Cortical Mechanisms of Visual Recognition and Learning: A Hierarchical Kalman Filter Model*

Rajesh P. N. Rao
Department of Computer Science
University of Rochester
Rochester, NY 14627-0226
rao@cs.rochester.edu

Abstract

We describe a biologically plausible model of dynamic recognition and learning in the visual cortex based on the statistical theory of Kalman filtering from optimal control theory. The model utilizes a hierarchical network whose successive levels implement Kalman filters operating over successively larger spatial and temporal scales. Each hierarchical level in the network predicts the current visual recognition state at a lower level and adapts its own recognition state using the residual error between the prediction and the actual lower-level state. Simultaneously, the network also learns an internal model of the spatiotemporal dynamics of the input stream by adapting the synaptic weights at each hierarchical level in order to minimize prediction errors. The Kalman filter model respects key neuroanatomical data such as the reciprocity of connections between visual cortical areas, and assigns specific computational roles to the inter-laminar connections known to exist between neurons in the visual cortex. Previous work elucidated the usefulness of this model in explaining neurophysiological phenomena such as endstopping and other related extra-classical receptive field effects. In this paper, in addition to providing a more detailed exposition of the model, we present a variety of experimental results demonstrating the ability of this model to perform robust spatiotemporal segmentation and recognition of objects and image sequences in the presence of varying amounts of occlusion, background clutter, and noise.

Keywords: Recognition, Prediction, Learning, Internal Models, Visual Cortex, Kalman Filtering.

*Submitted for publication, March 1997.

1 Introduction

Vision is fundamentally a dynamic process. The images impinging on the retina are seldom comprised of a sequence of unrelated static signals but rather, reflect measurements of a coherent stream of events occurring in the distal environment. The regularity in the structure of the visual input stream stems primarily from the constraints imposed on visual events by various physical laws of nature in conjunction with the observer's own choices of actions on the immediate environment. Under such a setting, the goal of a visual system becomes one of estimating (and predicting) the "hidden" internal states of an observed dynamic system, in this case, the visual environment. Accurate estimation of the internal state of the external environment then becomes synonymous with accurate recognition of the input stimuli generated by the environment. More importantly, the ability to estimate current states and predict future states of the environment allows the organism to learn efficient visuomotor control programs and form useful cognitive plans for the immediate and distant future.

Perhaps the best known algorithm for accurate estimation and prediction of the internal state of an observed dynamic system is the Kalman filter [Kalman, 1960; Kalman and Bucy, 1961]. The Kalman filter is a linear dynamical system that attempts to mimic the behavior of an observed natural process. It does so by calculating, at each time instant, an optimal estimate of the current internal state of the observed process. The optimal state estimate maximizes the posterior probability of the state given the observed data (see Section 4). At each time instant, the filter uses its state estimate in conjunction with an internal model of the observed natural process to generate a prediction of the next expected input. Given the next input, the filter computes the difference (or *sensory residual error*) between its prediction and the actual input, and uses this residual to correct its estimate of the state. The new corrected estimate is then used to predict the next state, thereby completing one full cycle of filter operation.

Since its discovery about three decades ago, the Kalman filter has been applied successfully to a wide range of problems in fields as diverse as economics [Athans, 1974] and engineering [Cipra, 1993]. Early applications of the filter were predominantly in aerospace engineering. In 1969, it left an indelible mark on human history when it helped man set foot on the moon: the descent of the Apollo 11 lunar module to the surface of the moon was guided by a 21-state Kalman filter [Cipra, 1993]. Even today, Kalman filters form the heart of inertial navigation systems that safely guide commercial airplanes to their respective destinations.

In this paper, we describe a hierarchical Kalman filter based model of dynamic recognition and visual learning. As described elsewhere [Rao and Ballard, 1996a], the Kalman filter model of recognition can be regarded as a natural generalization of some previous schemes for appearance-based recognition such as principal component analysis (PCA) (cf. the *Eigenface* method of [Turk and Pentland, 1991] and the *Eigenspace* method of [Murase and Nayar, 1995]). It also shares the

favorable properties of some recently proposed learning algorithms [Olshausen and Field, 1996; Bell and Sejnowski, 1996] that have been shown to develop localized receptive fields similar to those of simple cells in the primary visual cortex from natural image inputs. Although Kalman filters have previously been used in computer vision (see, for example, [Blake and Yuille, 1992]), most of these applications have relied on hand-built dynamic models of restricted visual phenomena such as translating contours. The present model differs from these previous approaches by allowing dynamic internal models of arbitrary visual phenomena to be *learned* directly from the spatiotemporal input stream (Section 6). In addition, we show how the standard Kalman filter can be (a) made robust to occlusions, clutter, and noise (Section 7), and (b) extended to a hierarchical framework for modeling the ubiquitous class of phenomena that occur at multiple spatiotemporal scales (Section 8). Numerous examples are given to illustrate the recognition performance of the robust and hierarchical Kalman filter using a variety of objects and image sequences. In Section 9, we describe how a hierarchical Kalman filter may be implemented within the known neuronal circuitry of the visual cortex, and how the various components of the filter fit comfortably within the laminar structure of the cortex. In the final section (Section 10), we enumerate the strengths and weaknesses of the model, discuss a simple extension for making the model invariant to transformations such as translations, rotations, and scale, and briefly speculate on the possibility that a Kalman filter-like estimation algorithm may underlie the computational operation of other cortical areas such parietal, auditory, and motor regions as well.

2 Kalman Filtering: A Simple Example

We begin with an extremely simple example from elementary statistics that succinctly conveys the essence of Kalman filtering. Consider the problem of computing the arithmetic mean of a set of $t-1$ real number inputs, denoted by $I(1), I(2), \ldots, I(t-1)$. The arithmetic mean is then given by:

$$\widehat{r}(t-1) = \frac{I(1) + I(2) + \ldots + I(t-1)}{t-1} \tag{1}$$

Now suppose that we have been given a new input $I(t)$. Rather than recomputing the sum and dividing it by t:

$$\widehat{r}(t) = \frac{I(1) + I(2) + \ldots + I(t-1) + I(t)}{t} \tag{2}$$

one may use the equivalent "on-line" running average formula or *recursive estimation rule*:

$$\widehat{r}(t) = \widehat{r}(t-1) + \frac{1}{t}(I(t) - \widehat{r}(t-1)) \tag{3}$$

The reader can easily verify, by substituting the value of $\widehat{r}(t-1)$ from Equation 1 in the on-line formula above, that Equation 3 for updating the running average is in fact exactly the same as Equation 2. The advantage of the on-line formula (Equation 3) is that each input is used only

once to update the mean, thereby obviating the need for storage of the inputs $I(1), I(2), \ldots$. Furthermore, the equation only requires one difference, one division, and one addition, rather than an ever-increasing number of addition operations as in Equation 2.

Equation 2 captures the basic characteristics of *Kalman filtering*. It specifies how the previous estimate of the mean $\widehat{r}(t-1)$ is to be updated given a new input $I(t)$. This involves (a) calculating the difference or *sensory residual error* $(I(t)-\widehat{r}(t-1))$ between the new input and the previous estimate of the mean, (b) multiplying the difference by a *gain* $\frac{1}{t}$, which determines the weight accorded to the difference, and (c) adding the weighted residual error to the previous estimate. In other words, the old estimate is corrected by "filtering" the difference between the new sensory input and the old estimate using a gain term, and adding this sensory correction to the old estimate:

$$\text{New Estimate} \quad = \quad \text{Old Estimate} + \text{Gain} \times \text{Sensory Residual Error} \tag{4}$$

This is the canonical form of the Kalman filter. Equation 2 can be re-written in terms of Kalman filter terminology as follows:

$$\widehat{r}(t) \quad = \quad \widehat{r}(t-1) + N(t)(I(t) - \widehat{r}(t-1)) \tag{5}$$

$$N(t) \quad = \quad (1 + N(t-1)^{-1})^{-1} \tag{6}$$

where the initial conditions are given by $\widehat{r}(0) = 0$ and $N(1) = 1$. The gain term $N(t)$ (which is equal to $\frac{1}{t}$ in this case) is related to the *Kalman gain* in Kalman filter theory, and is recursively calculated at each time step using the value from the previous time step. As we shall see in subsequent sections, the Kalman gain is determined by the *covariance* of the process whose mean we have been concerned with in this section and which we estimated to be $\widehat{r}$ using the equations above.

We conclude the example by viewing the on-line Kalman filter formula for the mean (Equation 3) in the following equivalent fashion:

$$\widehat{r}(t) = \frac{t-1}{t}\widehat{r}(t-1) + \frac{1}{t}I(t) \tag{7}$$

Thus, the new estimate is simply a weighted average of the old estimate and the new input $I(t)$. In this simple example of calculating the arithmetic mean, the new inputs receive less and less weight $(\frac{1}{t})$ as we receive more and more inputs, signifying that our estimates $\widehat{r}$ for the mean are getting progressively more accurate. In the general case, however, the degree to which the sensory input influences the Kalman filter estimate is determined by the Kalman gain (Section 5), which, in general, does not necessarily decrease over time. Instead, it is usually an appropriate function of the on-going needs of the task at hand, as we shall see in Section 7.

3 Internal World Models and the Estimation Problem

There is a growing consensus among cognitive neuroscientists that the brain learns and maintains an internal model of the external world [Barlow, 1985; 1994], and that conscious experience involves

an active interaction between external sensory events and this internal modeling process [Picton and Stuss, 1994]. The concept of an internal world model and its relationship to sensory feedback has been a dominant theme in modern cognitive psychology [Neisser, 1967]. Neisser proposed an "analysis-by-synthesis" approach to perception, wherein an internal world model is adapted according to the sensory stimuli received by the perceiver. This idea is reminiscent of Mackay's epistemological automata [MacKay, 1956], which "perceives" by comparing its expectations of sensory inputs with the actual inputs. The evolutionary origins of this ongoing sensory-model comparison process can perhaps be traced back to the simple feedback loops of micro-organisms [Humphrey, 1992], where the "modeling" occurs at the organism's peripheral surface, as opposed to higher mammals, where this modeling presumably occurs at the level of the cerebral cortex.

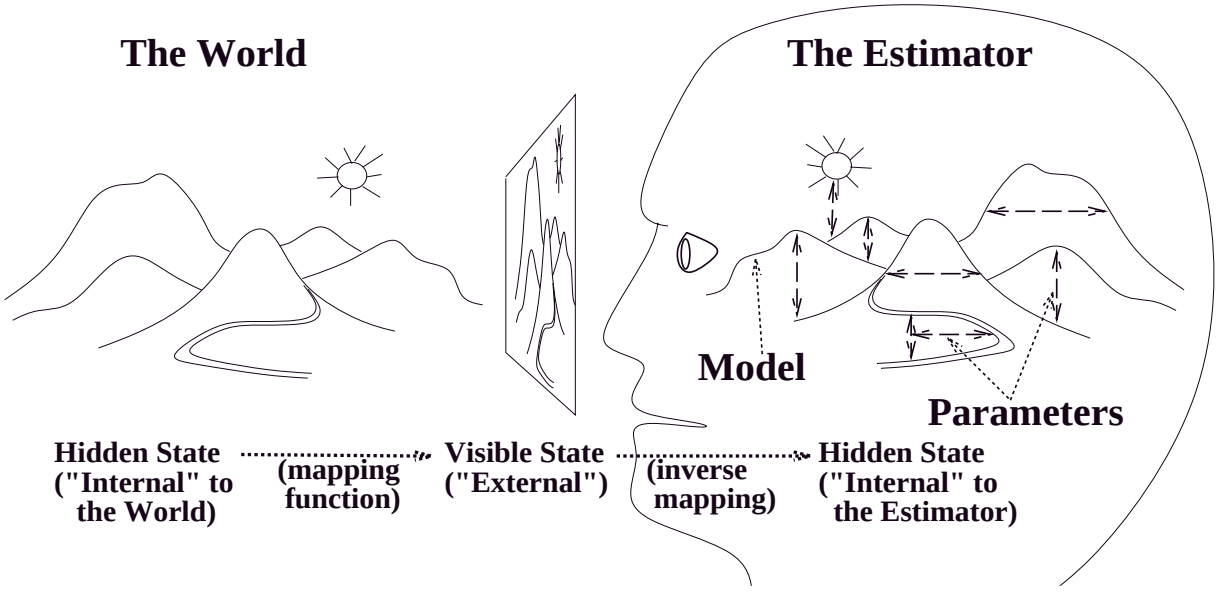
Figure 1 (a) depicts the problem faced by an organism perceiving the external world with the help of an internal model. The organism does not have access to the hidden internal states of the world that are causing its sensory experiences. Instead, it must solve the "inverse" problem of *estimating* these hidden state parameters using only the sensory measurements obtained from its various sensing devices in order to correctly interpret and understand the external world. Note that with respect to the cortex, the definition of an "external world" need not be restricted to sensory modalities such as vision or audition. The cortex may equally well build and use internal models of certain extracortical systems such as the various muscular systems responsible for executing body movements and perhaps even subcortical regions such as those involved in the expression of emotions. For the purposes of this paper, however, we shall be concerned mainly with internal models of the visual environment.

The use of an internal model begets two important questions: (a) what mathematical form does the internal model assume, and (b) how is this internal model learned and used by the organism during perception, action, and cognition? Figure 1 (b) shows the mathematical form of the internal model assumed by the Kalman filter[1]. Briefly, the Kalman filter assumes that a natural process in the external world can be modeled as a stochastic linear dynamical system. In particular, at any time instant t, the internal state of the given natural process is assumed to be characterized by a k-element *state vector* $\mathbf{r}(t)$. Although not directly accessible, this internal state vector is assumed to generate a measurable and observable output $\mathbf{I}(t)$ (for example, an image) according to:

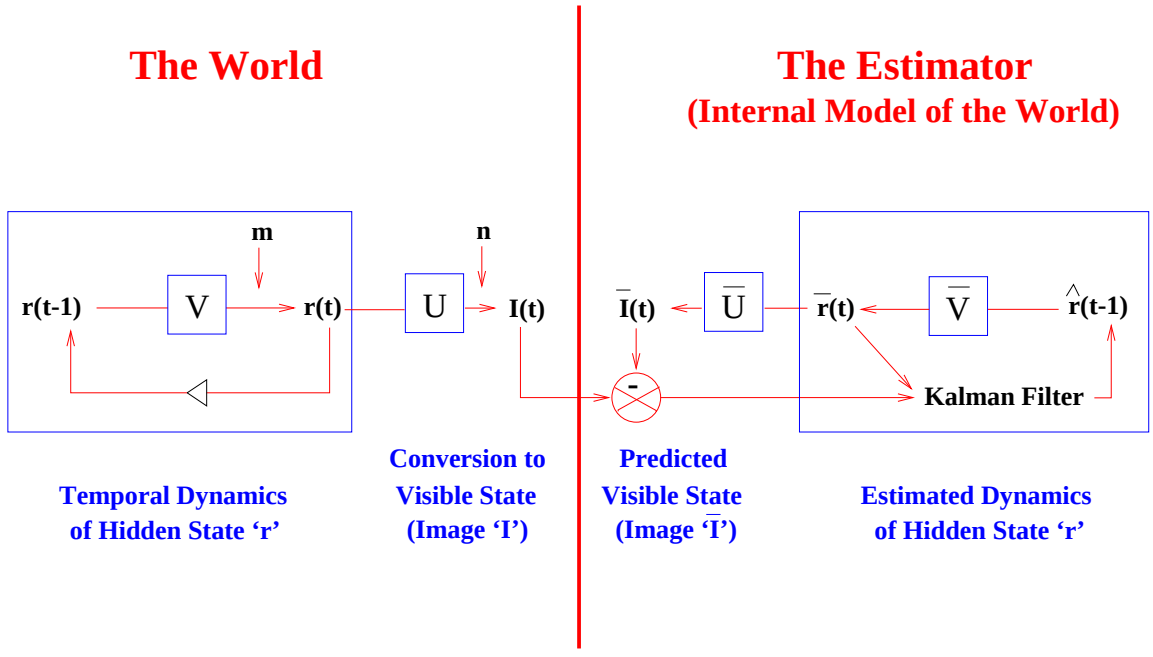
$$\mathbf{I}(t) = U\mathbf{r}(t) + \mathbf{n}(t) \tag{8}$$

where U is a (usually unknown) generative (or measurement) matrix that relates the $k \times 1$ internal state vector $\mathbf{r}(t)$ to the $n \times 1$ observable output vector $\mathbf{I}(t)$, and $\mathbf{n}(t)$ is a Gaussian stochastic noise process with mean zero and a covariance matrix given by $\Sigma = E[\mathbf{n}\mathbf{n}^T]$ (E denotes the expectation

[1] We shall be concerned here with the discrete Kalman filter, although similar results can be obtained for the continuous time version of the filter [Bryson and Ho, 1975].



(a)



(b)

Figure 1: **Internal World Models and the Problem of Hidden State Estimation** (a) conveys the essence of the general problem faced by an organism relying on an internal model of its environment (from [O'Reilly, 1996]). The underlying goal is to optimally estimate, at each time instant, the hidden internal state of the environment given only the sensory measurements **I**. (b) depicts a single-level Kalman filter solution to the estimation problem. The internal model is encoded jointly by the state transition matrix $\overline{V}$ and the generative matrix $\overline{U}$, and the filter uses this internal model to compute optimal estimates $\widehat{\mathbf{r}}$ of the current internal state $\mathbf{r}$ of the environment.

operator and T denotes transpose). Note that this is a sufficient description of $\mathbf{n}$ since a Gaussian distribution is completely characterized by its mean and covariance.

Equation 8 allows the modeling of a wide range of natural phenomena. For example, $\mathbf{I}(t)$ could represent a retinal image, generated by a set of physical "causes," as represented by $\mathbf{r}(t)$, inherent in the visual environment. These physical causes can, for instance, be related to various intrinsic attributes of the external stimulus, such as shape, illumination, and texture. The matrix U specifies how these attributes have been transformed to yield the measured image $\mathbf{I}(t)$. Any discrepancy between the actual image $\mathbf{I}(t)$ and the vector $U\mathbf{r}(t)$ is modeled using the stochastic noise vector $\mathbf{n}(t)$.

The second assumption made by the Kalman filter involves the dynamics of the state vector $\mathbf{r}(t)$ i.e. how the internal state of the observed process changes with time t. It is assumed that the transition from the internal state $\mathbf{r}(t-1)$ at time instant $t-1$ to the state $\mathbf{r}(t)$ at the next time instant can be modeled as:

$$\mathbf{r}(t) = V\mathbf{r}(t-1) + \mathbf{m}(t-1) \tag{9}$$

where V is a (usually unknown) *state transition (or prediction) matrix* and $\mathbf{m}$ is a Gaussian noise process with mean $\overline{\mathbf{m}}(t)$ and covariance $\Pi = E[(\mathbf{m} - \overline{\mathbf{m}})(\mathbf{m} - \overline{\mathbf{m}})^T]$. In other words, the matrix V is used to characterize the dynamic behavior of the observed system over the course of time. Any differences between the actual internal state $\mathbf{r}(t)$ and the prediction from the previous time step $V\mathbf{r}(t-1)$ is modeled as the stochastic noise vector $\mathbf{m}(t-1)$.

For example, in an aerospace application, one might use V to represent the available knowledge regarding how a missile or an aircraft is maneuvering over time. More generally, when observing a dynamic physical system, one might specify how parameters such as acceleration and velocity are presumably changing over time using appropriate physical equations of motion. In an economic system, one might use V to encode knowledge regarding the dynamic behavior of a set of parameters related to the stock market over time.

The astute reader has perhaps already inferred that the choice of the matrices U and V depends crucially on the representation $\mathbf{r}$ of the presumed internal state of the modeled process. As mentioned above, traditional applications of the Kalman filter have used anthropomorphic characterizations of natural phenomena, making use of known physical laws of nature to fix a priori the matrices U and V based on a convenient state representation $\mathbf{r}$ (denoting velocity, acceleration, etc.). However, if one were to use the above framework for characterizing arbitrary dynamic phenomena, one has to answer the two related questions: (1) for an arbitrary internal state representation $\mathbf{r}$, how are the corresponding matrices U and V to be estimated? (2) given estimates for matrices U and V, how can one find an estimate of the corresponding state $\mathbf{r}$? The solution, as pursued in the next two sections, is to define an appropriate optimization function and minimize this function to obtain estimates $\widehat{\mathbf{r}}$, $\widehat{U}$, and $\widehat{V}$ of $\mathbf{r}$, U, and V. Note that the estimates $\widehat{U}$, and $\widehat{V}$ together encode an internal model of the

world. This internal model generates momentary state estimates $\widehat{\mathbf{r}}(t)$ denoting interpretations (with respect to the internal model) of observed dynamic phenomena occurring in the external world.

4 An Optimization Function for Estimation

In this section, we describe how the parameters $\mathbf{r}$, U, and V in the Kalman filter can be estimated and learned directly from input data. The goal is to define an appropriate optimization function which can be minimized to yield optimal estimates for $\mathbf{r}$, U, and V. For the present purposes, assume that we know the true values of U and V, and we therefore wish to find, at each time instant, an optimal estimate $\widehat{\mathbf{r}}(t)$ of the current state $\mathbf{r}(t)$ of the observed process using only the measurable inputs $\mathbf{I}(t)$.

Suppose that we have already computed a prediction $\overline{\mathbf{r}}$ of the current state $\mathbf{r}$ based on prior data. In particular, let $\overline{\mathbf{r}}(t)$ be the mean of the current state vector *before* measurement of the input data $\mathbf{I}$ at the current time instant t. The corresponding covariance matrix is given by $E[(\mathbf{r}-\overline{\mathbf{r}})(\mathbf{r}-\overline{\mathbf{r}})^T] = M$. A common optimization function whose minimization yields an estimate for $\mathbf{r}$ is the *least-squares criterion*:

$$J_1 = \sum_{i=1}^{n} \left(\mathbf{I}^i - U^i\mathbf{r}\right)^2 + \sum_{i=1}^{k}(\mathbf{r}^i - \overline{\mathbf{r}}^i)^2 = (\mathbf{I}-U\mathbf{r})^T(\mathbf{I}-U\mathbf{r}) + (\mathbf{r}-\overline{\mathbf{r}})^T(\mathbf{r}-\overline{\mathbf{r}}) \tag{10}$$

where the superscript i denotes the ith element or row of the superscripted vector or matrix. For example, in the case where $\mathbf{I}$ represents an image, the value for $\mathbf{r}$ that minimizes this quadratic function is the value that (a) yields the smallest sum of pixelwise differences (residual errors) between the image $\mathbf{I}$ and its reconstruction $U\mathbf{r}$ obtained using the matrix U, and (b) is also as close as possible to the prediction $\overline{\mathbf{r}}$ computed from prior data.

The quadratic optimization function above is a special case of the more general *weighted least-squares criterion* [Bryson and Ho, 1975]:

$$J = (\mathbf{I}-U\mathbf{r})^T\Sigma^{-1}(\mathbf{I}-U\mathbf{r}) + (\mathbf{r}-\overline{\mathbf{r}})^T M^{-1}(\mathbf{r}-\overline{\mathbf{r}}) \tag{11}$$

This weighted least-squares criterion becomes meaningful when interpreted in terms of the stochastic model described in the previous section. Recall that the measurement equation 8 was characterized in terms of a Gaussian with mean zero and covariance Σ. Also, as given in the previous paragraph, $\mathbf{r}$ follows a Gaussian distribution with mean $\overline{\mathbf{r}}$ and covariance M. Thus, it can be shown that J is simply the sum of the negative log of the (Gaussian) probability of generating the data $\mathbf{I}$ given the state $\mathbf{r}$, and the negative log of the (Gaussian) prior probability of the state $\mathbf{r}$:

$$J = (-\log P(\mathbf{I}|\mathbf{r})) + (-\log P(\mathbf{r})) \tag{12}$$

The first term in the above equation follows from the fact that $P(\mathbf{I}|\mathbf{r}) = P(\mathbf{I},\mathbf{r})/P(\mathbf{r}) = P(\mathbf{n},\mathbf{r})/P(\mathbf{r}) = P(\mathbf{n})$, assuming $P(\mathbf{n},\mathbf{r}) = P(\mathbf{n})P(\mathbf{r})$. Now, note that the *posterior* probability of the state given the the input data is given by (using Bayes theorem):

$$P(\mathbf{r}|\mathbf{I}) = P(\mathbf{I}|\mathbf{r})P(\mathbf{r})/P(\mathbf{I}) \tag{13}$$

By taking the negative log of both sides (and ignoring the term due to $P(\mathbf{I})$ since it is a fixed quantity), we can conclude that minimizing J is exactly the same as maximizing the posterior probability of the state $\mathbf{r}$ given the input data $\mathbf{I}$.

5 The Kalman Filter

The optimization function J formulated in the previous section can be minimized to find the optimal value $\hat{\mathbf{r}}$ of the state $\mathbf{r}$ by setting $\frac{\partial J}{\partial \mathbf{r}} = 0$:

$$-U^T\Sigma^{-1}(\mathbf{I} - U\hat{\mathbf{r}}) + M^{-1}(\hat{\mathbf{r}} - \overline{\mathbf{r}}) = 0 \tag{14}$$

which yields:

$$(U^T\Sigma^{-1}U + M^{-1})\hat{\mathbf{r}} = M^{-1}\overline{\mathbf{r}} + U^T\Sigma^{-1}\mathbf{I} \tag{15}$$

Using the substitution $N(t) = (U^T\Sigma^{-1}U + M^{-1})^{-1}$ and rearranging the terms in the above equation, we obtain the *Kalman filter* equation:

$$\hat{\mathbf{r}}(t) = \overline{\mathbf{r}}(t) + N(t)U^T\Sigma(t)^{-1}(\mathbf{I}(t) - U\overline{\mathbf{r}}(t)) \tag{16}$$

Note the similarity between this equation and the one for computing the running average (Equation 5). Both are of the form:

$$\text{New Estimate} = \text{Old Estimate} + \text{Gain} \times \text{Sensory Residual Error} \tag{17}$$

The gain matrix $K(t) = N(t)U^T\Sigma(t)^{-1}$ is known as the Kalman gain. It determines the weight given to the sensory residual in correcting the old estimate $\overline{\mathbf{r}}$. Note that this gain is determined by the covariances Σ and M, and therefore effectively trades off the prior estimate $\overline{\mathbf{r}}$ against the sensory input $\mathbf{I}$ according to the uncertainties in these two sources. This become clear if one rewrites the Kalman filter equation as:

$$\hat{\mathbf{r}}(t) = N(t)M^{-1}\overline{\mathbf{r}}(t) + N(t)U^T\Sigma(t)^{-1}\mathbf{I}(t) \tag{18}$$

Thus, the Kalman filter estimate $\hat{\mathbf{r}}$ is essentially a weighted average of the prior estimate $\overline{\mathbf{r}}$ and the new sensory input $\mathbf{I}$ (compare with Equation 7).

The Kalman filter estimate $\hat{\mathbf{r}}$ is in fact the *mean* of the Gaussian distribution of the state $\mathbf{r}$ *after* measurement of $\mathbf{I}$ [Bryson and Ho, 1975]. The matrix N, which performs a form of divisive normalization, can likewise be shown to be the corresponding *covariance* matrix. Recall that $\overline{\mathbf{r}}$ and M were the mean and covariance *before* measurement of $\mathbf{I}$. We can now specify how these quantities can be updated over time:

$$\overline{\mathbf{r}}(t) = V\hat{\mathbf{r}}(t-1) + \overline{\mathbf{m}}(t-1) \tag{19}$$

$$M(t) = VN(t-1)V^T + \Pi(t-1) \tag{20}$$

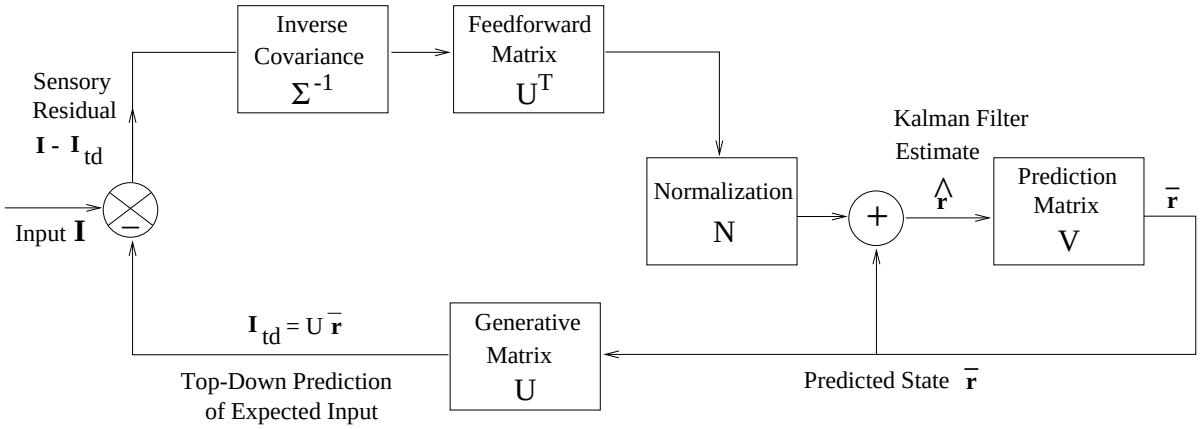


Figure 2: **Schematic Diagram of the Kalman Filter**.

The above equations *propagate* the estimates of the mean and covariance ($\hat{\mathbf{r}}$ and N respectively) forward in time to generate the predictions $\overline{\mathbf{r}}$ and M for the next time instant.

In summary, the Kalman filter predicts one step into the future using Equation 19, obtains the next sensory input $\mathbf{I}(t)$, and then corrects its prediction $\overline{\mathbf{r}}(t)$ using the sensory residual $(\mathbf{I}(t) - U\overline{\mathbf{r}}(t))$ and the Kalman gain $K(t) = N(t)U^T\Sigma^{-1}$. This yields the corrected estimate $\hat{\mathbf{r}}(t)$ for the new mean of the distribution, which is then used to make the next state prediction $\overline{\mathbf{r}}(t+1)$. The covariance matrices corresponding to $\overline{\mathbf{r}}$ and $\hat{\mathbf{r}}(t)$ are updated in an analogous fashion. Figure 3 depicts this evolution of the conditional Gaussian probability density function of the state over time according to the Kalman filter equations.

Before concluding this section, let us briefly return to the running average example from Section 2 to better understand the Kalman filter equations above. In that example, we were concerned with estimating a constant scalar quantity r using an increasing number of measurements I. Thus, the measurement equation can be modeled as:

$$I(t) = r(t) + n(t) \tag{21}$$

where n has mean zero and a variance $\Sigma = 1$. Note that in this case, $U = 1$. The quantity r remains the same over time, so the dynamics can be modeled simply as:

$$r(t) = r(t-1) \tag{22}$$

Note that in this case, $V = 1$ and $\Pi = 0$. Thus, the Kalman filter update equations are given by:

$$\hat{r}(t) = \overline{r}(t) + N(t)(I(t) - \overline{r}(t)) \tag{23}$$

$$\overline{r}(t) = \hat{r}(t-1) \tag{24}$$

$$N(t) = (1 + M(t)^{-1})^{-1} \tag{25}$$

$$M(t) = N(t-1) \tag{26}$$

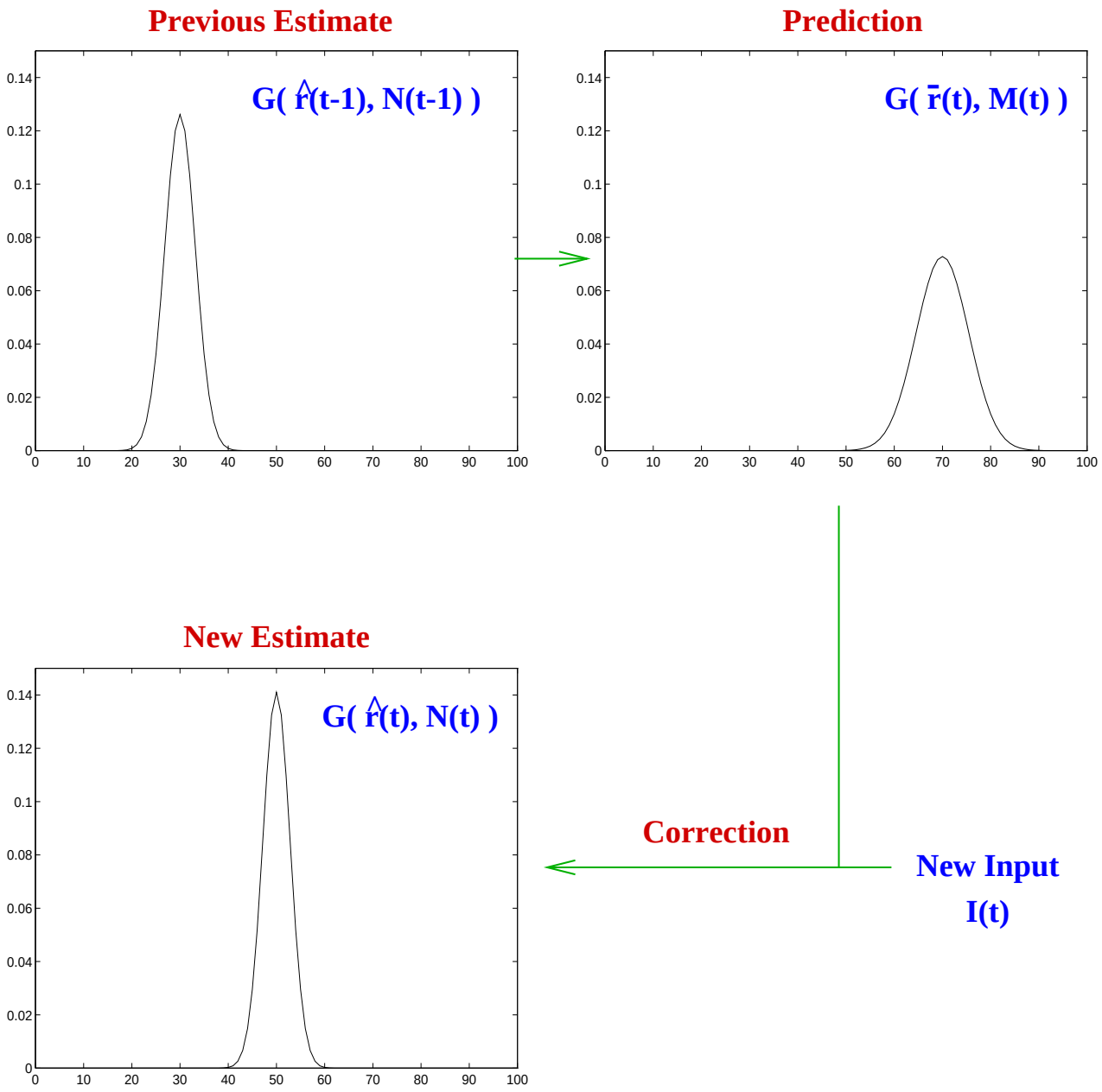


Figure 3: **Propagation of Probability Density in the Kalman filter**. The estimate of the state at time $t-1$ is a Gaussian density with mean $\hat{\mathbf{r}}(t-1)$ and covariance $N(t-1)$. These values are used to predict the mean $\bar{\mathbf{r}}(t)$ and covariance $M(t)$ for the next time step (Equations 19 and 20). Note that this generally results in an increase in uncertainty, as suggested by the increase in variance of the Gaussian bump in the figure. The input at the next time step is used to correct $\bar{\mathbf{r}}$ and M according to the Kalman filter equations, resulting in a new mean $\hat{\mathbf{r}}(t)$ and new covariance $N(t)$. This process is repeated for each subsequent time step.

which reduces to exactly the same two equations (Equation 5 and 6) that we derived in Section 2 for computing the running average.

6 Learning an Internal Model

The previous section derived the Kalman filter for estimating the state $\mathbf{r}$, assuming that the measurement (or generative) matrix U and the state transition (or prediction) matrix V were known. As noted previously, these matrices together encode the filter's internal model of the observed dynamic process. Traditionally, engineers have used hand-coded dynamic models, picking values for U and V according to the physics of the dynamic system or other forms of a priori knowledge of the task at hand [Ayache and Faugeras, 1986; Blake and Yuille, 1992; Broida and Chellappa, 1986; Dickmanns and Mysliwetz, 1992; Hallam, 1983; Matthies *et al.*, 1989; Pentland, 1992]. However, in complex dynamic environments, the formulation of such hand-coded models becomes increasingly difficult. An interesting alternative [Rao and Ballard, 1996a] is to initialize the matrices U and V to small random values, and then adapt these values in response to input data, thereby *learning* an internal model of the input environment. We explore this alternative in this section.

6.1 Learning the Measurement Matrix

The starting point for deriving "learning rules" for U and V is the observation that given a Kalman filter estimate $\hat{\mathbf{r}}$ for the state $\mathbf{r}$ based on some prior values for U and V, one can update the prior values for U and V using two additional update equations that together minimize a joint optimization function J. First, let $\mathbf{u}$ and $\mathbf{v}$ denote the vectorized forms of the matrices U and V respectively. For example, the $n \times k$ generative matrix U can be collapsed into an $nk \times 1$ vector $\mathbf{u} = [U^1 U^2 \ldots U^n]^T$ where U^i denotes the ith row of U. We assume these vectors stochastically drift over time according to:

$$\mathbf{u}(t) = \mathbf{u}(t-1) + \mathbf{n}_u(t-1) \quad (27)$$

$$\mathbf{v}(t) = \mathbf{v}(t-1) + \mathbf{n}_v(t-1) \quad (28)$$

where $\mathbf{n}_u$ and $\mathbf{n}_v$ are stochastic noise processes with mean $\overline{\mathbf{n}}_u$ and $\overline{\mathbf{n}}_v$, and covariances given by Π_u and Π_v respectively. Note that unlike the dynamics of $\mathbf{r}$, these equations for $\mathbf{u}$ and $\mathbf{v}$ do not employ a state transition matrix since the physical relationships encoded by the matrices U and V are assumed to be relatively stable, being perturbed only by random stochastic noise over time.

As in the case of $\mathbf{r}$, let $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$ be estimates of $\mathbf{u}$ and $\mathbf{v}$ calculated from prior data. Thus, $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$ represent the means of the Gaussian distributions for $\mathbf{u}$ and $\mathbf{v}$ before the measurement of the current input $\mathbf{I}$. The corresponding covariances are given by $P = E[(\mathbf{u} - \overline{\mathbf{u}})(\mathbf{u} - \overline{\mathbf{u}})^T]$ and $Q = E[(\mathbf{v} - \overline{\mathbf{v}})(\mathbf{v} - \overline{\mathbf{v}})^T]$. We can then redefine the optimization function J as:

$$\begin{aligned} J = {} & (\mathbf{I} - U\mathbf{r})^T \Sigma^{-1} (\mathbf{I} - U\mathbf{r}) + (\mathbf{r} - \overline{\mathbf{r}})^T M^{-1} (\mathbf{r} - \overline{\mathbf{r}}) + (\mathbf{u} - \overline{\mathbf{u}})^T P^{-1} (\mathbf{u} - \overline{\mathbf{u}}) \\ & + (\mathbf{v} - \overline{\mathbf{v}})^T Q^{-1} (\mathbf{v} - \overline{\mathbf{v}}) \end{aligned} \quad (29)$$

In the above, note that we can substitute $(\mathbf{I} - U\mathbf{r}) = (\mathbf{I} - R\mathbf{u})$ where R is the $n \times nk$ matrix given by:

$$R = \begin{bmatrix} \mathbf{r}^T & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{r}^T & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{r}^T \end{bmatrix} \tag{30}$$

As we did in the case of $\mathbf{r}$, by setting $\frac{\partial J}{\partial \mathbf{u}} = 0$ and solving for the optimal estimate $\hat{\mathbf{u}}$ of $\mathbf{u}$, we obtain the following Kalman filter-based "learning rule" for the mean and covariance of $\mathbf{u}$ after measurement of input $\mathbf{I}$:

$$\hat{\mathbf{u}}(t) = \overline{\mathbf{u}}(t) + N_u(t)R(t)^T\Sigma(t)^{-1}(\mathbf{I}(t) - R(t)\overline{\mathbf{u}}(t)) \tag{31}$$

$$N_u(t) = (P(t)^{-1} + R(t)^T\Sigma(t)^{-1}R(t))^{-1} \tag{32}$$

where $\overline{\mathbf{u}}(t) = \hat{\mathbf{u}}(t-1) + \overline{\mathbf{n}}_u(t-1)$ and $P(t) = N_u(t-1) + \Pi_u(t-1)$. Note the close similarities between these equations and the Kalman filter equations for $\mathbf{r}$ that were derived in Section 5.

6.2 Learning the State Transition Matrix

For deriving the update equations for $\mathbf{v}$ from the optimization function J, we define a $k \times k^2$ matrix $\widehat{R}(t)$ as:

$$\widehat{R}(t) = \begin{bmatrix} \hat{\mathbf{r}}(t)^T & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{r}}(t)^T & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \hat{\mathbf{r}}(t)^T \end{bmatrix} \tag{33}$$

where $\hat{\mathbf{r}}(t)$ is the Kalman filter state estimate at time t. This allows us to rewrite the state transition step (Equation 19) as:

$$\begin{aligned} \overline{\mathbf{r}}(t+1) &= V(t)\hat{\mathbf{r}}(t)) + \overline{\mathbf{n}}(t) \\ &= \widehat{R}(t)\mathbf{v}(t) + \overline{\mathbf{n}}(t) \end{aligned} \tag{34}$$

Substituting the right hand side of this equation for $\overline{\mathbf{r}}$ in the optimization function J and setting $\frac{\partial J}{\partial \mathbf{v}} = 0$, we obtain the following learning rule for the mean and covariance of $\mathbf{v}$:

$$\hat{\mathbf{v}}(t) = \overline{\mathbf{v}}(t) + N_v(t)\widehat{R}(t)^T M(t)^{-1}\left[\mathbf{r}(t+1) - \overline{\mathbf{r}}(t+1)\right] \tag{35}$$

$$N_v(t) = (Q(t)^{-1} + \widehat{R}(t)^T M(t)^{-1}\widehat{R}(t))^{-1} \tag{36}$$

where $\overline{\mathbf{v}}(t) = \hat{\mathbf{v}}(t-1) + \overline{\mathbf{n}}_v(t-1)$ and $Q(t) = N_v(t-1) + \Pi_v(t-1)$. Note that in this case, the estimate of V is corrected using the prediction error $(\mathbf{r}(t) - \overline{\mathbf{r}}(t))$, which denotes the difference between the actual state and the predicted state.

6.3 Convergence of the Learning Scheme

An interesting question is the issue of convergence of the overall filtering/learning scheme involving $\mathbf{r}$, U, and V. Fortunately, one can appeal to the well-known Expectation-Maximization (EM) algorithm from statistics [Dempster *et al.*, 1977] and allow the overall scheme to converge by choosing appropriate values for the state $\mathbf{r}$ in the above learning rules for $\mathbf{u}$ and $\mathbf{v}$ (note that in the above rules, we did not specify values for $\mathbf{r}(t)$ (comprising $R(t)$ in Equation 31 and $\mathbf{r}(t+1)$ in Equation 35).

The EM algorithm suggests that in the case of static input stimuli ($\overline{\mathbf{r}}(t) = \widehat{\mathbf{r}}(t-1)$), one may use $\mathbf{r}(t) = \widehat{\mathbf{r}}$ when updating the estimate for $\mathbf{u}$, where $\widehat{\mathbf{r}}$ is the converged optimal state estimate for the given static input. In the case of dynamic (time-varying) stimuli, the EM algorithm prescribes the use of $\mathbf{r}(t) = \widehat{\mathbf{r}}(t|N)$, which is the optimal temporally *smoothed* state estimate [Bryson and Ho, 1975] for time t ($\leq N$), given input data for each of the time instants $1, \ldots, N$. Unfortunately, the smoothed state estimate requires knowledge of future inputs and is computationally quite expensive. For the experimental results described in this paper, we approximated the smoothed estimates by their on-line counterparts $\widehat{\mathbf{r}}(t)$ when updating the matrices U and V during training.

6.4 Static Recognition Examples

In the first experiment intended to evaluate the proposed learning and recognition framework, we tested the ability of the filter to learn and recognize static $3D$ objects based solely on their appearance. Grayscale images of size 105×105 pixels depicting five $3D$ objects were used for training the filter (Figure 4 (a)). The generative matrix U was of size 11025×5, implying that the filter was forced to generate predictions based on only five basis images that form the columns of U. The result is a significant reduction in dimensionality, from the 11025-dimensional input image space to a 5-element state vector $\mathbf{r}$. The elements of the matrix U were initialized to small random values and each column normalized to length one, as were the input images. For learning static inputs, the prediction matrix V is unnecessary since we may use $\overline{\mathbf{r}}(t) = \widehat{\mathbf{r}}(t-1)$ and $M(t) = N(t-1)$. This results in an iterative Kalman filter that converges, after a few iterations, to the optimal state estimate $\widehat{\mathbf{r}}$ for a given static input. After convergence of the filter for each input, the matrix U was updated according to Equation 31. Figures 4 and 5 summarize the ongoing effects of this training process.

After training, the filter was tested on images of various objects (Figure 6). The behavior of the trained filter on objects that it has encountered previously was as expected, with almost zero residual errors at all pixels, indicating correct prediction and recognition (Figure 6 (a)). The filter shows a moderate ability to generalize to occluded or incomplete inputs, such as in (b). A more robust alternative for handling such cases is discussed in Section 7. Perhaps the most interesting test case is (c), where the input image contains an object very similar to a training object. The prediction is that of the training object closest in appearance to the input stimulus (in this case, the doll). Such behavior may aid various processes concerned with *categorization* of novel input stimuli. On the other hand, the residual image (rightmost image) accentuates the differences between the training

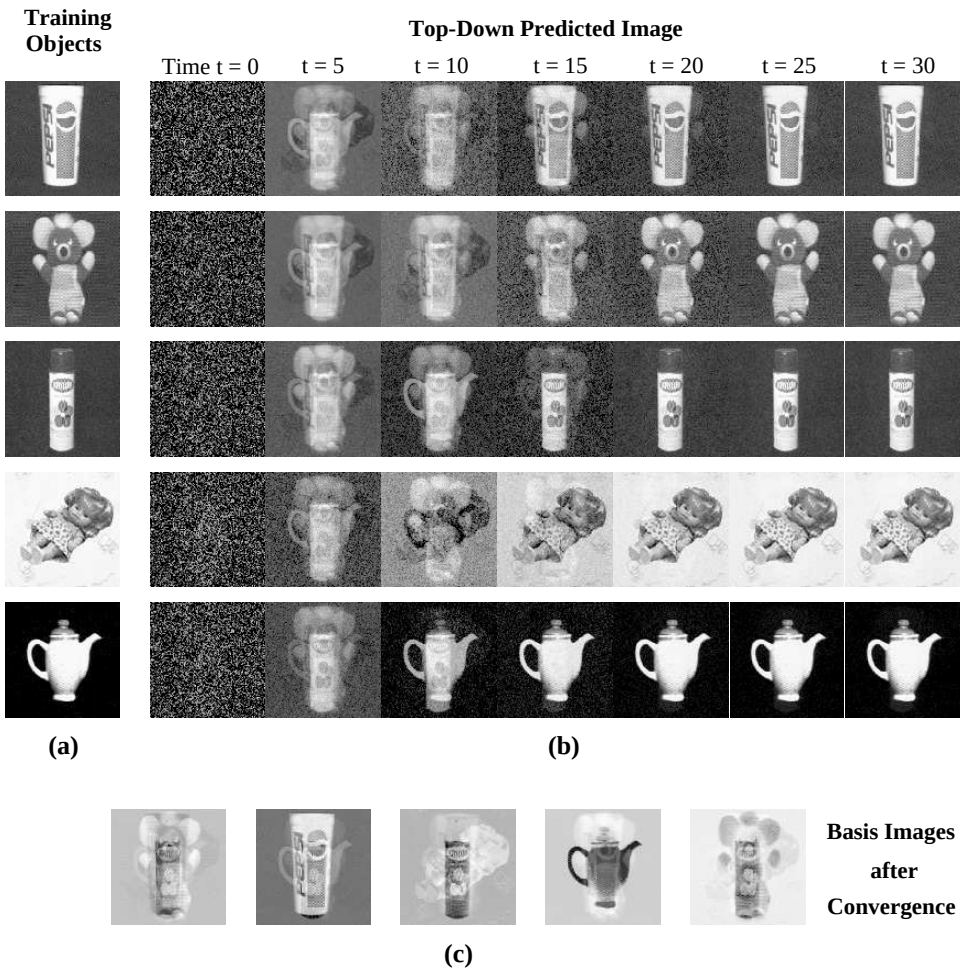


Figure 4: **Example 1: Learning Internal Models of Objects.** (a) The five objects used for training a Kalman filter whose matrix U was initialized to random values. (b) The evolution of the learning process, showing a relatively rapid increase in prediction accuracy after each exposure to the input stimuli. Figure 5 summarizes this learning process over time. (c) The basis images (columns of U) learned by the filter after convergence to stable values. Different linear combinations of these basis images, weighted according to the state vector $\overline{\mathbf{r}}(t)$, give rise to different approximations of the input images, as shown in (b).

object and the new stimulus, preventing a misclassification of the new stimulus as the training object. Such false positive errors have been the bane of many purely feedforward recognition systems, which are unable to "invert" their recognition estimates and verify their hypotheses. A final example demonstrating the ability of the filter to function as a *novelty detector* is shown in Figure 6 (d). Here, a completely novel object was input to the filter, which generates an "average" image with relatively large residual errors at a number of pixel locations. Large residual errors in general imply that the presented stimulus is novel. If the novel stimulus is deemed to be important, it can be made part of the filter's repertoire of known objects by allowing the residual errors to drive the adaptation of the matrix U as specified by Equation 31.

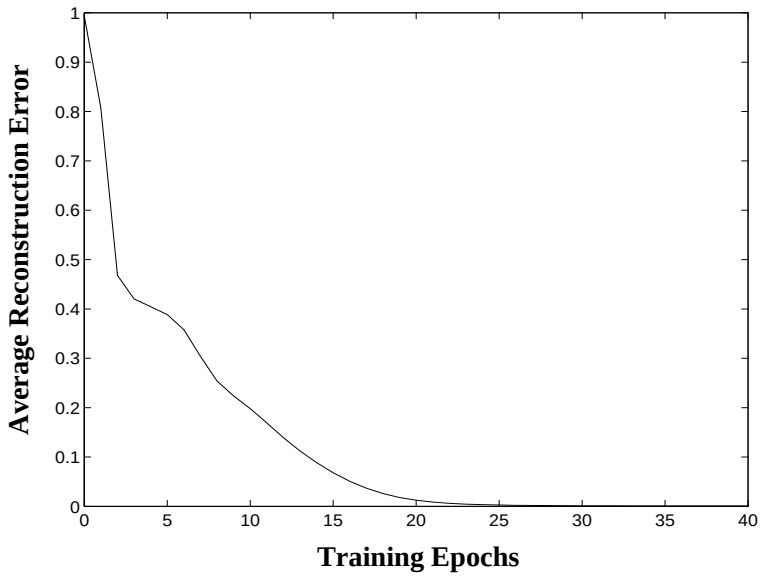


Figure 5: **Learning Curve for Example 1.** The graph shows the average error in image reconstruction (or prediction error), measured as sum of squared pixel-wise errors, across the five training objects as a function of number of exposures to the set of objects.

6.5 View-Based Recognition of 3D Objects

In a second experiment, we evaluated the ability of the filter to recognize $3D$ objects using a view-based approach [Poggio and Edelman, 1990; Murase and Nayar, 1995]. Thirty-six views of two different $3D$ objects, each view 10° azimuth apart from the next, were used for training the filter (Figure 7 (a)). For computational efficiency, only the 32×32 image patches from the central image region were used for training (other regions are assumed to be analyzed by neighboring modules - see Section 8). The matrix U was of size 1024×50.

As shown in Figure 7, training images produce accurate predictions (reconstructions) with low residuals (top two rows). An intermediate view that was 5° from the nearest training view generated a moderately accurate interpolated prediction (middle row). This was apparently sufficient for the 100% recognition rate that was obtained in this simple case for 36 different testing views of each object, each test view being 5° away from the nearest training view. The second to last row depicts how the effect of occlusions spreads globally [Leonardis and Bischof, 1996], as seen in the mediocre prediction and relatively large residuals at many locations. This is handled via robust estimation (Section 7). Finally, a completely novel object generates an "average" image, and large residuals as in the example in the previous section.

6.6 Dynamic Recognition Examples

In the third experiment, we evaluated the filter's ability to learn internal models of dynamic stimuli. Three image sequences were used for training the filter (Figure 8): (1) a horizontal bar moving downwards, (2) a vertical bar moving to the right, and (3) an expanding circle. Each sequence

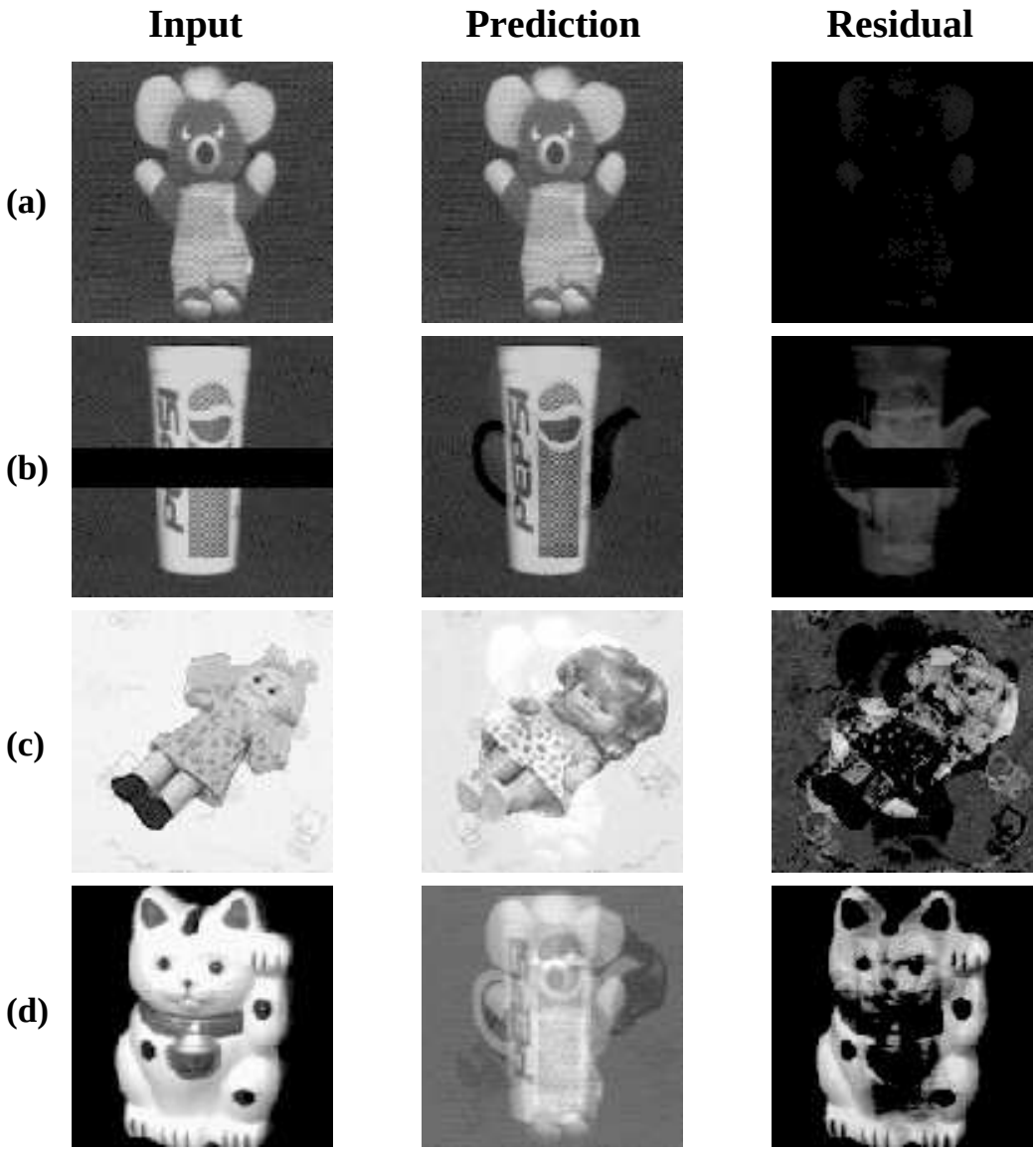


Figure 6: **Using Internal Models for Object Recognition.** The internal models of objects learned in Figure 4 were tested by using various input images and observing the response of the filter. (a) When an object in the training set is input (left), the prediction generated is an almost exact reconstruction of the input image (middle), with small residual errors at all image locations (dark image on the right). (b) Inputs with missing data are handled gracefully, the predicted image being that of the closest training object. However, some artifacts can also be observed (middle image) with some residual errors in prediction (right). Missing data and occlusions are dealt with in Section 7 (see Figures 15 and 16). (c) A novel object (a doll) that resembles a training object (another doll) causes the filter to predict the closest resembling training object (the doll). The residual errors (on the right) highlight the differences between the two similar objects. (d) A completely novel object results in a prediction resembling a mixture of the training images, with large residual errors. These residuals can be used to learn the new object in case it is deemed relevant to the recognition system, using Equation 31.

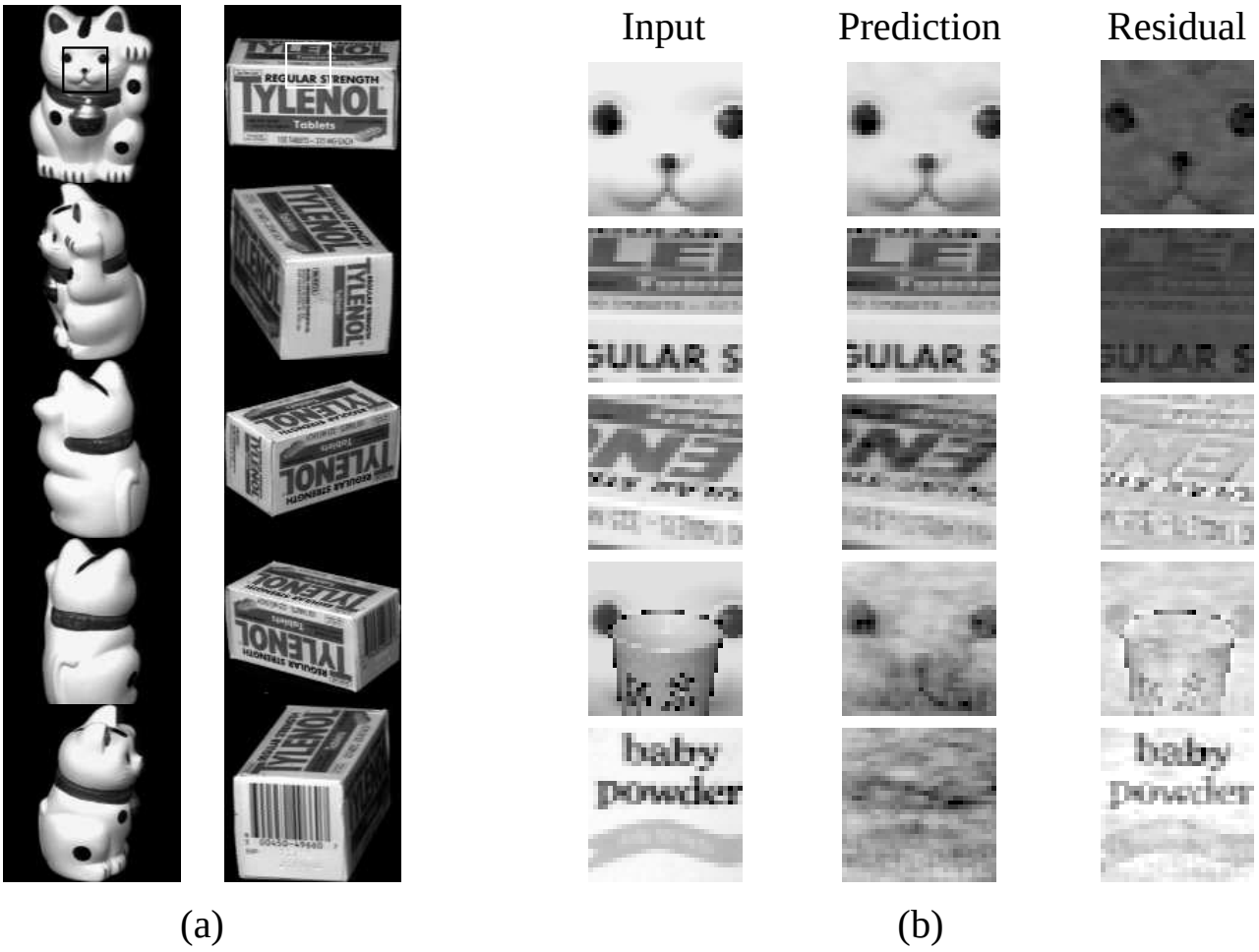


Figure 7: **Example 2: View-Based Recognition of 3D Objects.** (a) shows 5 of the 36 training views used for learning the generative matrix U for two different $3D$ objects. The trained filter was then tested on 36 intermediate views for each object. Only the image region demarcated by the box was used for training to preserve computational efficiency; other regions are assumed to be analyzed by neighboring modules (see Section 8). (b) shows some examples of the responses generated by the trained filter.

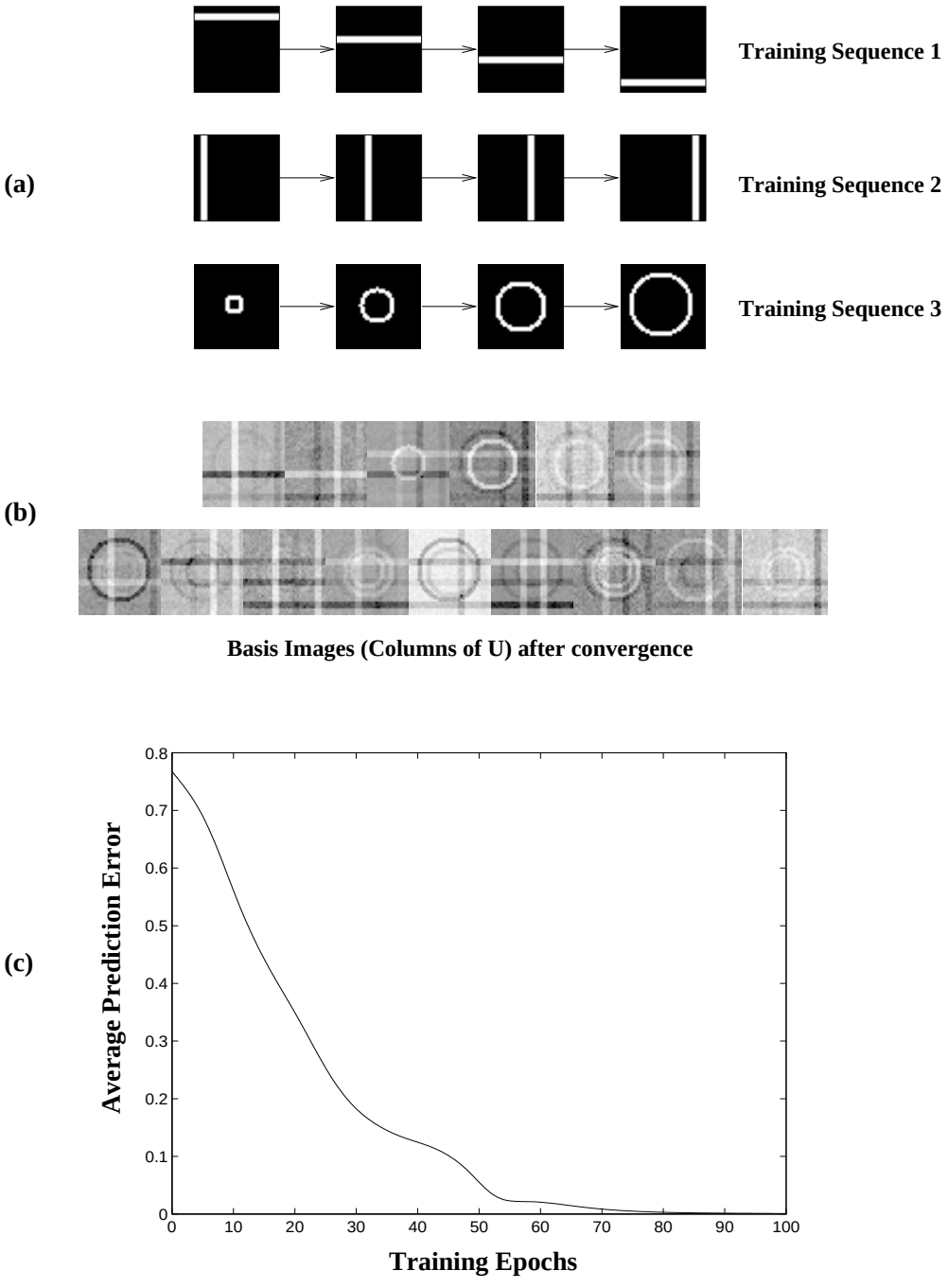


Figure 8: **Example 3: Learning Internal Models of Dynamic Stimuli.** (a) shows the three image sequences used for training the filter. (b) shows the basis images learned by the filter after being exposed to the training stimuli. (c) depicts the learning curve, a plot of the average prediction error (sum of squared pixel-wise errors) as a function of the number of exposures to the training sequences.

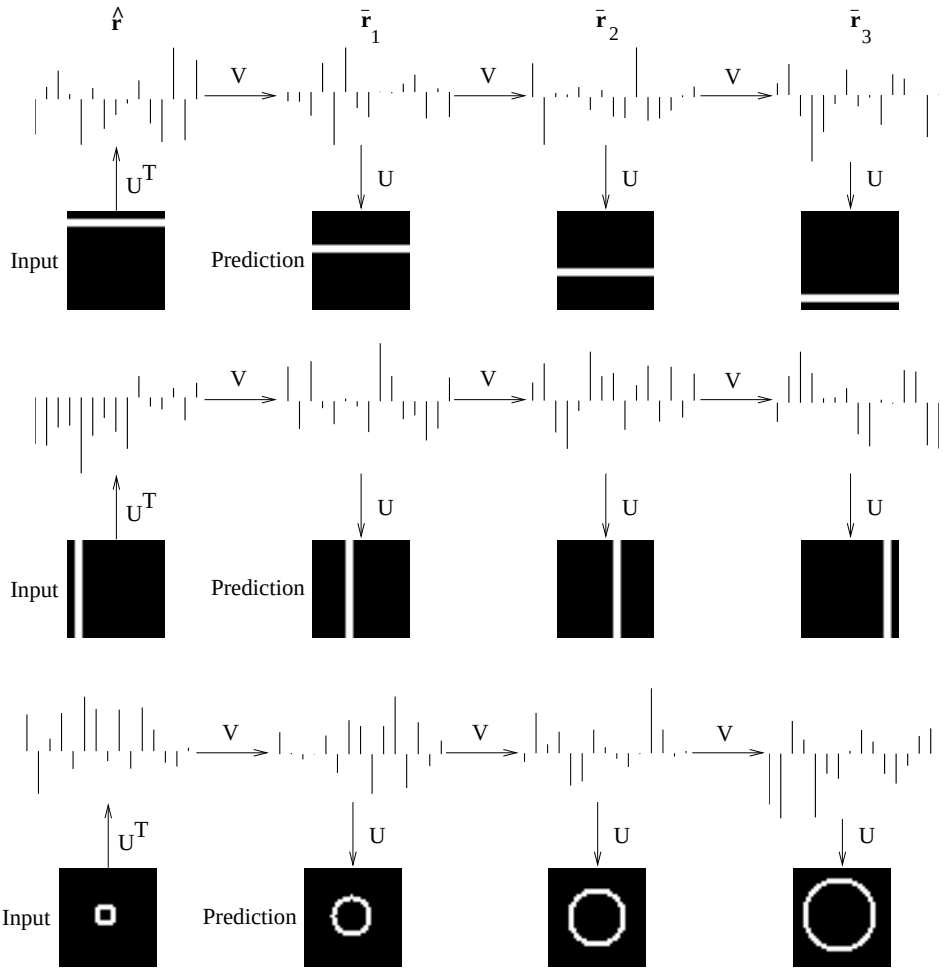


Figure 9: **Using Internal Models to Predict Input Sequences.** The figure depicts the process of generating three consecutive predictions, given a single input image (extreme left), for each of the three training stimuli of Figure 8. The vectors $\hat{\mathbf{r}}$ and $\overline{\mathbf{r}}$ are shown as histograms, with positive values denoted by bars oriented upwards and negative values by bars oriented downwards. The first image generates an estimate $\hat{\mathbf{r}}$ via the feedforward matrix U^T. This estimate is used to generate the subsequent predictions $\overline{\mathbf{r}}$ by cycling through the prediction matrix V. Each $\overline{\mathbf{r}}$ can be translated to its corresponding image via the generative matrix U.

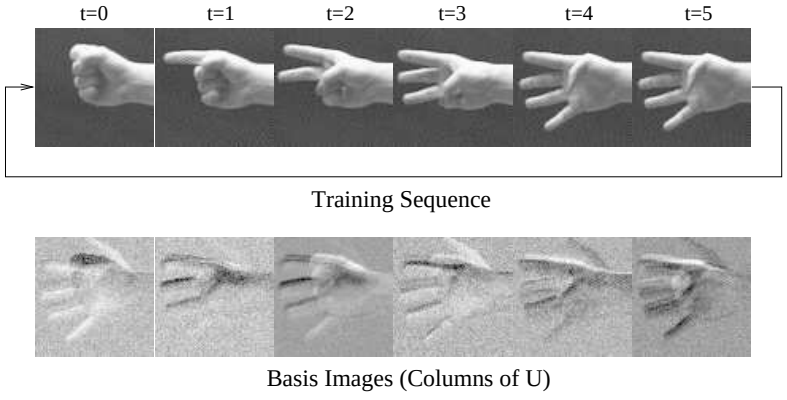


Figure 10: **Example 4: Learning Sequences of Gestures.** The top row shows a cyclic image sequence of hand gestures used to train a Kalman filter. The bottom row shows six of the fifteen basis images (columns of the matrix U) learned after exposure to the cyclic training sequence.

consisted of four 38×38 images. The generative matrix U and the prediction matrix V were initialized to random 1444×15 and 15×15 matrices respectively. These matrices were adapted according to Equations 31 and 35 during repeated exposures to the training sequences. Figure 8 (c) depicts this learning process in terms of the reduction in the average prediction error across the three sequences over time. The final learned basis images, which form the columns of U, are shown in (b).

Figure 9 illustrates the process by which the trained filter uses its internal model and associated internal representations to generate successive predictions. For each of the three training image sequences, an initial input image is multiplied by the "feedforward" matrix U^T to generate an estimate $\hat{\mathbf{r}}$ of the initial state (see Figure 2). For each sequence, this vector is shown in the figure as a histogram, with positive values in the vector represented as bars above the horizontal, negative values as bars below the horizontal. Successive multiplications of this state estimate with the learned prediction matrix V generate successive predictions of the state at future time steps. These predictions are depicted in the figure as histograms marked $\overline{\mathbf{r}}_1$, $\overline{\mathbf{r}}_2$, and $\overline{\mathbf{r}}_3$. Each of these internal state representations can be in turn be translated to their corresponding image representations by multiplying them with the learned generative matrix U, as shown in the figure for each time step.

The fourth experiment was intended to verify the ability of the filter to learn internal models of more complex and possibly articulated stimuli. A filter was trained on an image sequence depicting a set of hand gestures (Figure 10). Each image was grayscale and of size 75×75 pixels. The matrices U and V (of size 5625×15 and 15×15 respectively) were initialized to small random values, before training using Equations 31 and 35. Some of the basis images (columns of U) obtained after training are shown in Figure 10 (bottom row).

Figure 11 illustrates the prediction and recognition of the gesture sequence using the learned internal model. The filter was initialized to a random state vector, causing large residual errors (third row) at the initial time step. The errors are however corrected rapidly due to the Kalman

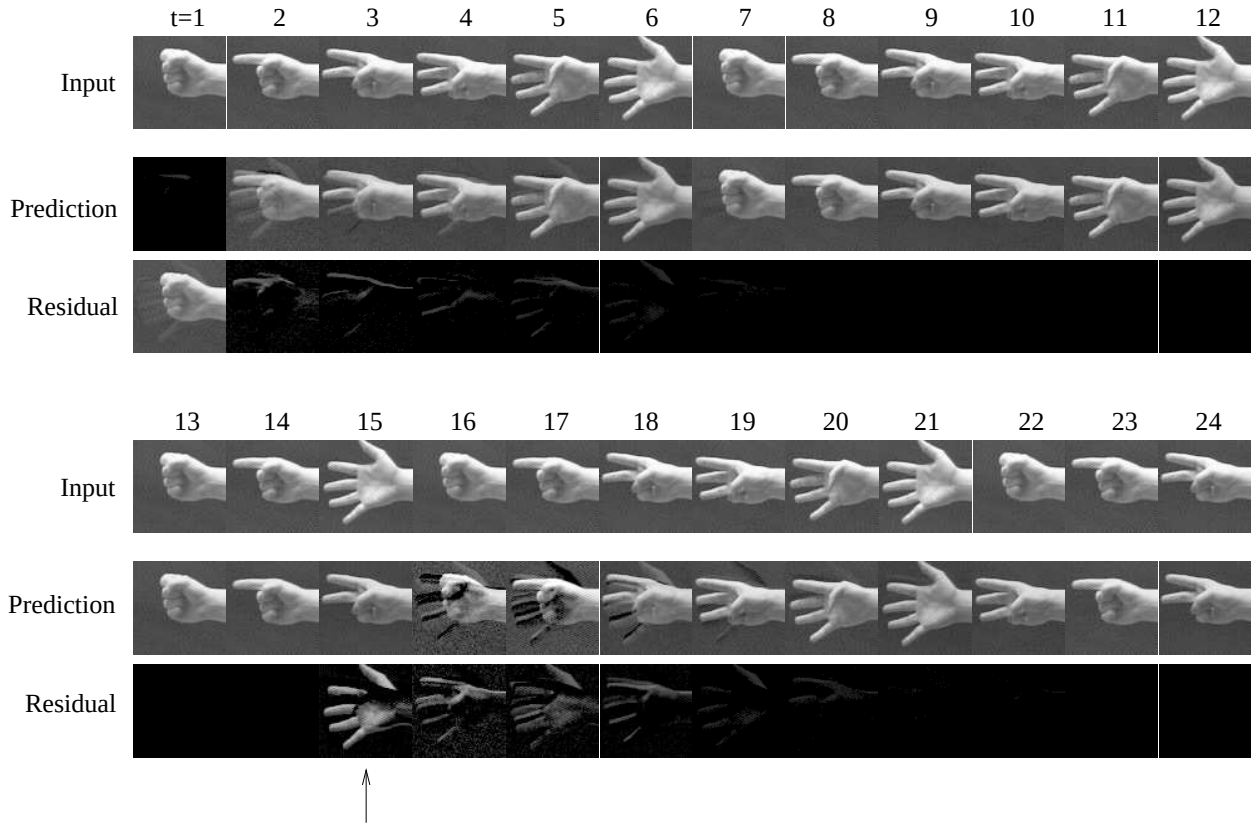


Figure 11: **Tracking and Recognizing Gestures.** (Top panel) When initialized to a random state, the filter rapidly corrects its prediction (second row) such that the initially large residual errors (third row) become appreciably small within the first few time steps. (Bottom panel) If the sequence is abruptly interrupted and another part of the sequence inserted (time step 15 as marked by the arrow), the relatively large residual errors cause the filter to immediately correct itself within the next two or three time steps, allowing accurate predictions of the interposed stimuli at subsequent time steps.

filter dynamics, resulting in relatively accurate predictions at subsequent time steps. An interesting exercise, marked by the arrow in the lower panel of Figure 11, is to abruptly interrupt the input sequence with an unexpected subsequence. This causes a large residual image due to the unexpected stimulus, but the filter soon corrects itself and begins to recognize and track the new interposed sequence, as evident from the accurate predictions and low residual errors in the subsequent time steps.

6.7 Hidden State and Perceptual Aliasing

In the final experiment, we investigated how the filter handles the pervasive problem of "hidden state" [McCallum, 1996] or "perceptual aliasing" [Whitehead and Ballard, 1991; Chrisman, 1992] in partially observable environments. This problem has received much attention in the reinforcement learning community (for example, see [Kaelbling *et al.*, 1996]). The essence of the problem lies in the fact that *a given observation of the environment by itself might be insufficient to determine the corresponding state of the environment.* A simple example of this problem is given in Figure 12 (a),

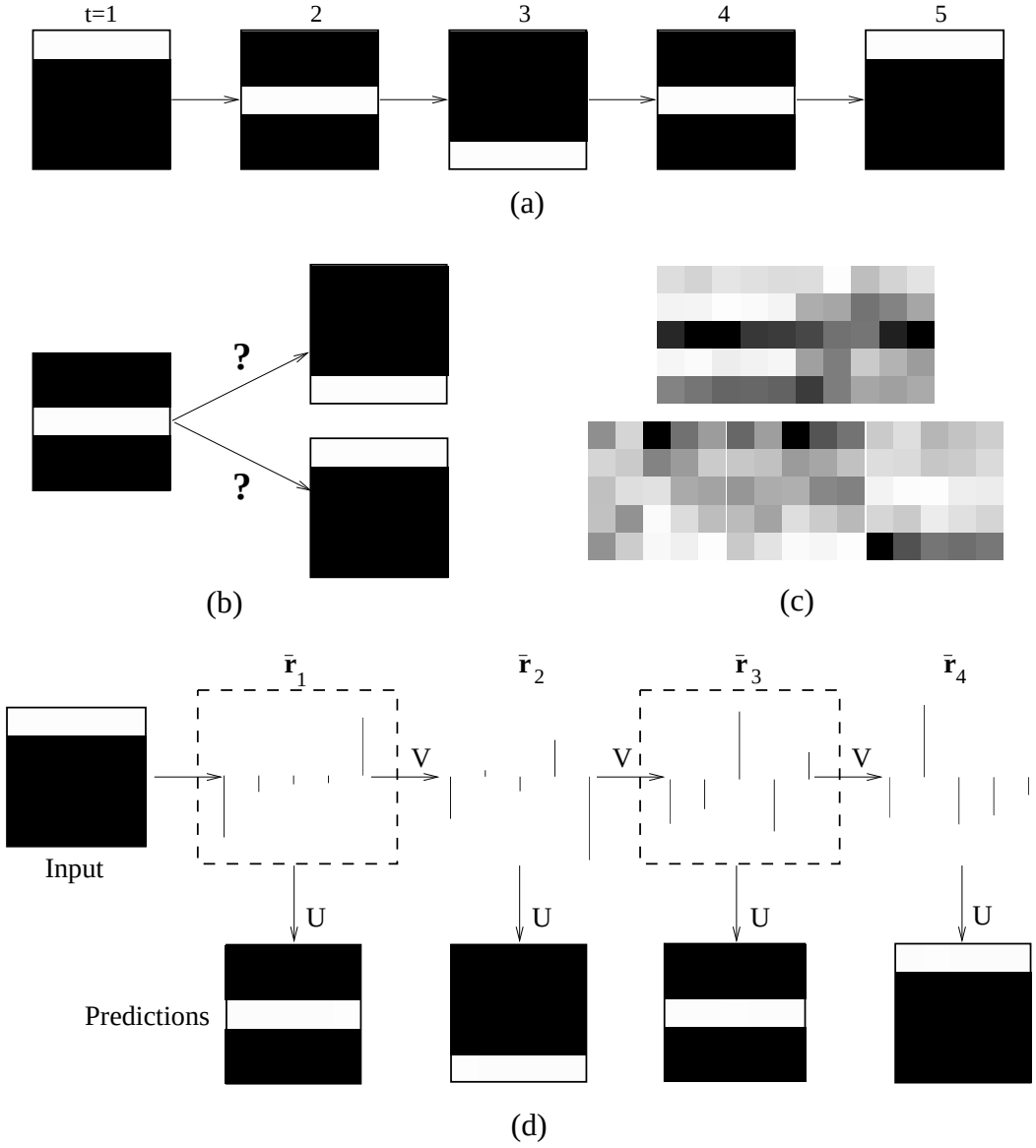


Figure 12: **Learning to Disambiguate Aliased Inputs.** (a) shows an image sequence depicting a horizontal bar, first moving down and then up. Note that the same image is encountered at time steps 2 and 4, but different images are to be predicted at the next time step, as shown in (b). This is the problem of perceptual aliasing/hidden state, where the current input alone is insufficient to determine the current state and predict the next input. (c) and (d) show how the adaptive filter derived in the previous section handles this problem. The five basis functions (columns of U) learned by the filter are shown in (c). Using these basis images (matrix U) and the learned prediction matrix V, we see in (d) how the filter has learned two different internal state representations $\overline{\mathbf{r}}_1$ and $\overline{\mathbf{r}}_3$ of the same (aliased) image at time steps 2 and 4. This allows the filter to disambiguate the aliased input and accurately predict the two very different stimuli at the next time step in each of the two cases.

which depicts a horizontal bar that first moves down and then up. Note that the observations made at time steps 2 and 4 are exactly the same, but in one case, the state is that of moving down while in the other, it is that of moving up. The observed image by itself is insufficient to determine the current state of the input environment and the next prediction to be generated (Figure 12 (b)). One needs to make use of prior contextual information in order to correctly predict the next input.

Figures 12 (c) and (d) show how a Kalman filter with a 5-element state vector can learn to disambiguate the aliased inputs. The five basis images (columns of U) learned by the filter, after several exposures to the input training sequence, is shown in (c). The learned matrices U and V together allow the filter to disambiguate the identical inputs at time steps 2 and 4, as shown in (d). The histograms within the dotted boxes represent the filter's state predictions $\overline{\mathbf{r}}_1$ and $\overline{\mathbf{r}}_3$ for time steps 2 and 4. The significant differences between these two vectors indicate that the filter has learned to represent the aliased inputs as two different states, allowing very different predictions $\overline{\mathbf{r}}_2$ and $\overline{\mathbf{r}}_4$ at the next time steps when multiplied by V. However, despite these differences, the representations $\overline{\mathbf{r}}_1$ and $\overline{\mathbf{r}}_3$ were learned in such a manner that they generate the same image when multiplied by the generative matrix U.

7 Robust Kalman Filters

The optimization function J used in the previous sections for deriving the Kalman filter was a quadratic function of the residual errors $(\mathbf{I} - U\mathbf{r})$. A quadratic optimization function is however susceptible to *outliers* (or gross errors) i.e. data points that lie far away from the majority of the data points in $\mathbf{I}$ [Huber, 1981]. For example, in the case where $\mathbf{I}$ represents an input image, occlusions, background clutter, and other forms of noise may cause many pixels in $\mathbf{I}$ to deviate significantly from corresponding pixels in the predicted image $U\mathbf{r}$ of an object of interest contained in the image $\mathbf{I}$. These deviating pixels need to be treated as outliers and discounted for in the minimization process in order to get an accurate estimate of the state $\mathbf{r}$.

The field of *robust statistics* [Huber, 1981] suggests some useful techniques for preventing gross outliers from influencing the solution to an estimation problem. A commonly used technique is *M-estimation* (Maximum likelihood type estimation), which involves minimizing a function of the form:

$$J' = \sum_{i=1}^{n} \rho\left(\mathbf{I}^i - U^i\mathbf{r}\right) \tag{37}$$

where ρ is a function that increases much less rapidly than the square. This ensures that large residual errors (which correspond to outliers) do not influence the optimization of J' as much as they would in a quadratic function. Note that when ρ equals the square function, we obtain the quadratic error function we previously used in J. More interestingly, suppose we define ρ in terms of a diagonal matrix S as follows:

$$J' = (\mathbf{I} - U\mathbf{r})^T S(\mathbf{I} - U\mathbf{r}) \tag{38}$$

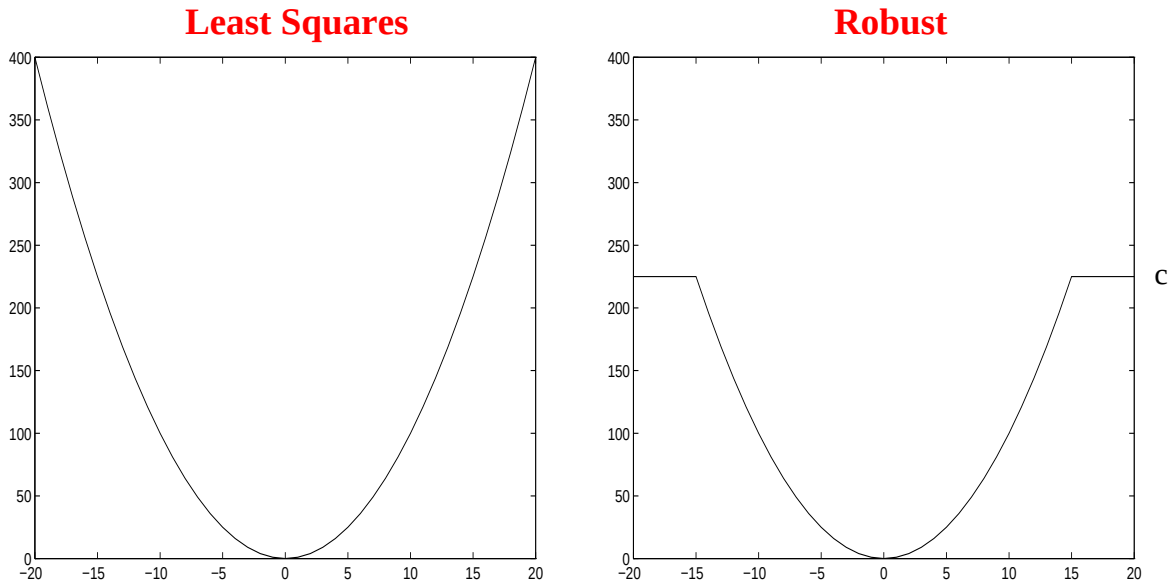


Figure 13: **Least Squares versus Robust Optimization.** The robust optimization function clips large residual errors i.e. those exceeding the threshold c to the constant saturation value c, thereby preventing the corresponding outliers in the input data from influencing the optimization process.

where the diagonal entries $S^{i,i}$ determine the weight accorded to the corresponding data residual $(\mathbf{I}^i - U^i\mathbf{r})$. A simple but attractive choice for these weights is the non-linear function given by:

$$S^{i,i} = \min\left\{1, c/(\mathbf{I}^i - U^i\mathbf{r})^2\right\} \tag{39}$$

where c is a threshold parameter. To understand the behavior of this function, note that S effectively clips the ith summand in J' to a constant saturation value c whenever the ith squared residual $(\mathbf{I}^i - U^i\mathbf{r})^2$ exceeds the threshold c; otherwise, the summand is set equal to the squared residual. Figure 13 contrasts this robust optimization function with the standard least squares optimization function.

By substituting $\Sigma^{-1} = S$ in the optimization function J (Equation 11), we can rederive the Kalman filter update equations. The resulting *robust Kalman filter* for updating the state estimate is given by:

$$\widehat{\mathbf{r}}(t) = \overline{\mathbf{r}}(t) + N(t)U^T G(t)(\mathbf{I} - U\overline{\mathbf{r}}(t)) \tag{40}$$

$$\overline{\mathbf{r}}(t) = V\widehat{\mathbf{r}}(t-1)) + \overline{\mathbf{m}}(t-1) \tag{41}$$

where $N(t) = (U^T G(t)U + M(t)^{-1})^{-1}$, $M(t) = VN(t-1)V^T + \Pi(t-1)$, and $G(t)$ is an $n \times n$ diagonal matrix whose diagonal entries at time instant t are given by:

$$G^{i,i} = \begin{cases} 0 & \text{if } (\mathbf{I}^i(t) - U^i\overline{\mathbf{r}}(t))^2 > c(t) \\ 1 & \text{otherwise} \end{cases}$$

G can be regarded as the sensory residual gain or "gating" matrix, which determines the (binary) gain on the various components of the incoming sensory residual error. By effectively excluding any

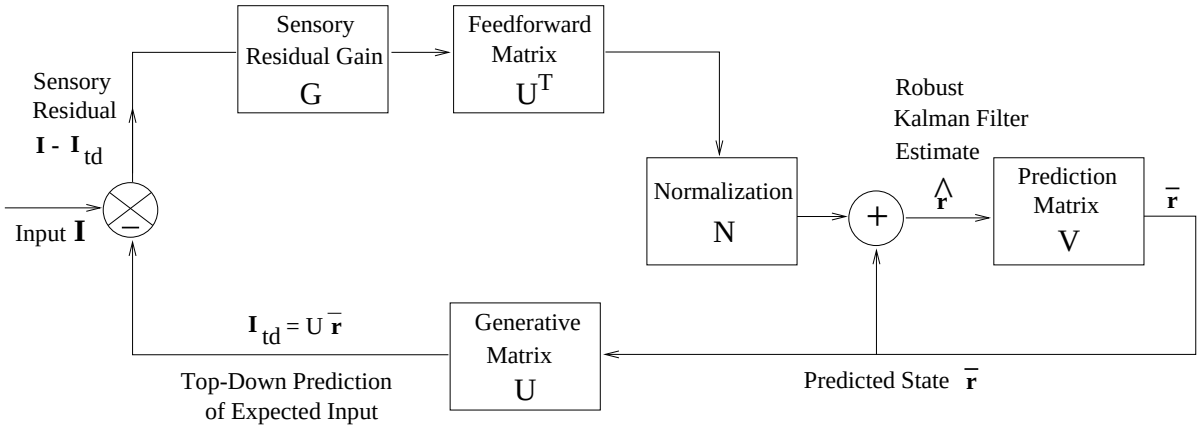


Figure 14: **Schematic Diagram of the Robust Kalman Filter**. The sensory residual gain or gating matrix G is a non-linear function of the current residuals. It effectively filters out any high residuals, thereby preventing outliers in input data $\mathbf{I}$ from influencing the robust Kalman filter estimate $\widehat{\mathbf{r}}$.

high residuals, G allows the Kalman filter to ignore the corresponding outliers in the input $\mathbf{I}$, thereby enabling it to robustly estimate the state $\mathbf{r}$. Figure 14 depicts the operation of the robust Kalman filter and its various components.

7.1 Robust Static Recognition Examples

To evaluate the robust filter, we used a filter trained on the same two objects that were used in Figure 7. During robust filtering and recognition, the outlier threshold c was initialized to the sum of the mean plus k standard deviations of the current distribution of squared residual errors $(\mathbf{I}^i - U^i\mathbf{r})^2$, where k was initialized to an appropriately large value (e.g. $k = 3$). The value of k was gradually decreased during each iteration in order to allow the filter to refine its robust estimate by gradually pruning away the outliers, as the filter converges to a single object estimate. After convergence, the diagonal of the matrix G contains zeros in the image locations containing the outliers and ones in the remaining locations. Figure 15 (a) depicts how the robust form of the filter can reject outliers and produce an accurate prediction of an occluded object (compare with Figure 7).

Interestingly, the outliers (white) produce a crude *segmentation* of the occluder, which can subsequently be used to focus "attention" on the occluder and recover its identity. In particular, an *outlier mask* $\mathbf{m}$ can be defined by taking the complement of the diagonal of G (i.e. $\mathbf{m}^i = 1 - G^{i,i}$). By replacing the diagonal of G with $\mathbf{m}$ in Equation 40 and repeating the estimation process, one can obtain robust estimates of the image region(s) that were previously treated as outliers. Such a two-step recognition process is depicted in Figure 15 (b), where the image is a combination of the two training objects in Figure 7. The filter first recognizes the "dominant" object, which was generally observed to be the object occupying a larger area of the input image or possessing regions with higher contrast. The outlier mask $\mathbf{m}$ is subsequently used for extracting the identity of the second object (lower arrow).

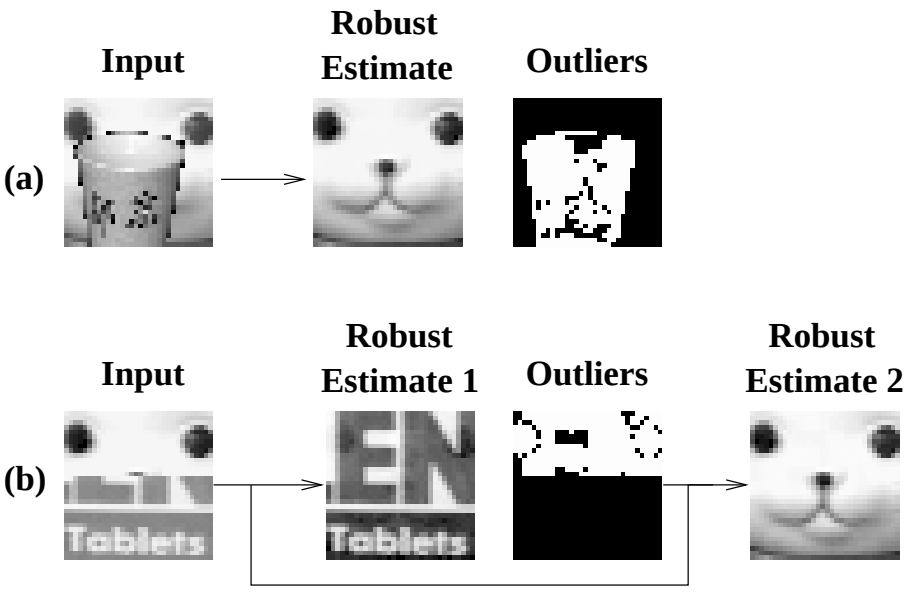


Figure 15: **Robust Recognition: Example 1**. (a) depicts the robust estimation of object identity in the presence of an occlusion. The portions of the input image treated as outliers (the diagonal of the gating matrix G) are shown in white in the rightmost image. (b) demonstrates the case where the input contains combinations of the training objects (same objects as in Figure 7). The filter first converges to one of the objects (the "dominant" one in the image). The identity of the second object is then retrieved using the complement of the outlier mask produced during the recognition of the first object.

A second example using images with slightly more complex forms of occlusions and clutter is shown in Figure 16. Static grayscale images of size 65×105 depicting two $3D$ objects were used for training a filter with the matrix U of size 6825×5 (Figure 16 (a)). As shown in (b), the filter was successful in segmenting and recognizing the training object in spite of considerable occlusion and background clutter. The case where one training object is occluding another is shown in (c). Both objects were successfully recognized by the robust filter. The standard least-squares Kalman filter was unable to resolve either of the two objects as shown in the image at the extreme right.

7.2 Robust Dynamic Recognition Examples

To test robustness during spatiotemporal recognition, we used the trained filter from example 3 (Figure 8). In the first test, we added uniformly distributed additive noise to the images in the expanding circle sequence. The robustness parameter c was set to the sum of the mean plus 1.5 standard deviations of the current distribution of squared residual errors. As shown in Figure 17 (a), the robust filter produced relatively accurate predictions of the noisy images, when initialized with the first image of the expanding circle sequence.

A more interesting case involving ambiguous stimuli is shown in Figure 17 (b) and (c). The input in this case is comprised of a sequence of three images, each containing both a horizontal *and* a vertical bar. Note that the filter was trained on both a horizontal bar moving downwards as well as a vertical bar moving rightwards (Figure 8). Given ambiguous stimuli containing both these

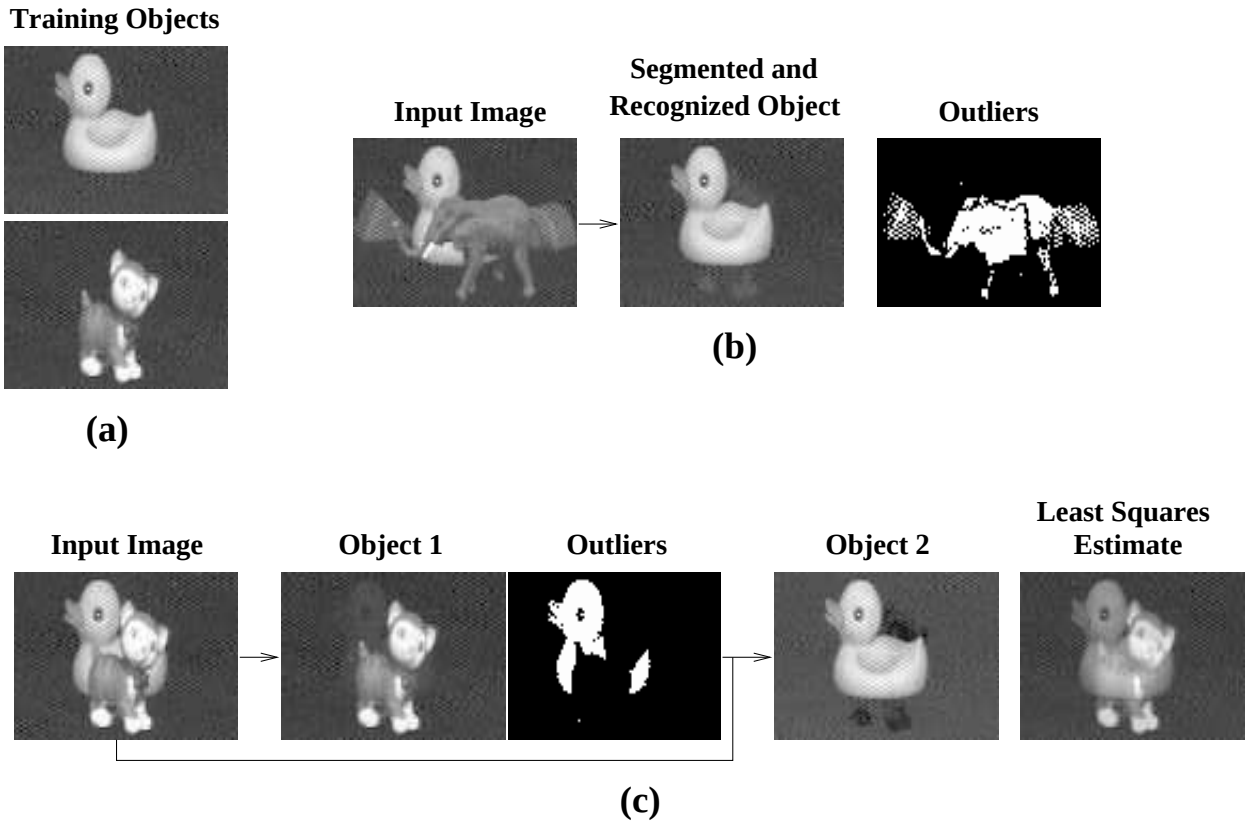


Figure 16: **Robust Recognition: Example 2**. (a) Images used to train the robust filter. (b) Occlusions, background clutter, and other forms of noise are treated as outliers (white regions in the third image, depicting the diagonal of the gating matrix G). This allows the filter to simultaneously segment and recognize the training object, as indicated by the accurate reconstruction (middle image) of the training image based on the final robust state estimate. (c) In the more interesting case of the training objects occluding each other, the filter converges to one of the objects (the "dominant" one in the image). The second object is recognized by taking the complement of the outliers (diagonal of G) and repeating the filtering process (third and fourth images). The fifth image is the image reconstruction obtained from the standard (least squares derived) Kalman filter estimate, showing an inability to resolve or recognize either of the two objects.

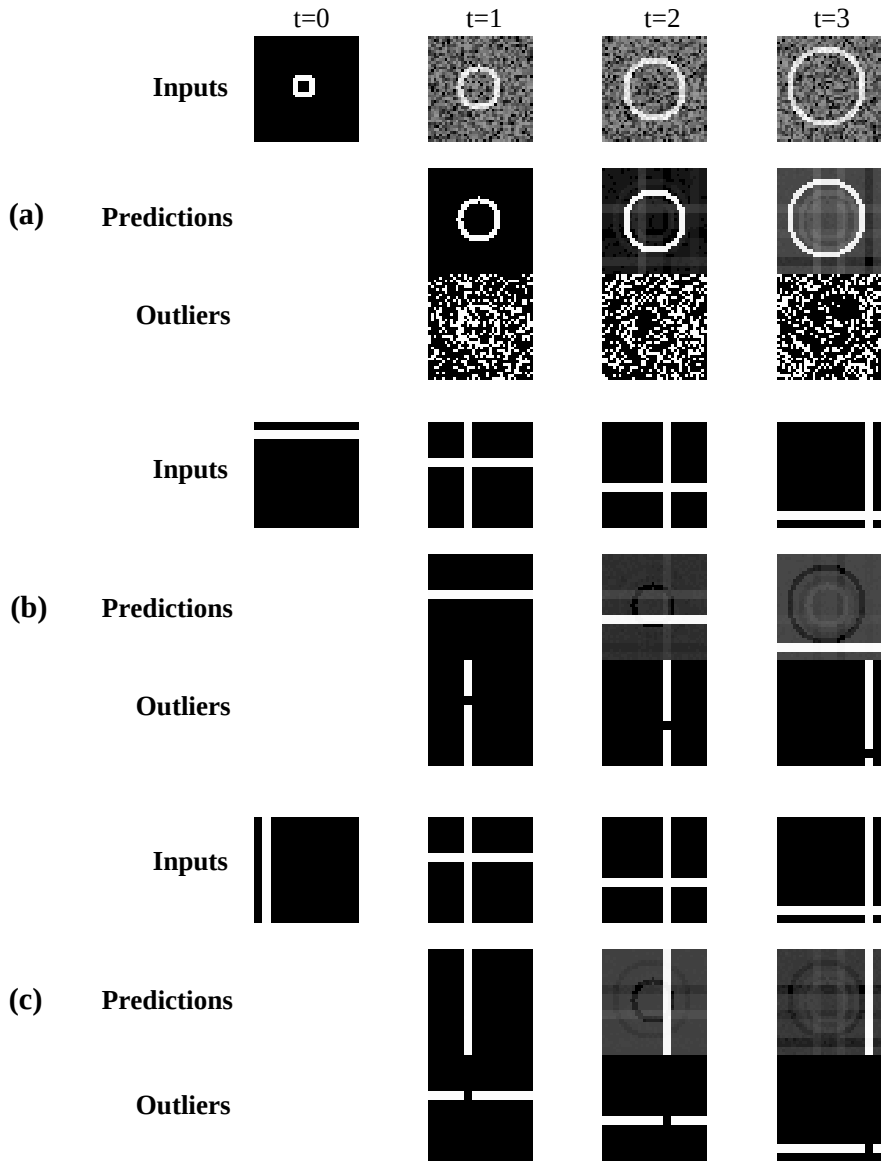


Figure 17: **Robust Segmentation and Recognition of Noisy and Ambiguous Spatiotemporal Stimuli**. The trained filter from Figure 8 was tested for robustness. (a) demonstrates the ability of the robust filter to tolerate additive white noise in the input images by treating noisy pixels as outliers. (b) and (c) show how the same ambiguous stimuli at time steps $t = 1$ through $t = 3$ are interpreted differently based on the initial "priming" input. In case (b), the stimulus is interpreted as a horizontal bar moving downwards, where as in case (c), it is interpreted as a vertical bar moving rightwards. The outliers reflect the corresponding parts of the input that were ignored during interpretation of the stimuli.

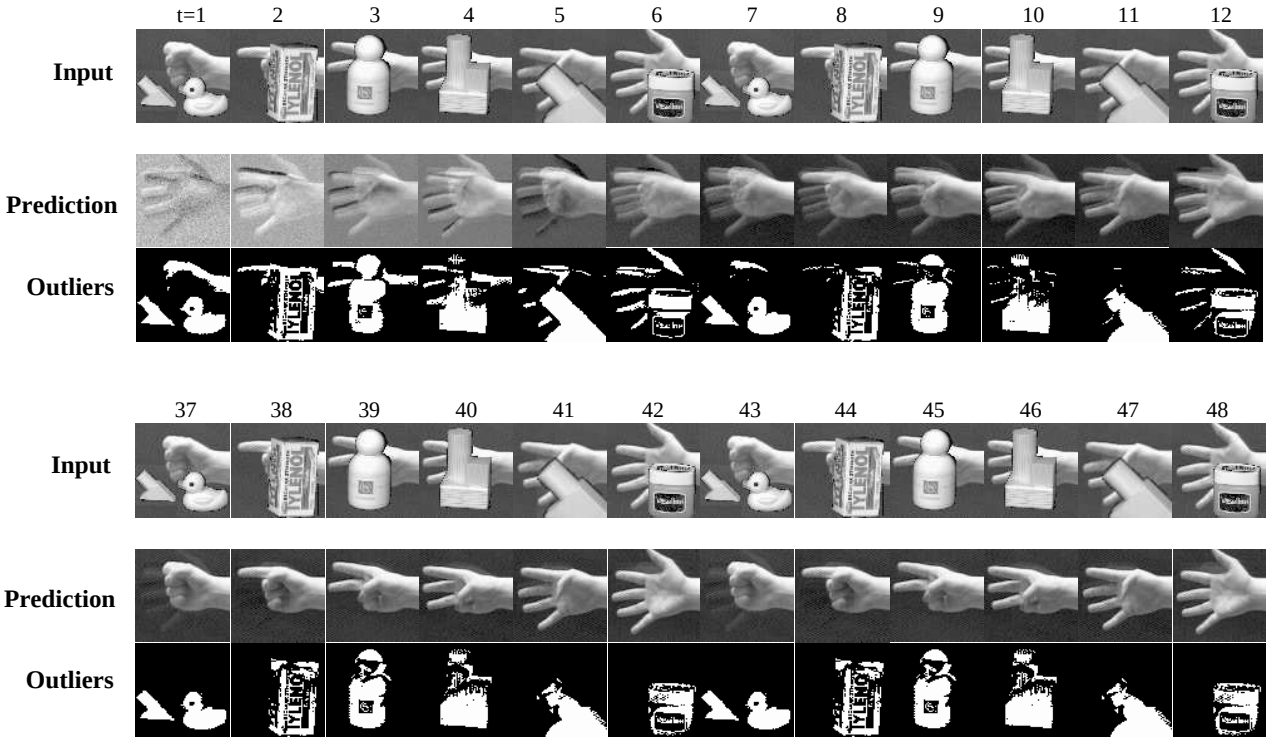


Figure 18: **Robust Segmentation and Recognition of Occluded Gestures**. The filter from Figure 10 that was trained on a cyclic image sequence of hand gestures was tested for robustness in the presence of occlusions and clutter. The top panel shows the initial transient phase of the robust filtering process (after starting with the leftmost occluded input). The bottom panel depicts the steady state behavior of the robust filter, showing relatively accurate predictions of the occluded hand gestures and a corresponding set of segmented outlier objects.

stimuli, the filter interprets the input differently depending on the initial "priming" input. As shown in Figure 17 (b), when the initial input is the first image from the horizontal bar sequence, the image sequence is interpreted as a horizontal bar moving downwards. On the other hand, when the initial priming input is a vertical bar as shown in (c), the filter interprets the sequence as a vertical bar moving rightwards and ignores the extraneous vertical bars by treating them as outliers.

In a second experiment, we tested the trained filter from Figure 10 on a cyclic image sequence of hand gestures with occlusions and clutter. The robustness parameter c was computed at each time instant as the sum of the mean plus 0.3 standard deviations of the current distribution of squared residual errors. The filter was initialized with the first occluded gesture image in the sequence. As shown in Figure 18, the filter exhibits an initial transient phase where the predictions are not completely accurate and the outliers are yet to be detected. However, after a few cycles of exposure to the occluded image sequence, the filter converged to stable estimates of the gesture images as shown in the bottom panel of Figure 18. The occluding objects were also successfully segmented as shown in the last row of images in the figure.

8 Hierarchical Kalman Filters

The Kalman filters derived in the previous sections used a single level stochastic model of the input generation process as defined by the matrices U and V. It is however highly unlikely that such a simple model can adequately describe complex natural phenomena such as the generation of visual images, sounds, and other sensations. Most natural phenomena manifest themselves over a multitude of spatial and temporal scales. For example, the rich class of stochastic processes possessing $1/f^{\beta}$ power spectra exhibit statistical and fractal self-similarities that can be satisfactorily captured only in a multiscale framework [Chou *et al.*, 1994]. Modeling such phenomena at a single spatial and/or temporal resolution generally leads to an incomplete and often incorrect understanding of the observed phenomenon. There has consequently been much recent interest in multiscale signal processing methods. Techniques such as image pyramids [Cantoni and Levialdi, 1986], wavelets [Daubechies, 1992], and scale-space theory [Lindeberg, 1994] have found wide applications in computer vision and image processing.

In this section, we propose a method for learning and using hierarchical internal models of natural dynamic phenomena in the external world. These internal models possess the following properties: (a) each hierarchical level uses the output state of its immediate predecessor as input, with only the lowest level operating directly on the sensory input, and (b) the hierarchical levels operate over progressively larger spatial and temporal contexts, thereby allowing the development of progressively more abstract spatiotemporal representations as one ascends the hierarchy. Such an arrangement allows the important aspects of the input environment to be encoded and interpreted succinctly at multiple spatial and temporal scales. An additional computational advantage of such a hierarchical scheme is the possibility of faster learning and faster convergence to the desired estimates as is often witnessed in multigrid methods for optimization [Hackbusch, 1985].

8.1 Modeling Top-Down Influences

The central issue when defining a hierarchical model is the specification of how "top-down" signals from a higher level influence the state at a lower level. For the current purposes, we shall pursue a simple generalization of the familiar single level model used in the Kalman filter derivations above, bearing in mind that other more complex interactions between higher and lower level signals may also envisioned [Dayan and Hinton, 1996].

Consider the first hierarchical-level. Recall that we used a generative model (measurement equation) of the form (Equation 8):

$$\mathbf{I} = U\mathbf{r} + \mathbf{n} \tag{42}$$

How should a higher level (for example, the second level) influence the state $\mathbf{r}$ at the first level? A convenient answer is for the higher level to use an identical generative model as the lower level, but instead of generating $\mathbf{I}$, the higher level generates a top-down prediction of the current state $\mathbf{r}$

at the lower level. In other words, the "top-down" information $\mathbf{r}_{td} = U^h \mathbf{r}^h$ from the higher-level is assumed to *constrain* the lower level state $\mathbf{r}$ according to:

$$\mathbf{r} = \mathbf{r}_{td}(t) + \mathbf{n}_{td}(t) \tag{43}$$

where $E(\mathbf{n}_{td}(t)) = 0$ and $E[\mathbf{n}_{td}(t)\mathbf{n}_{td}(t)^T] = \Sigma_{td}(t)$.

Note that this equation not only characterizes the top-down influences on the first level state $\mathbf{r}$ but also serves as the measurement equation for the higher level:

$$\mathbf{r} = U^h \mathbf{r}^h(t) + \mathbf{n}(t) \tag{44}$$

where $\mathbf{n}(t) = \mathbf{n}_{td}(t)$. In other words, Equation 43 and Equation 44 are one and the same, although it is used at two different levels with different interpretations. Comparing Equation 44 with the corresponding equation for the first level (Equation 8), we see that the only difference lies in the fact that the observable "input" $\mathbf{I}$ has been replaced by the state $\mathbf{r}$ at the previous level. Thus, for the second level, the covariance Σ is the same as the top-down covariance Σ_{td} at the first level.

Although the above equations concerned only the first two levels, identical equations can be formulated for each pair of adjacent levels for an arbitrary number of levels. An entire set of such equations defines a hierarchical generative (or measurement) model. Together with the state transition equations for each level, these equations form a hierarchical and stochastic model of an observed dynamic process, as determined by the states $\mathbf{r}$, and the matrices U and V at each hierarchical level of the model.

By analogy with Equations 29 and 38, we may define a new robust optimization function that takes into account the top-down information from a higher level:

$$\begin{aligned} J &= (\mathbf{I} - U\mathbf{r})^T S(\mathbf{I} - U\mathbf{r}) + (\mathbf{r} - \mathbf{r}_{td})^T S_{td}(\mathbf{r} - \mathbf{r}_{td}) + (\mathbf{r} - \overline{\mathbf{r}})^T M^{-1}(\mathbf{r} - \overline{\mathbf{r}}) \\ &\quad + (\mathbf{u} - \overline{\mathbf{u}})^T P^{-1}(\mathbf{u} - \overline{\mathbf{u}}) + (\mathbf{v} - \overline{\mathbf{v}})^T Q^{-1}(\mathbf{v} - \overline{\mathbf{v}}) \end{aligned} \tag{45}$$

where S is as defined in Equation 39 and $S_{td} = \Sigma_{td}^{-1}$ is taken to be a diagonal matrix whose elements are determined by the non-linear function:

$$S_{td}^{i,i} = \min\left\{1, c_{td}/(\mathbf{r}^i - \mathbf{r}_{td}^i)^2\right\} \tag{46}$$

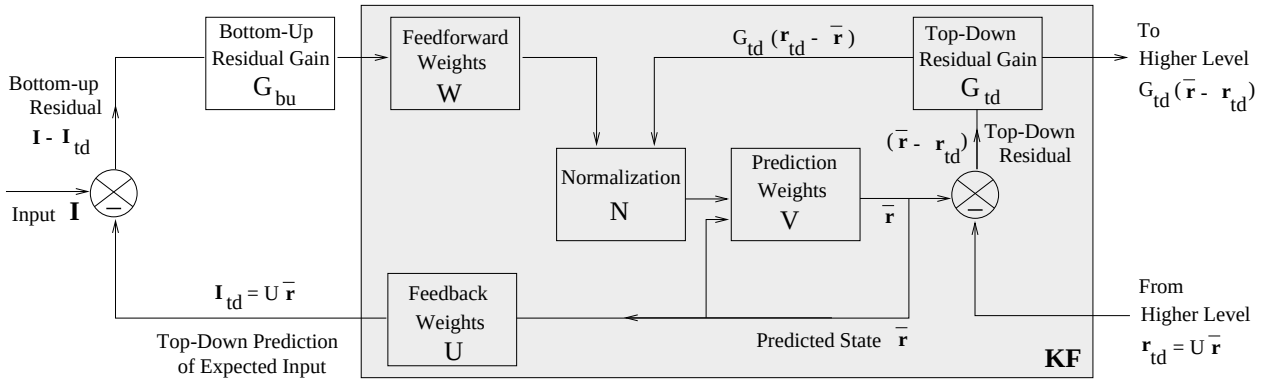
where c_{td} is a threshold parameter that prevents high top-down residuals (outliers) from influencing the optimization process.

8.2 Robust Hierarchical Kalman Filtering

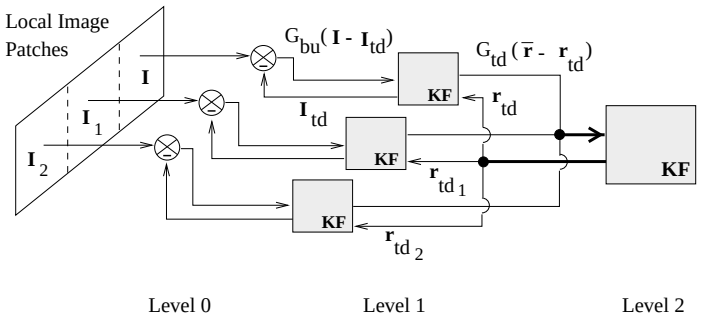
The robust hierarchical Kalman filter then assumes the form:

$$\widehat{\mathbf{r}}(t) = \overline{\mathbf{r}}(t) + N(t)U^T G(t)(\mathbf{I} - U\overline{\mathbf{r}}(t)) + N(t)G_{td}(t)(\mathbf{r}_{td}(t) - \overline{\mathbf{r}}(t)) \tag{47}$$

$$N(t) = (U^T G(t) U + G_{td}(t) + M^{-1}(t))^{-1} \tag{48}$$



(a)



(b)

Figure 19: **Schematic Diagram of a Robust Hierarchical Kalman Filter**. (a) Basic Kalman filter (KF) module (shaded box), showing two inputs (feedforward residual from lower level and feedback prediction from higher level) and two outputs (feedback prediction to lower level and feedforward residual to higher level). (b) Three-level hierarchical Kalman filter, composed of 3 level 1 KF modules, analyzing 3 local image regions and providing input to a single level 2 module.

where $\overline{\mathbf{r}}$, M and $G(t)$ are as defined in Section 7, and $G_{td}(t)$ is an $k \times k$ diagonal matrix whose entries at time instant t are given by:

$$G_{td}^{i,i} = \begin{cases} 0 & \text{if } (\overline{\mathbf{r}}^i(t) - \overline{\mathbf{r}}_{td}^i)^2 > c_{td}(t) \\ 1 & \text{otherwise} \end{cases}$$

The vector $\overline{\mathbf{r}}_{td}(t)$ is simply the top-down prediction $U^h\overline{\mathbf{r}}^h$ from the higher level at time t. Just as $G(t)$ filters out any high bottom-up residuals, $G_{td}(t)$ prunes away the high top-down residuals, thereby enabling the filter to robustly estimate the state $\mathbf{r}$ using both bottom-up and top-down information. It should also be noted that although the generative model and the dynamics at each level are linear, the coupling between any two levels at any time instant is a *non-linear* function of the current residuals. As a result, the hierarchy cannot be collapsed into a single level dynamic system. Figure 19 depicts the hierarchical filter and its various components.

8.3 Spatial Hierarchies

In the case of visual recognition, it may be computationally more efficient to allow the lowest level to analyze and predict only small image patches, allowing the next level to analyze and predict the states of a group of adjacent lower level modules. Such an arrangement becomes especially relevant when modeling the visual cortex, wherein the limited dendritic spread of a neuron allows only a small patch of the visual field to serve as input to a given neuron, and therefore, larger patches are analyzed only by a higher level receiving convergent inputs from a group of lower level neurons. This results in the appearance of larger neuronal receptive fields as one ascends the visual hierarchy [Van Essen, 1985; Maunsell and Newsome, 1987; Desimone and Ungerleider, 1989].

In the case of the hierarchical Kalman filter, analysis and prediction at increasing spatial scales can be achieved by allowing the higher level to estimate and predict the augmented vector formed by concatenating the state vectors $\overline{\mathbf{r}}$ of a set of adjacent modules at the lower level. In other words, we use $\mathbf{r}_{td} = U^h_{i:i+k-1}\mathbf{r}^h$, where $U^h_{i:i+k-1}$ represents the k rows from i through $i+k-1$ of the higher level generative matrix U^h (recall that the lower level state $\mathbf{r}$ is a $k \times 1$ vector and so is $\mathbf{r}_{td}$). Thus, to summarize, $\mathbf{r}^h$ represents the higher level state that generates a long vector given by $U^h\mathbf{r}^h$; this vector is split into smaller vectors $U^h_{i:i+k-1}\mathbf{r}^h$, which act as top-down constraints $\mathbf{r}_{td}$ on the various states $\mathbf{r}$ at the lower level.

Given the above arrangement, successively higher levels in the hierarchical filter analyze and predict over (exponentially) larger spatial extents. The highest level then has access to spatial information from the entire image, albeit in an abstract form, after having been processed by the lower levels. It can in turn influence processing at the lower levels via its generative (feedback) connections defined by the various matrices U at the different levels.

8.4 Temporal Hierarchies, Penalty Terms, and Temporal Decay

Just as successive levels in the hierarchical Kalman filter can be made to operate over successively larger spatial extents, it is possible to arrange for successive levels of the filter to predict lower level states based on successively longer temporal extents. The underlying idea is to include a *penalty term* in the optimization function J for each of the parameters $\mathbf{r}$, U and V. These penalty terms can be justified in a wide variety of settings:

- *Regularization Theory* [Poggio *et al.*, 1985; Girosi *et al.*, 1995]: Here, the penalty term is used to prevent the estimator from overfitting its estimate to the input data, thereby allowing the estimator to retain an ability to *generalize* to novel data that differs in some ways from the input data used during training. See [Girosi *et al.*, 1995] and references therein for further details regarding regularization theory.

- *Minimum Description Length (MDL) principle* [Rissanen, 1989; Zemel, 1994]: This is a formal information-theoretic formulation of the well-known Occam's Razor principle: "Given the choice between a set of possible explanations, pick the simplest one." Simply put, the MDL principle advocates balancing the cost of encoding the data given the use of a model with the cost of specifying the model itself. Cost of a discrete event x is defined in terms of the length of its encoding in bits, which is essentially $-\log P(x)$ (this gives us another interpretation of the $-\log P$ terms in J). The penalty terms for model parameters arise as a consequence of the calculation of the probability mass of the model parameters for use in their respective $-\log P$ based encoding terms in J (see [Rao and Ballard, 1996a] for details).

- *Seeking Higher-Order Statistical Correlations*: As proposed by [Olshausen and Field, 1996] and [Harpur and Prager, 1996], the addition of certain nonlinear penalty terms allows an estimator to seek more than pairwise correlations in the input data.

With the addition of penalty terms for the parameters $\mathbf{r}$, $\mathbf{u}$, and $\mathbf{v}$, the new optimization function becomes:

$$J_{new} = J + f(\mathbf{r}) + g(\mathbf{u}) + h(\mathbf{v}) \tag{49}$$

where J is the function from Equation 45 above and f, g, and h are nonlinear penalty functions. For example, some previously used penalty functions include $f(x) = g(x) = h(x) = \alpha x^2$ [Rao and Ballard, 1996a], $f(x) = \alpha \log(1+x^2)$ [Olshausen and Field, 1996], and $f(x) = \alpha |x|^{1/r}$ [Harpur and Prager, 1996] where the functions are applied to all components x of a given vector $\mathbf{x}$ and the results are summed in the optimization function. The resulting hierarchical Kalman filter that minimizes J_{new} is given by:

$$\widehat{\mathbf{r}}(t) \;=\; \overline{\mathbf{r}}(t) + N(t)U^T G(t)(\mathbf{I} - U\overline{\mathbf{r}}(t)) + N(t)G_{td}(t)(\mathbf{r}_{td}(t) - \overline{\mathbf{r}}(t)) - N(t)f'(\mathbf{r}(t)) \tag{50}$$

Note the addition of the *temporal decay* term $-N(t)f'(\mathbf{r}(t))$ in the Kalman filter due to the penalty term. Similar decays are introduced in the equations for $\mathbf{u}$ and $\mathbf{v}$.

By decreasing exponentially the decay function f for each successive level (for example, by decreasing the constant α exponentially at each successive level), one can make the higher level neurons decay at a slower rate than their level counterparts. As a result, one obtains a temporal hierarchy wherein higher levels have a longer time constant and therefore predict lower level states based on a larger temporal context. In other words, the lowest level modules possess the shortest "memories" while the higher levels use longer historical traces to make their predictions, taking into account events that occurred progressively further back in time.

8.5 Hierarchical Recognition Example

The hierarchical framework was experimentally tested using two simple sequences consisting of alternating facial expressions from four different persons (Figure 20). The image sequences were learned using the three-level hierarchical network depicted in Figure 19. At level 0, the 64×64 image was partitioned into four equal 32×32 sub-images. These four sub-images served as input to the four level 1 Kalman filter modules as shown in Figure 20 (b). The level 1 matrices U and V in each module were of size 1024×5 and 5×5 respectively. A single level 2 module was used to estimate the states of the four lower level modules. The level 2 matrix U was of size 20×5 and the matrix V was 5×5. All matrices were initialized to small random values before training the hierarchical filter on the image sequence.

Figure 20 shows the level 1 basis images (columns of the four level 1 U matrices) obtained after training two hierarchical filters on the two facial image sequences. The panel below illustrates the performance of the filters when confronted with noisy, incomplete, and occluded versions of the training image sequences. As shown in the figure, the filters were able to correctly predict and track the two sequences, once the outliers were detected and discounted for within the first four or five frames of the image sequence.

9 Neural Implementation in the Visual Cortex

The mammalian neocortex possesses a remarkable regularity in its neuroanatomical structure, as seen in the pattern of connections within and between different cortical areas [Creutzfeldt, 1977; Barlow, 1985]. For example, a ubiquitous feature of cortico-cortical connectivity is the reciprocity of connections between various cortical areas: if area A projects to area B, then area B almost invariably projects to area A [Rockland and Pandya, 1979; Felleman and Van Essen, 1991]. These connections typically terminate and originate in distinct cortical laminae depending on their source and destination, which allows one to designate one cortical area as being at a higher or lower hierarchical level with respect to another. For example, feedforward connections from a "lower" area generally originate in layer 2+3 (composed of layers 2 and 3) and terminate in layer 4 of a "higher" area, while

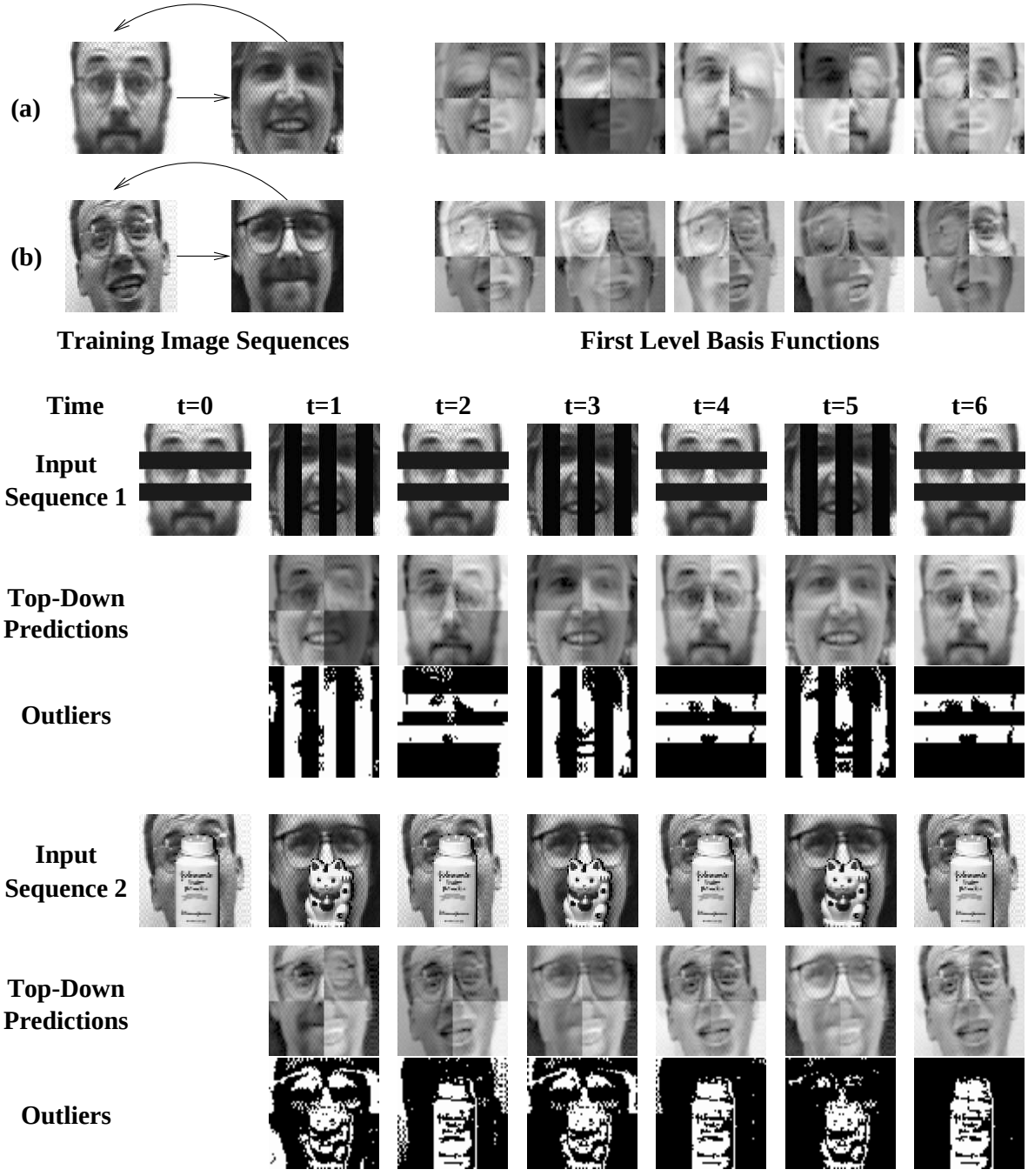
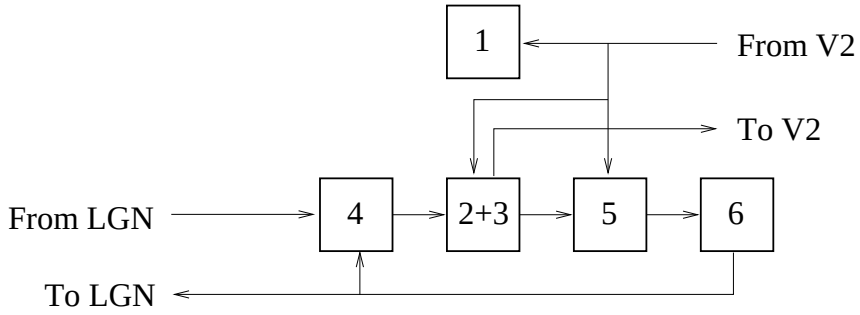


Figure 20: **Dynamic Recognition using Hierarchical Kalman Filters**. (a) and (b) show the two cyclic training image sequences and their corresponding basis images at level 1 of the hierarchical Kalman filter (for each sequence, the five columns of the four first level U matrices are shown as five composite images). A hierarchical architecture as shown in Figure 19 was used. (Below) Robust prediction and tracking of the two sequences in the presence of occlusions.



Interlaminar Connections in Primary Visual Cortex

Figure 21: **Laminar Connections in Primate V1** (after [Bolz *et al.*, 1989; Lund, 1981; Van Essen, 1985]). Some inter-laminar and cortical projections have been omitted to simplify the diagram. The horizontal connections among local neurons are not shown, although these may play an important role in mediating some of the local interactions due to the covariance terms in the Kalman filter. See text for details.

the feedback connections from the higher area explicitly avoid layer 4, terminating predominantly in layers 1, 2+3, 5 and 6 of the lower area. There also exist regularities in the intracortical connections between the different layers, as typified by the pattern of connections shown in Figure 21. In this section, we will focus on the visual cortex, and attempt to interpret its neuronal circuitry and neuroanatomical organization in terms of the hierarchical Kalman filter model.

The visual cortex has been previously characterized as a roughly hierarchical network composed of many distinct interconnected areas [Van Essen and Maunsell, 1983; Felleman and Van Essen, 1991]. This hierarchical characterization is based on the laminar patterns of origins and terminations of the connections between the different visual cortical areas. This hierarchical structure, together with the reciprocity of connections between areas and the distinctive laminar connections within a given area, make the visual cortex especially well-suited to implement a hierarchical Kalman filter-like prediction mechanism. For instance, the feedback connections from a higher area may carry the predictions $U\overline{\mathbf{r}}$ of lower level neural activities $\mathbf{I}$, while the feedforward connections may convey to the higher level the differences or *residuals* $(\mathbf{I} - U\overline{\mathbf{r}})$ between the predictions and the actual lower level activities. These residuals would allow the visual cortex to compute robust optimal estimates of visual events occurring in the distal environment, thereby enabling it to recognize these events and predict future events based on a hierarchical and distributed internal model of the visual environment. The internal model, as encoded by the Kalman filter parameters U and V, could be learned and refined by the organism during periods of exposure to the visual environment. Similar ideas have been suggested by a number of other authors in a variety of contexts [MacKay, 1956; Grossberg, 1976; Barlow, 1985; Harth *et al.*, 1987; Albus, 1991; Mumford, 1992; Pentland, 1992; Kawato *et al.*, 1993; Hinton *et al.*, 1995; Dayan *et al.*, 1995; Softky, 1996].

9.1 Neural Network Implementation of the Kalman Filter

To see how a Kalman filter-like prediction mechanism may be implemented by cortical neurons, first note that the basic operation required by the Kalman filter is the multiplication $A\mathbf{v}$ of a matrix A with a vector $\mathbf{v}$. In the standard neural implementation of this type of an operation [Churchland and Sejnowski, 1992], the matrix A represents the synaptic strength of neurons (each row represents the synapses of one neuron) and the components of the vector $\mathbf{v}$ denote the pre-synaptic inputs to these neurons. Each neuron computes a weighted sum of its pre-synaptic inputs according to the weights encoded by its synapses.

For example, consider the calculation of the top-down prediction signal $U\overline{\mathbf{r}}$ using a set of n neurons (the design can be applied to other components of the filter). The synapses of the ith neuron encode the values in the ith row of the matrix U. The output o_i of this neuron, given the pre-synaptic activity $\overline{\mathbf{r}}$, is the weighted sum:

$$o_i = \sum_{j=1}^{k} U_{ij}\overline{r}_j \tag{51}$$

This output can, for instance, be encoded as the spike rate of the neuron, although more complex schemes employing timing information may also be used [Abeles, 1991]. The axonal output vector $U\overline{\mathbf{r}}$ in this case forms the Kalman filter's prediction of the next expected input $\mathbf{I}$ at the lower level. The neurons representing U can therefore be regarded as "feedback" neurons, translating the signal $\overline{\mathbf{r}}$ at a higher area to the lower abstraction level using the values in U. The subsequent residual error $(\mathbf{I} - U\overline{\mathbf{r}})$ that is required by the filter can be generated by feedback-mediated *inhibition* of the input $\mathbf{I}$ as shown in Figure 22 (a). This residual error signal is then successively processed by neurons representing the "bottom-up" gain matrix G, the "feedforward" matrix $W = U^T$, and the "normalization" matrix N of the Kalman filter. Recall that these three matrices together comprise the Kalman gain matrix $K = NU^TG$ that is multiplied with the incoming sensory residual signal. From the Kalman filter equation 40, we see that this weighted residual signal needs to be added to the prior estimate $\overline{\mathbf{r}}$. This recursive summation operation can be readily implemented by a set of integrate-and-fire neurons, performing this summation over time. The resulting output is processed by the prediction neurons V, which generate the next predicted state $\overline{\mathbf{r}}$ as their axonal output.

The preceding sequence of neural operations amounts to implementing the complete Kalman filter equation, combining equations 40 and 41:

$$\overline{\mathbf{r}}(t) = V\left(\overline{\mathbf{r}}(t-1) + NWG(\mathbf{I}(t-1) - U\overline{\mathbf{r}}(t-1))\right) + \overline{\mathbf{n}}(t-1) \tag{52}$$

The predicted state vector $\overline{\mathbf{r}}$ is then transmitted to the lower level by the feedback neurons, whose synapses U transform the higher level state $\overline{\mathbf{r}}(t)$ into the lower level signal $U\overline{\mathbf{r}}(t)$. This signal, which is at the same abstraction level as the lower level inputs, is then used to inhibit the next set of incoming signals $\mathbf{I}(t)$.

The neural implementation of the interactions with the top-down signals $\mathbf{r}_{td}$ is similar. These predictive signals generate the top-down residual error $(\overline{\mathbf{r}} - \mathbf{r}_{td})$ via inhibition of the signals $\overline{\mathbf{r}}$. This residual is then successively processed by neurons representing the "top-down" gain matrix G_{td} and the normalization matrix N (see Figure 22 (a)). Recall from Equation 47 that the neurons representing the normalization matrix N also integrate the bottom-up weighted residual $WG(\mathbf{I} - U\overline{\mathbf{r}}(t-1))$. The output of the neurons representing N is then used to correct the current state estimate and generate the next state prediction $\overline{\mathbf{r}}(t)$ as in Equation 52 above.

The neurons representing the gain matrices G and G_{td} may be regarded as *gating* neurons [Jordan and Jacobs, 1994], which compute the values in G or G_{td} based on their inputs (the residuals). These outputs then serve to inhibit the incoming residual signals according to whether these signals represent outliers or not. Although we only explored binary gain matrices in Section 7, one may alternatively use other nonlinear functions that gradually attenuate the incoming residuals rather than using a hard threshold as we did in Section 7. Another interesting alternative is to use non-diagonal gain matrices S and S_{td}. For example, a natural generalization of Equation 39 is:

$$S^{i,j} = \min\left\{1, c/|(\mathbf{I}^i - U^i\mathbf{r})(\mathbf{I}^j - U^j\mathbf{r})|\right\} \tag{53}$$

where $|\cdot|$ denotes absolute value. Such a function would allow neighboring components of the residual vectors $(\overline{\mathbf{r}} - \mathbf{r}_{td})$ and $(\mathbf{I} - U\overline{\mathbf{r}})$ to play an active role in the estimation of each component of $\widehat{\mathbf{r}}$, thereby determining the degree of *lateral interactions* between neighboring neurons in the Kalman filter network.

9.2 Laminar Implementation in the Visual Cortex

Figure 22 shows how a neural implementation of the different components of a hierarchical Kalman filter can fit comfortably within the laminar structure of a given cortical area. For the sake of concreteness, we consider here the case of primary visual cortex (or V1). The model assumes that the feedback connections from V1 to V1's thalamic input site, the dorsal lateral geniculate nucleus (dLGN), convey predictions of the expected activities of dLGN relay neurons. These predictive signals (denoted by $U\overline{\mathbf{r}}$ in Figure 22) are assumed to inhibit, via inhibitory interneurons, the activities of their corresponding dLGN neurons to the extent that the predictions match the afferent retinal signals. Support for such an operation comes from the work of Murphy and Sillito [Murphy and Sillito, 1987] indicating that corticofugal feedback engages inhibitory mechanisms in the dLGN. Such an assumption is also supported by neuroanatomical evidence that feedback connections from the cortex contact inhibitory interneurons in the dLGN [Orban, 1992]. The resulting *residual* activities of the dLGN neurons are assumed to be conveyed to the cortex via the feedforward thalamo-cortical pathway.

On their way to the cortex, the residuals are modulated by a "bottom-up" gain (analogous to the gain G in the Kalman filter) which, for example, could be implemented by the inhibitory neurons of

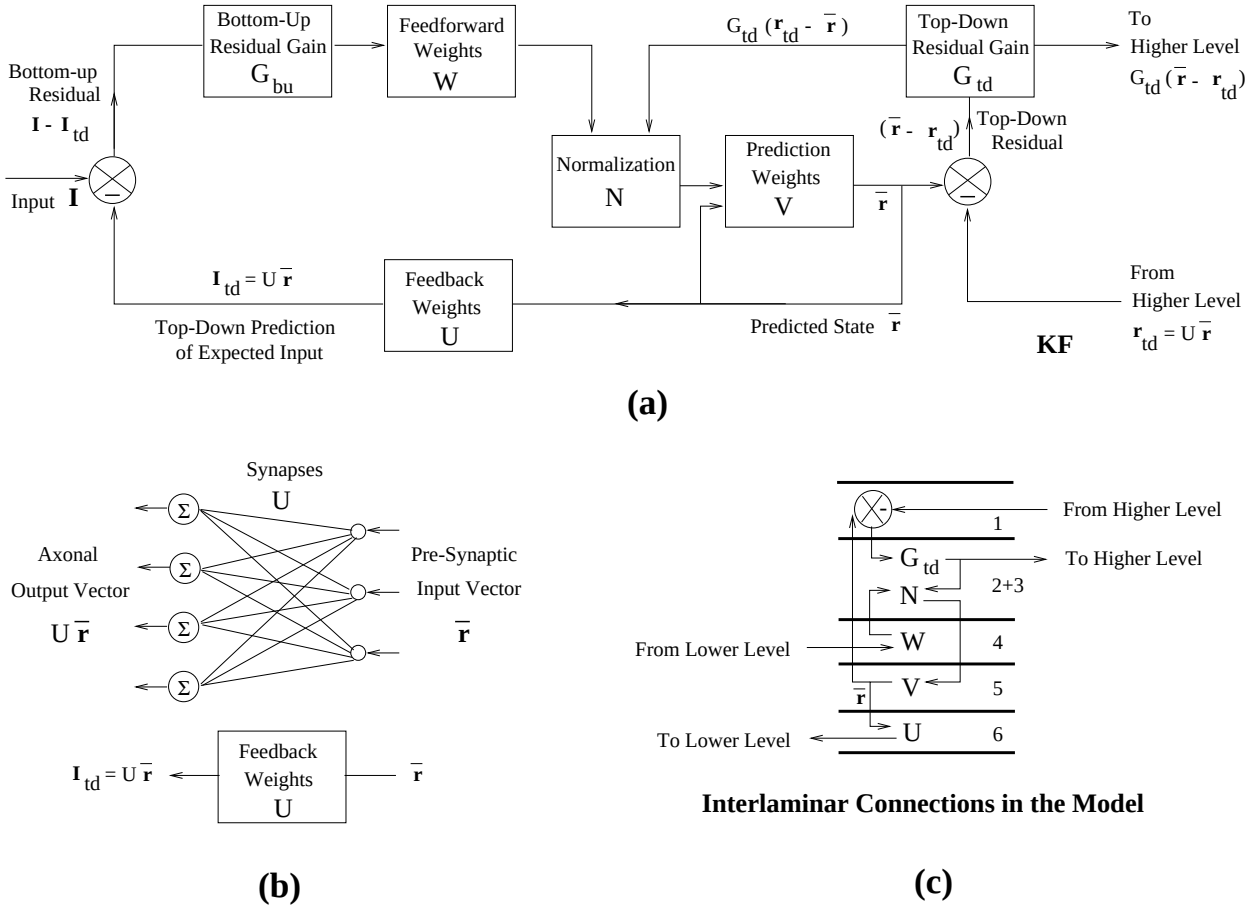


Figure 22: **Laminar Implementation of the Hierarchical Kalman filter model**. (a) shows the schematic diagram of the basic hierarchical Kalman filter module. (b) illustrates the neural implementation of the primitive operation required by the Kalman filter, namely, the multiplication of a matrix and a vector. The figure depicts the network for computing the feedback signals but the same design is applicable to all components of the filter. (c) shows a laminar implementation of the basic Kalman filter module in (a). This implementation conforms to many of the known neuroanatomical connections between laminae in primate V1 (compare with Figure 21).

the thalamic reticular complex (including the perigeniculate nucleus) [Crick, 1984]. The modulated residual signal is then linearly filtered through the synapses of layer 4 cells in V1, which would correspond to the feedforward weights $W = U^T$ in the Kalman filter (see Figure 22). The signal subsequently undergoes a normalization (corresponding to the matrix N in the Kalman filter). A plausible site for this normalization is layer 2+3 (composed of layers 2 and 3), given that this layer receives a substantial projection from layer 4 [Gilbert and Wiesel, 1981; Bolz *et al.*, 1989]. Normalization of responses has also been proposed by other authors to explain saturation of neural responses in V1 at high stimulus contrasts [Heeger *et al.*, 1996]. It has been suggested that such normalization of responses may occur as a result of biophysical membrane properties of neurons in the visual cortex [Carandini and Heeger, 1994]. The preceding sequence of neural processing suffices to compute the weighted Kalman residual $NWG(\mathbf{I} - U\overline{\mathbf{r}})$, which is used to correct the previous state estimate $\overline{\mathbf{r}}(t)$ at time instant t (see Equation 52). The corrected estimate then generates the next state

prediction $\overline{\mathbf{r}}(t+1)$ via the synapses V. Neurons in layer 5 appear to be ideally located for such a computation, given that layer 2+3 cells project extensively into layer 5 [Gilbert and Wiesel, 1981; Bolz *et al.*, 1989]. Furthermore, layer 5 projects to layer 6, which is known to be a major source of cortico-thalamic feedback [Gilbert and Wiesel, 1981]. Layer 6 neurons could therefore synaptically encode the feedback weights U of the Kalman filter and convey the feedback signal $U\overline{\mathbf{r}}(t+1)$ to the dLGN for predictive inhibition of the next input $\mathbf{I}(t+1)$.

A final issue is the role of cortico-cortical feedback from a higher visual area, for example, V2. As in the case of cortico-thalamic feedback, the model assumes that the higher area (V2) conveys predictions of the state at the lower area (V1). Just as in the bottom-up case, the predictive signals from V2 are assumed to generate residuals $(\overline{\mathbf{r}} - \mathbf{r}_{td})$ via interactions with inhibitory interneurons in the superficial layers. These residual activities are modulated by gating neurons in layer 2+3, which represent the top-down gain matrix G_{td}. The modulated top-down residuals $G_{td}(\overline{\mathbf{r}} - \mathbf{r}_{td})$ are assumed to be conveyed to two sites: (a) via axon collaterals to the layer 2+3 "normalization" neurons N, which integrate these top-down residuals with the bottom-up residuals $WG(\mathbf{I} - U\overline{\mathbf{r}})$ from layer 4, and (b) to the higher visual area (V2), where this signal serves as the bottom-up weighted residual for the Kalman filter at the higher level (see Figure 19 (b) and 22 (c)).

The above interpretation of cortical circuitry is partially supported by some recent experimental data on the role of feedback in mediating certain neurophysiological effects in the visual cortex. For instance, "complex" cells in layer 2+3 of the primary visual cortex have been known to exhibit highly nonlinear "extra-classical" effects due to stimuli that lie beyond the classical receptive field of the cell. A classical example is the phenomenon of endstopping, wherein the response of a neuron to an oriented bar falls off drastically as the bar is extended beyond the neuron's receptive field. One explanation of this highly nonlinear response suppression attributes the attenuation in response to lateral inhibition from neighboring cells. However, recent experiments by Zipser *et al.* [Zipser *et al.*, 1996] indicate that these extra-classical effects in layer 2+3 are seen only 80-100 milliseconds after stimulus onset. This is consistent with the idea of higher level feedback mediating these effects and inconsistent with a purely feedforward explanation such as lateral inhibition, since one would then expect these effects to appear almost simultaneously with stimulus onset. In the Kalman filter model, note that responses of the layer 2+3 neurons represent the weighted residual signal $G_{td}(\overline{\mathbf{r}} - \mathbf{r}_{td})$. Note that neurons at the higher level (V2) predict based on a larger spatial extent than a neuron at the lower level (V1), and these prediction get more accurate as the stimulus is extended beyond the classical receptive field of the lower level neuron up to the size of the higher level receptive field. Thus, the response of a layer 2+3 neuron at the lower level gets effectively suppressed due to diminishing residuals $(\overline{\mathbf{r}} - \mathbf{r}_{td})$ caused by better predictions $\mathbf{r}_{td}$ from the higher level of the lower level state $\overline{\mathbf{r}}$, as stimulus size is incrementally increased up to the size of the larger RF of the higher level neurons. Simulation results using a hierarchical Kalman filter network trained on natural images support this

interpretation of extra-classical effects in the visual cortex [Rao and Ballard, 1996b; 1996a]. In addition, preliminary neurophysiological results from experiments involving the inactivation of V2 also seem to indicate that feedback from a higher area plays a crucial role in mediating extra-classical effects in a lower area such as V1 [James *et al.*, 1995]. Further experiments are however needed to rigorously confirm many of the other functional interpretations of cortical circuitry suggested by the Kalman filter model.

10 Conclusions

Based on the assumption that the cortex builds and maintains an internal world model of its external environment, this paper derived a mathematically rigorous model of cortical function resting firmly on the well-established statistical principle of Kalman filtering. The cortex was viewed as a hierarchical predictor and adaptive estimator of the hidden internal state of the input environment. Update rules for prediction, robust estimation, and learning of internal models were derived from first principles using an optimization function based on maximizing the posterior probabilities of model parameters given the observed data. The model was experimentally tested using a variety of $3D$ objects in scenarios containing varying amounts of occlusion, clutter, and noise. A neural implementation of the model was described, and it was shown that the various components of the model can fit comfortably within the known neuroanatomical structure of the cortex. The flow of information between these components was also shown to be consistent with the pattern of inter-laminar connections in the visual cortex.

One obvious shortcoming of the model is the assumption of linearity when modeling the measurement and state transition processes (Section 3). Indeed, this is the primary weakness of the standard Kalman filter. It is therefore not surprising that non-linear alternatives such as the extended Kalman filter have been proposed [Maybeck, 1979]. The model presented here readily generalizes to the extended Kalman filter case, where the prediction step (Equation 19) can be made non-linear [Rao and Ballard, 1996a]. Unfortunately, the introduction of nonlinearities often complicates the corresponding estimation process, forcing the use of approximations (such as Taylor series based approximations) to make the mathematical derivations tractable. As a result, many important properties such as optimality and stability may be lost. In this paper, we explored the use of some limited forms of nonlinearities, such as making the covariance matrices nonlinear functions of the prediction errors in order to facilitate robust estimation, and adding nonlinear decay terms to the Kalman filter dynamics. These limited forms of nonlinearities may help ameliorate some of the weaknesses of the standard Kalman filter, while at the same time retaining its favorable properties. In the context of biological modeling, it has been argued that some cortical neurons may very well operate linearly in much of their dynamic range [Kohonen, 1988; Ferster, 1994]. Kohonen, in particular, argues that strong nonlinearities at the single neuron level are generally seen more often in evolutionarily older (subcortical) structures than the more recently

evolved neocortex. As explained in Section 9, many of the complex neural responses that appear nonlinear when a neuron is viewed in isolation can be explained within the hierarchical Kalman filter model as occurring due to feedback inhibition and other limited forms of local nonlinear interactions. Finally, we note that single units in the current model code both positive and negative quantities, whereas the cortex may employ multiple neurons for this purpose, thereby causing the appearance of a non-linearity. Such a sign-related nonlinearity may underlie cortical response properties such as direction-selectivity. Simulations are currently underway to verify the feasibility of this more detailed neuron-level model of the cortex.

Another possible limitation of the model is the assumption of Gaussian probability distributions when modeling the state and noise processes. Although such an assumption is partially supported by the Central Limit Theorem from statistics [Feller, 1968], the unimodality of the Gaussian distribution does not allow a simultaneous representation of multiple object hypotheses such as in a cluttered scene [Isard and Blake, 1996]. However, this limitation is handled in the present model in two ways: (a) the hierarchical version of the Kalman filter allows multiple hypotheses to reside concurrently in a distributed manner at various hierarchical levels, and (b) the robust version of the Kalman filter allows objects other than the primary one to be treated as outliers. These outliers can however be subsequently recognized in a sequential fashion as demonstrated in Section 7. A related limitation is the restriction to modeling (first-order) Markov processes, in which the next state is assumed to depend only on the preceding state (Equation 9). This is not as serious a limitation as it seems because (a) any finite-order Markov process, where the next state depends on a finite number of past states, can be represented as a first-order Markov process [Bryson and Ho, 1975], and (b) since the matrices U and V are not fixed but can be adapted according to the input stimuli, many processes that may initially appear to be non-Markov can nevertheless be handled by the adaptive filter by finding appropriate U and V such that the resulting states disambiguate any aliasing in the input stream. A simple example of this adaptive search process in the context of an apparently non-Markovian input stream was given in Section 6.7.

An important problem that we have not addressed in this paper is the issue of transformation invariance: how can the state estimates calculated by the hierarchical Kalman filter ("What") be made invariant to object transformations ("Where") such as translations, rotations, and scale? A simple but attractive solution is to model the transformed image $\mathbf{I}(\mathbf{x})$ as a function of the original image $\mathbf{I}(\mathbf{0})$, where $\mathbf{x}$ is a vector denoting a distributed representation of the relative transformation ("Where") with respect to the original image. In particular, one can expand the transformed image $\mathbf{I}(\mathbf{x})$ in a Taylor series about an original reference point $\mathbf{0}$:

$$\mathbf{I}(\mathbf{x}) \;=\; \mathbf{I}(\mathbf{0}) + \frac{\partial \mathbf{I}(\mathbf{0})}{\partial \mathbf{x}}\mathbf{x} + \text{higher order terms} \tag{54}$$

For small transformations, the higher order terms can be safely ignored and their effect can be

modeled as stochastic noise:

$$\Delta \mathbf{I}(t) \;=\; \frac{\partial \mathbf{I}(\mathbf{0})}{\partial \mathbf{x}} \mathbf{x}(t) + \mathbf{n}(t) \tag{55}$$

where $\Delta \mathbf{I}(t) = \mathbf{I}(\mathbf{x}) - \mathbf{I}(\mathbf{0})$ and $\mathbf{n}$ is assumed to be a Gaussian noise process. The *Jacobian* matrix $\frac{\partial \mathbf{I}(\mathbf{0})}{\partial \mathbf{x}}$ can be approximated using a $n \times nk$ matrix D as:

$$\frac{\partial \mathbf{I}(\mathbf{0})}{\partial \mathbf{x}} \;=\; D\mathcal{I}_0 \tag{56}$$

where $\mathcal{I}_0$ is the $nk \times k$ matrix:

$$\mathcal{I}_0 = \begin{bmatrix} \mathbf{I}(\mathbf{0}) & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}(\mathbf{0}) & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{I}(\mathbf{0}) \end{bmatrix} \tag{57}$$

Thus, we obtain the following Kalman filter measurement (or generative) model for estimating the transformation state $\mathbf{x}$:

$$\Delta \mathbf{I}(t) \;=\; D\mathcal{I}_0 \mathbf{x}(t) + \mathbf{n}(t) \tag{58}$$

It is a straightforward exercise to formulate an optimization function based on the above equation and derive a Kalman filter update rule for the current transformation state $\mathbf{x}$ and a learning rule for the matrix D [Rao and Ballard, 1997]. Note that these rules require the original image $\mathbf{I}(\mathbf{0})$, which is the top-down predicted image representing the *current object hypothesis* ("What"), to be supplied by the object estimation network maintaining the current object state $\mathbf{r}$. Thus, the model for invariant recognition consists of two cooperating networks, one that estimates object identity $\mathbf{r}$ as given by the reference image $\mathbf{I}(\mathbf{0})$ and another that estimates the relative transformation $\mathbf{x}$. An especially favorable property of such an arrangement is that the estimate of object identity remains stable in the first network as the second network attempts to account for any transformations being induced in the image plane, appropriately conveying the type of transformation being induced in its estimate for $\mathbf{x}$. Another favorable property is that the transformation estimates $\mathbf{x}$ remain the same even when different objects are being transformed in an identical manner. This independence and decoupling of the transformation estimates $\mathbf{x}$ from object estimates $\mathbf{r}$ is crucial for learning general sensory-motor routines that can be uniformly applied across objects (for example, grasping a cup) without regard to object specific features (for example, texture or color of the cup) that are irrelevant to motor programming. Finally, the problem of large image transformations can be handled using a hierarchical estimation framework involving cooperative estimation of object state and relative transformation at multiple scales, similar to the hierarchical scheme described in Section 8. Interestingly, this computational dichotomy between the estimation of "what" and "where" parameters resembles

the well-known segregation between the ventral occipitotemporal and the dorsal occipitoparietal pathways observed in the primate neocortex [Ungerleider and Mishkin, 1982; Mishkin *et al.*, 1983; Van Essen and Maunsell, 1983; Van Essen, 1985].

Given that the cortex possesses roughly the same neuroanatomical input-output structure and pattern of connections across many different cortical areas [Creutzfeldt, 1977; Barlow, 1985; Pandya *et al.*, 1988], a crucial test for any putative model of the cortex is whether it is general enough to be uniformly applicable to different cortical areas without regard to the specific input modality. Although this paper was primarily concerned with visual recognition and estimation, it is not hard to see that the basic computational mechanisms underlying the Kalman filter model do not place any restrictions on the type of input signals being estimated. The model can thus be readily applied to other modalities such as audition and olfaction. On the other hand, the adaptive nature of the filter endows it with the ability to tune itself to the specific idiosyncrasies of the input modality. We have already seen how the same Kalman filter circuitry that was used for object state estimation can also be employed to compute transformation estimates, a function that is generally associated with the parietal cortex. One may also envision a hierarchical Kalman filter based architecture for the recognition of speech and other auditory signals, wherein successive levels of the hierarchy recognize signals spanning successively larger temporal intervals, such as phonemes, words and sentences. Such an architecture may allow modeling of the auditory system comprising of the medial geniculate nucleus (MGN) and the hierarchically organized areas of the primate auditory cortex (A1, A2, and related areas in the superior temporal gyrus) [Pandya *et al.*, 1988; Newman, 1988]. Similarly, one might attempt to view the motor cortical hierarchy, comprising the primary motor cortex (M1), supplementary motor area (SMA), premotor cortex, and related regions in prefrontal cortex [Pandya *et al.*, 1988; Felleman and Van Essen, 1991], as implementing a hierarchical prediction and estimation scheme that generates appropriate motor signals for intended body movements and receives as inputs the various afferent feedback signals from muscle spindles, Golgi tendon organs, and other motor receptors from corresponding parts of the body. Successive levels in this motor hierarchy would represent successively more abstract specifications of intended motor movements, which would presumably be planned at the highest levels in response to current sensory state estimates in conjunction with various emotion-based internal goals, motivations, and aspirations. Although computationally quite attractive, such a Kalman filter based integrated view of the cortex must necessarily remain speculative, until further experiments and simulations verify the tenability of such a hypothesis.

Acknowledgments

This research was supported by NIH/PHS research grant 1-P41-RR09283. The author is greatly indebted to Dana Ballard for many useful discussions, comments, and suggestions during the course of this work.

References

[Abeles, 1991] M. Abeles. *Corticonics: Neural Circuits of the Cerebral Cortex.* New York: Cambridge University Press, 1991.

[Albus, 1991] J.S. Albus. Outline for a theory of intelligence. *IEEE Trans. Systems, Man, and Cybernetics*, 21(3):473–509, 1991.

[Athans, 1974] M. Athans. The importance of Kalman filtering methods for economic systems. *Annals of Economic and Social Measurement*, 3:49–64, 1974.

[Ayache and Faugeras, 1986] N. Ayache and O.D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.

[Barlow, 1985] H.B. Barlow. Cerebral cortex as model builder. In *Models of the Visual Cortex*, pages 37–46. New York: John Wiley and Sons, 1985.

[Barlow, 1994] H. Barlow. What is the computational goal of the neocortex? In C. Koch and J.L. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 1–22. Cambridge, MA: MIT Press, 1994.

[Bell and Sejnowski, 1996] A.J. Bell and T.J. Sejnowski. The 'independent components' of natural scenes are edge filters. Submitted to *Vision Research*, 1996.

[Blake and Yuille, 1992] A. Blake and A. Yuille, editors. *Active Vision*. Cambridge, MA: MIT Press, 1992.

[Bolz *et al.*, 1989] J. Bolz, C.D. Gilbert, and T.N. Wiesel. Pharmacological analysis of cortical circuitry. *Trends in Neurosciences*, 12(8):292–296, 1989.

[Broida and Chellappa, 1986] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):90–99, 1986.

[Bryson and Ho, 1975] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control*. New York: John Wiley and Sons, 1975.

[Cantoni and Levialdi, 1986] V. Cantoni and S. Levialdi, editors. *Pyramidal Systems for Computer Vision (Proceedings of a NATO Advanced Research Workshop, Maratea, Italy, May 5–9, 1986)*, Berlin, 1986. Springer.

[Carandini and Heeger, 1994] M. Carandini and D. Heeger. Summation and division by neurons in visual cortex. *Science*, 264:1333–1336, 1994.

[Chou *et al.*, 1994] K.C. Chou, A.S. Willsky, and A. Benveniste. Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control*, 39(3):464–478, March 1994.

[Chrisman, 1992] L. Chrisman. Reinforcement learning with perceptual aliasing. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1992.

[Churchland and Sejnowski, 1992] P. Churchland and T. Sejnowski. *The Computational Brain*. Cambridge, MA: MIT Press, 1992.

[Cipra, 1993] B. Cipra. Engineers look to Kalman filtering for guidance. *SIAM News*, 26(5), 1993.

[Creutzfeldt, 1977] O.D. Creutzfeldt. Generality of the functional structure of the neocortex. *Naturwissenschaften*, 64:507–517, 1977.

[Crick, 1984] F. Crick. Function of the thalamic reticular complex: The searchlight hypothesis. *Proc. Natl. Acad. Sci.*, 81:4586–4590, 1984.

[Daubechies, 1992] I. Daubechies. *Ten lectures on wavelets*. CBMS-NSF Regional Conferences Series in Applied Mathematics. SIAM, Philadelphia, PA, 1992.

[Dayan and Hinton, 1996] P. Dayan and G.E. Hinton. Varieties of Helmholtz machine. *Neural Networks*, 9(8):1385–1403, 1996.

[Dayan *et al.*, 1995] P. Dayan, G.E. Hinton, R.M. Neal, and R.S. Zemel. The Helmholtz machine. *Neural Computation*, 7:889–904, 1995.

[Dempster *et al.*, 1977] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, 39:1–38, 1977.

[Desimone and Ungerleider, 1989] R. Desimone and L.G. Ungerleider. Neural mechanisms of visual processing in monkeys. In F. Boller and J. Grafman, editors, *Handbook of Neuropsychology*, volume 2, chapter 14, pages 267–299. New York: Elsevier, 1989.

[Dickmanns and Mysliwetz, 1992] E.D. Dickmanns and B.D. Mysliwetz. Recursive 3D road and relative ego-state recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):199–213, 1992.

[Felleman and Van Essen, 1991] D.J. Felleman and D.C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

[Feller, 1968] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, New York, 1968.

[Ferster, 1994] D. Ferster. Linearity of synaptic interactions in the assembly of receptive fields in cat visual cortex. *Current Opinion in Neurobiology*, 4:563–568, 1994.

[Gilbert and Wiesel, 1981] C.D. Gilbert and T.N. Wiesel. Laminar specialization and intracortical connections in cat primary visual cortex. In *The Organization of the Cerebral Cortex*, pages 163–191. Cambridge, MA: MIT Press, 1981.

[Girosi *et al.*, 1995] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[Grossberg, 1976] S. Grossberg. Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, 23:187–202, 1976.

[Hackbusch, 1985] W. Hackbusch. *Multi-grid methods and applications*. Berlin: Springer-Verlag, 1985.

[Hallam, 1983] J. Hallam. Resolving observer motion by object tracking. In *Proc. of 8th International Joint Conf. on Artificial Intelligence*, volume 2, pages 792–798, 1983.

[Harpur and Prager, 1996] G.F. Harpur and R.W. Prager. Development of low-entropy coding in a recurrent network. *Network*, 7:277–284, 1996.

[Harth *et al.*, 1987] E. Harth, K.P. Unnikrishnan, and A.S. Pandya. The inversion of sensory processing by feedback pathways: A model of visual cognitive functions. *Science*, 237:184–187, 1987.

[Heeger *et al.*, 1996] D.J. Heeger, E.P. Simoncelli, and J.A. Movshon. Computational models of cortical visual processing. *Proc. National Acad. Sciences*, 93:623–627, 1996.

[Hinton *et al.*, 1995] G.E. Hinton, P. Dayan, B.J. Frey, and R.M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[Huber, 1981] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[Humphrey, 1992] N. Humphrey. *A History of the Mind*. New York: Simon and Schuster, 1992.

[Isard and Blake, 1996] M. Isard and A. Blake. Contour tracking by stochastic propogation of conditional density. In *Proc. of ECCV*, pages 343–356, 1996.

[James *et al.*, 1995] A.C. James, J.M. Hupe, S.L. Lomber, B. Payne, P. Girard, and J. Bullier. Feedback connections contribute to center-surround interactions in neurons of monkey areas V1 and V2. *Soc. Neuro. Abstr.*, 21:904, 1995.

[Jordan and Jacobs, 1994] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[Kaelbling *et al.*, 1996] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[Kalman and Bucy, 1961] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME J. Basic Eng.*, 83:95–108, 1961.

[Kalman, 1960] R.E. Kalman. A new approach to linear filtering and prediction theory. *Trans. ASME J. Basic Eng.*, 82:35–45, 1960.

[Kawato *et al.*, 1993] M. Kawato, H. Hayakawa, and T. Inui. A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network*, 4:415–422, 1993.

[Kohonen, 1988] T. Kohonen. *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, second edition, 1988.

[Leonardis and Bischof, 1996] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proc. of CVPR*, pages 453–458, 1996.

[Lindeberg, 1994] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Netherlands: Kluwer Academic Publishers, 1994.

[Lund, 1981] J.S. Lund. Intrinsic organization of the primate visual cortex, area 17, as seen in Golgi preparations. In *The Organization of the Cerebral Cortex*, pages 105–124. Cambridge, MA: MIT Press, 1981.

[MacKay, 1956] D.M. MacKay. The epistemological problem for automata. In *Automata Studies*, pages 235–251. Princeton, NJ: Princeton University Press, 1956.

[Matthies *et al.*, 1989] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[Maunsell and Newsome, 1987] J.H.R. Maunsell and W.T. Newsome. Visual processing in monkey extrastriate cortex. *Annual Review of Neuroscience*, 10:363–401, 1987.

[Maybeck, 1979] P.S. Maybeck. *Stochastic Models, Estimation, and Control (Vols. I and II)*. New York: Academic Press, 1979.

[McCallum, 1996] R.A. McCallum. Hidden state and reinforcement learning with instance-based state identification. *IEEE Trans. on Systems, Man and Cybernetics*, 26(3):464–473, 1996.

[Mishkin *et al.*, 1983] M. Mishkin, L.G. Ungerleider, and K.A. Macko. Object vision and spatial vision: Two cortical pathways. *Trends in Neuroscience*, 6:414–417, 1983.

[Mumford, 1992] D. Mumford. On the computational architecture of the neocortex. II. The role of cortico-cortical loops. *Biological Cybernetics*, 66:241–251, 1992.

[Murase and Nayar, 1995] H. Murase and S.K. Nayar. Visual learning and recognition of 3D objects from appearance. *IJCV*, 14:5–24, 1995.

[Murphy and Sillito, 1987] P.C. Murphy and A.M. Sillito. Corticofugal feedback influences the generation of length tuning in the visual pathway. *Nature*, 329:727–729, 1987.

[Neisser, 1967] U. Neisser. *Cognitive Psychology*. New York: Appleton-Century-Crofts, 1967.

[Newman, 1988] J.D. Newman. Primate hearing mechanisms. In H.D. Steklis and J. Erwin, editors, *Comparative Primate Biology, Volume 4: Neurosciences*, pages 469–499. New York: Alan R. Liss, Inc., 1988.

[Olshausen and Field, 1996] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[Orban, 1992] G.A. Orban. *Neuronal Operations in the Visual Cortex*. New York: Springer Verlag, 1992.

[O'Reilly, 1996] R.C. O'Reilly. *The LEABRA model of neural interactions and learning in the neocortex*. PhD thesis, Department of Psychology, Carnegie Mellon University, 1996.

[Pandya *et al.*, 1988] D.N. Pandya, B. Seltzer, and H. Barbas. Input-output organization of the primate cerebral cortex. In H.D. Steklis and J. Erwin, editors, *Comparative Primate Biology, Volume 4: Neurosciences*, pages 39–80. New York: Alan R. Liss, Inc., 1988.

[Pentland, 1992] A.P. Pentland. Dynamic vision. In G.A. Carpenter and S. Grossberg, editors, *Neural Networks for Vision and Image Processing*, pages 133–159. Cambridge, MA: MIT Press, 1992.

[Picton and Stuss, 1994] T.W. Picton and D.T. Stuss. Neurobiology of conscious experience. *Current Opinion in Neurobiology*, 4:256–265, 1994.

[Poggio and Edelman, 1990] T. Poggio and S. Edelman. A network that learns to recognize 3D objects. *Nature*, 343:263–266, 1990.

[Poggio *et al.*, 1985] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[Rao and Ballard, 1996a] R.P.N. Rao and D.H. Ballard. Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, 9(4):805-847, 1997. Also, Technical Report 96.2, National Resource Laboratory for the Study of Brain and Behavior, Department of Computer Science, University of Rochester, 1996.

[Rao and Ballard, 1996b] R.P.N. Rao and D.H. Ballard. The visual cortex as a hierarchical predictor. Technical Report 96.4, National Resource Laboratory for the Study of Brain and Behavior, Department of Computer Science, University of Rochester, September 1996.

[Rao and Ballard, 1997] R.P.N. Rao and D.H. Ballard. Kalman filter networks for invariant recognition, motion, and stereo. Technical Report 97.2, National Resource Laboratory for the Study of Brain and Behavior, Department of Computer Science, University of Rochester, March 1997.

[Rissanen, 1989] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.

[Rockland and Pandya, 1979] K.S. Rockland and D.N. Pandya. Laminar origins and terminations of cortical connections of the occipital lobe in the rhesus monkey. *Brain Research*, 179:3–20, 1979.

[Softky, 1996] W.R. Softky. Unsupervised pixel-prediction. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 809–815. Cambridge, MA: MIT Press, 1996.

[Turk and Pentland, 1991] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[Ungerleider and Mishkin, 1982] L. Ungerleider and M. Mishkin. Two cortical visual systems. In D. Ingle, M. Goodale, and R. Mansfield, editors, *Analysis of Visual Behavior*, pages 549–585. Cambridge, MA: MIT Press, 1982.

[Van Essen and Maunsell, 1983] D.C. Van Essen and J.H.R. Maunsell. Hierarchical organization and functional streams in the visual cortex. *Trends in Neuroscience*, 6:370–375, 1983.

[Van Essen, 1985] D.C. Van Essen. Functional organization of primate visual cortex. In A. Peters and E.G. Jones, editors, *Cerebral Cortex*, volume 3, pages 259–329. Plenum, 1985.

［Whitehead and Ballard, 1991］S.D. Whitehead and D.H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7(1):45–83, 1991.

［Zemel, 1994］R.S. Zemel. *A Minimum Description Length Framework for Unsupervised Learning*. PhD thesis, Department of Computer Science, University of Toronto, 1994.

［Zipser *et al.*, 1996］K. Zipser, V.A.F. Lamme, and P.H. Schiller. Contextual modulation in primary visual cortex. *Journal of Neuroscience*, 16(22):7376–7389, 1996.