

Figure 2.1: Schematic representations of the receptive fields of two typical simple cells in primary visual cortex. The dark coloration represents the part of the receptive field responsive to illumination which is darker than background, the light coloration represents responsiveness to lighter than background illumination. The receptive field on the left is therefore responsive to an oriented step change in illumination; the one on the right is responsive to an oriented line. Based on a figure in [Miller, 1995].

The goal of our modeling efforts is to offer an information theoretic argument as to why simple cells are tuned for a specific spatial frequency and orientation and to propose a mechanism for developing their specificities based on sensory experience with the visual world. Much work has already been done on this topic, which will be summarized next, and the final section of this chapter includes a discussion of our model in the context of the literature.

2.1.2 Models of simple cell receptive fields

In the following brief survey of previous modeling of simple cell receptive fields, we divide the models into two categories: *phenomenological* and *functional.* A phenomenological model of simple cell receptive fields proposes a mechanism which produces patterns which resemble simple cell receptive fields independent of any functional role; in other words, the model is able to replicate the phenomena without providing an argument for its purpose. In a functional model of receptive fields, the mechanism for the generation of receptive fields is based on their proposed function. Our model falls into the latter category.

In typical phenomenological model, such as the work of [Linsker, 1986], simple cells receive random visual input with particular correlation structure, and their synapses are modified according to a Hebbian learning rule made competitive by multiplicative or sub-

tractive weight decay. Analysis of this type of model in [Miller and Mackay, 1994] has shown that the resulting receptive fields arise from modes of the correlation structure of the random input. While such models may provide explanations of prenatal development, their use of random input is unrealistic for animals experiencing a sensory world. Furthermore, development is treated as a separate issue from functional operation.

Of the functional models that have been proposed, the most significant is the recent work of [Olshausen and Field, 1996]. These authors propose that the computational purpose of simple cells is to form an efficient linear basis for representing images. Their measure of efficiency is "sparseness," meaning that any particular image should be represented with as few basis coefficients as possible. Their model is trained on a database of real images using various nonlinear measures of sparseness, and the resulting receptive fields qualitatively resemble those of real simple cells. As this model is the most similar in the literature to our own, we return to discuss it further in the final section.

2.1.3 Chapter outline

The remainder of this chapter is divided into six sections. Since the simple cell receptive field model is based on assumptions about the statistical character of images, the first section is both a general introduction to the statistical modeling of images and an exposition of our specific statistical model of images as being composed of visual features. In the second section, we establish the theoretical basis for our model of the function of simple cells by deriving the optimal detection strategy for visual features, assuming their form is known. The section concludes by describing the model in terms of specific neurobiological mechanisms. In the third section, the specific form of the features is assumed to be unknown, and a constrained Hebbian learning rule is derived to determine the form of features from example images. This Hebbian rule serves as our model for the development of simple cell receptive fields. In the fourth section, the model is elaborated to accommodate variation in lighting conditions and sensor noise. The fifth section contains results from training on a database of photographs, and the model simple cell receptive fields are shown to be qualitatively similar to those observed experimentally. The last section discusses the model in terms of its relation to the work of other authors, the mathematical techniques used, and its potential for further application.

For definitions of terms and notation in this chapter, please refer to Section 1.1. As men-

tioned in that section, all derivations below use one-dimensional spatial signals to simplify notation, but the results naturally generalize to the two dimensions of real images.

2.2 Statistical signal models of images

Perhaps the most powerful and successful technique for describing the highly complex structure of images is to model an image as if it were the output of a stochastically driven process. This section begins with the introduction of the prototypical structure of statistical signal models, followed by two specific models. The first is the *standard model*, which forms the basis for the methods of classical linear signal processing. After presenting it, we demonstrate its limitations as a model of real images. We then propose a more detailed model of real images, the *features model*, which will be fully developed in following sections as a basis for explaining the development and function of simple cell receptive fields.

2.2.1 The prototypical model

The prototypical statistical signal model is that one or more independent, white, stochastic signal sources known as *innovations processes* are passed through corresponding linear *innovations filters* and then summed together to produce the *observed process*; the observed process is the image seen by the observer. The model is illustrated in schematically in Figure 2.2, and the notational conventions which will be used to describe the various parts of the model are summarized in the table below.

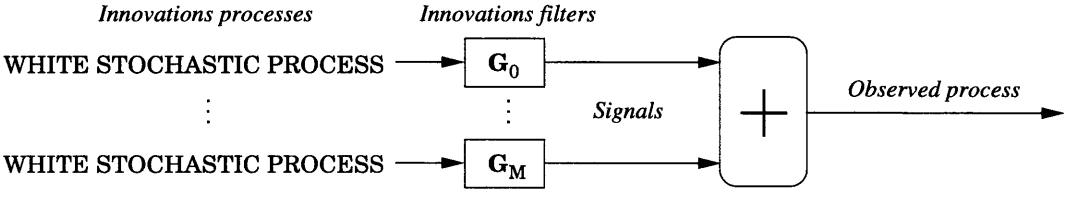


Figure 2.2: The prototypical statistical signal model. Independent, white, stochastic innovations processes drive corresponding innovations filters which are summed together to produce the observed process.

i_m	Innovations process m (a white, stochastic process)
G_m	Spatial frequency response of innovations filter m (a linear filter)
g_m	Impulse response of innovations filter m (a linear filter)
s_m	Signal m (innovations process m passed through innovations filter m)
z	Observed process (sum of the signals, the image seen by the observer)

Using this notation, the equations describing the model can be written as follows. Each of the innovations processes i_m is passed through an innovations filter to generate signal s_m:

$$s_m(x) = g_m(x) * i_m(x) \tag{2.1}$$

where "$*$" denotes convolution and x is one-dimensional spatial position. The observations process z is the sum of all of the signals s_m:

$$z(x) = \sum_m s_m(x) \tag{2.2}$$

For any particular model of this structure, two items must be specified based on our *a priori* hypotheses about the mechanisms underlying image generation: the number of innovations processes and filters and the stochastic nature of the individual innovations processes. The spatial frequency response or impulse response functions of the innovations filters may then be adjusted so that the model best fits a representative set of real image data.

The innovations processes are the sources of information in the image, and therefore an important statistical signal processing operation is to optimally estimate the innovations processes given the observed process. Optimality is defined in the minimum mean squared error sense, *i.e.* the optimal estimator produces the minimum mean squared error between its estimate of an innovations process and the true innovations process. The optimal innovations estimator can be used to perform many fundamental signal processing tasks such as redundancy reduction, noise suppression, and in the context of the proper image model, feature detection.

For each of the two models considered below (the standard model and the features model), we are therefore interested in specifying the following:

- The number of innovations processes and filters.

- The stochastic nature of the individual innovations processes.
- How to adjust the spatial frequency response or impulse functions of the innovations filters to best fit real image data.
- The optimal innovations estimator.

2.2.2 The standard model

The standard model is the simplest possible form of the prototype outlined in the last section. It has one white Gaussian innovations process i_0 with unit variance and a single innovations filter G_0, as illustrated in Figure 2.3.

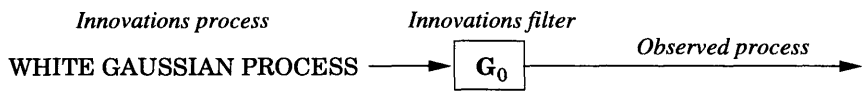


Figure 2.3: The standard model. One white, Gaussian innovations process drives the linear innovations filter G_0.

As discussed in the previous section, the model must be fit to real image data by adjusting the frequency response of the innovations filter, $G_0(j\omega_x)$. The power spectrum of images predicted by the standard model is

$$S_{zz}(\omega_x) = |G_0(j\omega_x)|^2 \tag{2.3}$$

The innovations filter $G_0(j\omega_x)$ can therefore be fit using the estimated power spectrum of real images $\hat{S}_{zz}(\omega_x)$:

$$G_0(j\omega_x) = |\hat{S}_{zz}(\omega_x)|^{\frac{1}{2}} \tag{2.4}$$

As has been reported in the literature [Field, 1994], by using a database of photographs to estimate S_{zz}, one finds that $G_0(j\omega_x)$ is approximately proportional to $\frac{1}{\omega_x}$.

The final task is to specify the optimal innovations estimator. For the standard model, the estimator is a linear filter, as illustrated in Figure 2.4. The filter has the form

$$H_0(j\omega_x) = \frac{1}{G_0(j\omega_x)} \tag{2.5}$$

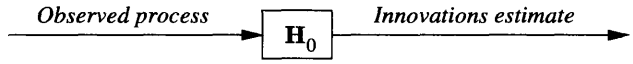


Figure 2.4: The optimal innovations estimator for the standard model. Given the observed process of the standard model, the optimal estimate of the innovations process is the linear filter H_0.

While the standard image model has been successfully used in many applications, it has a significant limitation: the signals it describes lack the semantic content of real images. We hypothesize that an important aspect of real images is that they contain visual *features*, which we define with the following:

DEFINITION *A feature is a specific spatial pattern which can be described as being present or absent in any given image location.*

By this broad definition, a feature may be something as simple as a line segment or as complex as a human face.

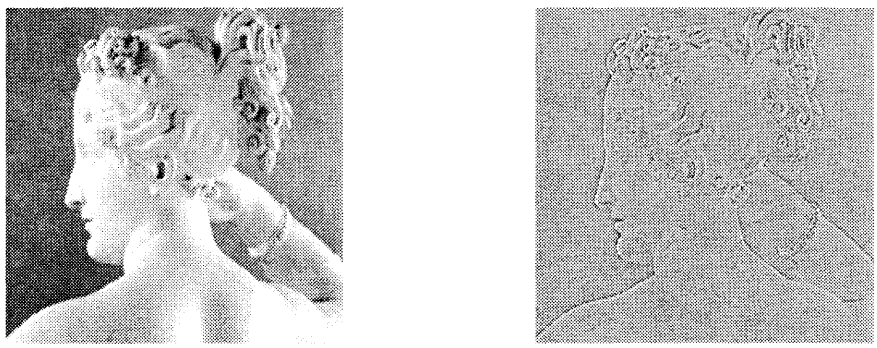


Figure 2.5: An image analyzed in the context of the standard model. *Left:* A sample real image. *Right:* The image passed through the innovations estimator derived from the average power spectrum of a database of photographs (refer to (2.5)).

To demonstrate this particular deficiency of the standard model, it was used to analyze the sample real image shown in Figure 2.5. The image was passed through the optimal innovations estimator H estimated from the average power spectrum of a database of photographs. If the standard model were a complete model of real images, the resulting innovations estimate would be white, Gaussian noise. However, even though a significant amount of structure has been removed from the image by the innovations estimator, the semantic content of the image remains. We propose that real images require an alternative,

more detailed model which supplements the standard model with stochastic processes which explicitly add features.

2.2.3 The features model

Central to the definition of a feature is the idea that a binary judgment can be made about whether it is present or absent at any given image location. The features model therefore augments the standard model with a set of M Poisson impulse innovations processes to generate M distinct visual features. Recalling the definition from Section 1.1.1, an impulse process is the sum of uniquely shifted Dirac delta functions:

$$i_m(x) = \sum_j \delta(x - x_j), \text{ where } x_j \neq x_k \text{ if } j \neq k \tag{2.6}$$

In a Poisson impulse process, the positions of the impulses are statistically independent. Although in the prototype statistical signal model asserts that the innovations processes be white, we relax this slightly for the Poisson process to allow a nonzero mean. If the mean of a particular Poisson process is λ, its power spectrum is white plus a DC term:

$$\lambda + \lambda^2 \delta(\omega_x) \tag{2.7}$$

When passed through a linear innovations filter G_m, an impulse process produces a series of shifted images of the impulse response of G_m:

$$\sum_j g_m(x - x_j) \tag{2.8}$$

The feature is thus only present in the image at spatial locations $\{x_j\}$. In the features model, the impulse responses of the additional innovations filters define the characteristic form of each feature. Referring to the examples of features mentioned in the previous section, the impulse response of an individual filter might be as simple as a line segment or as complex as a human face. We denote the combination of an impulse process and its corresponding innovations filter as a *feature generator*. The more feature generators the model uses, the less necessary the white Gaussian innovations process.

In the context of the features model, the task of detecting the presence or absence of a particular feature can be interpreted as optimally estimating the innovations processes

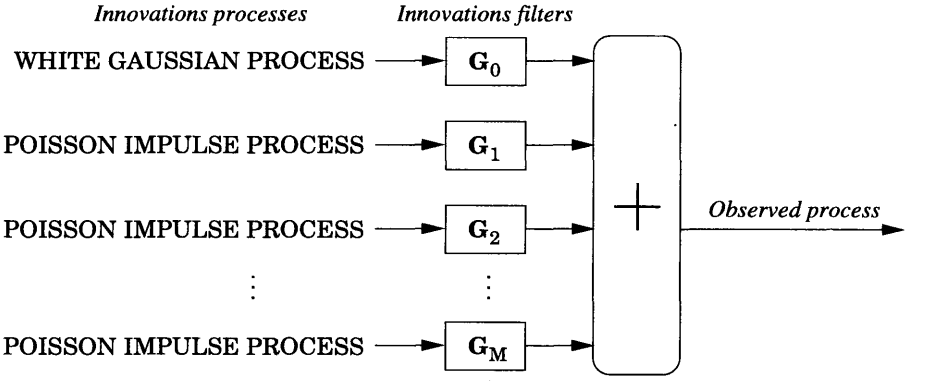


Figure 2.6: The features model. The standard model is augmented with Poisson impulse innovations processes and their corresponding innovations filters which have impulse responses in the shape of various visual features.

generating that feature. Due to the binary nature of the impulse innovations processes, the optimal estimator involves the use of a nonlinear classifier. It is our hypothesis that the simple cells of visual cortex act as such estimators.

In accordance with our discussion in Section 2.2.1, there are two important aspects to specify in the model. First, in the next section we derive the form of the optimal innovations estimators which serve as feature detectors. Second, in the following section we specify a method for fitting the model's parameters, and thus the parameters of the innovations estimators, to real image data.

2.3 The optimal innovations estimator for the features model

In this section, we derive the optimal innovations estimator for the features model in four stages of increasing complexity. In the first stage, the optimal innovations estimator is derived for a features model reduced to the Gaussian innovations process and a single impulse innovations process with no innovations filters. The second stage adds an innovations filter for the impulse process, the third stage adds the innovation filter for the Gaussian innovations process, and in the fourth stage the estimator is derived for the complete features model. Much of the mathematical development outlined below is well known in *detection theory*, which is discussed in detail in [McDonough and Whalen, 1995].

One important practical difference between the models discussed in this section and the

features model as originally described is that all signals are assumed to be bandlimited due to physical constraints. The images are therefore represented by discrete samples of their luminance amplitude, known as *pixels*, and the impulse innovations processes are bandlimited impulse processes composed of shifted Kronecker delta functions:

$$\sum_i \delta[k - k_i], \text{ where } k_i \neq k_j \text{ if } i \neq j \tag{2.9}$$

We make a series of five constraining assumptions about the innovations processes and filters for the purpose of mathematical tractability. The utility of these assumptions is demonstrated in the results section by using the model on real image data.

2.3.1 The first stage

The first stage of deriving the optimal innovations estimator begins with only one of the bandlimited, binary valued innovations processes, i_1, whose two amplitude values are zero and one. The probability of a sample with amplitude one ($i_1[k] = 1$) is parameterized by λ, and the probability of zero ($i_1[k] = 0$) is thus $1 - \lambda$. The parameter λ must be known *a priori*. Each sample of i_1 is chosen independently, so that knowledge of $i_1[k]$ does not bias the probability distribution of $i_1[k+1]$.

The observed signal is the sum of this innovations process and the Gaussian innovations process, i_0. We assume for now that i_0 has a variance of one, and so each sample of the observed process z is therefore a sum of $i_1[k]$ and an independently drawn, zero mean, Gaussian random variable $i_0[k]$ of variance one:

$$z[k] = i_1[k] + i_0[k] \tag{2.10}$$

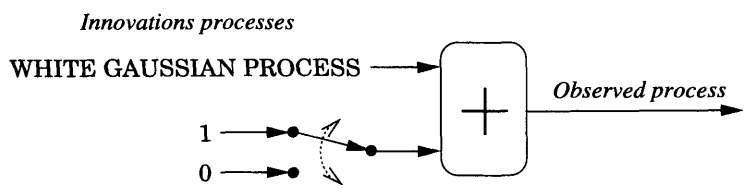


Figure 2.7: The observed process of the first stage. A single binary innovations process, the innovations process of the standard model, and no innovations filters.

Given $z[k]$, the receiver must estimate $i_1[k]$ with minimum error. This is the same scenario as the communications problem posed in Section 1.2.1; as before, the threshold function $\Theta(x;\theta)$ serves as the optimal estimator. This scenario is diagrammed in Figure 2.7.

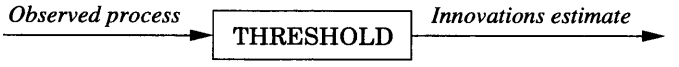


Figure 2.8: The optimal innovations estimator of the first stage. A threshold function.

Paralleling the discussion in Section 1.2.1, we denote $i_1[k] = 0$ by D_0 and $i_1[k] = 1$ by D_1. The value of θ should be set so that $\Theta(i_1[k] + i_0[k];\theta)$ will be one when

$$p(D_1|z[k]) > p(D_0|z[k]) \tag{2.11}$$

and zero otherwise. Unlike the derivation in Section 1.2.1 $p(D_1) \neq p(D_0)$; in fact, $p(D_1) = \lambda$ and $p(D_0) = 1 - \lambda$. The conditional probability distributions for $z[k]$ are therefore

$$p(z[k]|D_0) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{z[k]^2}{2\sigma^2} \tag{2.12}$$

$$p(z[k]|D_1) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{(z[k]-1)^2}{2\sigma^2} \tag{2.13}$$

The optimal θ satisfies the equation

$$p(D_1|\theta) = p(D_0|\theta) \tag{2.14}$$

Using Bayes' rule and substituting the Gaussian probability distributions above:

$$\frac{1-\lambda}{\sqrt{2\pi}} \exp -\frac{\theta^2}{2} = \frac{\lambda}{\sqrt{2\pi}} \exp -\frac{(\theta-1)^2}{2} \tag{2.15}$$

Solving for θ,

$$\theta = \frac{1}{2} + \ln \frac{1-\lambda}{\lambda} \tag{2.16}$$

If $\lambda = \frac{1}{2}$, then $\theta = \frac{1}{2}$, and therefore this more general result agrees with the specific case derived in Section 1.2.1. Using the above expression for θ, the complete optimal innovations estimator can be written as

$$\Theta(z[k];\theta) \tag{2.17}$$

2.3.2 The second stage

The next stage of complexity is to pass the innovations process i_1 through the innovations filter G_1 to produce s_1. Estimating i_1 will therefore involve examining multiple pixels of z. In order to constrain the number of pixels N that must be examined, we constrain G_1 to have a finite impulse response, which is the same as constraining visual features to have a finite spatial extent in the image.

ASSUMPTION 1 *The impulse response of each feature generator is only nonzero for N pixels, where N is a finite number.*

The observed process z of the second stage is the sum of s_1 and the Gaussian innovations process:

$$z[k] = s_1[k] + i_0[k] \tag{2.18}$$

where s_1 is defined in (2.1).

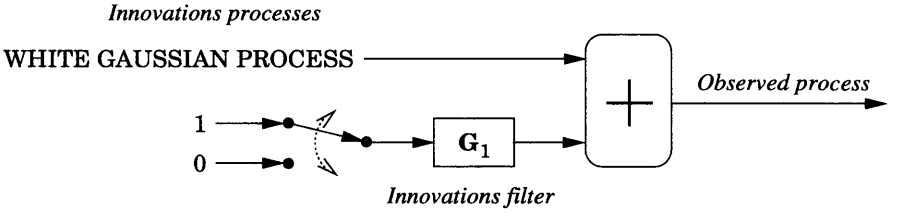


Figure 2.9: The observed process of the second stage. The binary innovations process i_1 is passed through an innovations filter G_1, then added to the white Gaussian innovations process i_0 to produce the observed process.

The optimal innovations estimator has two parts. The first is a linear filter H_1 called a *matched filter*, which maximizes the ratio of its response to the signal s_m to its response to the Gaussian innovations process i_0. The second is a threshold operator to perform a Bayesian classification just as in the first stage. This scenario is schematically illustrated in Figure 2.10. The expression for the matched filter is

$$H_1(j\omega) = G_1^*(j\omega) \tag{2.19}$$

which is a filter whose impulse response is the spatial reversal of the impulse response of G_1. The derivation for this result can be found in [McDonough and Whalen, 1995]. We present a geometric argument, using a vector representation of signals[1]. Using N-dimensional vectors, where N is chosen to be the length of the impulse response of the filter G_1, the vector representation of G_1 is

$$\mathbf{g}_1 = \begin{bmatrix} g[0] \\ g[1] \\ \vdots \\ g[N-1] \end{bmatrix} \tag{2.20}$$

Using this, (2.18) may be rewritten as

$$z[k] = \mathbf{g}_1 \cdot \mathbf{i}_1[k] + i_0[k] \tag{2.21}$$

Ignoring for the moment other possible values for $\mathbf{i}_1[k]$, the estimator should distinguish the following two cases:

$$D_1\text{: } \mathbf{i}_1[k] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \qquad D_0\text{: } \mathbf{i}_1[k] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.22}$$

If the estimator can accurately distinguish these two cases, it can estimate the value of $i[k-N+1]$. Since it must make this estimation based on the observed process z, note that the observed vector in the two cases is

$$D_1\text{: } \mathbf{z}[k] = \mathbf{h}_1 + \mathbf{i}_0[k] \qquad D_0\text{: } \mathbf{z}[k] = \mathbf{i}_0[k] \tag{2.23}$$

[1]The vector representation of signals is defined in Section 1.1.5

where $\mathbf{h}_1$ is an N-dimensional vector whose elements are the elements of $\mathbf{g}$ in reverse order ($\mathbf{h}_1$ is the vector representation of the matched filter):

$$\mathbf{h}_1 = \begin{bmatrix} g[N-1] \\ g[N-2] \\ \vdots \\ g[0] \end{bmatrix} \tag{2.24}$$

Just as in the previous section, D_1 is more likely to have occurred when

$$p(D_1|\mathbf{z}[k]) > p(D_0|\mathbf{z}[k]) \tag{2.25}$$

Making a Bayesian substitution, the two sides of (2.25) are equal at the classification boundary:

$$p(\mathbf{z}[k]|D_1)p(D_1) = p(\mathbf{z}[k]|D_0)p(D_0) \tag{2.26}$$

Since the probability of $d[k-N+1]$ being one is λ, $p(D_1)$ is also λ and $p(D_0) = 1-\lambda$. Furthermore, because i_0 is white, the probability distributions of the two cases are that of uncorrelated, multivariate Gaussian random vectors:

$$p(\mathbf{z}[k]|D_0) = \frac{1}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\mathbf{z}[k]) \cdot (\mathbf{z}[k])}{2} \tag{2.27}$$

$$p(\mathbf{z}[k]|D_1) = \frac{1}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\mathbf{z}[k]-\mathbf{h}_1) \cdot (\mathbf{z}[k]-\mathbf{h}_1)}{2} \tag{2.28}$$

Making the appropriate substitutions in (2.26) and solving:

$$\mathbf{h}_1 \cdot \mathbf{z}[k] = \frac{1}{2}\mathbf{h}_1 \cdot \mathbf{h}_1 + \ln \frac{1-\lambda}{\lambda} \tag{2.29}$$

If θ is set to

$$\theta = \frac{1}{2}\mathbf{h}_1 \cdot \mathbf{h}_1 + \ln \frac{1-\lambda}{\lambda} \tag{2.30}$$

the complete optimal innovations estimator can be written as

$$\Theta(\mathbf{h}_1 \cdot \mathbf{z}[k]; \theta) \tag{2.31}$$

which combines passing the observed process through the matched filter ($\mathbf{h}_1 \cdot \mathbf{z}[k]$) and the threshold operation. The innovations estimator is illustrated schematically in the block diagram of Figure 2.10 and geometrically in Figure 2.11.

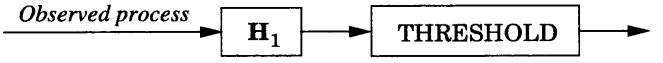


Figure 2.10: The optimal innovations estimator of the second stage. The estimator is a linear *matched filter* followed by a threshold acting as a nonlinear classifier.

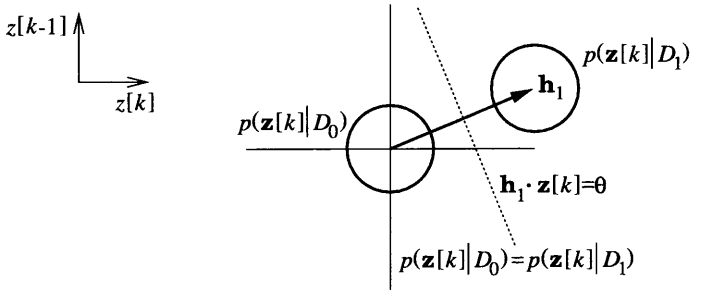


Figure 2.11: The geometric illustration of the matched filter in two dimensions. The space shown is the vector space of a two dimensional $\mathbf{z}[k]$, so that the abscissa is $z[k]$ and the ordinate is $z[k-1]$. The circles mark the equiprobability contours at one standard deviation of $p(\mathbf{z}[k]|D_0)$ and $p(\mathbf{z}[k]|D_1)$. The vector $\mathbf{h}_1$ represents the matched filter; applying the filter is equivalent to projecting $\mathbf{z}$ onto $\mathbf{h}_1$. The dotted line is the graph of $p(\mathbf{z}[k]|D_0) = p(\mathbf{z}[k]|D_1)$, which also marks the decision plane defined by applying the threshold function to the output of the matched filter.

Returning to the earlier caveat of ignoring other values of $\mathbf{i}[k]$, there are two cases which are of concern. The first is values of $\mathbf{i}[k]$ in which the one value is not the last element of the vector. For example,

$$\mathbf{i}_1[k] = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.32}$$

The classifier should assign this to the D_0 case, since a single feature should not be detected at multiple locations. The second case of concern is more than one occurrence of a feature within a single $\mathbf{i}_1[k]$, which would appear as multiple amplitude one elements. For example,

$$\mathbf{i}_1[k] = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{2.33}$$

This case should be classified as D_1, since the feature is actually present. In order to simplify these two cases, we make the following assumption:

ASSUMPTION 2 *The impulse response of each feature generator is orthogonal to all spatial translations of itself of more than κ pixels, where $\kappa \ll N$.*

By this assumption, the inherent structure of the feature's pattern causes the classifier to act correctly in both situations. If it results in significant misclassifications for shifts of less than κ the results of the detector can be averaged and subsampled.

2.3.3 The third stage

The next stage of complexity is to add the innovations filter G_0 for the white, Gaussian innovations process of the standard model, as illustrated in Figure 2.12. The observed process z of the third stage is the sum of s_1 and s_0:

$$z[k] = s_1[k] + s_0[k] \tag{2.34}$$

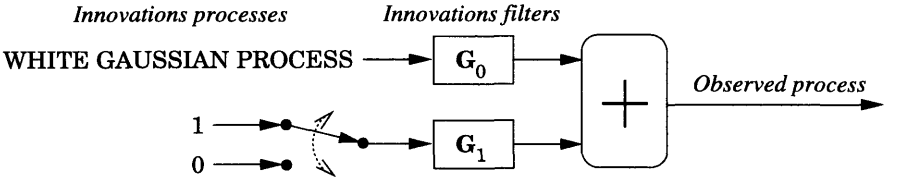


Figure 2.12: The observed process of the third stage.

The matched filter of the last section is not the optimal estimator for i_1 given this observed process because the samples of $s_0[k]$ are correlated and the mathematical derivation relied on the assumption of a white Gaussian process. See Figure 2.13 for a geometrical illustration of this point.

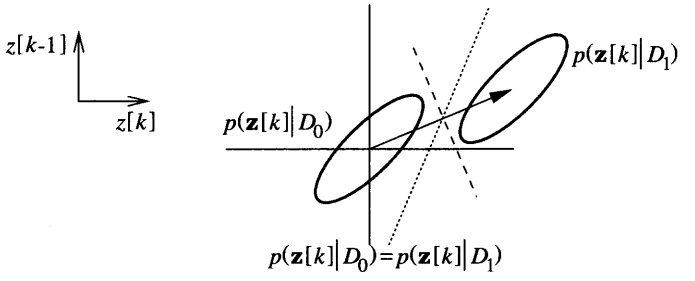


Figure 2.13: The geometric illustration of the why the matched filter developed in Section 2.3.2 does not work when the Gaussian process is correlated (nonwhite). The elipses mark the equiprobability contours at one standard deviation of $p(\mathbf{z}[k]|D_0)$ and $p(\mathbf{z}[k]|D_1)$. The dotted line is the graph of $p(\mathbf{z}[k]|D_0) = p(\mathbf{z}[k]|D_1)$, and the dashed line is the decision boundary that would be formed if a matched filter were applied directly; since these two lines do not coincide, the matched filter is not optimal. This situation may be transformed into the one shown below in Figure 2.14 by first passing z through a whitening filter, after which a matched filter may be applied.

The optimal estimator for the third stage is the *generalized matched filter*, which is a linear filter best understood as a combination of two filters: a *whitening filter* and a matched filter. The whitening filter W is designed to transform the present problem into the matched filter problem by converting the signal s_0 into a white, uncorrelated process. Its form is thus:

$$W(j\omega) = \frac{1}{|G_0(j\omega)|} \tag{2.35}$$

We will use the tilde to denote signals after they have passed through the whitening filter. For example, the signal s_0 after it has passed through the whitening filter is $\tilde{s}_0$:

$$\tilde{s}_0[k] = w[k] * s_0[k] \tag{2.36}$$

and this signal, as expected, has a white power spectrum:

$$S_{\tilde{s}_0\tilde{s}_0}(\omega) = 1 \tag{2.37}$$

Also note that the variance of $\tilde{s}_0$ is now one by definition.

The result of passing the observed process z through the whitening filter is

$$\tilde{z}[k] = \tilde{s}_0[k] + \tilde{s}_1[k] \tag{2.38}$$

Since $\tilde{s}_0$ is a white Gaussian process, the matched filter from last section applies. The difference is that the matched filter must now match the impulse response of G_1 after passing through W:

$$\tilde{H}_1(j\omega) = \Big(W(j\omega)G_1(j\omega)\Big)^* \tag{2.39}$$

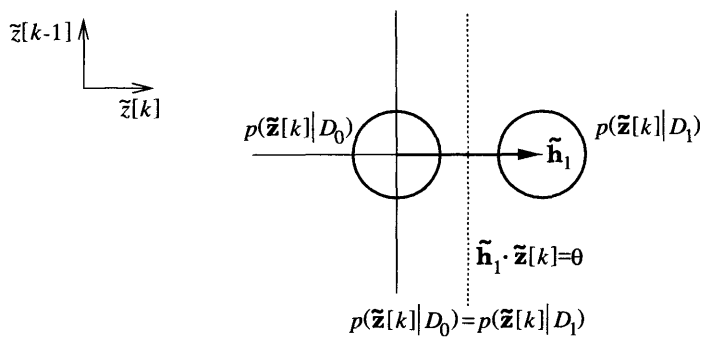


Figure 2.14: The geometric illustration of using a matched filter after applying a whitening filter to the situation illustrated in Figure 2.13. After the whitening filter is applied, it is the same geometric scenario shown in Figure 2.11.

This is illustrated geometrically in Figure 2.14. The generalized matched filter is formed from the chaining together of the whitening filter and the matched filter:

$$H_1(j\omega) = \tilde{H}_1(j\omega)W(j\omega) \tag{2.40}$$

Just as in the previous section, a threshold function applied to the output of the generalized matched filter is the optimal Bayesian classifier. This is illustrated in Figure 2.15. The conditional probability distributions for $\tilde{\mathbf{z}}[k]$ are

$$p(\tilde{\mathbf{z}}[k]|D_0) = \frac{1}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\tilde{\mathbf{z}}[k]) \cdot (\tilde{\mathbf{z}}[k])}{2} \tag{2.41}$$

$$p(\tilde{\mathbf{z}}[k]|D_1) = \frac{1}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\tilde{\mathbf{z}}[k] - \tilde{\mathbf{h}}_1) \cdot (\tilde{\mathbf{z}}[k] - \tilde{\mathbf{h}}_1)}{2} \tag{2.42}$$

The value of θ is

$$\theta = \frac{1}{2}\tilde{\mathbf{h}}_1 \cdot \tilde{\mathbf{h}}_1 + \ln \frac{1-\lambda}{\lambda} \tag{2.43}$$

The complete optimal innovations estimator can therefore be written as

$$\Theta(\tilde{\mathbf{h}}_1 \cdot \tilde{\mathbf{z}}[k]; \theta) \tag{2.44}$$

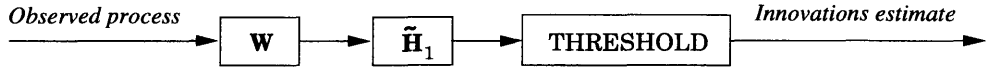


Figure 2.15: The innovations estimator of the third stage. The generalized matched filter is a whitening filter followed by a postwhitened matched filter.

2.3.4 The fourth stage

The fourth stage is the complete features model with all innovations processes and their innovations filters. Aside from the white, Gaussian innovations process i_0 of the standard model, there are M bandlimited Poisson, impulse innovations processes $i_{1...M}$, as illustrated in Figure 2.16.

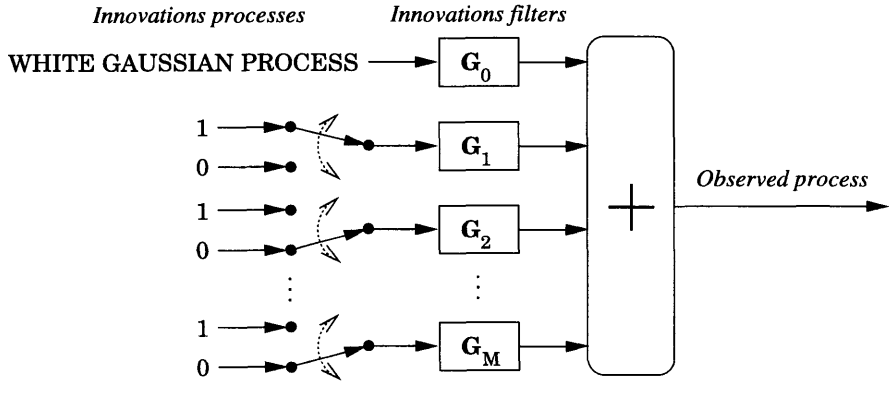


Figure 2.16: The observed process of the fourth stage, the complete bandlimited features model. This is a bandlimited version of Figure 2.6.

The observed process of the complete features model is the sum of all innovations processes passed through their innovations filters:

$$z[k] = \sum_{m=0}^{M} s_m[k] \tag{2.45}$$

In order to simplify the design of the optimal innovations estimator, we assume that there is no exact spatial coincidence of features:

ASSUMPTION 3 *No more than one feature is initiated at any given spatial location.*

Mathematically stated, this assumes that at most one of the binary innovations processes has an amplitude of one at any given sample index:

$$\sum_{m=1}^{M} i_m[k] \in \{0, 1\} \tag{2.46}$$

We denote the event that $i_m[k] = 0$ for all m as D_0, and that $i_m[k] = 1$ as D_m. The probabilities of these events must sum to one:

$$\sum_{m=0}^{M} p(D_m) = 1 \tag{2.47}$$

Based on Assumption 3, the optimal innovations estimator must solve the following equation for m at every pixel:

$$\max_{m=0\ldots M} p(D_m|\mathbf{z}[k]) \tag{2.48}$$

By making a Bayesian substitution, this is the same as maximizing

$$\max_{m=0\ldots M} p(D_m)p(\mathbf{z}[k]|D_m) \tag{2.49}$$

In order to solve this equation, the probabilities $p(D_m)$ of $i_m[k]$ being one must be specified. In the first three stages, there was only a single necessary *a priori* parameter λ. However, with M impulse innovations processes, there are potentially M *a priori* parameters to set. We limit it to a single parameter with the following assumption:

ASSUMPTION 4 *All features occur with the same per sample frequency* λ.

From Assumption 4, $p(D_{1...M}) = \lambda$, and therefore $p(D_0) = 1 - \lambda M$. Next, the probability $p(\mathbf{z}[k]|D_m)$ must be calculated. The exact expression for this is complicated, but it can be simplified with two approximations which rely on assuming that there a large number of feature generators ($M \gg 1$). In order to state these assumptions, we define $n_m[k]$ as

$$n_m[k] = \sum_{j=0, j\neq m}^{M} s_m \tag{2.50}$$

This allows the observation process to be rewritten as

$$z[k] = s_m[k] + n_m[k] \tag{2.51}$$

for any given m. The first approximation is that there are enough statistically independent innovations processes so that by the central limit theorem, the sum of signals s_j converges to a nonwhite, Gaussian process. Therefore, $n_m[k]$ is approximately a nonwhite, Gaussian process. Just as for the generalized matched filter, this process can be decorrelated with whitening filter W_m:

$$W_m(j\omega) = \frac{1}{|S_{n_m n_m}(\omega)|^{\frac{1}{2}}} \tag{2.52}$$

Furthermore, since the probabilities $p(D_m)$ sum to one, λ must be less than $\frac{1}{M}$. If M is large, each feature therefore contributes little to the power spectrum of the observed process. The second approximation is that the whitening filter W_m is approximately the same as the whitening filter for the entire observed process:

$$W_m(j\omega) \approx W(j\omega) = \frac{1}{|S_{zz}(\omega)|^{\frac{1}{2}}} \tag{2.53}$$

The combination of the two approximations is that if the observed process is passed through a whitening filter, the conditional probability distributions for $p(\tilde{\mathbf{z}}[k]|D_m)$ are approximately

$$p(\tilde{\mathbf{z}}[k]|D_0) \approx \frac{1}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\tilde{\mathbf{z}}[k]) \cdot (\tilde{\mathbf{z}}[k])}{2} \tag{2.54}$$

$$p(\mathbf{z}[k]|D_m) \approx \frac{1}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\tilde{\mathbf{z}}[k] - \tilde{\mathbf{h}}_m) \cdot (\tilde{\mathbf{z}}[k] - \tilde{\mathbf{h}}_m)}{2} \tag{2.55}$$

which can be substituted into (2.49). The next step in solving (2.49) is to find the maximum among $p(D_{1...M}|\tilde{\mathbf{z}}[k])$:

$$\max_{m=1...M} \frac{\lambda}{(2\pi)^{\frac{N}{2}}} \exp \frac{-(\tilde{\mathbf{z}}[k] - \tilde{\mathbf{h}}_m) \cdot (\tilde{\mathbf{z}}[k] - \tilde{\mathbf{h}}_m)}{2} \tag{2.56}$$

which is the same as minimizing

$$\min_{m=1...M} \tilde{\mathbf{z}}[k] \cdot \tilde{\mathbf{z}}[k] - 2\tilde{\mathbf{z}}[k] \cdot \tilde{\mathbf{h}}_m + \tilde{\mathbf{h}}_m \cdot \tilde{\mathbf{h}}_m \tag{2.57}$$

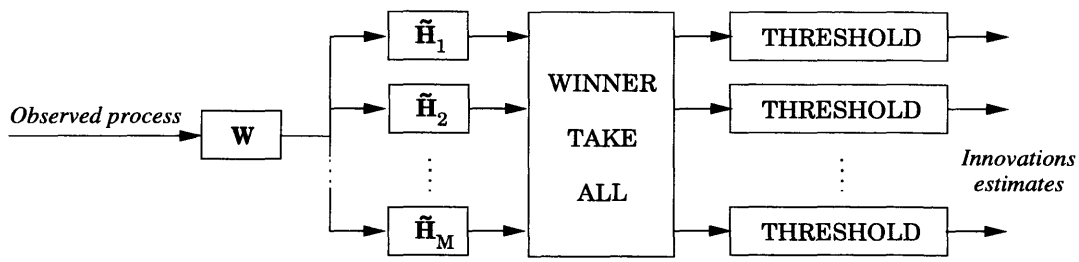


Figure 2.17: The optimal innovations estimator for the fourth stage, the complete bandlimited features model. The observed process is whitened, then processed by a bank of matched filters. The largest matched filter output is selected by the WINNER TAKE ALL block and is then passed through a threshold function.

A geometric interpretation of (2.57) is that it is determining which of the vectors $\tilde{\mathbf{h}}_m$ is closest to $\tilde{\mathbf{z}}[k]$. When finding this minimum, $\tilde{\mathbf{z}}[k] \cdot \tilde{\mathbf{z}}[k]$ can be ignored because it is constant. In order to further simplify the computation, we make the following assumption:

ASSUMPTION 5 *The postwhitened impulse responses of the feature generators have the same power.*

This can be stated mathematically as

$$||\tilde{\mathbf{h}}_m|| = \Lambda \text{ for all } m \tag{2.58}$$

where Λ is a constant. By this assumption, $\tilde{\mathbf{h}}_m \cdot \tilde{\mathbf{h}}_m = \Lambda^2$ for all m. Therefore, minimizing (2.57) is the same as

$$\max_{m=1...M} \tilde{\mathbf{h}}_m \cdot \tilde{\mathbf{z}}[k] \tag{2.59}$$

Using the value of m which satisfies (2.59), the second step in solving (2.49) is to find maximum of $p(D_0|\tilde{\mathbf{z}}[k])$ and $p(D_m|\tilde{\mathbf{z}}[k]$. This can be accomplished with the threshold operation, where the threshold parameter θ is given by

$$\theta = \frac{1}{2}\Lambda^2 + \ln\frac{1-\lambda M}{\lambda} \tag{2.60}$$

Combining this with (2.59), the entire optimal innovations estimator for the features model can be written as

$$\Theta(\max_{m=1\ldots M} \tilde{\mathbf{h}}_m \cdot \tilde{\mathbf{z}}[k]; \theta) \tag{2.61}$$

which is illustrated schematically in Figure 2.17, where the WINNER TAKE ALL block is a device which finds which signal among its N inputs has the maximum amplitude at each sample k and passes that signal through unchanged; it outputs a zero amplitude sample for all other signals. This winner-take-all operation implements the solving for m in (2.59). A geometric illustration of the estimator is shown in Figure 2.18.

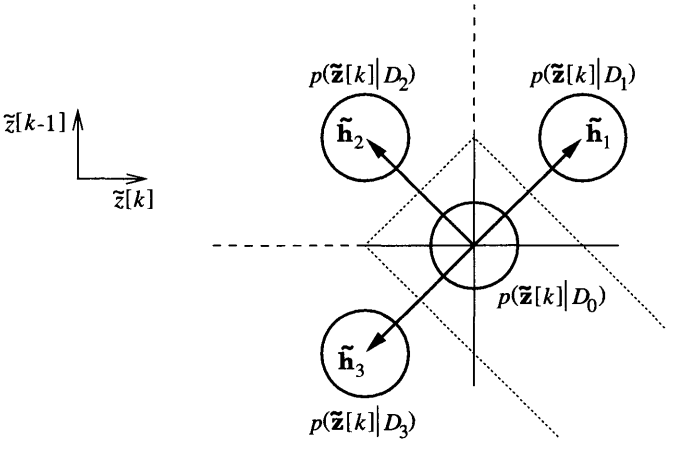


Figure 2.18: The geometric illustration of the optimal innovations estimator for a features model with three feature generators. The axes represent the components of the two dimensional, postwhitened, observed process $\tilde{\mathbf{z}}$. As before, the circles mark the equiprobability contours at one standard deviation of $p(\tilde{\mathbf{z}}[k]|D_m)$. The dashed lines are the decision boundaries formed by the winner-take-all operation, and the dotted lines are the decision boundary formed by the action of the threshold function.

2.3.5 Neurobiological mechanisms

The optimal innovations estimator described in the previous section has four distinct computational steps: a whitening filter, a matched filter, a winner take all operation and a threshold function. We next hypothesize a corresponding neurobiological mechanism responsible for each step. The image transduced by the retina is the observed process z. The contrast enhancing circuitry of the retina can be interpreted as the whitening filter W applied to z, so the retinal output is $\tilde{z}$. In the cortex, we propose that there are M simple cells which act as matched filters at each retinotopic location; each simple cell filters its input with the synaptic weightings specified by the vector $\tilde{\mathbf{h}}_m$. The winner-take-all operation can be effected by lateral inhibition among simple cells, and the threshold function can be implemented by the current threshold for spike output. We believe all of these mechanisms to be neurobiologically plausible.

In the next section, we describe a learning algorithm for the setting of the simple cell synaptic weights to the proper values of $\tilde{\mathbf{h}}_m$ based on sensory experience with example images.

2.4 Fitting the features model to real images

In the previous section, we derived the optimal innovations estimator for the features model assuming the innovations filters were known. In this section, the innovations filters are assumed to be unknown and must be estimated from real image data. The innovations filters need only be estimated indirectly insofar as they determine the matched filters of the optimal innovations estimators. In other words, the values of $\tilde{\mathbf{h}}_m$ must be computed from image data.

Before the model can be fit with image data, three parameters must be set *a priori*: θ (the threshold value), M (the number of features) and N (the number of dimensions of the vector representation of $\tilde{\mathbf{z}}$). In order to demonstrate that these are the only parameters that must be set *a priori*, we first define a normalized version of the matched filters $\tilde{\mathbf{h}}_m$:

$$\mathbf{w}_m = \frac{\tilde{\mathbf{h}}_m}{||\tilde{\mathbf{h}}_m||} \tag{2.62}$$

This normalizes the length of $\mathbf{w}_m$ to one: $||\mathbf{w}_m|| = 1$. If θ is redefined as

$$\theta = (\frac{1}{2}\Lambda^2 + \frac{1}{2}\ln\frac{1-\lambda M}{\lambda})\Lambda \tag{2.63}$$

then (2.61) can be rewritten as

$$\Theta(\max_{m=1...M} \mathbf{w}_m \cdot \tilde{\mathbf{z}}[k]; \theta) \tag{2.64}$$

Therefore, if N, M and θ are set *a priori*, the only parameters the optimal innovations estimator must compute from real data are the vectors $\mathbf{w}_m$. Note that for the visual cortex, the three *a priori* parameters can be specified genetically: N is the number of retinotopic dendritic inputs to each simple cell, M is the ratio of the number of simple cells to retinotopic inputs, and θ is the current threshold for somatic spiking.

The features model predicts that the probability distribution of the post-whitened observed process $\tilde{\mathbf{z}}$ is composed of $M+1$ Gaussian clusters, one at the origin and M corresponding to the individual features, as shown in Figure 2.18. The clusters are centered on the vectors $\tilde{\mathbf{h}}_m$, and therefore an algorithm designed to find clusters in observed data can determine the proper values for $\tilde{\mathbf{h}}_m$ (or its normalized counterpart, $\mathbf{w}_m$). The computationally simplest on-line method for locating clusters is the k-means clustering algorithm [Gersho and Gray, 1992]. The k-means algorithm begins by randomly distributed k cluster centers. Incoming data vectors are classified as belonging to the nearest cluster, where proximity is measured by the Euclidean distance between the data vector and the cluster center. After a data vector is assigned to a particular cluster, the center of the cluster is moved slightly toward the data vector.

In the context of the features model, this translates into the M matched filters $\mathbf{w}_m$ being initialized to random values, the classification step simply being the normal operation of the optimal innovations estimator, and the weight vectors being updated with the following rule:

$$\mathbf{w}_m \longmapsto \mathbf{w}_m + \mu\tilde{\mathbf{z}} \tag{2.65}$$

where μ is a learning rate parameter. There is one significant difference that the training algorithm must have from the standard k-means algorithm. Since all the vectors are assumed to be of length one, this length must be maintained during training. In other words, the vectors $\mathbf{w}_m$ are constrained to move on the surface of the unit hypersphere in N-dimensional

space. The following update rule has the same effect as (2.65), but due to the term in the denominator, the vector remains normalized to a length of one:

$$\mathbf{w}_m \longmapsto \frac{\mathbf{w}_m + \mu\tilde{\mathbf{z}}}{||\mathbf{w}_m + \mu\tilde{\mathbf{z}}||} \tag{2.66}$$

However, this relatively complicated to implement computationally, and implausible in neural hardware. We will therefore use the approximation technique proposed in [Oja, 1982] to derive a simpler, neurobiologically plausible rule. Assuming that μ is so small that $O(\mu^2)$ terms are negligible, an approximation to $||\mathbf{w}_m + \mu\tilde{\mathbf{z}}||$ is

$$(||\mathbf{w}_m||^2 + 2\mu\mathbf{w}_m \cdot \tilde{\mathbf{z}} + O(\mu^2))^{\frac{1}{2}} \tag{2.67}$$

Using the approximation $\sqrt{1+x} \approx 1 + \frac{x}{2}$ and the fact that $||\mathbf{w}_m|| = 1$, (2.67) reduces to

$$1 + \mu\mathbf{w}_m \cdot \tilde{\mathbf{z}} \tag{2.68}$$

Substituting this result, (2.66) becomes

$$\mathbf{w}_m \mapsto \frac{\mathbf{w}_m + \mu\tilde{\mathbf{z}}}{1 + \mu\mathbf{w}_m \cdot \tilde{\mathbf{z}}} \tag{2.69}$$

This can be simplified by multiplying numerator and denominator by $1 - \mu\mathbf{w}_m \cdot \tilde{\mathbf{z}}$:

$$\frac{\mathbf{w}_m + \mu(\mathbf{w}_m \cdot \tilde{\mathbf{z}})\mathbf{w}_m + \mu\tilde{\mathbf{z}} + O(\mu^2)}{1 - O(\mu^2)} \tag{2.70}$$

which can be reduced to

$$\mathbf{w}_m + \mu(\tilde{\mathbf{z}} - (\mathbf{w}_m \cdot \tilde{\mathbf{z}})\mathbf{w}_m) \tag{2.71}$$

Equation (2.66) can thus be approximated with the simpler learning rule

$$\Delta\mathbf{w}_m = \mu(\tilde{\mathbf{z}} - (\mathbf{w}_m \cdot \tilde{\mathbf{z}})\mathbf{w}_m) \tag{2.72}$$

In a neurobiological context, note that $\tilde{\mathbf{z}}$ represents the retinotopic input, $\mathbf{w}_m \cdot \tilde{\mathbf{z}}$ is the weighted sum of all synaptic input, and $\mathbf{w}_m$ represents the current synaptic weights. Therefore, the update rule at each synapse requires the locally available value of its own input and

own weight, and the only nonlocal information necessary is the somatically computed sum of all synaptic inputs. This is therefore a neurobiologically plausible, Hebbian style learning rule.

2.5 Lighting level variation and sensor noise

There are two issues which we have so far ignored in our model but are of significant practical concern: luminance variation and sensor noise. Lighting level variation is fluctation in the light level illuminating the image; we model this as a statistically independent, non-negative gain term $\beta(x)$ multiplying the image signal. Sensor noise is the result of electrical noise in a photoreceptor and is modeled by an additive, white Gaussian noise source of variance σ_n^2. A schematic diagram illustrating these two additional model complexities is shown in Figure 2.19.

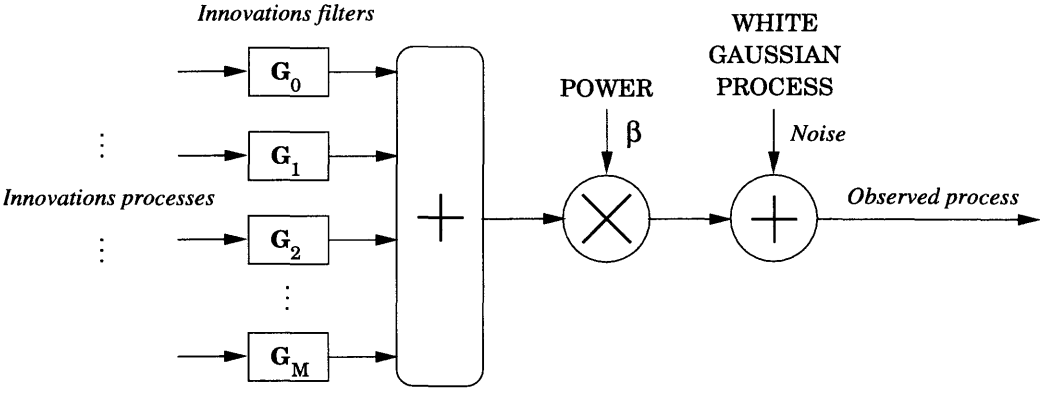


Figure 2.19: Adding lighting level variation and sensor noise to the model. The sum of the innovations processes passed through their corresponding innovations filters is multiplied by a statistically independent, non-negative gain term representing lighting level variation and is then summed with a white, Gaussian noise source representing the effects of sensor noise.

The difficulty presented by lighting level variation is that the matched filter strategy was developed with the assumption that signal power was known and constant. This was essential because it fixed the threshold value θ of the matched filter. However, since the lighting level multiplies the signal with an aribtrary gain factor $\beta(x)$, this gain must be estimated and divided out of the signal before it is passed through the optimal innovations estimator. This is illustrated in Figure 2.20.

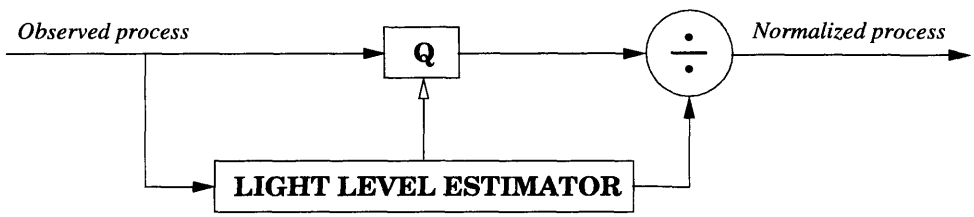


Figure 2.20: Countering the effects of lighting level variation by estimating the light level and dividing it out. Sensor noise is smoothed by the filter Q whose spatial frequency response is adjusting according to the estimated lighting level (represented by the arrow between the light level estimator and the filter Q).

In order to design a luminance estimator, we assume that the lighting level varies on a spatial scale greater than the spatial scale of the features. Without this assumption, lighting level and luminance variation due to the particular appearance of each feature would be confounded and impossible to separate. Since both signal mean and standard deviation are proportional to lighting level, either can be used as an estimate of lighting level, as long as they are estimated on a spatial scale larger than that of the features. For example, mean lighting level can be computed over P pixels if $P \gg N$, where N is the pixel size of the features.

The second problem to be addressed is the presence of sensor noise. The white, Gaussian sensor noise is assumed to have a constant power of σ_n^2, regardless of lighting level. At high light levels, its power is negligible compared to the image signal and may be ignored. However, when light levels are low, sensor noise significantly alters the expected spatial frequency spectrum of the observed process. The form of the linear smoothing filter which minimizes the noise power with the least impact on the signal is well known Jain [1989]. Its equation is

$$Q(\omega) = \frac{\beta^2 S_{zz}(\omega)^2}{\beta^2 S_{zz}(\omega) + \sigma_n^2} \tag{2.73}$$

This filter is adaptive because it depends on the measured value of β (hence the arrow in Figure 2.20 from the light level estimator to Q). Note that when the luminance value β is high, the smoothing filter Q is approximately one.

2.6 Results

In this section, we present the results of simulating the development of simple cell receptive fields using the learning algorithm presented in Section 2.4 and the response of the trained simple cells to sample images. The images used for training were from a database of face images from the MIT Media Lab used in Turk and Pentland [1991]. The database contained 432 images of the faces of sixteen different individuals taken with all possible combinations of three different head orientations, three viewing distances and three lighting levels. Each image was 128×128 pixels in dimension.

Before discussing the actual results, we first detail the numerical methods used for each step of the simulation. The lighting level normalization discussed in Section 2.5 was performed by the relatively crude procedure of normalizing the luminance variance of the entire image to one. To implement the whitening filter, discrete cosine transform (DCT) symmetric convolution was used because of its computational efficiency and reflective boundary conditions (as opposed to the periodic boundary conditions of the Fourier transform) [Martucci, 1994]. In order to estimate the whitening filter coefficients, all images in the database were cosine transformed, and the root means squared of the transform coefficients was calculated and inverted to form the whitening filter. This procedure is the cosine transform equivalent of estimating the power spectrum. Images were whitened by cosine transforming them, multiplying the image transform coefficients by the corresponding whitening filter coefficient, and inverse transforming. A noise level was estimated and used to form a smoothing filter, which was then combined with the whitening filter. A sample normalized, smoothed and whitened image from the MIT face database is shown in Figure 2.21. Note that in the whitened image, many of the pixel amplitudes are negative, so the luminance values were chosen such that grey represents a pixel amplitude of zero, black represents the most negative amplitude value and white the most positive. This convention is used in all of the figures below.

The model's *a priori* parameters were set as follows: receptive field size (corresponding to the parameter N) was chosen to be an 8×8 square pixel block, and the number of features M was set to 64. The threshold level was adjusted empirically.

Before training, all receptive fields were initialized to random Gaussian synaptic weight values, then the weights were normalized to a total length of one. During training, images

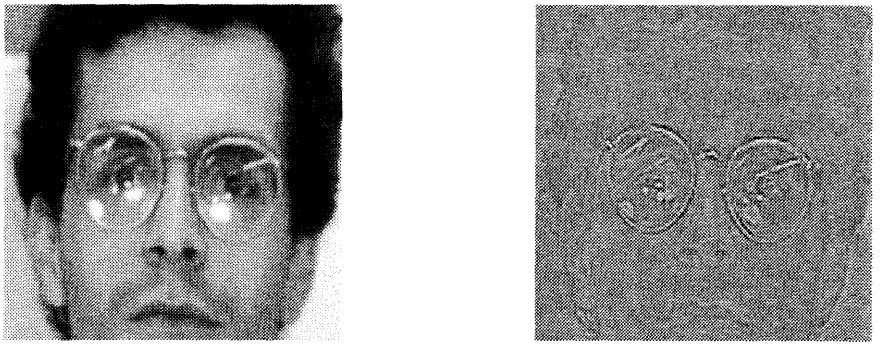


Figure 2.21: *Left:* An image from MIT Media Lab face database *Right:* The normalized, smoothed and whitened image. Because the post-whitened image has negatively valued pixel amplitudes, the luminance values were chosen such that grey represents a pixel amplitude of zero, black represents the most negative amplitude value and white the most positive. This convention is also used in the figures below.

were whitened, then analyzed in 8×8 pixel blocks centered on every pixel location in the image. All pixel blocks were analyzed by the 64 simulated simple cells. Since statistical translation invariance was assumed, it was not necessary to simulate a separate group of simple cells at each pixel. A typical result of training on 250 randomly selected images from the database is shown on the left in Figure 2.22. Training took approximately half an hour running on a 100MHz Pentium processor. The simulated receptive fields have an oriented, bandpass appearance which qualitatively agrees with physiologically observed simple cell receptive fields.

As can be seen from the codebook on the left in Figure 2.22, many of the receptive fields are spatial translates of one another. This translational redundancy is caused by the fact that spatial translates of a particular feature are considered by the model to be distinct features because the competition only occurs at each pixel location. This problem might be solved by extending the competition during learning to neighboring pixels; we believe that this is most likely the case in neurobiology. The problem was addressed in the simulation by shifting the receptive fields during training to maintain their centroids at the center point of the 8×8 pixel block. The result is shown on the right in Figure 2.22. This strategy produced greater variety of receptive fields and especially a greater variance in receptive field spatial frequency.

In order to ensure that these receptive fields were actually from features present in the

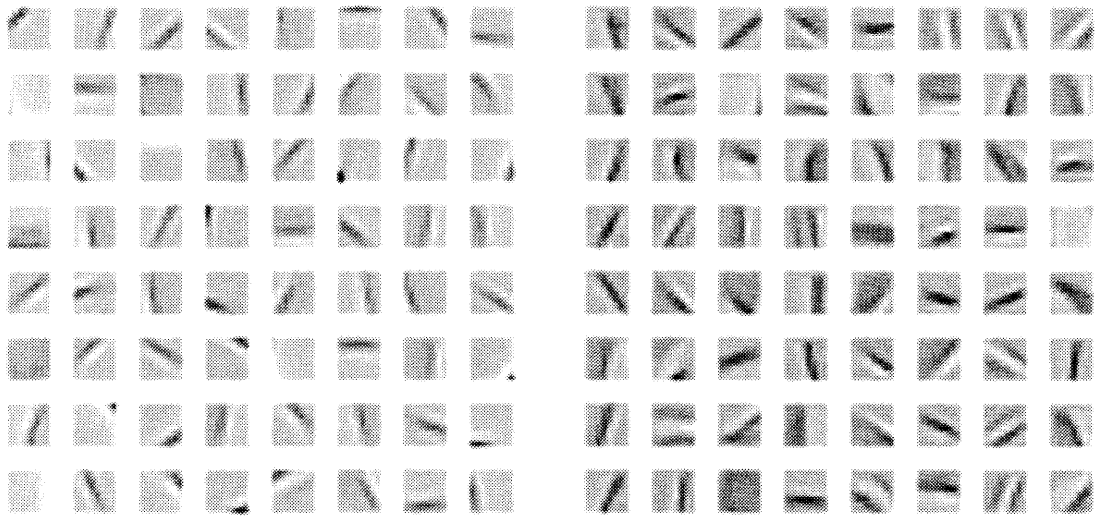


Figure 2.22: *Left:* The receptive fields of 64 simulated simple cells after training on a set of 250 randomly selected images from the MIT face database. Each receptive field has a dimension of 8×8 pixels. The images have been smoothed to eliminate pixelation, and as in Figure 2.21, grey represents zero pixel amplitude. *Right:* A set of 64 receptive fields which were automatically centered in their pixel blocks during training to avoid translational redundancy.

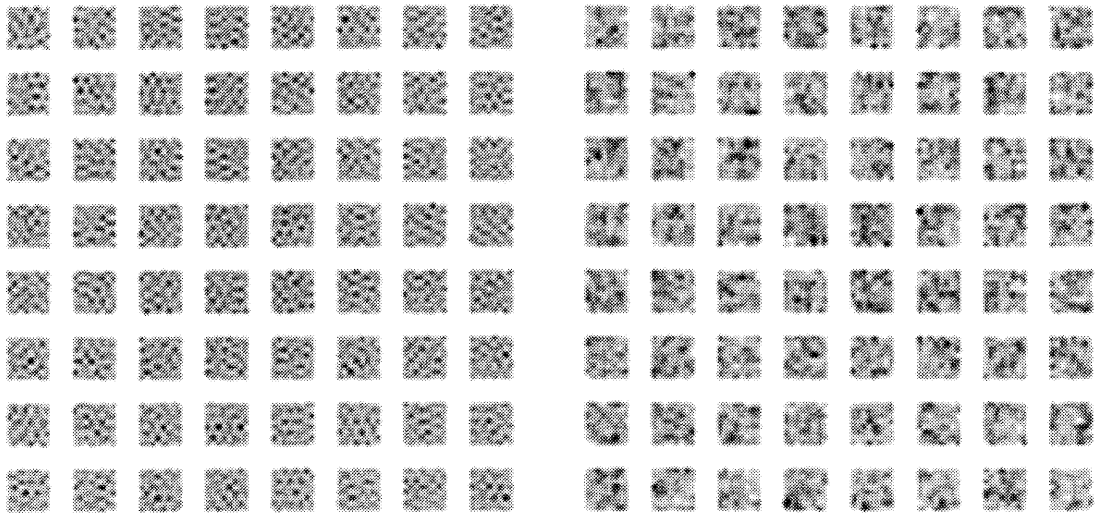


Figure 2.23: Control cases to ensure that the simulated simple cell receptive fields represent features in the data, not artifacts. *Left:* Result of training on images whose pixels were randomly scrambled. *Right:* Result of training on images whose pixels were scrambled, then passed through a $\frac{1}{\omega_x}$ filter so that they have the power spectrum of real images. Neither case produced the type of receptive fields shown in Figure 2.22, as would be expected.

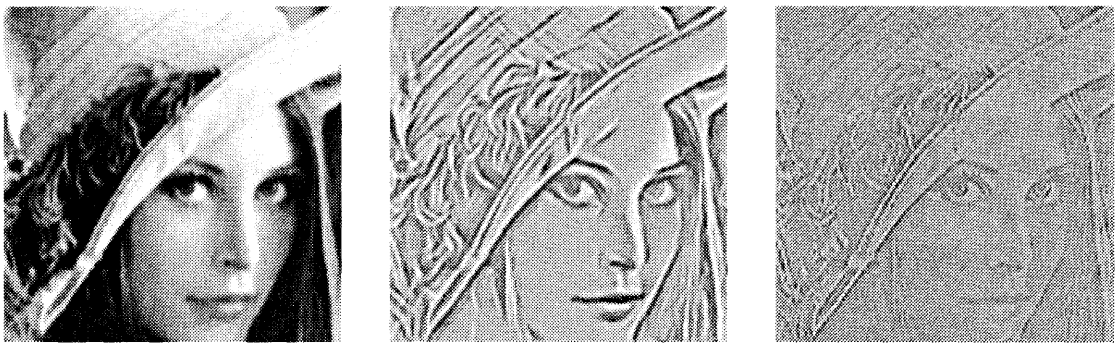


Figure 2.24: *Left:* A 128 × 128 section of the IEEE standard image, "Lena," which was not used in training. *Center:* Result of reconstructing the image from simulated simple cell responses (see text for explanation). *Right:* The normalized, smoothed and whitened image, shown for comparison.

data and not an artifact of the algorithm itself, two control cases were also run. In the first control case, the pixels of each image were scrambled before being passed to the simple cell simulation, which destroyed all but the first order statistics of the data. In the second control, the image pixels were first scrambled, then passed through a filter which gave them the same $\frac{1}{\omega_x}$ power spectrum of real images. Both training sessions produced noisy receptive fields of no discernible pattern, as would be expected; see Figure 2.23.

Since the model is of both development and function, we demonstrate the information filtering properties of the model simple cells by using their responses to reconstruct an image. The reconstruction procedure is performed by first passing the image through the simple cell simulation. At each image pixel location where a simple cell responded, the simple cell's receptive field is superposed at the corresponding pixel location in the reconstructed image. An example of this is shown in Figure 2.24. Note that there are relatively large regions of grey in the reconstructions corresponding to areas of the image to which no model simple cells responded, indicating flat or slowly varying luminance. A large array of examples is also shown in Figure 2.25, including fifteen of the face images from the MIT database and three images which were not used in training.

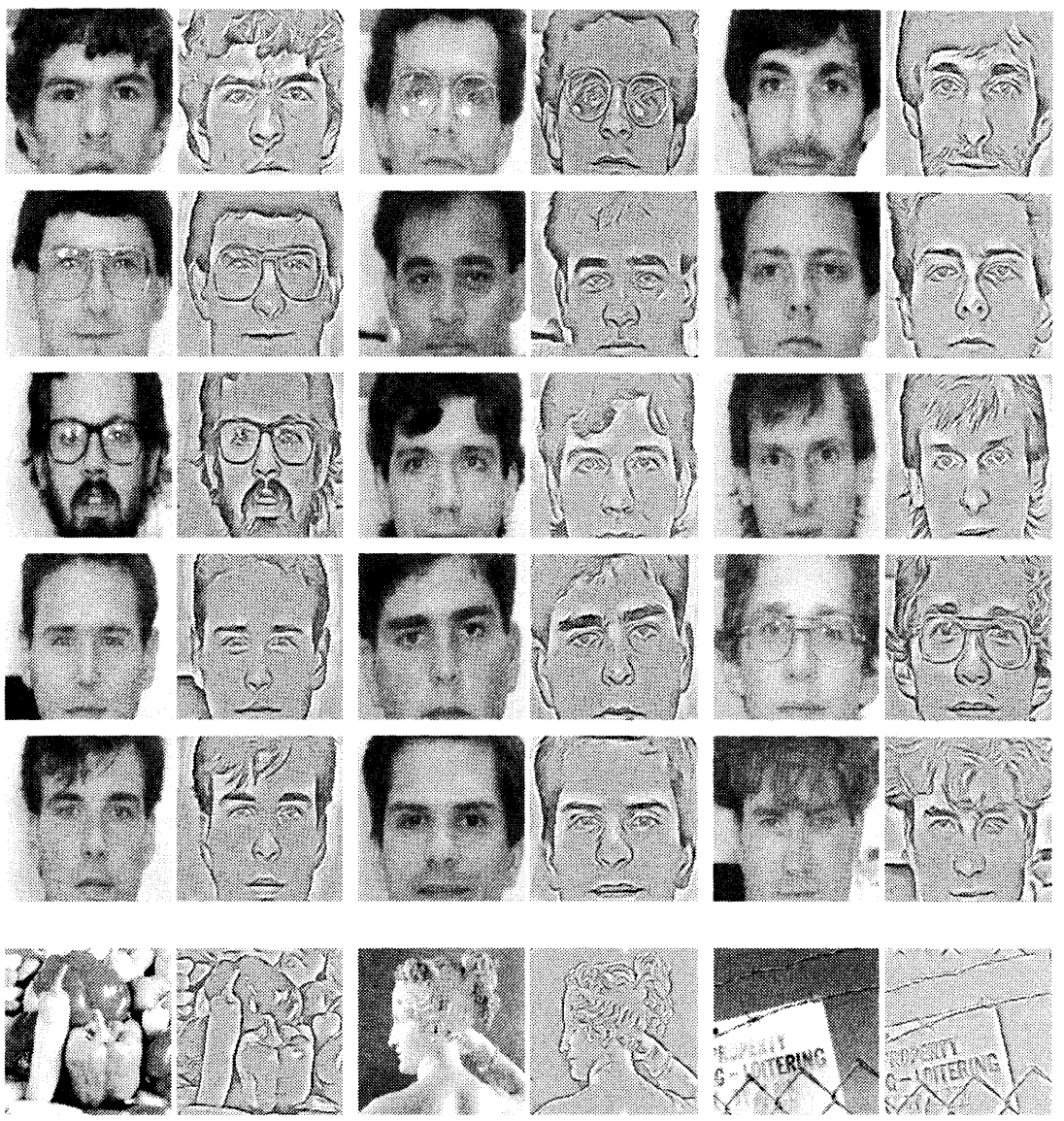


Figure 2.25: Eighteen different images paired with their reconstructions from simulated simple cell responses. The original image is shown on the left of each pair, the reconstruction on the right. The top fifteen images are from the MIT face database used for training; the bottom three images are not from the training set.

2.7 Discussion

In the preceding sections we presented a model for real images, derived mechanisms to analyze images based on this model, and proposed that these mechanisms have realistic neurobiological implementations, including the simple cells of mammalian primary visual cortex. In this section, we further compare our model to that of Field and Olshausen, discuss the mathematical techniques used, and speculate on the further development of this work.

In the model of [Olshausen and Field, 1996] (hereafter referred to as the "sparse coding model"), the objective is to produce a sparse, linear code for images. The winner-take-all mechanism of our features model can also be viewed as a sparse code; at the most, only one simple cell responds at each pixel. However, there is a fundamental difference between the features model and the sparse coding model: the assertion that images are composed of features which have the binary valued property of either being present or absent. The features model therefore uses a nonlinear threshold operation for feature detection. While the sparse coding model uses an nonlinear function to measure sparseness, the code itself is ultimately linear. Without an optimality criterion to guide its selection, the nonlinear measure of sparseness must be arbitrarily chosen. The sparse coding model also required at least a factor of one hundred more CPU cycles to train than the features model and required a complicated gradient descent technique.

The features model can be viewed as a mathematical hybrid of the detection theory technique of matched filtering and a cluster finding algorithm related to k-means clustering or vector quantization. Since none of the assumptions of the model were particularly specific to images, the techniques discussed potentially have wider applicability in self-organizing feature detection.

The prototypical statistical signal model we presented in Section 2.2.1 is also potentially more generally applicable. For example, wavelet or subband signal analysis [Strang and Nguyen, 1996] implies the signal model shown in Figure 2.26. In this case, the Gaussian innovations process of the standard model is augmented with a bank of innovations processes which switch between a white Gaussian source and zero. The innovations filters G_m divide the spectrum into frequency bands. This model asserts that signal energy is localized in both time (or space) by the switching of the innovations processes and frequency (or spatial

frequency) by the bandpass character of the innovations filters. We speculate that by using this model, an optimality criterion could potentially be derived for fitting the innovations filters G_m to real data.

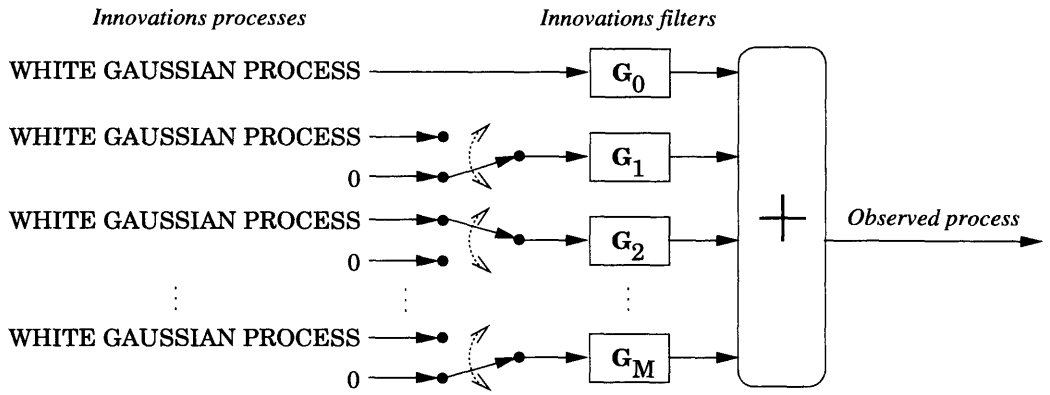


Figure 2.26: The subband statistical signal model. The white, Gaussian innovations process of the standard model is augmented by a bank of innovations processes which switch between a white Gaussian source and zero driving a bank of innovations filters which divide the spectrum into frequency bands.

Among the feature model's three required *a priori* parameters, the threshold value θ is dynamically adjustable. In the derivation of the optimal innovations estimator, the value of θ was chosen so that type I and type II misclassification errors would be equiprobable. However, as discussed in Section 1.2.1, the the two errors may be weighted asymmetrically depending on various circumstances. For example, type I (false alarm) errors may be highly undesirable, in which case the value of θ can be increased. Conversely, if it becomes more important to avoid type II (missed detection) errors, the value of θ can be lowered. Neurobiologically, this would correspond to raising or lowering the current threshold for somatic spiking, which can be effected by neuromodulators.

The most obvious improvement over the techniques employed in implementing the features model is to use a better lighting level normalization algorithm. We simply normalized the variance of total image luminance; a more sophisticated algorithm would estimate lighting level over a smaller spatial scale. As for a more realistic modeling of the neurobiology, a strict winner-take-all mechanism is probably not necessary, just some form of competition among filters.

Cortical processing of visual stimuli continues far beyond the level of simple cells. This

further processing could potentially be explained with the use of a more sophisticated statistical signal model. We speculate that the most significant weakness of the features model is that the innovations processes are assumed to be statistically independent. However, in a real image the presence of a feature at a particular location is predictive of the presence of features at neighboring locations. For example, the probability of two horizontally oriented edge features occurring at horizontally neighboring pixel locations is much higher than their independent probabilities would predict. The next level of processing would therefore involve detectors of combinations of features which we believe could be self-organized using similar methods to those presented for the features model.

Chapter 3 A Model of Spike Rate Coding

3.1 Introduction

This chapter describes a model of the coding of information in the nervous system by the rate of axonal voltage spikes. Assuming an integrate-and-fire mechanism for spike generation, we develop a quantization based model of rate coding and use it to derive the mathematical relationship between the amplitude and temporal resolution of a rate encoded signal. We elaborate the model to include integrator leak in the spike generation mechanism and show that it compactly combines coding and the computation of a threshold function.

3.1.1 Spike rate coding

One of the fundamental roles of a neuron's soma is to transform the analog valued ionic currents resulting from sensory transduction or dendritic computation into a series of voltage spikes suitable for axonal transmission. This conversion is motivated by the fact that digital communication can be conducted very accurately and very fast; if a signal is restricted to having two legal values, a chain of high gain amplifiers can transmit the signal rapidly over a long distance with high fidelity, as discussed in Section 1.2.1. In the axon, these amplifiers are voltage sensitive ion channels.

The relationship between the train of axonal voltage spikes and the somatic current which generated it are a controversial topic in neuroscience. The prevailing theory, first introduced in the 1920's by Edgar Adrian, is the rate coding hypothesis: information is coded as the rate or frequency of spikes [Kandel and Schwartz, 1991]. Recent theories of spike coding have argued against the rate hypothesis [Theunissen and Miller, 1995], but the arguments are vague since there is no mathematically rigorous definition of rate coding. For example, without a definition it is ambiguous over what time scale the spike rate should be estimated and what amplitude resolution the estimate produces. It is the intention of this chapter to propose a solid mathematical framework for a theory of rate coding.

Assuming the soma is well modeled as an *integrate-and-fire circuit* (IFC), we show that rate coding can be analyzed from a traditional electrical engineering perspective by treating

it as a specific strategy of amplitude quantization. Quantizing a signal s can be usefully modeled as adding *quantization noise* to s, and we can quantitatively describe the power spectrum of this noise. [Jayant and Noll, 1984] Decoding a rate code is then equivalent to reducing quantization noise, which can be accomplished using a low pass linear filter with a cutoff frequency at the signal's bandwidth Ω_s. This is equivalent to averaging the spike rate over a time scale of $\frac{\pi}{\Omega_s}$. A second method of reducing quantization noise is to average over the rate encodings of multiple, correlated signals. The two methods are known in a neuroscience context as *temporal integration* and *spatial integration*, respectively, and both can be implemented by the recipient of the spike rate code, the dendrite.[Kandel and Schwartz, 1991]

3.1.2 Chapter outline

The remainder of this chapter is divided into three sections. The first section begins with a review of basic theory of amplitude quantization, then develops a novel analysis of IFC rate coding as amplitude quantization. The power of the quantization formalism is that it places quantitative bounds on the amplitude resolution of rate encoded signals. In the final part of the first section, a model of the soma is presented and compared to the IFC.

One difference between an IFC and a real soma is that there can be significant leak in the somatic integrator. We demonstrate in the second section that an IFC with a leaky integrator can actually be used to simultaneously implement a rate encoder and the threshold function necessary for a matched filter; this economically combines the coding mechanism with a computational primitive discussed in the previous chapter. In the final section, we discuss the implications of our definition of rate coding on the tradeoff between temporal resolution and amplitude resolution, the interpretation of real neurophysiological data and alternative theories of spike coding.

For definitions of terms and notation in this chapter, please refer to Section 1.1.

3.2 The IFC as an amplitude quantizer

The goal of this section is to define the rate code implemented by the integrate-and-fire circuit (IFC) in terms of amplitude quantization. We therefore begin by introducing the idea of amplitude quantization and then develop the theory of IFC rate coding as a form of

amplitude quantization in the three stages. In the first stage, we analyze the ideal uniform quantizer in order to develop necessary techniques and methodology. In the second stage, we introduce the *sigma delta quantizer*, which is an ideal uniform quantizer with preintegration of its input and postdifferentiation of its output. While no direct physical implementation of the sigma delta quantizer exists, it serves as a useful tool for analyzing the IFC, which is defined in the third stage and shown to be functionally equivalent to the sigma delta quantizer. In the last part of this section, a model of the soma is presented and compared to the IFC.

3.2.1 Amplitude quantization

An amplitude quantizer converts an analog valued input signal s into a digital or *quantized* signal q whose amplitude values belong to a countable set of quantized levels. Since the quantized signal q is intended be an approximation of the input signal s, the quantizer selects each individual amplitude value $q(t)$ by choosing the quantized level which is closest to the amplitude value $s(t)$. A block diagram of the amplitude quantizer is shown in Figure 3.1.

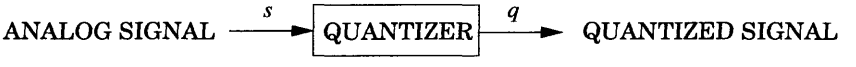


Figure 3.1: Block diagram of the amplitude quantizer. A continuously valued signal s is passed into a quantizer which converts it into a quantized signal q whose amplitudes belong to a countable set of quantized levels.

In the context of the neuron, the motivation for quantizing the dendritic current signal s with the IFC is so that it can be transmitted by the fast, highly reliable mechanisms of the axon. When neural computation is performed in the analog domain, the neuron which receives the quantized transmission must convert it back into analog form of s with as little information loss as possible. Since quantization can be usefully modeled as the addition of an uncorrelated noise source, the conversion of the quantized signal back into an analog signal can be accomplished with a linear filter, a role which can be fulfilled by the dendrite. This is illustrated in Figure 3.2.

The IFC is a member of a class of quantizers known as *uniform quantizers*. In a uniform quantizer, the N quantized levels are sequentially spaced apart by the *step size* parameter Δ. A useful analogy is to consider a ruler to be a uniform quantizer of distances, where the

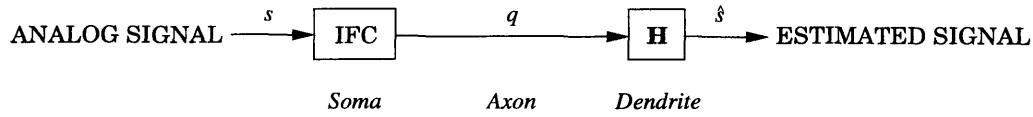


Figure 3.2: Block diagram of the integrate-and-fire circuit (IFC) used in the context of neural transmission. The analog valued dendritic current s is transformed by the IFC, a model of somatic spike rate encoding, into the quantized spike train q. The axon carries the spike train q to its destination, the target cell's dendrite, which is modeled as a linear filter H. The filter H has a low pass response, which serves to convert q into an estimate of s.

step size Δ is the spacing of the ruler's markings. The *ideal uniform quantizer* has an infinite number of quantized levels: all integer multiples of Δ. Denoting the quantizing operation as Q, we can formally write the operation of the ideal uniform quantizer as

$$\mathsf{Q}(x) = \Delta\lfloor\frac{x}{\Delta} + \frac{1}{2}\rfloor \tag{3.1}$$

where $\lfloor\cdot\rfloor$ denotes the greatest integer operation [Jayant and Noll, 1984]. [1] Figure 3.3 illustrates the function $\mathsf{Q}(x)$ and the quantization of a sample waveform s.

Although the theory developed below is for one dimensional, time varying signals, we use the quantization of pixel luminance values in examples because it provides an excellent way to visualize the various effects of quantization. Each of the 512 horizontal scan lines of the image will be treated as if it were a one second long, time varying signal and independently quantized. Since there are 512 pixels in each scan line, the maximum frequency present in the signal is 256Hz. An advantage of using an image is that neighboring scan lines are naturally highly correlated, allowing for the demonstration of spatial integration. The image used throughout is the IEEE standard photograph "peppers," which is shown in Figure 3.4.

3.2.2 The ideal uniform quantizer

In the analysis of the ideal uniform quantizer below, we derive condition under which the nonlinear process of quantization can be successfully approximated as the addition of an independent noise source. This condition is an upper bound on the quantizer step size Δ. The analysis in this section is based on [Lipshitz et al., 1992] and [Gray and Stockham, 1993].

[1] The greatest integer operation $\lfloor x\rfloor$ returns the greatest integer which is less than or equal to x.

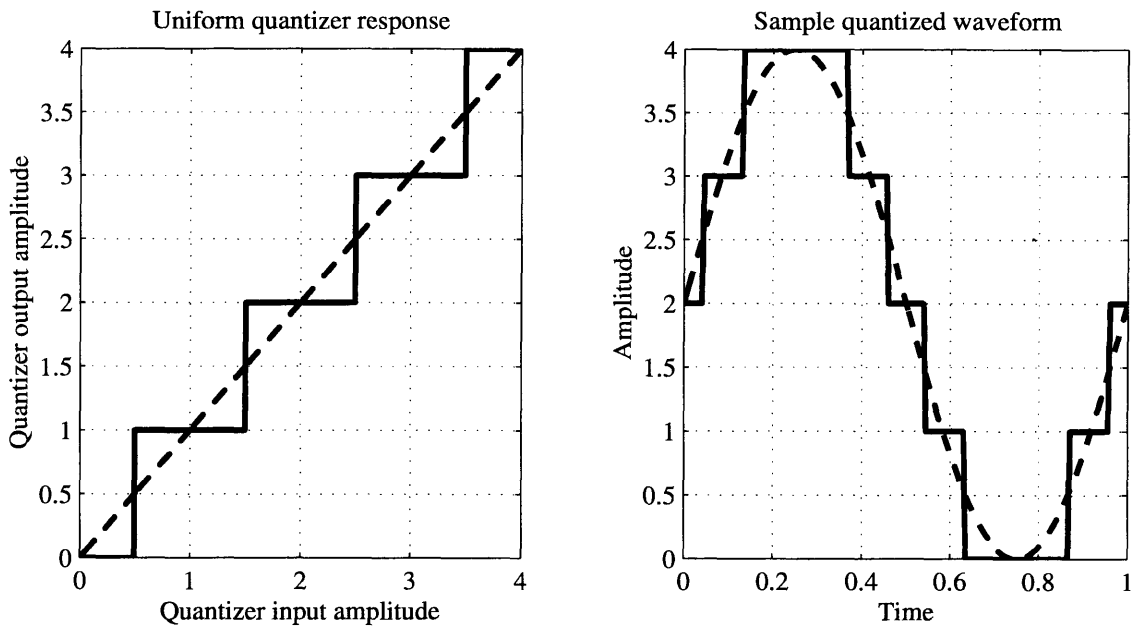


Figure 3.3: *Left:* The output of an ideal quantizer with step size Δ of one (solid line), plotted with the identity function for comparison (dashed line). *Right:* A sample waveform s (dashed line) plotted along with its quantized representation q (solid line).

In order to formally analyze the uniform quantizer, we define the *quantization noise* ε, to be the difference between the quantizer's input and output:

$$\varepsilon = \mathsf{Q}(x) - x \tag{3.2}$$

Note that for the ideal uniform quantizer, the value of ε is constrained to the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2})$. The quantized signal q can be represented as sum the signal s and the quantization noise ε, as illustrated in Figure 3.5.

$$q[k] = s[k] + \varepsilon[k] \tag{3.3}$$

Calling ε quantization *noise* is actually a misnomer at this point since the value of ε is a deterministic function of signal amplitude, as is illustrated in Figure 3.6. A more accurate label for ε would be signal-dependent distortion. Figure 3.7 illustrates the distorting effect of a uniform quantizer with a large Δ applied to the luminance amplitudes of an image.

We show in the following analysis that if the signal is mixed with additive noise before

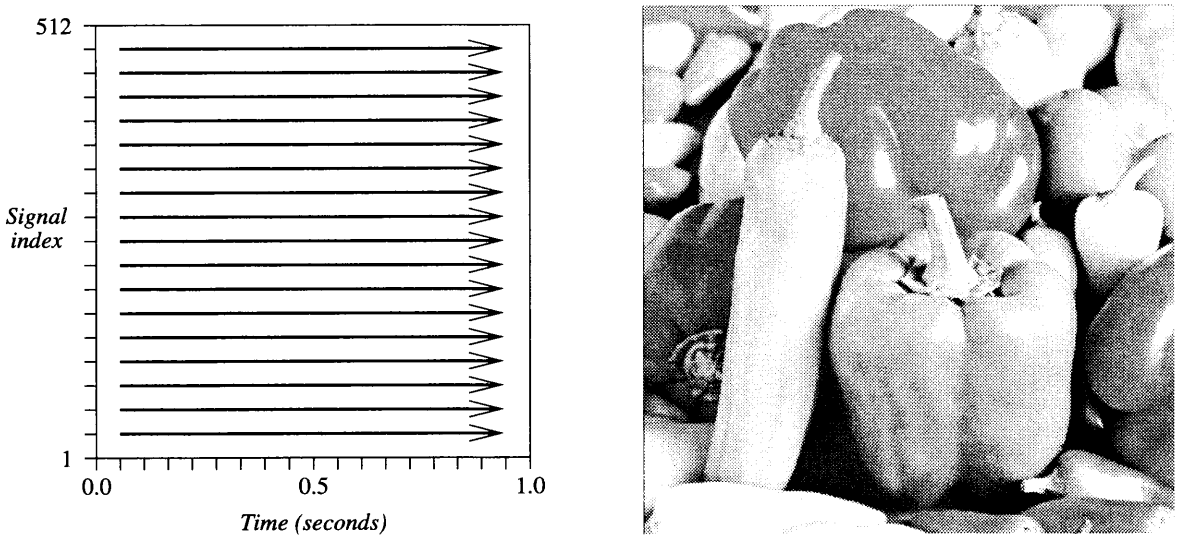


Figure 3.4: *Left:* Various effects of quantization are illustrated below by quantizing the pixel luminance values of an image. Each of the 512 horizontal scan lines of the image will be treated as if it were a one second long, time varying signal and independently quantized. *Right:* The IEEE standard image "peppers" which is the actual image used for the illustrations.

it is quantized, the quantization noise ε can be approximated as a signal-independent noise source. This noise can either be added artificially or, as in the case of the neuron, be inherent background noise in a physical system. The mixture of signal and noise before quantization is illustrated schematically in Figure 3.8.

The added noise is a Gaussian white noise source n with a mean of zero and a variance of σ_n^2. Furthermore, we assume that both the signal s and noise n are bandlimited, and the noise source has the greater of the two bandwidths: $\Omega_n > \Omega_s$. Since neither s and n have signal power over the frequency Ω_n, they can be represented using samples taken at a sampling interval of $\frac{\pi}{\Omega_n}$ seconds.

With the addition of noise, the relationship between s and ε becomes stochastic. The value of ε has a probability distribution, and because each noise sample $n[k]$ is an independent Gaussian distributed random variable, the distribution of $\varepsilon[k]$ is

$$p(\varepsilon[k]) = \sum_{i=-\infty}^{\infty} \frac{1}{\sigma_n\sqrt{2\pi}} \exp \frac{-(i\Delta - s[k])^2}{2\sigma_n^2} \tag{3.4}$$

Note that the distribution of $\varepsilon[k]$ is a function of the signal amplitude $s[k]$. However, the

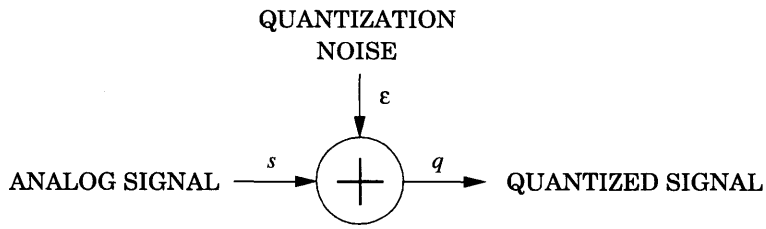


Figure 3.5: Block diagram of the equivalent model of quantization as the addition of quantization noise ε.

greater the amplitude of the noise relative to the quantizer step size Δ, the less influence the actual value of $s[k]$ has on $\varepsilon[k]$. This is illustrated in Figure 3.9. For σ_n which are small with respect to Δ, the value of ε is strongly influenced by $s[k]$. However, when $\sigma_n \geq \frac{\Delta}{2}$, the distribution of $\varepsilon[k]$ is approximately that of a uniformly distributed variable on the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2})$, independent of the value of $s[k]$:

$$\text{For } \sigma_n \geq \frac{\Delta}{2}, \quad p(\varepsilon[k]) \approx \begin{cases} \frac{1}{\Delta} & \text{if } \varepsilon \in [-\frac{\Delta}{2}, \frac{\Delta}{2}] \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

So for $\sigma_n \geq \frac{\Delta}{2}$, we approximate $\varepsilon[k]$ as being statistically independent of $s[k]$, and its label *quantization noise* is then appropriate. If it is possible to subtract the noise source n after quantization:

$$q[k] - n[k] = s[k] + \varepsilon[k] \tag{3.6}$$

the result $s[k] + \varepsilon[k]$ is well approximated as the signal s plus a white noise source uniformly distributed on the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2})$. Adding an artificial noise source, quantizing, then subtracting the noise source is known as *subtractive dither*; an example of its use is shown in Figure 3.10.

However, the scenario in which we are primarily interested is not as straightforward as subtractive dither. If the noise n represents background noise in a physical system, it is not possible to subtract it out. It is therefore important to characterize the sum of the noise n and the quantization noise ε. To do so, we define another variable, the *quantization error* e, to be sum of n and ε:

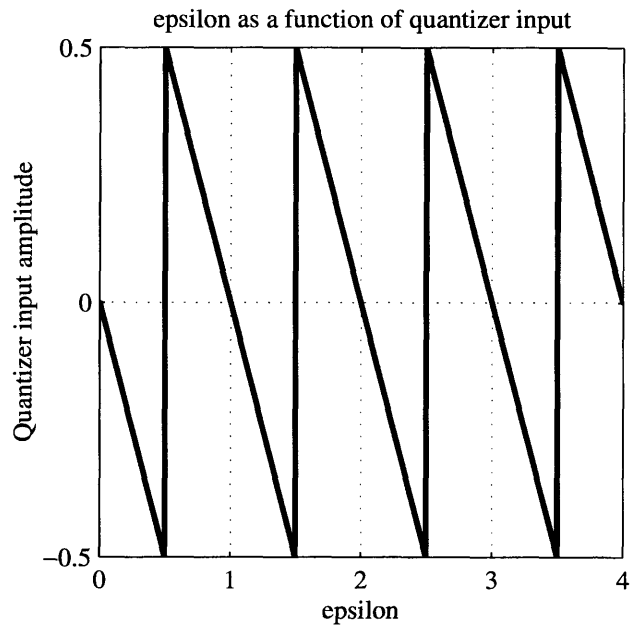


Figure 3.6: Graph of the value of quantization noise ε as a function of quantizer input for an ideal uniform quantizer with step size $\Delta = 1$.

$$e[k] = n[k] + \varepsilon[k] \tag{3.7}$$

The quantized signal q can then be represented as sum the signal s and the quantization error e, as illustrated in Figure 3.11.

$$q[k] = s[k] + e[k] \tag{3.8}$$

For the sake of clarity, a summary of the notation used in this section is presented in the table below:

s	Signal to be quantized
q	Quantizer output
n	Additive white, Gaussian noise source
ε	Quantization noise $\mathsf{Q}(x) - x$
e	Quantization error $n + \varepsilon$
Δ	Uniform quantizer step size
σ_n^2	Variance of noise source n

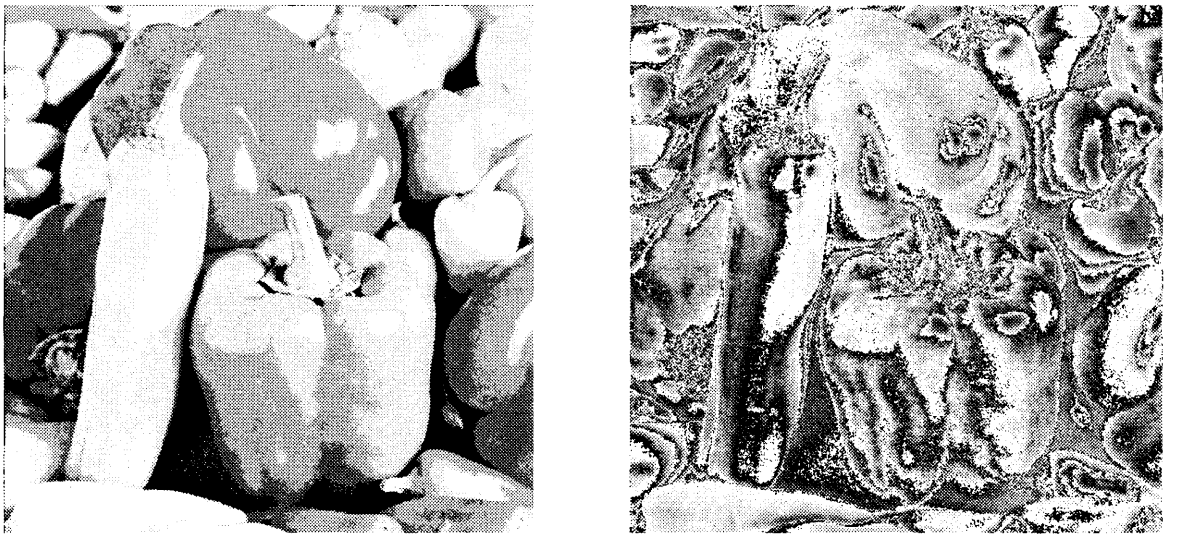


Figure 3.7: *Left:* Result of passing luminance amplitudes of each scan line of the image in Figure 3.4 through a five level uniform quantizer, illustrating the distorting effects of signal-dependent quantization noise ε. *Right:* The quantization noise ε itself, obtained by subtracting the original signal s from its quantized version q.

Although the quantization error e has a complicated relationship with the signal s, we show below that when $\sigma_n \geq \frac{\Delta}{2}$, the first and second-order statistics of the quantization error e are well approximated as being independent of the amplitudes the signal s. To second-order, the combination of the noise source n and the effect of the quantizer can then be approximated as a single, additive noise source.

The mean of $e[k]$ is

$$\Big\langle e[k] \Big\rangle = \sum_{i=-\infty}^{\infty} \frac{1}{\sigma_n \sqrt{2\pi}} \int_{\Delta(i-\frac{1}{2})}^{\Delta(i+\frac{1}{2})} (i\Delta - s[k]) \exp \frac{-(x - s[k])^2}{2\sigma_n^2} dx \tag{3.9}$$

which integrates to

$$\sum_{i=-\infty}^{\infty} \frac{i\Delta - s[k]}{2} \Big(\operatorname{erf} \frac{\Delta(j+\frac{1}{2})}{\sigma\sqrt{2}} - \operatorname{erf} \frac{\Delta(j-\frac{1}{2})}{\sigma\sqrt{2}} \Big) \tag{3.10}$$

Note that since n has a mean of zero, the mean of $e[k]$ is also the mean of $\varepsilon[k]$. As is illustrated in Figure 3.12, for $\sigma_n \geq \frac{\Delta}{2}$, we can make the approximation

$$\text{For } \sigma_n \geq \frac{\Delta}{2}, \quad \Big\langle e[k] \Big\rangle = \Big\langle \varepsilon[k] \Big\rangle \approx 0 \tag{3.11}$$

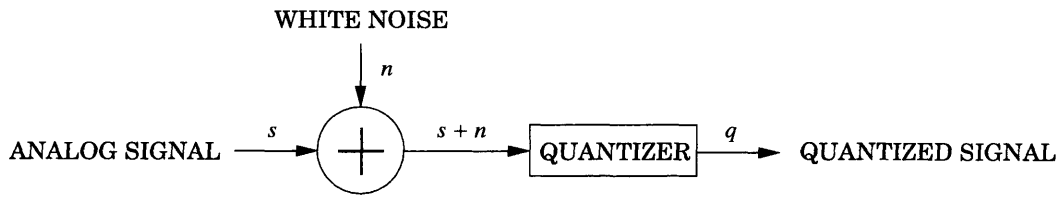


Figure 3.8: Block diagram of the noise added prior to quantization. A continuously valued signal s is summed with a zero mean, Gaussian white noise source before being fed into the quantizer. The noise may be artificially added or inherent background noise in a physical system.

For a complete second-order characterization of e, the autocorrelation $R_{ee}[m] = \langle e[k]e[k+m] \rangle$ must be characterized. For $m = 0$, this is the variance of $e[k]$, which can be expressed in terms of its component correlation terms:

$$\left\langle e[k]^2 \right\rangle = \left\langle \varepsilon[k]^2 \right\rangle + 2\left\langle \varepsilon[k]n[k] \right\rangle + \left\langle n[k]^2 \right\rangle \tag{3.12}$$

Since $\langle n[k]^2 \rangle$ is σ_n^2, and when $\sigma_n \geq \frac{\Delta}{2}$, the quantization noise ε is well approximated by a uniform random variable over an interval of width Δ, so

$$\text{For } \sigma_n \geq \frac{\Delta}{2}, \quad \left\langle \varepsilon[k]^2 \right\rangle \approx \frac{\Delta^2}{12} \tag{3.13}$$

The remaining term $2\langle \varepsilon[k]n[k] \rangle$ has the value

$$2\left\langle \varepsilon[k]n[k] \right\rangle = 2\sum_{i=-\infty}^{\infty} \frac{1}{\sigma_n\sqrt{2\pi}} \int_{\Delta(i-\frac{1}{2})}^{\Delta(i+\frac{1}{2})} (i\Delta - x)(x - s[k]) \exp\frac{-(x-s[k])^2}{2\sigma_n^2} dx \tag{3.14}$$

which can be reduced to

$$2\left(\sum_{i=-\infty}^{\infty} \frac{\Delta\sigma_n}{\sqrt{2\pi}} \exp\frac{-(\Delta(i+\frac{1}{2}) - s[k])^2}{2\sigma_n^2} \right) - \sigma_n^2 \tag{3.15}$$

As is shown in Figure 3.13, when $\sigma_n \geq \frac{\Delta}{2}$, we can make the approximation

$$\text{For } \sigma_n \geq \frac{\Delta}{2}, \quad 2\left\langle \varepsilon[k]n[k] \right\rangle \approx 0 \tag{3.16}$$

Using the two approximations above, the variance of $e[k]$ can be approximated as

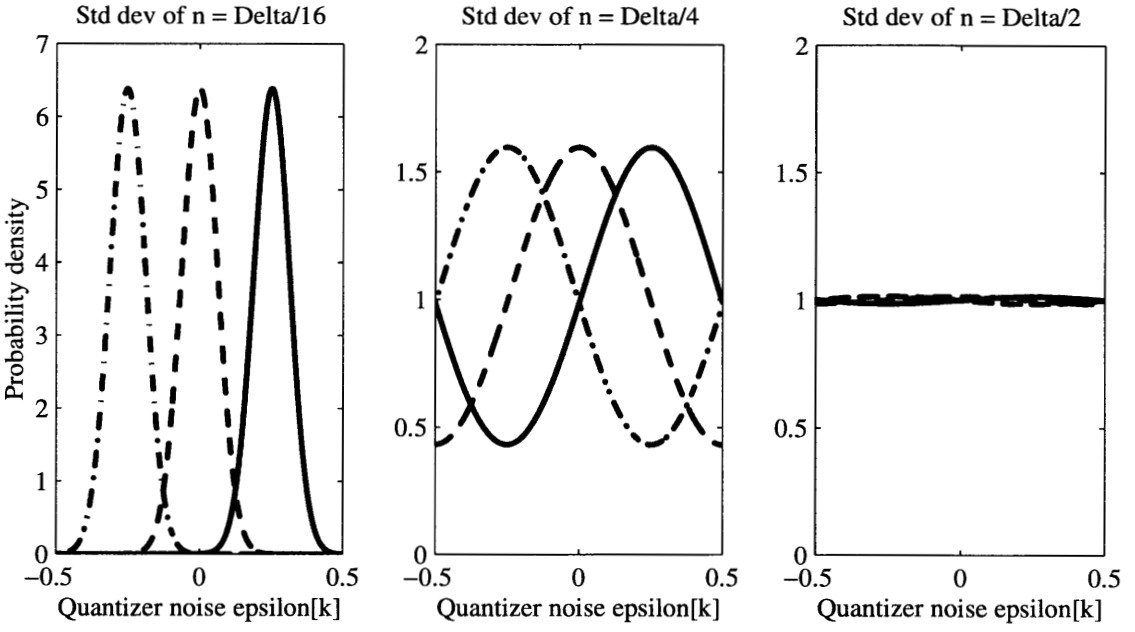


Figure 3.9: The distribution of quantization error $\epsilon[k]$ for a uniform quantizer with step size $\Delta = 1$ with three different standard deviations of the additive Gaussian noise source n: *Left:* $\sigma_n = \frac{\Delta}{16}$, *Middle:* $\sigma_n = \frac{\Delta}{4}$, *Right:* $\sigma_n = \frac{\Delta}{2}$. Each graph displays the distributions for three representative values of the signal amplitude $s[k]$: 0.25 (dash-dotted line), 0.5 (dashed line), and 0.75 (solid line). In the graph on the left, the quantization error is systematically dependent on the signal amplitude, but by the graph on the right, the quantization error is approximately statistically independent of the signal amplitude.

$$\text{For } \sigma_n \geq \frac{\Delta}{2}, \quad \Big\langle e[k]^2 \Big\rangle \approx \frac{\Delta^2}{12} + \sigma_n^2 \tag{3.17}$$

Next we consider the terms of $\langle e[k]e[k+m] \rangle$ for which $m \neq 0$. For $\sigma_n \geq \frac{\Delta}{2}$, the values of both $e[k+m]$ and $e[k]$ are approximately independent of the values of $s[k+m]$ and $s[k]$. Furthermore, different samples of the white noise source n are statistically independent. The samples of e are therefore approximately statistically independent, and so

$$\text{For } \sigma_n \geq \frac{\Delta}{2} \text{ and } m \neq 0, \quad \Big\langle e[k]e[k+m] \Big\rangle = \Big\langle e[k] \Big\rangle \Big\langle e[k+m] \Big\rangle \approx 0 \tag{3.18}$$

The entire autocorrelation of e is approximately

$$\text{For } \sigma_n \geq \frac{\Delta}{2}, \quad R_{ee}[m] = \Big\langle e[k]e[k+m] \Big\rangle \approx \begin{cases} \frac{\Delta^2}{12} + \sigma_n^2 & \text{if } m = 0 \\ 0 & \text{if } m \neq 0 \end{cases} \tag{3.19}$$

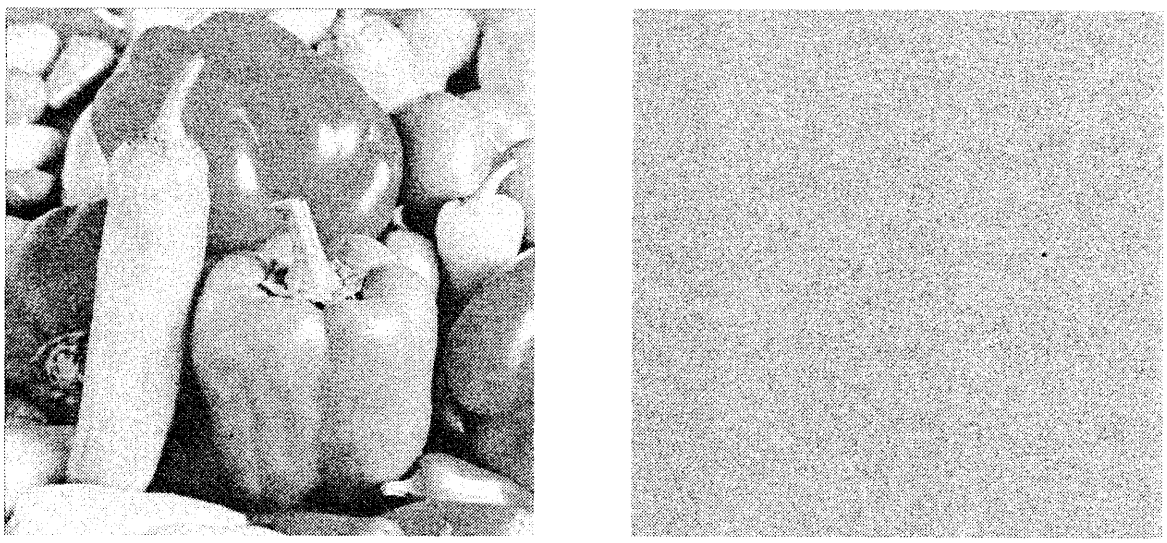


Figure 3.10: *Left:* Result of first adding zero mean, Gaussian white noise n with standard deviation $\frac{\Delta}{2}$ before passing the scan lines of the image through a five level uniform quantizer, then subtracting the noise n so that the only noise remaining is signal-independent quantization noise ε. Quantization noise in this case is well approximated by the addition of a white noise source uniformly distributed on the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2})$. *Right:* The quantization noise ε itself, obtained by subtracting both signal s and noise n from the quantized image q.

To second-order, when $\sigma_n \geq \frac{\Delta}{2}$, the error signal e is a zero mean white process with variance of $\sigma_n^2 + \frac{\Delta^2}{12}$ and a bandwidth of Ω_n. The validity of this approximation is illustrated in Figure 3.14. As discussed at the beginning of this section, the condition for which quantization can be approximated by the addition of an independent noise source is

$$\Delta \leq 2\sigma_n^2 \tag{3.20}$$

which we will refer to as the *uniform quantizer design rule.* The practical utility of this rule was demonstrated empirically in [Lipshitz et al., 1992].

We next discuss an important reason that a linear model of quantization is useful. If Ω_n is significantly larger than Ω_s, a low pass filter can reduce the quantization error. If the quantized signal q is passed through a low pass filter which rejects frequencies over Ω_s, the signal s will be unaffected, but since the power of the quantization noise e is distributed evenly between zero and Ω_n, its power will be reduced by the factor

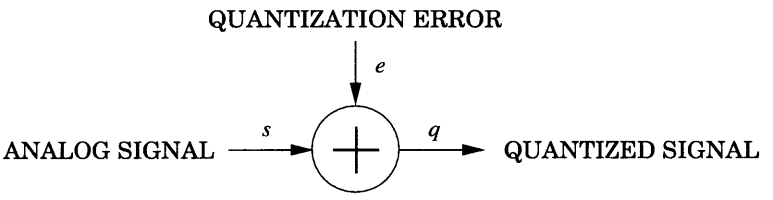


Figure 3.11: Block diagram of the equivalent model of the combination of the additive noise source and quantization as the addition of quantization error e.

$$\frac{\Omega_n}{\Omega_s} \tag{3.21}$$

which is a quantity known as the *temporal oversampling ratio.* To understand why this works, suppose that the temporal oversampling ratio was M. For every sample of s, there would be the equivalent of M samples of q, but each of those M samples contains a different random amplitude of quantization error e. A low pass filter can average those M samples together to reduce the power of the quantization error by a factor of M. If the quantization error were signal-dependent, a linear filter would not be nearly as successful in reducing its effects.

Similarly, suppose the signal s was quantized M times such that each time, the noise source n had a different random value. This number N is known as the *spatial oversampling ratio.* By the same logic as above, each of the N quantized versions of s would have different random quantization errors. Averaging together the N quantized signals would reduce the quantization error by a factor of N. This technique not only works for a single signal quantized multiple times, but also for the quantization of multiple signals which are highly correlated.

In the context of neuroscience, these two processes are known as *temporal integration* and *spatial integration*, respectively. Since they are independent mechanisms, they may be combined into *spatiotemporal integration* which can reduce total quantization noise power by a factor of NM. Spatiotemporal oversampling can be illustrated in Figure 3.15 by taking advantage of the fact that the neighboring scan lines of an image are highly correlated.

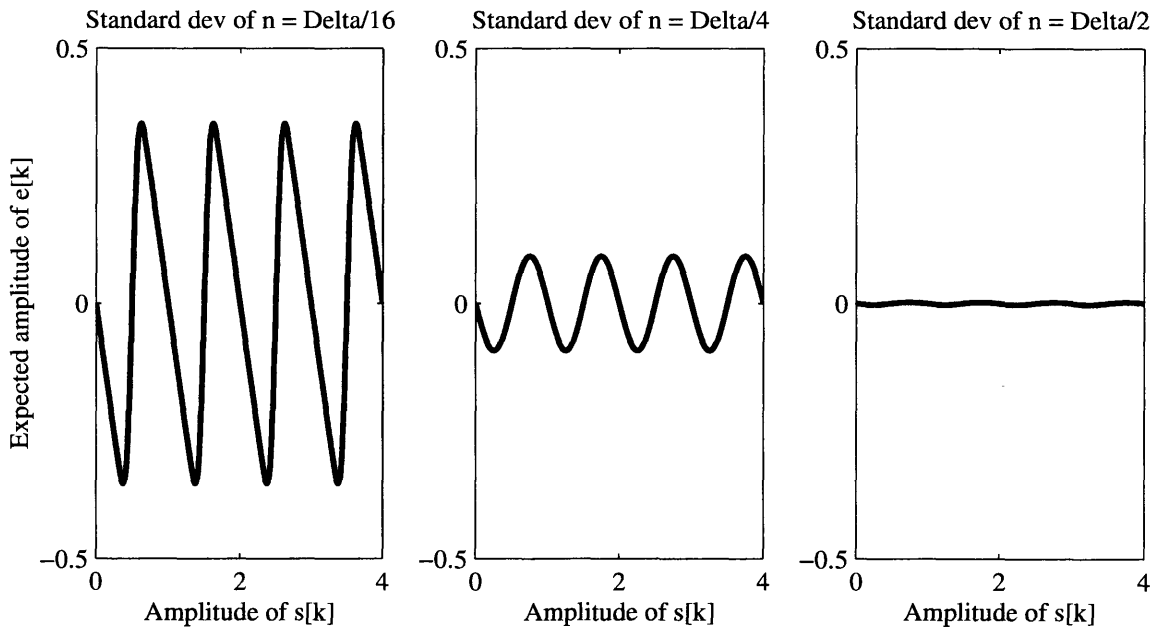


Figure 3.12: Graphs of the mean of quantization error $e[k]$ of an ideal uniform quantizer with step size $\Delta = 1$ as a function of signal amplitude $s[k]$ for three different noise levels: *Left:* $\sigma_n = \frac{\Delta}{16}$, *Middle:* $\sigma_n = \frac{\Delta}{4}$, *Right:* $\sigma_n = \frac{\Delta}{2}$. The graph on the left resembles Figure 3.6, but by the graph on the right, the mean is well approximated as zero.

3.2.3 The sigma delta quantizer

The next step toward a quantization theory of the IFC is a novel analysis of the *sigma delta quantizer* which we have developed. A sigma delta quantizer passes its input through a cascade of an integrator, an ideal uniform quantizer, and a differentiator, as shown in Figure 3.16. Given its use of an ideal integrator and an ideal uniform quantizer, the sigma quantizer is impossible to realize physically and is only useful as mathematical construction for the purpose of analysis. However, we show in the next section that there is a quantizer with a simple physical implementation which is functionally equivalent to the sigma delta quantizer: the integrate and fire circuit.

The sigma delta quantizer is a continuous time, open loop version of the sigma delta modulator, which is a common form of analog-to-digital converter [Candy and Temes, 1992]. The sigma delta modulator has been extensively discussed in the engineering literature, and while the approach taken below is related to the work of [Galton, 1992], the particular derivations, approximations and results presented are new.

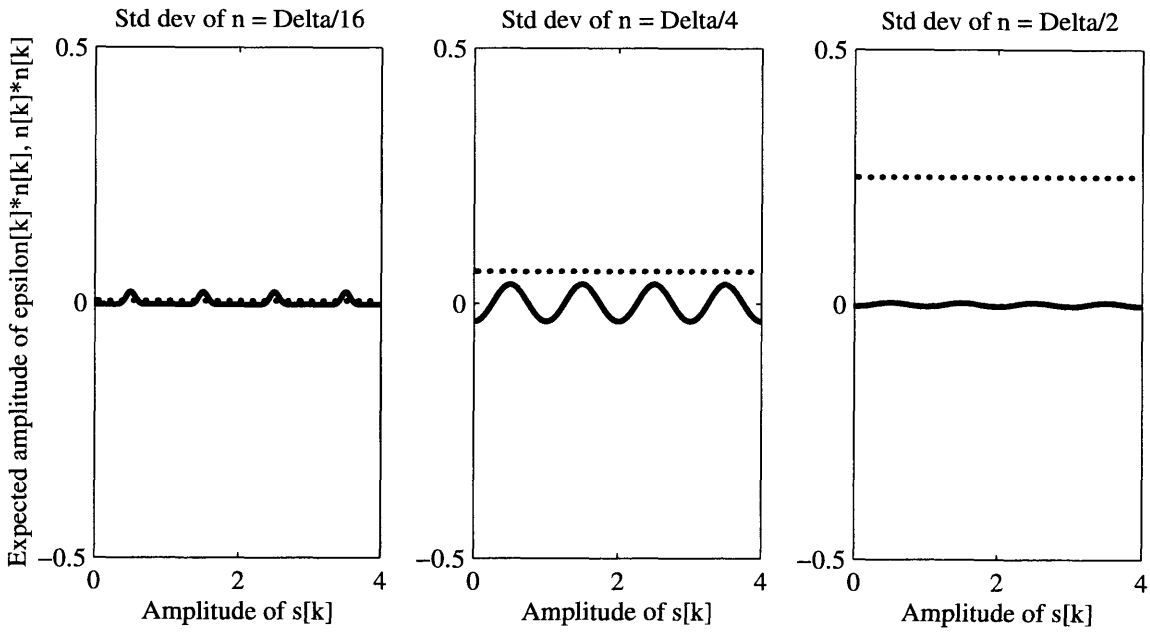


Figure 3.13: Graphs of the correlation of noise $n[k]$ and quantization noise $e[k]$ of an ideal uniform quantizer with step size $\Delta = 1$ as a function of signal amplitude $s[k]$ for three different noise levels: *Left:* $\sigma_n = \frac{\Delta}{16}$, *Middle:* $\sigma_n = \frac{\Delta}{4}$, *Right:* $\sigma_n = \frac{\Delta}{2}$. The solid line represents the correlation value, and the dotted line is drawn at σ_n^2. For the graph on the right, the correlation is well approximated as zero with respect to σ_n^2.

The sigma delta quantizer has the remarkable property that its quantization error is independent of the input signal to second-order for *any nonzero noise power.* However, the power spectrum of the quantization error is dependent on both the signal and the relative scale of σ_n and Δ, and this can be used to derive a criterion for selecting Δ based on σ_n and the bandwidth of the signal s.

In order to label all waveforms in the sigma delta quantizer, we denote the integrated signal and noise as $\bar{s}$ and $\bar{n}$, and the differentiated quantization noise as $\dot{\varepsilon}$:

$$\bar{s}(t) = \int_{-\infty}^{t} s(\tau)d\tau \tag{3.22}$$

$$\bar{n}(t) = \int_{-\infty}^{t} n(\tau)d\tau \tag{3.23}$$

$$\dot{\varepsilon}(t) = \frac{d}{dt}\varepsilon(t) \tag{3.24}$$

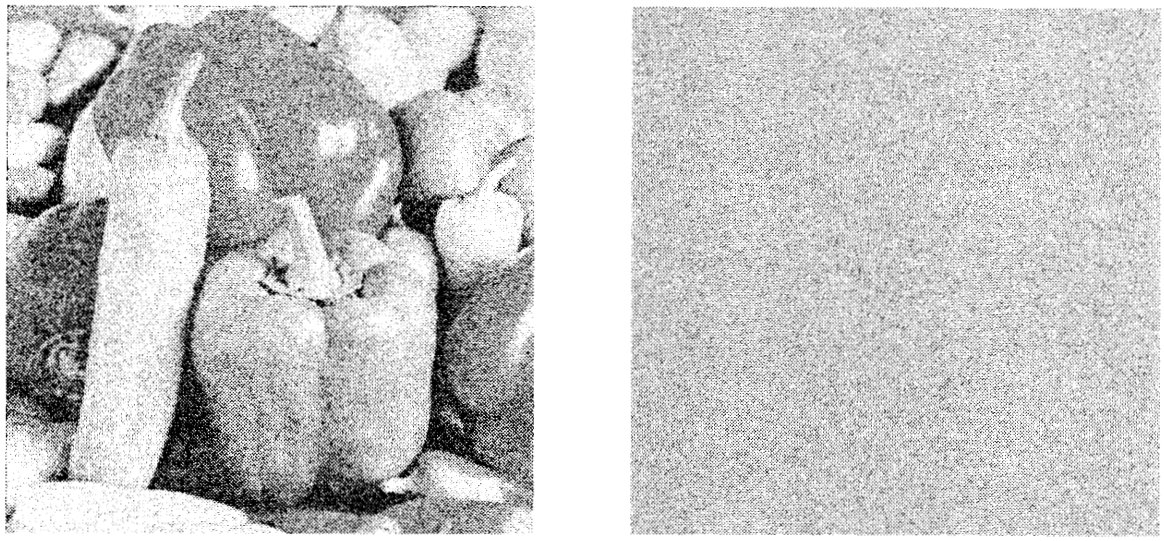


Figure 3.14: *Left:* Result of adding zero mean, Gaussian white noise n with standard deviation $\frac{\Delta}{2}$ before passing the scan lines of the image through a five level uniform quantizer, illustrating the sum of the signal s and quantization error e. While the quantization error has a complicated relationship with the signal, to second-order it is a statistically independent white noise source. *Right:* The quantization error itself, obtained by subtracting the signal s from the quantized signal q.

Figure 3.15: *Left:* Result of passing the quantized image from Figure 3.14 through a spatiotemporal linear filter designed assuming both a temporal oversampling ratio and a spatial oversampling ratio of four. This reduces the power of the quantization error in the resulting image by a total factor of sixteen times. *Right:* Result of passing the quantized image from Figure 3.7 through the same filter, illustrating that signal-dependent quantization error cannot be significantly reduced with linear filtering.

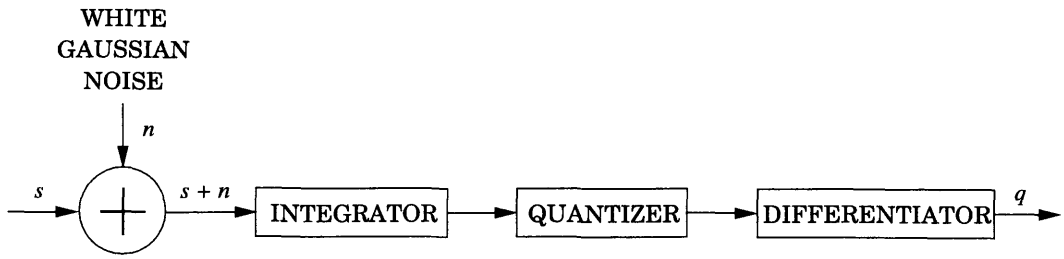


Figure 3.16: Block diagram of the sigma delta quantizer. A signal s plus additive Gaussian white noise n is integrated, quantized with an ideal uniform quantizer, then differentiated.

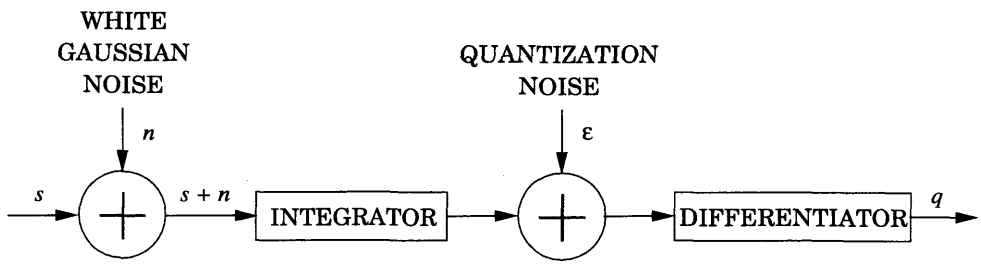


Figure 3.17: Block diagram of the equivalent model of the sigma quantizer with the quantizer replaced with an additive quantizer noise source ε, just as in Figure 3.5.

We first examine the structure of the output of the sigma delta quantizer, q. As the output of the integrator $\bar{s}+\bar{n}$ varies, it crosses the quantized levels of ideal uniform quantizer, producing instantaneous step changes of magnitude Δ in the quantizer's output value. The derivative of these step changes are delta functions scaled by either $+\Delta$ or $-\Delta$. The form of q is therefore

$$q(t) = \Delta \sum_i \delta(t_i) - \Delta \sum_j \delta(t_j) \tag{3.25}$$

where the times $\{t_i\}$ are when the integrator output is increasing across a quantized level, and the times $\{t_j\}$ are when the integrator output is decreasing across a quantized level:

$$\bar{s}(t_i) + \bar{n}(t_i) = \mathsf{Q}(\bar{s}(t_i) + \bar{n}(t_i)) \text{ and } \bar{s}(t_i) + \bar{n}(t_i) > 0 \tag{3.26}$$

$$\bar{s}(t_j) + \bar{n}(t_j) = \mathsf{Q}(\bar{s}(t_j) + \bar{n}(t_j)) \text{ and } \bar{s}(t_j) + \bar{n}(t_j) < 0 \tag{3.27}$$

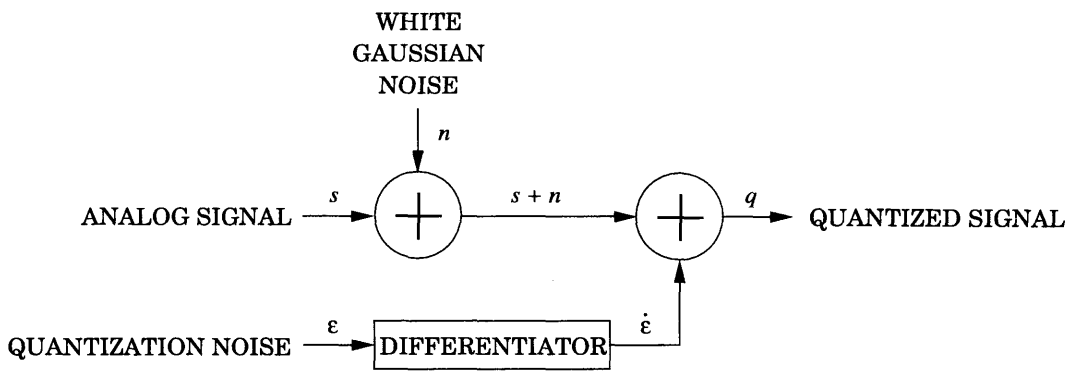


Figure 3.18: An alternative block diagram of the equivalent model shown in Figure 3.17. The quantizer noise ε is differentiated, then added to the signal s plus noise n.

Therefore the sigma delta quantizer output q is a valid quantized signal because it has three discrete levels: zero and the peak of a delta function scaled by either $+\Delta$ or $-\Delta$. If the input to the sigma delta quantizer is restricted to be non-negative, then q will only have two discrete levels and thus be a binary digital signal. We return to this point during the discussion the IFC itself.

Paralleling the analysis of the uniform quantizer, we can make an equivalent model of the sigma delta quantizer as adding quantization noise ε, as shown in Figure 3.17. At the output of this model, the added quantization noise is differentiated and the integration of the signal and noise is canceled by their subsequent differentiation. The output of the sigma delta quantizer q can be written as

$$q(t) = s(t) + n(t) + \dot{\varepsilon}(t) \tag{3.28}$$

This is illustrated schematically in Figure 3.18. As before, we are interested in evaluating the first and second-order statistics of the quantization error e, which for the sigma delta quantizer is

$$e(t) = n(t) + \dot{\varepsilon}(t) \tag{3.29}$$

Since differentiation is a linear operation with a DC response of zero, the mean of $\dot{\varepsilon}(t)$ must be zero:

$$\left\langle \dot{\varepsilon}(t) \right\rangle = 0 \tag{3.30}$$

so that the mean of e is zero:

$$\left\langle e(t) \right\rangle = \left\langle n(t) \right\rangle + \left\langle \dot{\varepsilon}(t) \right\rangle = 0 \tag{3.31}$$

The second-order statistics are more difficult to analyze. They can be conveniently expressed in terms of power spectra as

$$S_{ee}(\omega) = S_{nn}(\omega) + S_{n\dot{\varepsilon}}(\omega) + S_{\dot{\varepsilon}n}(\omega) + S_{\dot{\varepsilon}\dot{\varepsilon}}(\omega) \tag{3.32}$$

because the fact that differentiation is a linear operator can be used to rewrite (3.32) as

$$S_{ee}(\omega) = S_{nn}(\omega) + \omega^2 \Big(S_{\bar{n}\varepsilon}(\omega) + S_{\varepsilon\bar{n}}(\omega) + S_{\varepsilon\varepsilon}(\omega) \Big) \tag{3.33}$$

The second-order statistics of e can be evaluated by determining each of the four terms on the right side of the above equation. To that end, we assume that n is Gaussian white noise with infinite bandwidth. The autocorrelation and power spectrum of n are thus

$$R_{nn}(t) = \sigma_n^2 \delta(t) \tag{3.34}$$

$$S_{nn}(\omega) = \sigma_n^2 \tag{3.35}$$

Next, we must evaluate $S_{\varepsilon\bar{n}}(\omega)$ and $S_{\bar{n}\varepsilon}(\omega)$. In order to do that, the approximations in the previous section can be used to show that $\bar{n}$ and ε are uncorrelated. Those approximations were based on the standard deviation of the noise at the input of an ideal quantizer being at least $\frac{\Delta}{2}$. In the sigma delta quantizer, that noise is $\bar{n}$. The variance of $\bar{n}$ grows monotonically with time since

$$\left\langle \left(\int_{t_1}^{t_2} n(\tau) d\tau \right)^2 \right\rangle = |t_2 - t_1| \sigma_n^2 \tag{3.36}$$

from which can be written

$$\left\langle \left(\bar{n}(t+\tau) - \bar{n}(t)\right)^2 \right\rangle = |\tau|\sigma_n^2 \tag{3.37}$$

Assuming that the integrator value was constant before being switched on at some time t_0, after a transient lasting $\frac{\Delta^2}{4\sigma_n^2}$ seconds, $\sigma_{\bar{n}}$ will be greater than $\frac{\Delta}{2}$ from then on. This represents an important feature of the sigma delta quantizer, because *no matter what the value of* σ_n^2, after an initial transient ε and $\bar{n}$ can be considered uncorrelated:

$$\left\langle \varepsilon(t)\bar{n}(t) \right\rangle \approx 0 \tag{3.38}$$

and thus

$$S_{\bar{n}\varepsilon}(\omega) = S_{\varepsilon\bar{n}}(\omega) = 0 \tag{3.39}$$

The final term to define is $S_{\varepsilon\varepsilon}(\omega)$. Deriving an exact expression for this term is intractable since it has a very complicated dependence on the value of s. However, it is possible to place upper bounds on it as a function of Δ and σ_n.

To do so, we first evaluate $R_{\varepsilon\varepsilon}(0)$, which is the variance of ε. Again, using the approximation based on $\sigma_{\bar{n}} \geq \frac{\Delta}{2}$, from last section ε can be approximated as a random variable which is uniformly distributed over the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2})$. The variance is then

$$R_{\varepsilon\varepsilon}(0) = \left\langle \varepsilon(t)\varepsilon(t) \right\rangle \approx \frac{\Delta^2}{12} \tag{3.40}$$

We next evaluate $R_{\varepsilon\varepsilon}(\tau) = \langle \varepsilon(t)\varepsilon(t+\tau) \rangle$ for $\tau \neq 0$. This will be based on the probability distribution of $\varepsilon(t+\tau)$ given the value of $\varepsilon(t)$, which is equal to

$$p(\varepsilon(t+\tau)|\varepsilon(t)) = \sum_{i=-\infty}^{\infty} \frac{1}{\sigma_n\sqrt{2\pi|\tau|}} \exp \frac{-(i\Delta - u(t))^2}{2|\tau|\sigma_n^2} \tag{3.41}$$

where

$$u(t) = \bar{s}(t+\tau) - \bar{s}(t) + \varepsilon(t) \tag{3.42}$$

This probability distribution parallels (3.4) from the previous section. It is a function of the value of $u(t)$, but analogous to the analysis of (3.4), when

$$\sigma_n\sqrt{|\tau|} \geq \frac{\Delta}{2} \tag{3.43}$$

it is approximately independent of the value of $u(t)$ and therefore the value of $\varepsilon(t)$. Thus for $\tau \geq \frac{\Delta^2}{4\sigma_n^2}$, $\varepsilon(t+\tau)$ and $\varepsilon(t)$ can be approximated as independent random variables. Since they both have zero mean,

$$\text{For } |\tau| \geq \frac{\Delta^2}{4\sigma_n^2}, \quad R_{\varepsilon\varepsilon}(\tau) = \langle \varepsilon(t)\varepsilon(t+\tau)\rangle \approx 0 \tag{3.44}$$

This bound on $R_{\varepsilon\varepsilon}(\tau)$ can be translated into a bound on $S_{\varepsilon\varepsilon}(\omega)$. The power spectrum in terms of the autocorrelation is

$$S_{\varepsilon\varepsilon}(\omega) = \frac{1}{\pi}\int_0^\infty R_{\varepsilon\varepsilon}(\tau)\cos\omega\tau d\tau \tag{3.45}$$

The integration of $R_{\varepsilon\varepsilon}(\tau)$ need only be computed where the autocorrelation is nonzero, which it is only approximately for $\tau < \frac{\Delta^2}{4\sigma_n^2}$:

$$S_{\varepsilon\varepsilon}(\omega) \approx \frac{1}{\pi}\int_0^{\frac{\Delta^2}{4\sigma_n^2}} R_{\varepsilon\varepsilon}(\tau)\cos\omega\tau d\tau \tag{3.46}$$

We can now place an upper bound on this integral:

$$S_{\varepsilon\varepsilon}(\omega) \approx \frac{1}{\pi}\int_0^{\frac{\Delta^2}{4\sigma_n^2}} |R_{\varepsilon\varepsilon}(\tau)||\cos\omega\tau| d\tau \leq \frac{1}{\pi}\int_0^{\frac{\Delta^2}{4\sigma_n^2}} R_{\varepsilon\varepsilon}(0) d\tau \tag{3.47}$$

$$\frac{1}{\pi}\int_0^{\frac{\Delta^2}{4\sigma_n^2}} R_{\varepsilon\varepsilon}(0)d\tau = \left(\frac{\Delta^2}{4\pi\sigma_n^2}\right)\left(\frac{\Delta^2}{12}\right) = \frac{\Delta^4}{48\pi\sigma_n^2} \tag{3.48}$$

Substituting these results into (3.33) for the power spectrum of the quantization error, an upper bound can be set on $S_{ee}(\omega)$:

$$S_{ee}(\omega) \leq \sigma_n^2 + \omega^2\frac{\Delta^4}{48\pi\sigma_n^2} \tag{3.49}$$

which is independent of the signal s. The contributions to quantization error of two terms in the above equation are equal when

$$\omega^2\frac{\Delta^4}{48\pi\sigma_n^2} = \sigma_n^2 \tag{3.50}$$

which can be solved for ω:

$$\omega = \frac{\sigma_n^2}{\Delta^2\sqrt{48\pi}} \tag{3.51}$$

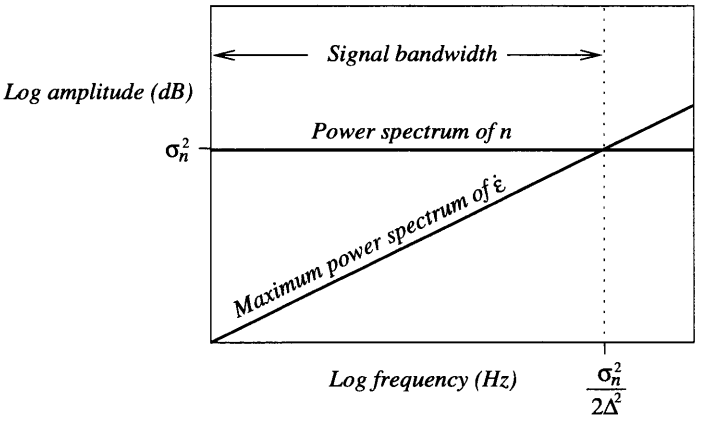


Figure 3.19: A schematic diagram of the power spectra of the noise n and the differentiated quantization noise $\dot{\epsilon}$. The white noise has a flat power spectrum, while the upper bound on the quantization noise power spectrum grows as $\omega^2\frac{\Delta^4}{48\pi\sigma_n^2}$. If the signal is bandlimited below the point where the two power spectra cross, the quantization noise (which is the sum of n and $\dot{\epsilon}$) is guaranteed to be less than twice the power of the noise n.

If the bandwidth of the signal Ω_s lies below this frequency ω, the noise added by the quantization is guaranteed to be at most equal to the power of n. A linear filter H can be used to reject all frequencies above Ω_s without affecting the signal. The step size Δ can be reasonably set to be

$$\Delta = \sigma_n\sqrt{\frac{\sqrt{48\pi}}{\Omega_s}} \tag{3.52}$$

which we will refer to as the *sigma delta quantizer design rule.* Note that the signal bandwidth is specified in radians per second. If the signal bandwidth is expressed in cycles per second (BW_s), this equation is approximately

$$\Delta = \sigma_n\sqrt{\frac{2}{BW_s}} \tag{3.53}$$

It is worth emphasizing that although the quantization noise $\varepsilon(t)$ is independent of both the signal $s(t)$ and noise $n(t)$, *its power spectrum is signal-dependent.* However, the power

spectrum never exceeds the above upper bound, and following the design rule limits the power of ε under the power of the noise n.

The upper bound on $S_{\varepsilon\varepsilon}$ was a based on assuming that the autocorrelation was constant when $|\tau| \leq \frac{\Delta^2}{4\sigma_n^2}$, and zero elsewhere. The actual autocorrelation decays smoothly to zero, and therefore the bound predicts a total autocorrelation power considerably higher than we typically find in simulation; it can overestimate the true value by as much as a factor of ten. However, even so that would only affect the above equation by underestimating Δ by a factor of $\sqrt[4]{10} \approx 1.78$.

3.2.4 The integrate-and-fire circuit

We next describe the IFC itself. The output of the sigma delta quantizer can be expressed as

$$q(t) = s(t) + n(t) + \frac{d}{dt}\varepsilon(t) \tag{3.54}$$

which, with minor manipulation, can be rearranged to

$$\int_{-\infty}^{t} \Big(s(\tau) + n(\tau) - q(\tau)\Big) d\tau = -\varepsilon(t) \tag{3.55}$$

Subtracting q from the integrator input would therefore make the integrator's output equal to $-\varepsilon(t)$, which is bounded on an interval of width Δ. This would mean that the integrator output would only span a range of Δ, and the quantizer need only detect when the integrator's output reached the bounds of this range. If the input signal is restricted to be non-negative, the quantizer would only need to detect when the integrator output reached the upper bound of its range.

This is the idea behind the integrate-and-fire circuit. The IFC integrates its input and passes the result into a *spike generator*. The spike generator produces a delta function (multiplied by the step size) $\Delta\delta(t - t_i)$ at the times t_i when its input reaches the value Δ. This combines the functionality of the ideal uniform quantizer and differentatiator cascade of the sigma delta quantizer. A negative feedback loop subtracts q from the input of the integrator, as described by (3.55). A block diagram of the IFC is shown in Figure 3.20.

Since the IFC is functionally equivalent to the sigma delta quantizer, the equation (3.52) applies for the choice of step size Δ as a function of σ_n and Ω_s. We illustrate this with

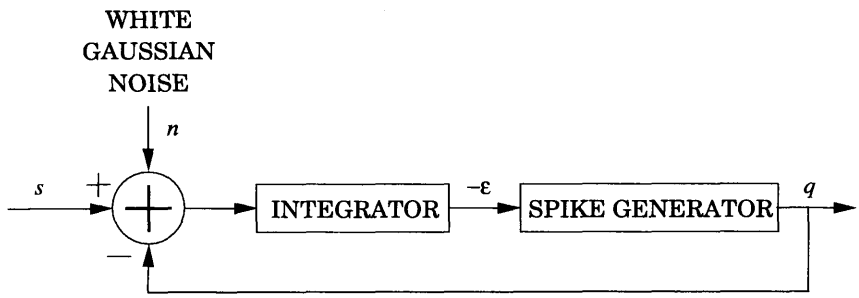


Figure 3.20: Block diagram of the integrate-and-fire circuit.

an example using the familiar "peppers" image. In Figure 3.21, we quantized the image scan lines using an IFC with a step size Δ of one. Since the image has essentially no noise, the upper bound on the power spectrum of $\dot{\varepsilon}$ is very large, allowing for the possibility of a significant signal-dependent quantization error power spectrum. As can be seen from the image, there are striking signal-dependent distortions which appear as vertical periodic banding, indicating large peaks in the power spectrum of the quantization error.

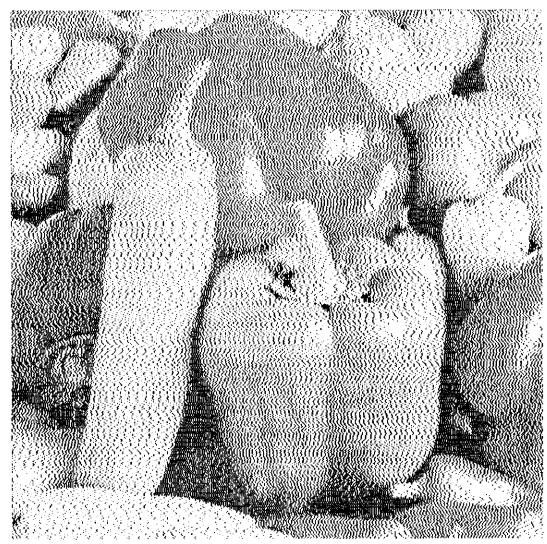


Figure 3.21: Result of passing each horizontal scan line of the "peppers" image through an IFC. The image has virtually no noise, and consequently the IFC encoding generates a high level of signal dependent distortion, seen most evidently in the encoded image's vertical banding patterns.

This distortion can be reduced by following the sigma delta quantizer design rule. With a step size Δ of one, the design rule is satisfied if

$$\sigma_n = \sqrt{\frac{BW_s}{2}} \tag{3.56}$$

For this example, we arbitrarily set the signal bandwidth to be one fourth the maximum frequency of the image scan lines: 64Hz. This fixes σ_n to be $\sqrt{32} \approx 5.66$. When this noise level is added to the image before passing it through the IFC, the result is the image on the left in Figure 3.22. Note that there are no longer any vertical bands, indicating a lack of significant signal dependence of the quantization error power spectrum.

As mentioned in the previous section, the design rule is based on a conservative bound. The step size Δ can often be set $\sqrt[4]{10} \approx 1.78$ lower (or equivalently, σ_n can be set lower by the same factor). To illustrate this, we set σ_n to be $\frac{\sqrt{32}}{\sqrt[4]{10}} \approx 3.18$ to generate the image on the right in Figure 3.22. Lowering σ_n much below this value begins to produce noticeable distortion.

Just as was discussed for the uniform quantizer, spatiotemporal linear filtering can be used to reject quantization error and restore the original signal as best as possible. This is the procedure used by a recipient of an IFC encoded transmission in order to decode the binary valued, quantized representation into the original, analog valued signal. In a neural context, this spatiotemporal filter can be implemented in the dendrite. As we pointed out before, if quantization error is signal dependent, a linear filter will typically have very limited success reducing it. Strictly temporal filtering is illustrated in Figure 3.23 in which each scan line is passed through an IFC, then filtered with a low pass filter with a cutoff frequency at the signal bandwidth. Spatiotemporal filtering is illustrated in Figure 3.24.

A final issue to discuss about the IFC is the bandwidth of the quantized output q. We have so far described q as a series of delta functions, which has infinite bandwidth. However, since the decoder low pass filters q at the signal bandwidth Ω_s, there is no need to preserve an infinite bandwidth in the representation of q. A finite bandwidth pulse shape can be used to signify the spike times of q. The most significant consideration is the reason for generating q in the first place: its binary nature must be preserved for the purpose of fast, accurate transmission. Therefore, the temporal extent of the pulse must be small enough so that successive pulses do not overlap. The minimum interspike interval is set by the length of time it takes the maximum signal level to integrate to Δ, which is just

Figure 3.22: *Left:* Result of first adding zero mean, Gaussian white noise n before passing each horizontal scan line of the image from Figure 3.4 through an IFC using the design rule from the last section. *Right:* Same as the image on the left, but using less noise. See text for details.

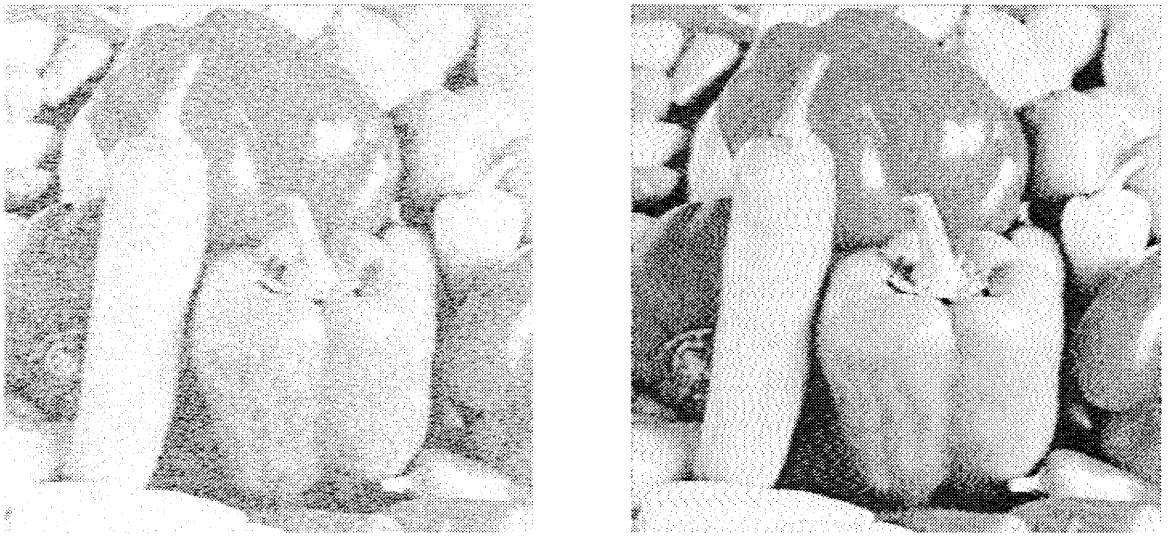


Figure 3.23: *Left:* Result of temporally integrating each horizontal scan line from the IFC encoded image on the right in Figure 3.22 with a low pass filter whose cutoff frequency is at the bandwidth of s, thus reducing the power of quantization error. *Right:* Result of doing the same with the image from Figure 3.21, illustrating that a linear filter provides little reduction for signal dependent quantization error.

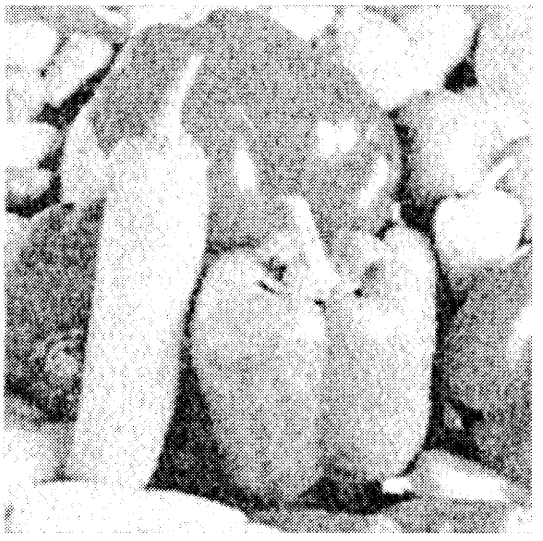


Figure 3.24: Result of spatiotemporally integrating the image IFC encoded image on the left in Figure 3.22, exploiting the correlation redundancy of neighboring scan lines to reduce quantization error even further than in Figure 3.23.

$$\frac{\Delta}{s_{max}} \tag{3.57}$$

where s_{max} is the maximum signal level. This determines the temporal resolution of q:

$$\Omega_q \geq \frac{\pi s_{max}}{\Delta} \tag{3.58}$$

This equation implies that if q were represented as a sampled signal, the sampling interval must be at least the minimum interspike interval. In other words, there should be at most one pulse per sampling interval.

3.2.5 The soma as an IFC

Since our intention is to model the coding mechanism of the neuron soma, we next present an actual model of the soma and compare it to the IFC. A basic circuit model of a neuron's soma is shown in Figure 3.25. The variable current source supplying the input signal $s + n$ represents total dendritic current. The resistor and battery represent the aggregate of all somatic potassium channels; the conductance value g is the total potassium conductance, and E is the potassium reversal potential. The capacitor serves as a model for the soma's

bilipid membrane. The voltage sensitive switch is an idealization of the spike generation zone in the axon hillock. The switch remains open until the voltage v reaches $E + \Delta$, which is the threshold voltage for spike initiation. When the voltage reaches threshold, a momentary closing of the switch sets v from $E + \Delta$ to E. We denote the times that the switch closes as the set $\{t_i\}$. At each closing, an axonal voltage spike is generated, which we represented with a temporally shifted delta function. The output spike train $q(t)$ can therefore be written

$$q(t) = \sum_i \delta(t - t_i) \tag{3.59}$$

This spike generation mechanism is instantaneous and has no refractory period, allowing spikes to occur arbitrarily frequently; in order to make this a more realistic model of the soma, an upper limit could be set on the firing rate. The values of E, C, g and Δ are assumed to be constant for a given soma.

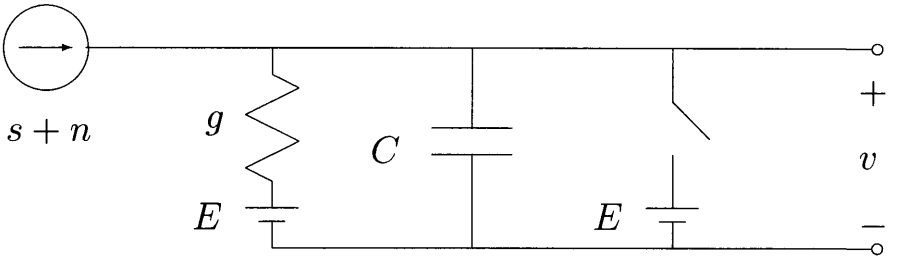


Figure 3.25: The circuit model of the soma. A variable current source provides the input signal $s + n$. When v reaches the threshold voltage $E + \Delta$, the switch closes momentarily setting v back to E and simultaneously, a delta function spike is produced in the output signal q (not shown).

Using $q(t)$, the equation for the above circuit can be written

$$\frac{d}{dt}v(t) = \frac{s(t) + n(t)}{C} - \frac{g}{C}(v(t) - E) - q(t) \tag{3.60}$$

Since E represents a constant offset, it can be set to zero without loss of generality. Since the capacitance of the soma is fixed, we can also consider C to be a scaling constant. Again without loss of generality, we can normalize this constant to one. Therefore, reduced to its essence and solved for $q(t)$, (3.60) becomes

$$q(t) = s(t) + n(t) - \frac{d}{dt}v(t) - gv(t) \tag{3.61}$$

which, with the substitution

$$v(t) = -\varepsilon(t) \tag{3.62}$$

is the same as the IFC, except for the extra term $gv(t)$ representing the leak in the integrator. Therefore, if the leak is not significant, the soma is well modeled as an IFC. The effect of significant leak will be addressed in the next section.

3.3 The effect of integrator leak

As discussed in the previous section, one of the important ways in which the somatic integrator differs from perfect integration by a leak current $gv(t)$. This cannot be directly accomodated by modification of the sigma delta quantizer, nor can we find the same type of closed form expressions as in the analysis of Section 3.2.3.

In order to develop an approximate analysis of integrator leak, we first note that since $v(t)$ is never greater than Δ, the maximum value of the leak current is $g\Delta$. Therefore, if

$$s(t) + n(t) \geq g\Delta \tag{3.63}$$

it is still guaranteed that $v(t)$ will increase and reach the value Δ. However, if this condition is not met, $v(t)$ will not be able to reach Δ and the soma will never spike. Therefore, the leak current $g\Delta$ sets a *threshold* for currents capable of spike generation. This suggests two qualitatively different modes of operation, spiking and non-spiking, depending on whether the IFC input is above or below threshold.

We hypothesize that the threshold can be used for computational purposes such as the matched filter used in feature detection discussed in the previous chapter. In this context, we next discuss the behavior of the leaky IFC for the cases of the signal being entirely below threshold, entirely above threshold, and transitioning from below to above threshold.

3.3.1 Below threshold

If $s(t) + n(t)$ remains below threshold, the system is entirely linear. The integrator output is a low pass filtered version of $s(t) + n(t)$, where the cutoff frequency is determined by g:

$$v(t) = -\varepsilon(t) = \Big(s(t) + n(t)\Big) * e^{-gt}u(t) \tag{3.64}$$

where $u(t)$ is the unit step function. Since the value of $v(t)$ never reaches Δ, there are no spikes and thus

$$q(t) = 0 \tag{3.65}$$

None of the analysis of the sigma delta quantizer holds since the quantization noise $\varepsilon(t)$ is exactly determined by $s(t) + n(t)$. However, since there is no quantizer output, it is more appropriate to consider this in the context of a computationally relevant thresholding mechanism than in the context of coding.

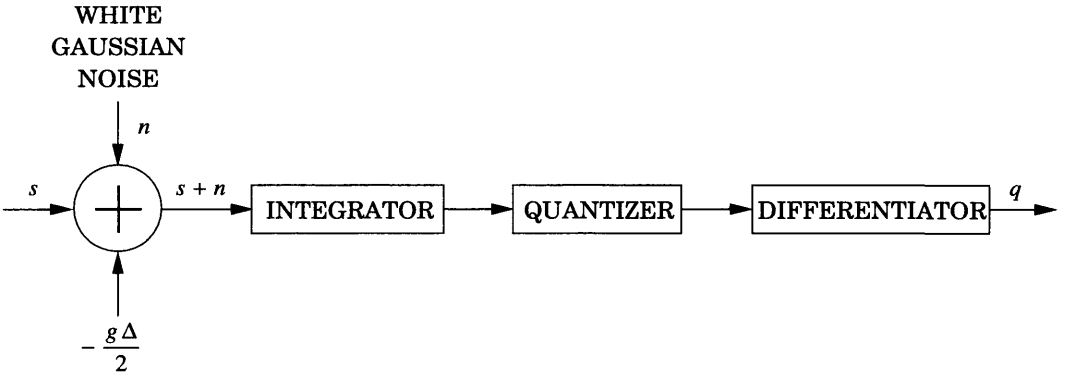


Figure 3.26: Block diagram of the above threshold, leaky integrator SDM approximation as a sigma quantizer with a constant of $\frac{g\Delta}{2}$ subtracted from the input.

3.3.2 Above threshold

If $s(t) + n(t)$ remains consistently above threshold, $v(t)$ will reach Δ and spikes will be generated, and thus the leaky integrator will perform the integration function necessary for the proper operation of an IFC. If it does indeed work as an IFC, then $v(t) = -\varepsilon(t)$ will be well approximated as a uniformly distributed random variable with a mean of $\frac{\Delta}{2}$, and the

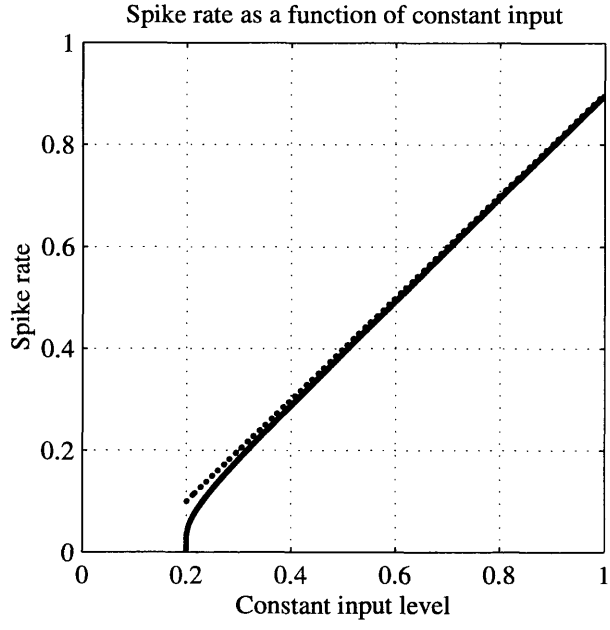


Figure 3.27: A graph of spike rate as a function of the level of a constant input. The relevant parameters are $\Delta = 1, g = 0.2$. The solid line shows the true spike rate for a leaky integrator SDM and the dotted line shows the equivalent model approximation. Soon after threshold, the approximation is very accurate.

following will hold:

$$\left\langle \frac{dv(t)}{dt} \right\rangle = \left\langle s(t) + n(t) - q(t) \right\rangle - \frac{g\Delta}{2} \tag{3.66}$$

We therefore propose that a good approximation for the leaky IFC with above threshold input is a sigma delta quantizer with input

$$s(t) + n(t) - \frac{g\Delta}{2} \tag{3.67}$$

This equivalent model is shown schematically in Figure 3.26. To test the accuracy of the equivalent model, we can use it to predict the leaky IFC response to constant input. From the leaky integrator equation, it can be derived that for a constant input $s = s_{DC}$, the average spike rate is [Scharstein, 1980]

$$\frac{g}{\ln(1 - \frac{g\Delta}{s_{DC}})} \tag{3.68}$$

The equivalent model would predict from (3.67) that the average spike rate should be

$$\frac{s_{DC}}{\Delta} - \frac{g\Delta}{2} \tag{3.69}$$

as long as s_{DC} is over threshold. A comparison of the true leaky IFC spike rate (3.68) and its approximation (3.69) is graphed in Figure 3.27. The equivalent model approximation is accurate soon after the input level exceeds the threshold value.

The equivalent model can also be tested by simulation using a real signal. This is illustrated in Figure 3.28, which shows that qualitatively very similar results are produced by using a leaky IFC and the equivalent model.

3.3.3 Transitioning between above and below threshold

Having characterized the response of the leaky IFC to signals which are always either above or below threshold, we next discuss the response to signals which transition across the threshold. The case of the signal amplitude transitioning from above threshold to below is straightforward: the behavior of the leaky IFC switches from the spiking mode to the non-spiking mode. However, when the signal amplitude transitions from below threshold to above, the situation is more complicated.

When the signal amplitude is below threshold, $\varepsilon(t)$ is correlated with $s(t) + n(t)$, contradicting a central assumption of the analysis of the sigma delta quantizer. However, when the signal amplitude rises above threshold, the assumption of independence between ε, s and n will become valid after a transient period due to the renewed integration of n. The transient period is at most $\frac{\Delta^2}{4\sigma_n^2}$ seconds long, but will typically be shorter since the value of ε is randomized to some degree by n even below threshold. Since ε is also differentiated at the output, the net effect is typically seen as a small, rapid overshoot in average spike rate, as illustrated in Figure 3.29.

If the signal is only above threshold for brief enough periods of time, ε can never statistically decouple from s and n. We believe that this accounts for the results observed in [Mainen and Sejnowski, 1995]. This may also be a crucial mechanism for codes which rely on precise spike timing, such as [Hopfield, 1995].

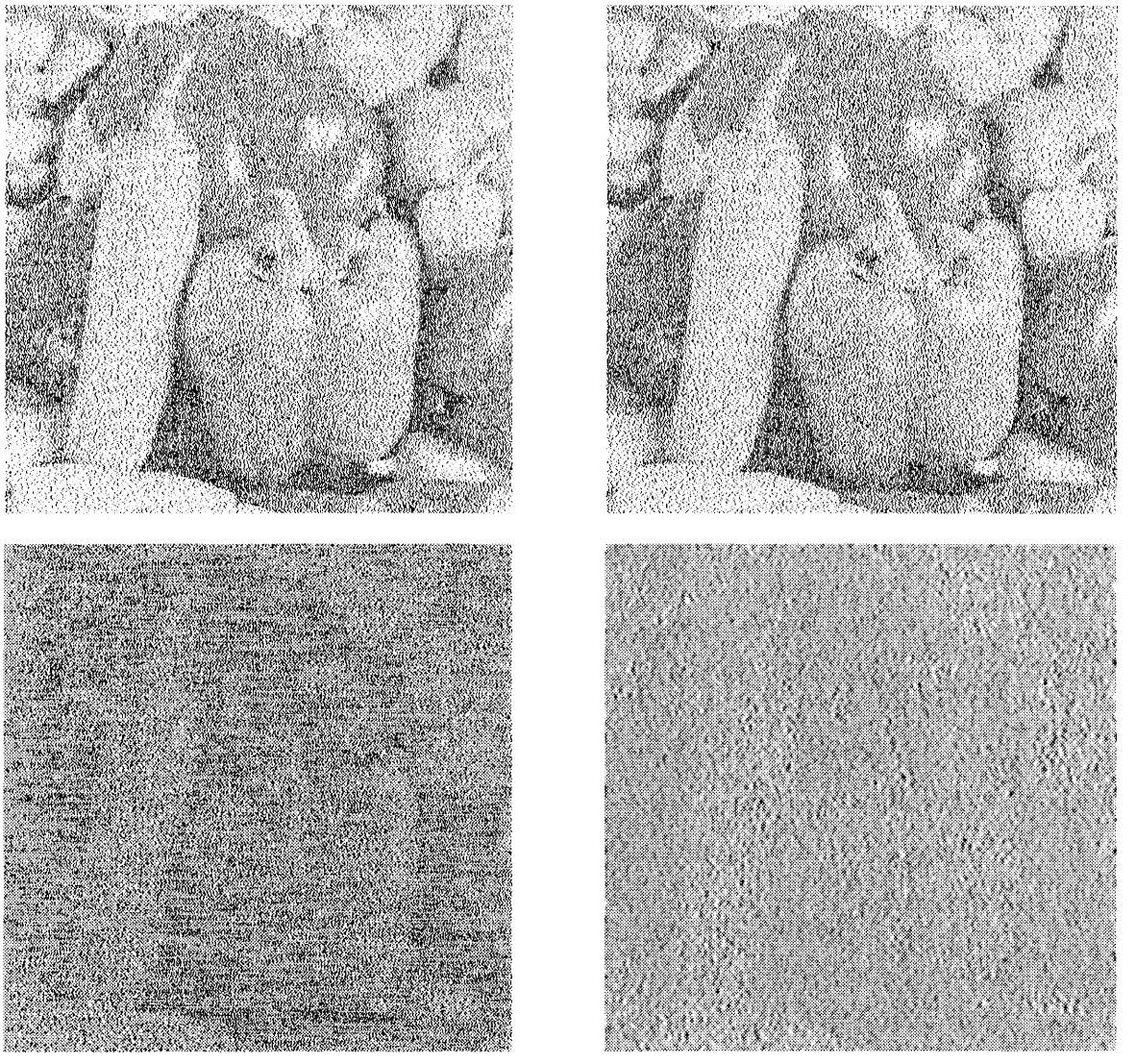


Figure 3.28: *Top left:* Result of using an IFC with a leaky integrator to encode the "peppers" image, which was offset so that all luminance values were above threshold, then the same amount of noise was added as for the image on the right in Figure 3.23. The leak parameter g was 0.2, and the signal amplitudes ranged between 0.2Δ and 0.6Δ. *Top right:* Result of passing the same image through the leak approximation by subtracting 0.1Δ from the luminance values before passing it through a nonleaky IFC. Note that the two images appear qualitatively similar. *Bottom left:* The result of subtracting the two top images to quantify their similarity. The difference can be mostly characterized as noise, but a faint outline of the peppers is visible, representing the error in the approximation of the equivalent model. The difference is most noticeable at the lowest signal levels (for example, the long pepper on the left is the most visible). *Bottom right:* The difference passed through the same spatiotemporal integration filter used in Figure 3.24.

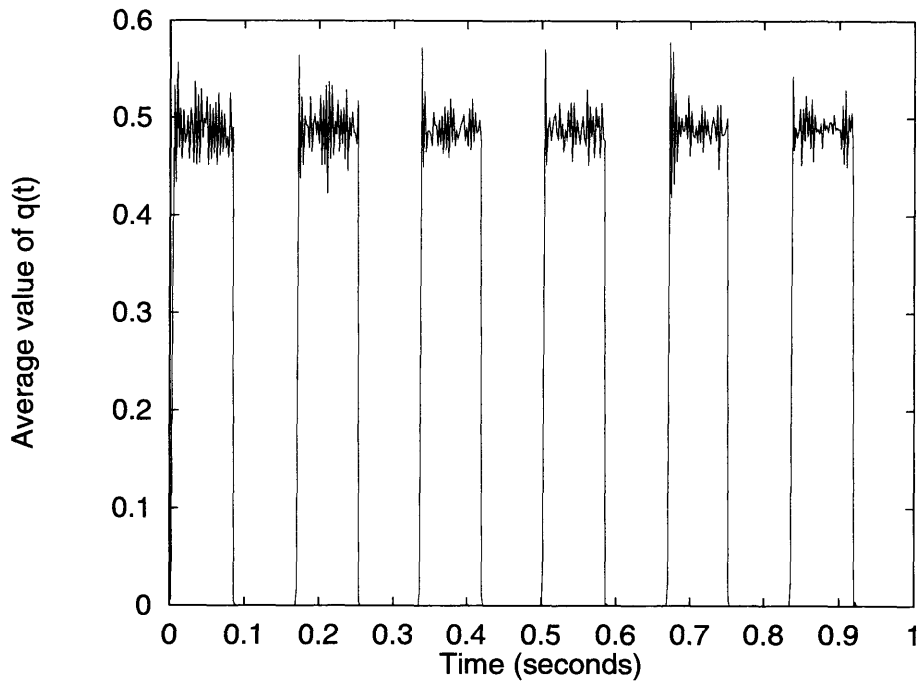


Figure 3.29: Result of averaging repeated trials of using a leaky IFC to encode a 6Hz square wave whose amplitude alternated between zero and three times the threshold value. This illustrates that there is a small, rapid overshoot which occurs when the signal transitions from below to above threshold.

3.4 Discussion

In the preceding sections, we set forth a mathematically explicit definition of IFC rate coding. The conversion of an analog signal into a series of spikes was shown to be usefully modeled as the addition of quantization noise; decoding the spikes into an estimate of the original analog signal can therefore be accomplished with a spatiotemporal linear filter designed to reduce quantization noise. In a neuroscience context, the IFC encoding is performed by a neuron soma, and the decoding is implemented as spatiotemporal integration in a dendrite.

Our sigma delta quantizer design rule quantitatively defined a tradeoff between the amplitude resolution and temporal resolution of a signal to be rate encoded. For an IFC with a given step size Δ, the ratio of the standard deviation of the noise σ_n and the square root of the signal bandwidth Ω_s must remain constant. The standard deviation of the noise is inversely proportional to a measure of amplitude resolution, the signal-to-noise ratio, and bandwidth is a measure of temporal resolution. Therefore, for an IFC the product of am-

plitude resolution and the square root of temporal resolution remains constant. This is analogous to the gain-bandwidth product of a linear amplifier. Assuming that the step size, noise power and signal bandwidth are fixed, the maximum spike rate is proportional to the maximum input signal level. Therefore, amplitude resolution of a rate encoded signal is also linearly proportional to the maximum spike rate.

Since a neuron may receive at least hundreds of inputs, its spatial oversampling ratio is likely to be quite large. Thus even if the amplitude resolution of any particular spike coded input may be very low, spatial averaging over all inputs could produce a result with high amplitude resolution if the inputs are correlated. Since learning is thought to occur by Hebbian mechanisms, it is probably the case that the inputs to a neuron are correlated. We further speculate that due to the metabolic cost of spiking and synaptic transduction, neurons rate code at the minimum necessary amplitude resolution and rely on spatiotemporal integration to produce a decoded signal with the necessary amplitude resolution.

We believe that a common misconception about spike coding is the false dichotomy between a rate code and a timing code. This is due to the lack of understanding that a rate code can define a time scale over which rates are computed. For example, analysis of experimentally derived spike trains such as the well known work of William Bialek [Bialek et al., 1991] are often used to argue against a rate code. However, Bialek's decoding mechanism was a linear filter, exactly as would be optimal from our analysis for a rate code. The results obtained by Bialek in the fly visual system can therefore be interpreted as a rate code with high temporal resolution but very low amplitude resolution.

Coding is also too often treated as a separate issue from computation. If the summation of inputs is an integral part of the computation a neuron is performing, as it is in the simple cell model of the previous chapter, there is no distinction between this computation and decoding a rate code. Furthermore, a leaky IFC simultaneously implements both a computationally relevant threshold and spike rate encoding. In a system as efficiently designed as we believe biology to be, coding and computation should be closely interrelated.

Chapter 4 Conclusion

In the course of this thesis, we have presented two models which seem relatively unrelated, but have some interesting common themes. For example, our model of visual feature detection can be interpreted as the demodulation of digitally modulated analog signals and our model of spike coding can be interpreted as the analog modulation of digital signals.

Another common theme of the two models is that they can both be interpreted in terms of quantization. The visual feature detector quantizes the multidimensional vector representation of an image patch, whereas the spike coder quantizes a time varying analog signal. Quantization involves a combination of linear processing and discretizing nonlinearities such as a threshold or winner-take-all mechanism.

Many of the neural network style approaches to computational problems are based on continuous function approximation with nonlinear basis functions. However, the calculation of the basis function values and the implemention of the necessary parameter fitting algorithms are computationally intricate. Quantization based approaches involve simpler computational elements and training algorithms, both of which are neurophysiologically realistic. While some work has been done, we believe that the application of the ideas and techniques of quantization to neural modeling remains relatively unexplored.

We strongly feel that the paradigm of information processing is crucial to understanding the fundamental principles of the brain, and that significant progress in neuroscience will only result from a well coordinated combination of physiological and theoretical results. We hope that the models presented in this thesis will help toward this goal.

Bibliography

W. Bialek, F. Rieke, R. R. D. Vansteveninck, and D. Warland. Reading a neural code. *Science*, 252(5014):1854–1857, 1991.

J. C. Candy and G. C. Temes. *Oversampling Delta-Sigma Data Converters.* IEEE Press, Piscataway, NJ, 1992.

D. J. Field. What is the goal of sensory coding. *Neural Computation*, 6(4):559–601, 1994.

Ian Galton. *An Analysis of Quantization Noise in $\Delta\Sigma$ Modulation and its Application to Parallel $\Delta\Sigma$ Modulation.* PhD thesis, California Institute of Technology, 1992.

A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic, Boston, 1992.

R. M. Gray and T. G. Stockham. Dithered quantizers. *IEEE Transactions on Information Theory*, 39(3):805–812, 1993.

J. J. Hopfield. Pattern-recognition computation using action-potential timing for stimulus representation. *Nature*, 376(6535):33–36, 1995.

A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

N. S. Jayant and P. Noll. *Digital Coding of Waveforms.* Prentice Hall, Englewood Cliffs, NJ, 1984.

E. R. Kandel and J. H. Schwartz. *Principles of Neural Science.* Elsevier, New York, 3 edition, 1991.

R. Linsker. From basic network principles to neural architecture - emergence of orientation-selective cells .2. *Proc. Natl. Acad. Sci. USA*, 83(21):8390–8394, 1986.

S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy. Quantization and dither - a theoretical survey. *J. Audio Engineering Soc.*, 40(5):355–375, 1992.

Z. F. Mainen and T. J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506, 1995.

S. A. Martucci. Symmetrical convolution and the discrete sine and cosine transforms. *IEEE Transactions on Signal Processing*, 42(5):1038–1051, 1994.

R. N. McDonough and A. Whalen. *Detection of Signals in Noise.* Academic Press, San Diego, 2 edition, 1995.

K. D. Miller. Receptive fields and maps in the visual cortex: Models of ocular dominance and orientation columns. In E. Domany, J. L. van Hemmen, and K. Shulten, editors, *Models of Neural Networks III*, New York, 1995. Springer-Verlag.

K. D. Miller and D. J. C. Mackay. The role of constraints in hebbian learning. *Neural Computation*, 6(1):100–126, 1994.

E. Oja. A simplified neuron model as a principal component analyzer. *J. Mathematical Biol.*, 15(3):267–273, 1982.

B. A. Olshausen and D. J. Field. Sparse coding of natural images produces localized, oriented, bandpass receptive fields. *Nature (in press)*, 1996.

A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

A. V. Oppenheim, A. S. Willsky, and I. T. Young. *Signals and Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1983.

A. Papoulis. *Probability, Random Variables and Stochastic Processes.* McGraw-Hill, New York, 3 edition, 1991.

H. Scharstein. Input-output relationship of the leaky-integrator neuron model. *Journal of Mathematical Biology*, 8:403–420, 1980.

G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesly-Cambridge Press, Wellesly, MA, 1996.

F. Theunissen and J. P. Miller. Temporal encoding in the nervous system: A rigorous definition. *Journal of Computational Neuroscience*, 2:149–162, 1995.

M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neurosci.*, 3(1):71–86, 1991.

B. A. Wandell. *Foundations of Vision.* Sinauer Associates, Sunderland, MA, 1995.