PERGAMON

Vision Research 39 (1999) 1963–1989

Vision
Research

# An optimal estimation approach to visual perception and learning[☆]

Rajesh P.N. Rao *

*The Salk Institute, Sloan Center for Theoretical Neurobiology and Computational Neurobiology Laboratory, 10010 N. Torrey Pines Road, La Jolla, CA 92037, USA*

Received 1 April 1997; received in revised form 16 April 1998

**Abstract**

How does the visual system learn an internal model of the external environment? How is this internal model used during visual perception? How are occlusions and background clutter so effortlessly discounted for when recognizing a familiar object? How is a particular object of interest attended to and recognized in the presence of other objects in the field of view? In this paper, we attempt to address these questions from the perspective of Bayesian optimal estimation theory. Using the concept of generative models and the statistical theory of Kalman filtering, we show how static and dynamic events occurring in the visual environment may be learned and recognized given only the input images. We also describe an extension of the Kalman filter model that can handle multiple objects in the field of view. The resulting robust Kalman filter model demonstrates how certain forms of attention can be viewed as an emergent property of the interaction between top–down expectations and bottom–up signals. Experimental results are provided to help demonstrate the ability of such a model to perform robust segmentation and recognition of objects and image sequences in the presence of occlusions and clutter. © 1999 Elsevier Science Ltd. All rights reserved.

*Keywords:* Visual recognition; Perceptual learning; Attention; Segmentation; Prediction; Kalman filtering

## 1. Introduction

Vision is fundamentally a dynamic process. The images impinging on the retina are seldom comprised 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 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.

In this paper, we describe a statistical theory of vision based directly on the assumptions that (a) vision is a stochastic, dynamic process and (b) the task of visual perception is to optimally estimate visual events and on a longer time scale, learn efficient internal models of the dynamic visual environment given only the input images. Optimality is defined in a Bayesian manner in terms of maximizing the posterior probability of generating the observed visual data, given a prior estimate of the state and the current input image. Using linear models for the dynamics of the state and for the generation of images from a given state, we derive equations for state estimation that are shown to implement the well-known Kalman filter (Kalman, 1960; Kalman & Bucy, 1961) from optimal control theory (Bryson & Ho, 1975). The Kalman filter is essentially a linear dynamical system that attempts to mimic the

[☆] This work was performed at the Department of Computer Science, University of Rochester, Rochester, NY 14627, USA.

* Corresponding author. Fax: +1-619-587-0417; e-mail: rao@salk.edu.

PII: S0042-6989(98)00279-X

behavior of an observed natural process. It does so by calculating, at each time instant, an optimal estimate of the current state of the observed process. This state estimate is used in conjunction with an internal model of the observed 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 iteration of the filter.

In the context of vision, the Kalman filter model can be regarded as a natural generalization of some previous schemes for appearance-based vision based on principal component analysis (PCA) (cf. the *Eigenface* method of Turk & Pentland, 1991) and the *Eigenspace* method of Murase & Nayar, 1995). It also shares the favorable properties of some recently proposed learning algorithms (Olshausen & Field, 1996; Bell & Sejnowski, 1997) that have been shown to develop localized receptive fields similar to those of simple cells in the primary visual cortex from natural image inputs (see Rao & Ballard, 1997a for more details). Although Kalman filters have previously been used in computer vision (see, for example, Blake & Yuille, 1992), many of these applications have relied on hand-built dynamic models of restricted visual phenomena such as translating contours. The present approach differs from these previous approaches in allowing dynamic internal models of visual phenomena to be learned on-line directly from the spatiotemporal input stream (Section 5). In addition, we show how the standard Kalman filter can be made robust to occlusions, clutter, and noise (Section 6). In Section 7, we provide experimental results showing how a visual system can:

1. Learn internal models of static 3D objects and dynamic stimuli given only their input images.
2. Use the learned internal models for (a) recognition; (b) categorization; (c) hypothesis verification; (d) novelty detection and subsequent learning, and (e) prediction.
3. Learn efficient internal representations to combat the problem of perceptual aliasing.
4. Recognize and segment objects in the presence of occlusions and background clutter.
5. Attend to a particular object of interest in the presence of other objects or noise in the input stream.
6. Interpret an ambiguous input stimulus in two different ways depending on an initial 'priming' input.

We conclude in Section 8 by discussing the strengths and weaknesses of the model and suggest possible directions for future research.

## 2. Problem statement and previous approaches

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 & 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 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.

Fig. 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 the definition of an 'external world' need not be restricted to sensory modalities such as vision or audition. One may equally well build and use internal models of, for instance, the various muscular systems responsible for executing various types of body movements. For the purposes of this paper, however, we shall be concerned 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? Perhaps the simplest mathematical form one can ascribe to an internal model is to assume a *linear generative model* for the process underlying the generation of sensory inputs. In particular, at any time instant $t$, the internal state of the given input generating process is assumed to be characterized by a $k$-element *internal 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 of $n$ pixels) according to:

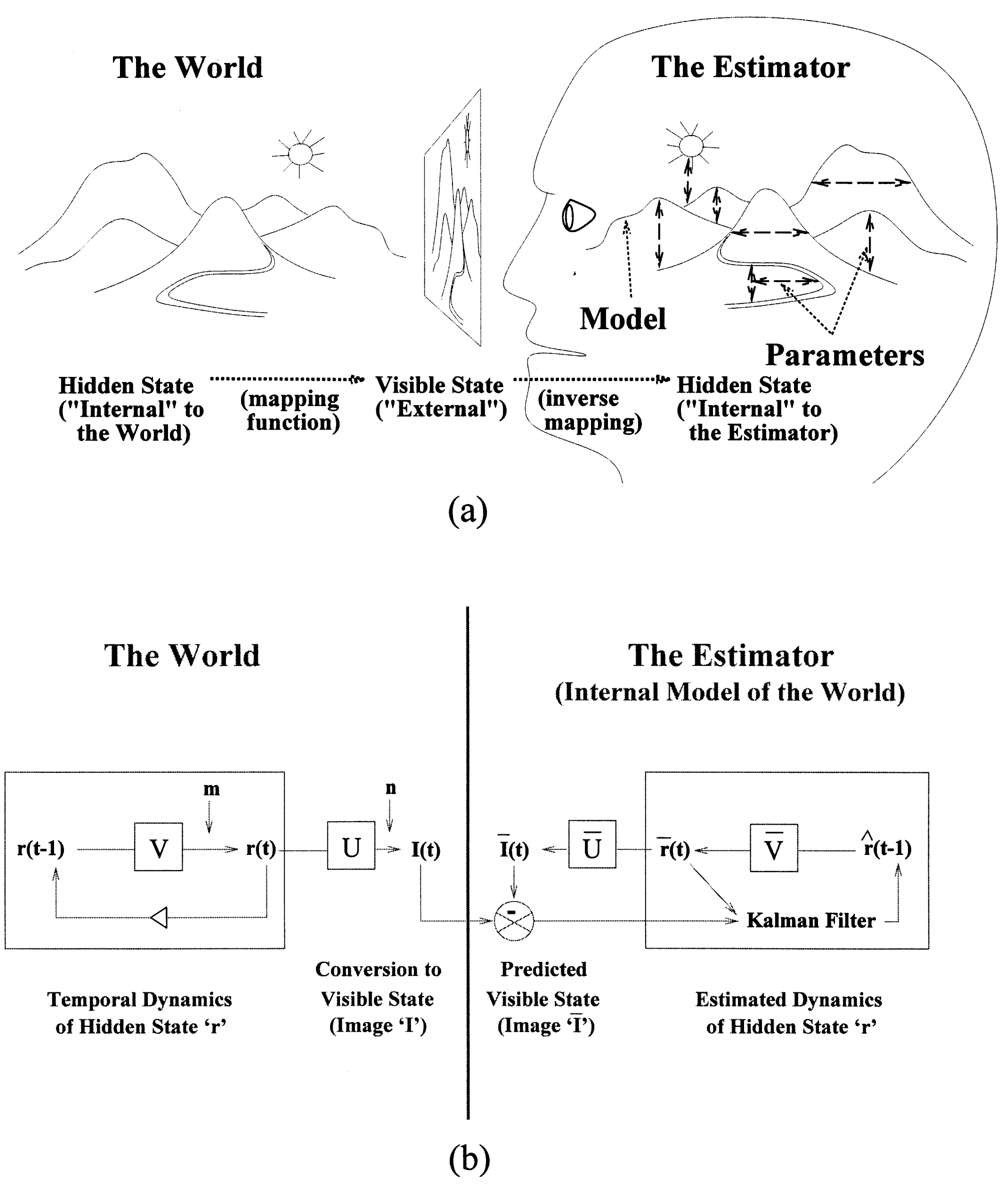
The World
The Estimator
Model
Parameters
Hidden State ("Internal" to the World)
(mapping function)
Visible State ("External")
(inverse mapping)
Hidden State ("Internal" to the Estimator)
(a)
The World
The Estimator
(Internal Model of the World)
m
n
r(t-1)
V
r(t)
U
I(t)
Ī(t)
Ū
r̄(t)
V̄
r̂(t-1)
-
Kalman Filter
Temporal Dynamics of Hidden State 'r'
Conversion to Visible State (Image 'I')
Predicted Visible State (Image 'Ī')
Estimated Dynamics of Hidden State 'r'
(b)

Fig. 1. Internal world models and the problem of optimal estimation of hidden state. (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 Kalman filter-based solution to the estimation problem. The internal model is encoded jointly by the state transition matrix $\bar{V}$ and the generative matrix $\bar{U}$, and the filter uses this internal model to compute optimal estimates $\hat{\mathbf{r}}$ of the current internal state **r** of the environment.

$$\mathbf{I}(t) = U\mathbf{r}(t) \tag{1}$$

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 $\mathbf{I}(t)$.

In the case of vision, $\mathbf{I}(t)$ could represent a retinal image, generated by a set of physical 'causes,' as represented by $\mathbf{r}(t)$, that are intrinsic to the visual environment. These physical causes may be related to various intrinsic attributes of the stimulus, such as shape, illumination, and texture. The matrix $U$ specifies how these attributes have been transformed to yield the measured image $\mathbf{I}(t)$. The $k$ columns of the matrix $U$ can be regarded as a set of $k$ basis vectors for representing the input images. The $k$ values in the state vector $\mathbf{r}(t)$ act as coefficients for these basis vectors, determining how much weight to assign to each basis vector $U_j$ for a given input image $\mathbf{I}(t)$:

$$\mathbf{I}(t) = \sum_{j=1}^{k} U_j r_j(t)$$

Given the generative model above, the goal of a sensory system becomes one of optimally estimating the state vector $\mathbf{r}(t)$ for any given input $\mathbf{I}(t)$ and on a longer time scale, learning an internal model of the input generating process by learning appropriate basis vectors within the matrix $U$. In a neural setting, the estimate for the basis vector matrix $U$ is assumed to be stored within the synaptic weights (or efficacies) of neurons in a network while the state vector $\mathbf{r}(t)$ is assumed to denote presynaptic neuronal responses (or firing rates).

### 2.1. Previous approaches

There has been recent interest in appearance-based approaches to computational vision. These differ significantly from traditional 3D model-based or geometry-based approaches (Huttenlocher & Ullman, 1987; Lowe, 1987; Lamdan & Wolfson, 1988; Grimson, 1990), which have typically been limited to representing restricted types of geometric objects. In the appearance-based approach, the need for explicit 3D geometric models of objects is avoided by extracting object representations directly from the input images. For example, Buhmann, Lades and Malsburg (1990) use a set of Gabor filters to form composite feature detectors called 'jets,' whose responses to input images are used in an elastic graph-matching strategy for recognition. Daugman (1993) uses multiscale 2-D Gabor wavelets to generate long 256-byte 'iris codes' for a human eye which he uses in a scheme for personal identity verification. Viola (1996) describes a recognition system that uses the responses of a statistically motivated set of 'complex' local features. Rao and Ballard (1995) use steerable Gaussian derivative filters at multiple scales for object identification and location using an active vision system. Mel (1996) has proposed an object recognition system called SEEMORE which employs 'receptive field' histograms for recognition, partly inspired by the work of Swain and Ballard on color histograms (Swain & Ballard, 1991). The 'receptive fields' are comprised of a large number of local color and edge/curvature detectors. A similar approach based on the notion of local receptive fields has independently been explored by Schiele and Crowley (1996). Schmid and Mohr (1996) use differential invariants rather than spatial features or filters and extract responses from salient 'keypoints' in a given scene. Poggio, Edelman and colleagues have used radial basis function networks for learning and recognizing wire objects and faces (Poggio & Edelman, 1990; Brunelli & Poggio, 1993). Nelson and Selinger (1998) report good results on a large database of 3D shapes using 2D boundary fragments in the context of an associative memory.

In many of the above appearance-based approaches, the features or spatial filters are fixed and not learned from input images. In terms of the generative model in Eq. (1), this reduces to using a fixed set of basis vectors within the generative matrix $\mathbf{U}$. These basis vectors (spatial filters or features) are selected a priori based on certain favorable mathematical or biological properties. The feature vector obtained by convolving a given input image with these basis vectors is used for the purpose of recognition. In terms of Eq. (1), this feature vector corresponds to an estimate of the state vector $\mathbf{r}(t)$ given a particular choice of basis vectors. The accuracy and usefulness of this estimate is determined primarily by the choice of the hand-picked basis vectors. As a result, the recognition system is prone to failure in cases where the hand-picked basis functions do not match the statistics of the input images. A recognition system can overcome this problem if it is endowed with the ability to autonomously tailor its basis vectors to match the statistics of its input stream, allowing it to learn and maintain an efficient internal model of its input environment.

### 2.2. Principal component analysis (PCA)

A popular technique that does allow the learning of basis vectors for efficiently representing input images is principal component analysis (PCA) (Chatfield & Collins, 1980; Jolliffe, 1986; Ballard, 1997), also known as the Karhunen–Loéve transform. Consider the problem of encoding a collection of $n \times 1$ input vectors $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_p$ using an $n \times k$ matrix $U$. One solution is to choose the columns of $U$ to be the first $k$ dominant eigenvectors (in terms of maximal eigenvalues) of the input covariance matrix $E(\mathbf{I}\mathbf{I}^T)$ as computed from zero-mean samples of input data ($E$ denotes the expectation operator and $T$ denotes transpose). This comprises the core of the 'eigenface' technique of Turk and Pentland (1991) and the eigenspace method of Murase and Nayar (1995). Alternately, assuming $P < n$, one can compute the singular value decomposition (SVD) of the matrix of input vectors to directly obtain the principal component basis vectors without having to compute the covariance matrix. This approach is utilized by Black and Jepson (1998) in their eigen-tracking approach. In either case, the columns of $U$ are orthogonal to each other and the estimate of the state vector $\mathbf{r}$ corresponding to a given input $\mathbf{I}$ can be computed as a simple linear feedforward function of the input:

$$\mathbf{r} = U^T\mathbf{I} \tag{2}$$

Since $k$ is generally much smaller than $n$, the state vector $r$ is an efficient compressed representation of the input image. A reconstruction of the input image $\hat{\mathbf{I}}$ can be generated from $\mathbf{r}$ by using the following relation which simply inverts the transformation in Eq. (2):

$$\hat{\mathbf{I}} = U\mathbf{r} \tag{3}$$

It is well-known that the eigenvector matrix $U$ minimizes the pixel-wise expected reconstruction error function:

$$J(U) = \sum_{i=1}^{n} (\mathbf{I}^i - U^i\mathbf{r})^2 = (\mathbf{I} - U\mathbf{r})^T(\mathbf{I} - U\mathbf{r})$$

(where $U^i$ denotes the $i$th row of the matrix $U$) over all inputs subject to the constraint that the columns of $U$ are orthogonal, $\mathbf{r}$ being specified as in Eq. (2).

In summary, PCA transforms an original set of variables (for example, image pixels) to a new set of *uncorrelated* variables (principal components) which are derived in the order of decreasing importance (i.e. decreasing variance). The uncorrelated variables are linear combinations of the original variables. The primary goal of PCA is parsimony. The hope is that the first few components will account for most of the variation in the original data; further analysis can then proceed on this new, smaller set of variables, thereby reducing the effective dimensionality of the data. Viewed geometrically, the transformation is just a rotation in the space spanned by the original variables.

### *2.3. Principal weaknesses of PCA*

PCA has been used in recent years for tasks such as face recognition, object recognition, pose estimation and tracking (Turk & Pentland, 1991; Murase & Nayar, 1995; Black & Jepson, 1998). Implicit in these applications of PCA to vision and image processing is the assumption that it provides an adequate description of the statistical process generating the input images.

Unfortunately, a number of recent studies on the statistics of natural images suggest that PCA may be unsuitable for describing natural image distributions (Field, 1994; Olshausen & Field, 1996). Although PCA achieves optimal linear data compression, it has several serious shortcomings that limit its use in vision and signal processing:

1. The basis vectors obtained from PCA are constrained to be mutually orthogonal whereas the mechanisms underlying the generation of natural data are often best described by using nonorthogonal basis vectors (Field, 1994; Olshausen & Field, 1996).
2. The internal state or response vector $\mathbf{r}$ need not be a purely one-shot linear feedforward function of the basis vectors and the input image as it is in the case of PCA (Eq. (2)). This is especially true in cases where the basis vectors are not mutually orthogonal or when there is top–down information from a higher hierarchical level that can influence the state at a lower level (cf. Rao & Ballard, 1997a).
3. PCA-based methods have typically been applied to the analysis of static images and it is not clear how these methods can be extended to the spatiotemporal case for prediction and learning of image dynamics directly from the input stream.
4. Principal component methods are suitable only when the data are well described by Gaussian clouds. Recent work by Field (1994) and others strongly suggest that natural image data cannot be satisfactorily described in this manner.
5. PCA requires the number of basis vectors to be less than the dimensionality of the input space. This means that overcomplete representations cannot be learned (see Simoncelli, Freeman, Adelson and Heeger (1992), Olshausen and Field (1997), Lewicki and Sejnowski (1998)) for arguments regarding the need for overcomplete representations in visual processing).
6. More importantly, PCA can only capture linear pairwise statistical dependencies in the input stream. However, natural scenes are rife with higher-order statistical structure that cannot be accounted for by linear pairwise statistics (Olshausen & Field, 1996).

## 3. The optimal estimation approach

In this section, we suggest a statistical framework that allows one to overcome some of the main limitations of PCA-based approaches. In particular, we define a generative model based directly on the assumption that vision is a stochastic, dynamic process. This in turn allows one to view the task of visual perception as one of optimally estimating visual events and on a longer time scale, learning efficient internal models of the visual environment. Optimality is defined in a Bayesian manner in terms of maximizing the posterior probability of generating the observed visual data, given a prior estimate of the state, the model parameters, and the current input image. Unlike PCA, the basis vectors are not constrained to be orthogonal and the state vector is viewed as a free parameter that can be optimized to suit the choice of the basis vectors. In addition, the approach allows one to tailor a possibly overcomplete set of non-orthogonal basis vectors to match input distributions by allowing one to choose appropriate prior distributions for the parameters (see Harpur & Prager, 1996; Olshausen & Field, 1996; Rao & Ballard, 1997a; Lewicki & Sejnowski, 1998 for more details).

### *3.1. Spatiotemporal generative model*

Fig. 1(b) shows the mathematical form of the spatiotemporal generative model we will be concerned with. Briefly, we assume that a natural process in the external world can be modeled as a stochastic linear dynamical system. At any time instant $t$, the internal state vector $\mathbf{r}(t)$ is assumed to generate a measurable and observable output $\mathbf{I}(t)$ (for example, an image) according to:

$$\mathbf{I}(t) = U\mathbf{r}(t) + \mathbf{n}(t) \tag{4}$$

where $U$ is the generative (or measurement) matrix and $\mathbf{n}(t)$ is a Gaussian stochastic noise process with mean zero and a covariance matrix given by $\Sigma = E[\mathbf{n}\mathbf{n}^T]$. Note that this is a sufficient description of $\mathbf{n}$ since a Gaussian distribution is completely characterized by its mean and covariance.

In addition to specifying how the internal state of the observed process generates a spatial image, we also need to specify how the internal state itself changes with time $t$. We assume 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:

$$\mathbf{r}(t) = V\mathbf{r}(t-1) + \mathbf{m}(t-1) \quad (5)$$

where $V$ is a (usually unknown) *state transition* (*or prediction*) *matrix* and $\mathbf{m}$ is a Gaussian noise process with mean $\bar{\mathbf{m}}(t)$ and covariance $\Pi = E[(\mathbf{m}-\bar{\mathbf{m}})(\mathbf{m}-\bar{\mathbf{m}})^T]$. In other words, the matrix $V$ is used to characterize the dynamic behavior of the observed system over 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 $\mathbf{m}(t-1)$.

The choice of the matrices $U$ and $V$ depends crucially on the representation $\mathbf{r}$ of the presumed internal state of the modeled process. Traditional applications of the above spatiotemporal generative model (such as Kalman filter-based applications) 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 $\mathbf{r}$? A solution that we pursue herein is to define an appropriate optimization function and minimize this function to obtain estimates $\hat{\mathbf{r}}$, $\hat{U}$, and $\hat{V}$ of $\mathbf{r}$, $U$, and $V$. Note that the estimates $\hat{U}$ and $\hat{V}$ together encode an internal model of the world. This internal model generates momentary state estimates $\hat{\mathbf{r}}(t)$ denoting interpretations (with respect to the internal model) of observed dynamic phenomena occurring in the external world.

### 3.2. An optimization function

The parameters $\mathbf{r}$, $U$, and $V$ in the spatiotemporal generative model above can be estimated and learned directly from input data if we can define an appropriate optimization function which can be minimized with respect to $\mathbf{r}$, $U$, 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 $\hat{\mathbf{r}}(t)$ 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 $\bar{\mathbf{r}}$ of the current state $\mathbf{r}$ based on prior data. In particular, let $\bar{\mathbf{r}}(t)$ 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 $E[(\mathbf{r}-\bar{\mathbf{r}})(\mathbf{r}-\bar{\mathbf{r}})^T] = M$. A common optimization function whose minimization yields an estimate for $\mathbf{r}$ is the *least-squares criterion*:

$$J_1 = \sum_{i=1}^{n} (\mathbf{I}^i - U^i\mathbf{r})^2 + \sum_{i=1}^{k} (\mathbf{r}^i - \bar{\mathbf{r}}^i)^2$$
$$= (\mathbf{I} - U\mathbf{r})^T(\mathbf{I} - U\mathbf{r}) + (\mathbf{r} - \bar{\mathbf{r}})^T(\mathbf{r} - \bar{\mathbf{r}})$$

where the superscript $i$ denotes the $i$th element or row of the superscripted vector or matrix. 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 $\bar{\mathbf{r}}$ computed from prior data.

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

$$J = (\mathbf{I} - U\mathbf{r})^T\Sigma^{-1}(\mathbf{I} - U\mathbf{r}) + (\mathbf{r} - \bar{\mathbf{r}})^T M^{-1}(\mathbf{r} - \bar{\mathbf{r}}) \quad (6)$$

This criterion becomes meaningful when interpreted in terms of the stochastic model described in the previous section. Recall that the measurement Eq. (4) was characterized in terms of a Gaussian noise process with mean zero and covariance $\Sigma$. Note also that $\mathbf{r}$ follows a Gaussian distribution with mean $\bar{\mathbf{r}}$ 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 $\mathbf{r}$ (ignoring constant terms):

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

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 input data is given by (using Bayes theorem):

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

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 $\mathbf{I}$.

## 4. Optimal estimation and prediction

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 $\partial J/\partial \mathbf{r} = 0$. This results in:

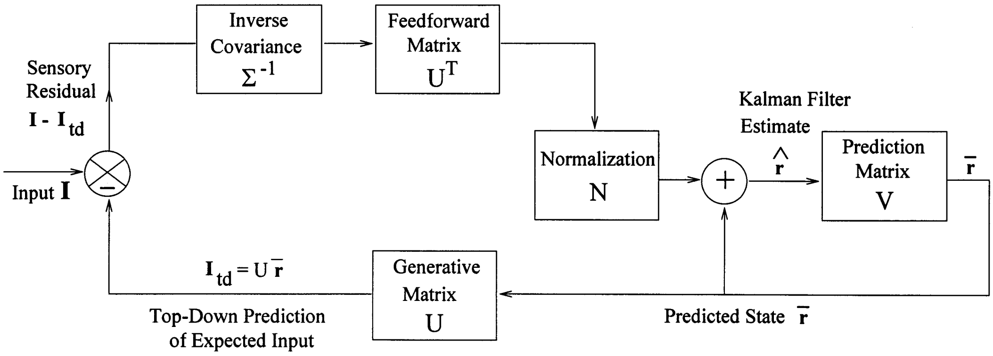

Fig. 2. Schematic diagram of the Kalman filter.

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

which yields:

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

Using the substitution $N(t) = (U^T\Sigma^{-1}U + M^{-1})^{-1}$ and rearranging the terms in the above equation, we obtain the following equation which implements the well-known *Kalman filter* from optimal control theory (Bryson & Ho, 1975):

$$\hat{\mathbf{r}}(t) = \bar{\mathbf{r}}(t) + N(t)U^T\Sigma(t)^{-1}(\mathbf{I}(t) - U\bar{\mathbf{r}}(t)) \tag{7}$$

Fig. 2 shows a schematic diagram of the Kalman filter. The Kalman filter estimate $\hat{\mathbf{r}}$ is the *mean* of the Gaussian distribution of the state $\mathbf{r}$ *after* measurement of $\mathbf{I}$ (Bryson & Ho, 1975). The matrix $N$, which performs a form of divisive normalization, can likewise be shown to be the corresponding *covariance* matrix. Recall that $\bar{\mathbf{r}}$ and $M$ were the mean and covariance *before* measurement of $\mathbf{I}$. These quantities are updated as follows:

$$\bar{\mathbf{r}}(t) = V\hat{\mathbf{r}}(t-1) + \bar{\mathbf{m}}(t-1) \tag{8}$$

$$M(t) = VN(t-1)V^T + \Pi(t-1) \tag{9}$$

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

In summary, the Kalman filter predicts one step into the future using Eq. (8), obtains the next sensory input $\mathbf{I}(t)$, and then corrects its prediction $\bar{\mathbf{r}}(t)$ using the sensory residual $(\mathbf{I}(t) - U\bar{\mathbf{r}}(t))$ and the gain matrix $K(t) = N(t)U^T\Sigma\,(t)^{-1}$ as in Eq. (7). 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 $\bar{\mathbf{r}}(t+1)$. The covariance matrices corresponding to $\bar{\mathbf{r}}$ and $\hat{\mathbf{r}}(t)$ are updated in an analogous fashion. Fig. 3 illustrates this evolution of the conditional Gaussian probability density function of the state over time according to the Kalman filter equations.

### *4.1. Running average example*

To understand the Kalman filter equation in perhaps its simplest form, consider the following rule for computing the average of a set of $t$ real number inputs $I(1)$, $I(2), \ldots, I(t-1), I(t)$:

$$\hat{r}(t) = (I(1) + I(2) + \ldots + I(t))/t \tag{10}$$

This equation can be rewritten as:

$$\hat{r}(t) = \hat{r}(t-1) + \frac{1}{t}\,(I(t) - \hat{r}(t-1)) \tag{11}$$

where $\hat{r}(t-1)$ is the average of the first $t-1$ numbers i.e. $\hat{r}(t-1) = (I(1) + I(2) + \ldots + I(t-1))/(t-1)$. Note that Eq. (11) is simply a recursive form of Eq. (10) and can be rewritten in terms of Kalman filter terminology as:

$$\hat{r}(t) = \hat{r}(t-1) + N(t)(I(t) - \hat{r}(t-1)) \tag{12}$$

$$N(t) = (1 + N(t-1)^{-1})^{-1} \tag{13}$$

where the initial conditions are given by $\hat{r}(0) = 0$ and $N(1) = 1$. In this simple case, since we are estimating a constant scalar quantity $r$ using an increasing number of measurements $I(t)$, the generative model is simply $I(t) = r(t) + n(t)$ where $n$ has mean zero and a variance $\Sigma = 1$, and $U = 1$. The dynamics are $r(t) = r(t-1)$, with $V = 1$ and $\Pi = 0$. Thus, the Kalman filter update equations are given by:

$$\hat{r}(t) = \bar{r}(t) + N(t)(I(t) - \bar{r}(t))$$

$$\bar{r}(t) = \hat{r}(t-1)$$

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

$$M(t) = N(t-1)$$

which reduces to exactly the same two equations (Eqs. (12) and (13)) above for computing the running average.

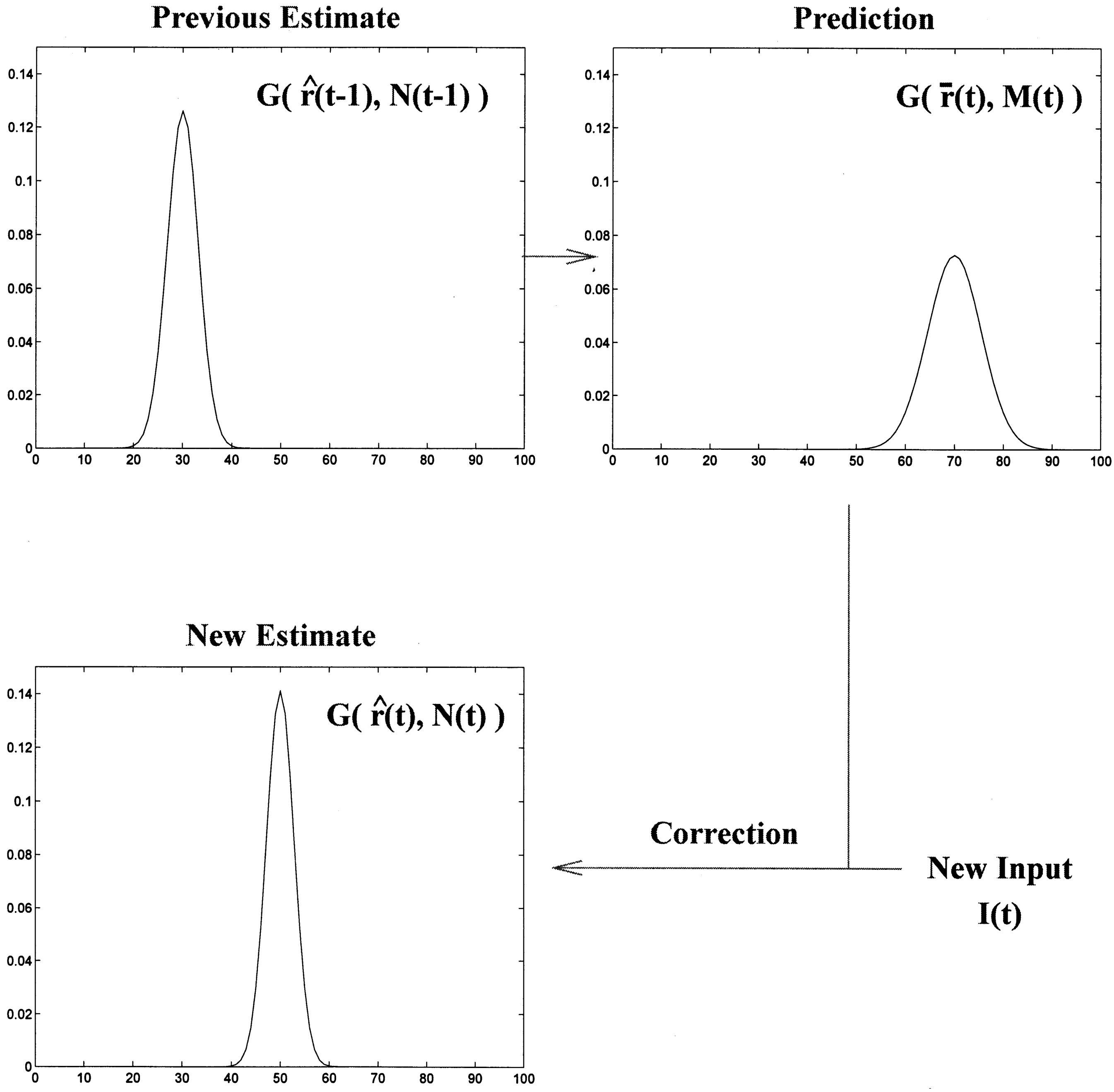

Fig. 3. Propagation of conditional probability density in the Kalman filter. The estimate of the state at time $t-1$ is the Gaussian density $P(\mathbf{r}|\mathbf{I})$ with mean $\hat{\mathbf{r}}(t-1)$ and covariance $N(t-1)$. These values are used to predict the mean $\bar{\mathbf{r}}(t)$ and covariance $M(t)$ for the next time step (Eqs. (8) and (9)). Note that this generally results in an increase in uncertainty, as suggested by the increase in variance of the Gaussian density in the figure. The input at the next time step is used to correct $\bar{\mathbf{r}}$ and $M$ according to the Kalman filter equation, resulting in a new conditional density with mean $\hat{\mathbf{r}}(t)$ and covariance $N(t)$. This process is repeated for each subsequent time step.

### *4.2. General form of the Kalman filter*

The Kalman filter equation and the running average rule are both of the form:

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

The gain matrix $K(t) = N(t)U^T\Sigma(t)^{-1}$ in Eq. (7) is known as the Kalman gain. It determines the weight given to the sensory residual in correcting the old estimate $\bar{\mathbf{r}}$. Note that this gain is determined by the covariances $\Sigma$ and $M$, and therefore effectively trades off the prior estimate $\bar{\mathbf{r}}$ against the sensory input $\mathbf{I}$ according to the uncertainties in these two sources. This become clear if one rewrites the Kalman filter (Eq. (7)) as:

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

Thus, the Kalman filter estimate $\hat{\mathbf{r}}$ is essentially a weighted average of the prior estimate $\bar{\mathbf{r}}$ and the new sensory input **I**. In the case of the running average example, Eq. (11) can be rewritten as:

$$\hat{r}(t) = \frac{t-1}{t}\hat{r}(t-1) + \frac{1}{t}I(t)$$

In this simple case, the new inputs receive less and less weight $(1/t)$ as we receive more and more inputs, signifying that our estimates $\hat{r}$ 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 matrix, which does not necessarily decrease over time. Instead, it is usually an appropriate function of the on-going needs of the task at hand (for example, see Section 6).

### *4.3. Simplified filter used in the experiments*

The general form of the Kalman filter (Eqs. (7)–(9)) involves computing matrix inverses and maintaining the state covariance matrix over time. This can become computationally very intensive even for images of moderate sizes, besides making the method susceptible to numerical instabilities. Fortunately, some simplifying assumptions can be made in the generative model to make the method more tractable. The noise covariance matrix can be assumed to be diagonal and scalar in many cases: $\Sigma = \sigma^2$. Also, rather than recomputing the state covariance $N(t)$ at each time step (involving two matrix inverses), one may approximate this covariance with a constant fixed and possibly scalar value $N_0$. The Kalman filter equations then reduce to:

$$\hat{\mathbf{r}}(t) = \bar{\mathbf{r}}(t) + \frac{N_0}{\sigma^2}U^T(\mathbf{I}(t) - U\bar{\mathbf{r}}(t)) \tag{14}$$

$$\bar{\mathbf{r}}(t) = V\hat{\mathbf{r}}(t-1) + \bar{\mathbf{m}}(t-1) \tag{15}$$

Note that using a constant gain $N_0/\sigma^2$ in the above equation can be interpreted as performing gradient descent on the optimization function of Eq. (6), the chosen value for the gain dictating the rate of descent towards a minimum (cf. Daugman, 1988; Pece, 1992; Olshausen & Field, 1997; Rao & Ballard, 1997a).

## 5. Learning internal models

The previous section derived the equation for estimating the state **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 an 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 (Hallam, 1983; Ayache & Faugeras, 1986; Broida & Chellappa, 1986; Matthies, Kanade & Szeliski, 1989; Blake & Yuille, 1992; Dickmanns & Mysliwetz, 1992; Pentland, 1992). However, in complex dynamic environments, the formulation of such hand-coded models becomes increasingly difficult. A more tractable alternative is to initialize the matrices $U$ and $V$ to small random values, and then adapt these values on-line in response to input data, thereby *learning* an internal model of the input environment.

### *5.1. Learning the measurement matrix*

The starting point for deriving 'learning rules' for $U$ and $V$ is the observation that given the optimal estimate $\hat{\mathbf{r}}$ for the state **r** based on some prior values for $U$ and $V$, one can obtain new estimates for $U$ and $V$ using two additional update equations that together minimize a joint optimization function $J$. Firstly, let **u** and **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 $\mathbf{u} = [U^1U^2\ldots U^n]^T$ where $U^i$ denotes the $i$th row of $U$. We assume these vectors stochastically drift over time according to:

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

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

where $\mathbf{n}_u$ and $\mathbf{n}_v$ are stochastic noise processes with mean $\bar{\mathbf{n}}_u$ and $\bar{\mathbf{n}}_v$, and covariances given by $\Pi_u$ and $\Pi_v$, respectively. Note that unlike the dynamics of **r**, these equations for **u** and **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 **r**, let $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ be estimates of **u** and **v** calculated from prior data. Thus, $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ represent the means of the Gaussian distributions for **u** and **v** before the measurement of the current input **I**. The corresponding covariances are given by $P = E[(\mathbf{u}-\bar{\mathbf{u}})(\mathbf{u}-\bar{\mathbf{u}})^T]$ and $Q = E[(\mathbf{v}-\bar{\mathbf{v}})(\mathbf{v}-\bar{\mathbf{v}})^T]$. We can then redefine the optimization function $J$ as:

$$J = (\mathbf{I} - U\mathbf{r})^T\Sigma^{-1}(\mathbf{I} - U\mathbf{r}) + (\mathbf{r}-\bar{\mathbf{r}})^TM^{-1}(\mathbf{r}-\bar{\mathbf{r}}) + (\mathbf{u}-\bar{\mathbf{u}})^TP^{-1}(\mathbf{u}-\bar{\mathbf{u}}) + (\mathbf{v}-\bar{\mathbf{v}})^TQ^{-1}(\mathbf{v}-\bar{\mathbf{v}}) \tag{16}$$

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:

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

As we did in the case of $\mathbf{r}$, by setting $\partial J/\partial \mathbf{u} = 0$ 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 $\mathbf{I}$:

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

$$N_u(t) = (R(t)^T\Sigma(t)^{-1}R(t) + P(t)^{-1})^{-1} \qquad (17)$$

where $\bar{\mathbf{u}}(t) = \hat{\mathbf{u}}(t-1) + \bar{\mathbf{n}}_u(t-1)$ 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 4. Also note that the learning occurs 'on-line' i.e. all the inputs to the system need not be provided a priori (as in some approaches relying on PCA/SVD) but rather, the system continues to learn as it encounters new inputs, the degree of learning being controlled by the gain term $N_u(t)R(t)^T\Sigma(t)^{-1}$.

### 5.2. 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 $\hat{R}(t-1)$ as:

$$\hat{R}(t-1) = \begin{bmatrix} \hat{\mathbf{r}}(t-1)^T & 0 & \cdots & 0 \\ 0 & \hat{\mathbf{r}}(t-1)^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \hat{\mathbf{r}}(t-1)^T \end{bmatrix}$$

where $\hat{\mathbf{r}}(t-1)$ is the Kalman filter state estimate at time $t-1$. This allows us to rewrite the state transition step (Eq. (8)) as:

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

$$= \hat{R}(t-1)\mathbf{v}(t-1) + \bar{\mathbf{m}}(t-1)$$

Substituting the right hand side of this equation for $\bar{\mathbf{r}}$ in the optimization function $J$ and setting $\partial J/\partial \mathbf{v} = 0$, we obtain the following on-line update rule for the mean and covariance of $\mathbf{v}$ at time $t-1$:

$$\hat{\mathbf{v}} = \bar{\mathbf{v}} + N_v\,\hat{R}^T M^{-1}[\mathbf{r}(t) - \mathbf{r}'(t)]$$

$$N_v = (\hat{R}^T M^{-1}\hat{R} + Q^{-1})^{-1}$$

where $\bar{\mathbf{v}}(t-1) = \hat{\mathbf{v}}(t-2) + \bar{\mathbf{n}}_v(t-2)$, $\mathbf{r}'(t) = \hat{R}(t-1)\bar{\mathbf{v}}(t-1) + \bar{\mathbf{m}}(t-1)$, and $Q(t-1) = N_v(t-2) + \Pi_v(t-2)$. Note that in this case, the estimate of $V$ is corrected using the prediction error $(\mathbf{r}(t) - \mathbf{r}'(t))$, which denotes the temporal difference between the actual state and the predicted state at time $t$ (cf. Kaelbling, Littman & Moore, 1996).

### 5.3. Simplified learning rules for U and V

The learning rules for $U$ and $V$ can be simplified by assuming zero mean noise, diagonal noise covariances and constant values for the $U$ and $V$ covariances, as was done in Section 4.3 for $\mathbf{r}$. This results in the following equations for the update of $U$ and $V$:

$$\hat{U}(t) = \bar{U}(t) + \alpha[\mathbf{I}(t) - \bar{U}(t)\mathbf{r}(t)]\mathbf{r}(t)^T \qquad (18)$$

$$\hat{V}(t-1) = \bar{V}(t-1) + \beta[\mathbf{r}(t) - \mathbf{r}'(t)]\hat{\mathbf{r}}(t-1)^T \qquad (19)$$

where $\bar{U}(t) = \hat{U}(t-1)$, $\bar{V}(t-1) = \hat{V}(t-2)$, $\mathbf{r}'(t) = \bar{V}(t-1)\hat{\mathbf{r}}(t-1) + \bar{\mathbf{m}}(t-1)$ and $\alpha = N_u/\sigma^{-2}$and $\beta = N_v/M$ are positive constants (learning rates) governing the rate of descent towards a minimum of the optimization function given by Eq. (16). Substituting the learned values $\bar{U}$ and $\bar{V}$ in the Kalman filter equations from Section 4.3, we obtain:

$$\hat{\mathbf{r}}(t) = \mathbf{r}'(t) + \frac{N_0}{\sigma^2}\,\bar{U}(t)^T(\mathbf{I}(t) - \bar{U}(t)\mathbf{r}'(t)) \qquad (20)$$

$$\mathbf{r}'(t) = \bar{V}(t-1)\hat{\mathbf{r}}(t-1) + \bar{\mathbf{m}}(t-1) \qquad (21)$$

The above learning rules and filter equations were used in the experiments described in Section 7 with appropriate values for the parameters $\alpha$, $\beta$ and $N_0/\partial^{-2}$ and with $\hat{\mathbf{r}}(0) = \mathbf{0}$ and random initial conditions for $\bar{U}$ and $\bar{V}$.

### 5.4. Convergence of the learning scheme

An interesting question is the issue of convergence of the overall filtering/learning scheme involving $\mathbf{r}$, $U$, and $V$. Note that in Eqs. (18) and (19) above, we did not specify values for $\mathbf{r}(t)$. The Expectation–Maximization (EM) algorithm from statistics (Dempster, Laird & Rubin, 1977) suggests that in the case of static input stimuli $(\bar{\mathbf{r}}(t) = \hat{\mathbf{r}}(t-1))$, one may use $\mathbf{r}(t) = \hat{\mathbf{r}}$ when updating the estimate for $U$, where $\hat{\mathbf{r}}$ 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 $\mathbf{r}(t) = \hat{\mathbf{r}}(t|N)$, which is the optimal temporally *smoothed* state estimate (Bryson & Ho, 1975) for time $t$ $(\leq N)$, given input data for each of the time instants 1,..., $N$. 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, using $\mathbf{r}(t) = \hat{\mathbf{r}}(t)$ in Eqs. (18) and (19) for updating the matrices $U$ and $V$.

Although the function $J$ is convex in each of the parameters $\mathbf{r}$, $U$, and $V$ individually, the function is no longer convex as a joint function of these variables. As a result, convergence to a global minimum is not assured. However, the internal model thus learned can still be used to estimate the state $\mathbf{r}$ and our experimental results suggest that these state estimates are often suffi-

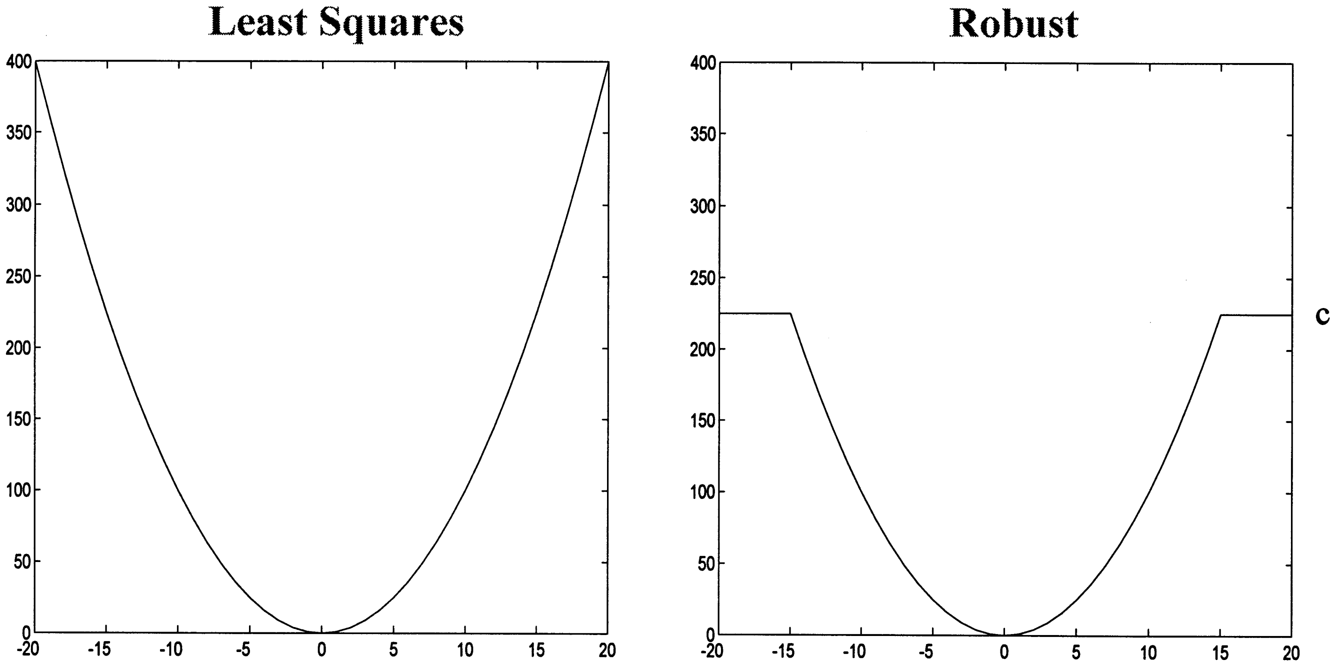

Fig. 4. Least squares versus robust optimization. The robust optimization function clips large residual errors i.e. those exceeding the threshold $c$ to the constant saturation value $c$, thereby preventing the corresponding outliers in the input data from influencing the optimization process.

cient for the purposes of visual prediction and recognition. We have found that the method converges as long as there are enough degrees of freedom available, as determined by the dimensionality of the state vector, for the method to learn unambiguous representations of the input stream (see Section 7.4).

## 6. Robust estimation, attention and segmentation

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 **I** (Huber, 1981). For example, in the case where **I** represents an input image, occlusions, background clutter, and other forms of noise may cause many pixels in **I** to deviate significantly from corresponding pixels in the predicted image $U\mathbf{r}$ of an object of interest contained in the image **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 **r**.

The field of *robust statistics* (Huber, 1981) provides 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:

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

where $\rho$ 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 $\rho$ equals the square function, we obtain the quadratic error function we previously used in $J$. More interestingly, suppose we define $\rho$ in terms of a diagonal matrix $S$ as follows:

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

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:

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

where $c$ is a threshold parameter. To understand the behavior of this function, note that $S$ effectively clips the $i$th summand in $J'$ to a constant saturation value $c$ whenever the $i$th squared residual $(\mathbf{I}^i - U^i\mathbf{r})^2$ exceeds the threshold $c$; otherwise, the summand is set equal to the squared residual. Fig. 4 contrasts this robust optimization function with the standard least squares optimization function.

By substituting $\Sigma^{-1} = S$ in the optimization function $J$ (Eq. (6)), we can rederive the Kalman filter update equations. The resulting *robust Kalman filter* for updating the state estimate is given by:

$$\hat{\mathbf{r}}(t) = \bar{\mathbf{r}}(t) + N(t)U^T G(t)(\mathbf{I} - U\bar{\mathbf{r}}(t)) \quad (22)$$

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

$$G^{i,i} = 0 \text{ if } (\mathbf{I}^i(t) - U^i\bar{\mathbf{r}}(t))^2 > c(t)$$

$$G^{i,i} = 1 \text{ otherwise} \quad (23)$$

Note that in the above expression, we approximated $\mathbf{r}(t)$ with its best available estimate $\bar{\mathbf{r}}(t)$. Similarly, in the experiments, we used the learned estimates $\bar{U}^i$ and $\bar{V}$ for $U^i$ and $V$, respectively, and a constant value $N_0$ for $N(t)$ as described in Section 4.3. Although these approximations might result in robust estimates that are not necessarily globally optimal, the experimental results using these choices were surprisingly good as described in Section 7.

The correlates of visual attention in the model are brought about by the non-linear operation of the matrix $G$ on the estimation process. $G$ 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 high residuals, $G$ allows the model to ignore the corresponding outliers in the input $\mathbf{I}$, thereby enabling it to robustly estimate the state $\mathbf{r}$. By ignoring the outliers, the recognition system is able to 'focus attention' on a familiar object and estimate its identity in the presence of occlusions and background clutter. This is illustrated with concrete examples in the experimental results section.

To understand how the model can perform segmentation, consider the case where the image contains two familiar objects, one occluding the other. During the robust estimation process, the 'dominant' object (generally, the one in the foreground) is estimated and recognized first, and the remaining parts of the input image are treated as outliers. These outliers in turn contain a crude *segmentation* of the occluder and they can thus be used to subsequently 'focus attention' on the occluder to 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 Eq. (22) and repeating the estimation process, one can obtain robust estimates of the image region(s) that were previously treated as outliers. Such a sequential recognition process is somewhat similar to the process of 'switching attention' from one object to another in a visual scene. This process can in principle be carried out until all regions of the image have been segmented and recognized. We illustrate this process with concrete examples in the experimental results section.

## 7. Experimental results

### 7.1. 2D Recognition

To illustrate the ability of the Kalman filter model to learn and recognize static objects based solely on their appearance, we used grayscale images of size 105 × 105 pixels, depicting five 3D objects, for training the model (Fig. 5(a)). The generative matrix $U$ was of size $11025 \times 5$. The model was thus forced to generate predictions (reconstructions) of the input images based on only five basis images that form the columns of $U$, resulting in a significant reduction in dimensionality, from the 11025-dimensional input image space to a five-element state vector $\mathbf{r}$ (this is not very surprising given that only five training objects were used). The elements of the matrix $U$ were initialized to small random values and each column was normalized to length one, as were the input image vectors. For learning static inputs, the prediction matrix $V$ is the identity matrix since we may use $\bar{\mathbf{r}}(t) = \hat{\mathbf{r}}(t-1)$ and $M(t) = N(t-1)$. Furthermore, as described in Sections 4.3 and 5.3, we used scalar variances for the various covariance matrices and approximated the Kalman gain $N(t)U^T\Sigma(t)^{-1}$ with the simpler form $(N_0/\sigma^2)U^T$, where $N_0/\sigma^2$ was set to 0.2 with $\bar{\mathbf{m}} = 0$. This results in an iterative Kalman filter that converges, after a few iterations, to the optimal state estimate $\hat{\mathbf{r}}$ for a given static input. After convergence of the model for each input, the matrix $U$ was updated according to the simplified learning rule in Eq. (18). The learning rate $\alpha$ was initialized to 0.8 and subsequently decreased by dividing with 1.08 after each pass through the training set of images. Figs. 5 and 6 summarize the ongoing effects of this training process. After training, the model was tested on images of various objects (Fig. 7). The behavior of the model on objects that it has encountered previously was as expected, with almost zero reconstruction errors at all pixels, indicating correct prediction and recognition (Fig. 7(a)). The model shows a moderate ability to generalize to occluded or incomplete inputs, such as in (b).

A better alternative for handling such cases is to use the robust form of the Kalman filter as we shall illustrate in Section 7.5. 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 the *categorization* of novel input stimuli and the assignment of inputs to their closest object classes. On the other hand, the residual image (rightmost image) accentuates the differences between the training object and the new stimulus, preventing a mis-identification of the new stimulus as

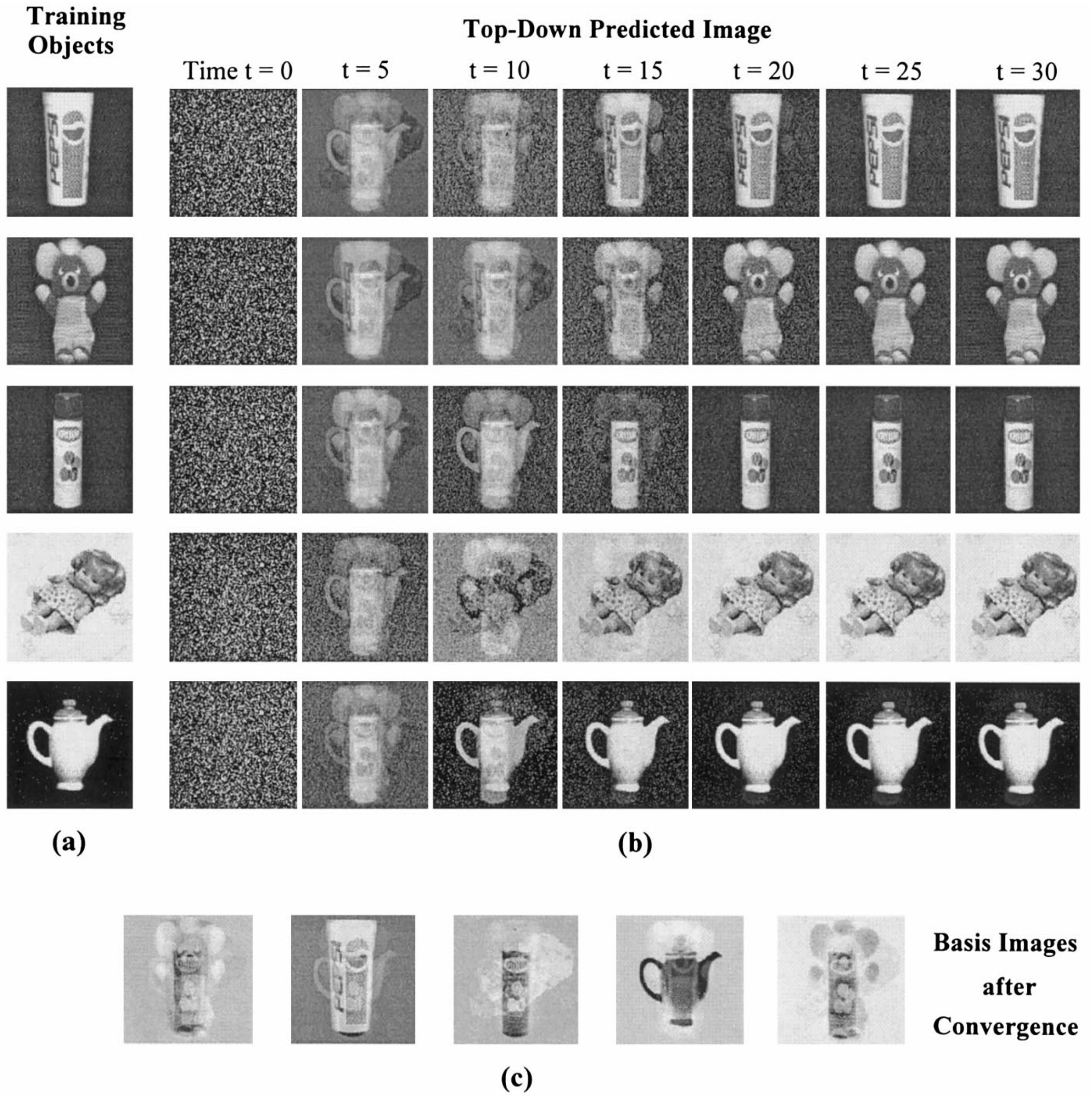

Fig. 5. Learning internal models of objects. (a) The five objects used for training a Kalman filter whose matrix $U$ was initialized to random values. (b) The evolution of the learning process, showing a relatively rapid increase in prediction accuracy after each exposure to the input stimuli. Fig. 6 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 $\bar{\mathbf{r}}(t)$, give rise to different approximations of the input images, as shown in (b).

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 Fig. 7(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 at many locations in general imply that the presented stimulus is novel. If the novel stimulus is deemed to be behaviorally important, it can be made part of the model's repertoire of known objects by allowing the residual errors to drive the adaptation of

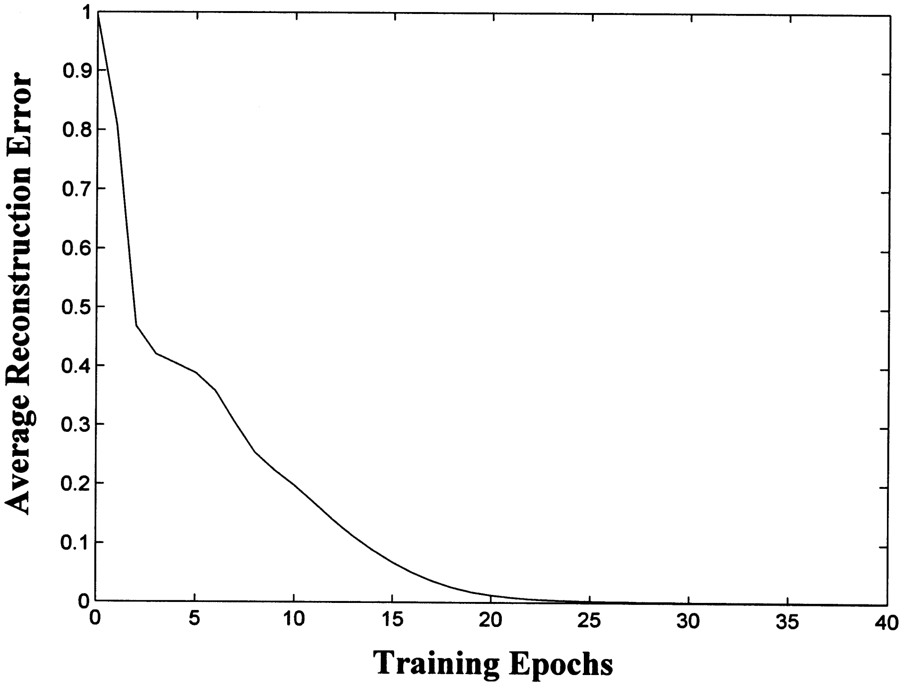

Fig. 6. Learning curve for Experiment 1. The graph shows the average error in image reconstruction (or prediction error), measured as sum of squared pixel-wise errors, across the five training objects as a function of number of exposures to the set of objects.

the matrix $U$ as specified by the learning rule in Eq. (18).

### 7.2. View-based recognition of 3D objects

In a second experiment, we evaluated the ability of the model to recognize 3D objects by training it on 36 2D views of two objects, each view 10° azimuth apart from the next (Fig. 8(a)). Such an approach, which is similar to Tarr and Pinker's multiple-views-plus-transformation (MVPT) theory of recognition (Tarr & Pinker, 1989), has also been advocated by Poggio and Edelman (1990); Edelman and Poggio (1991) and others (Murase & Nayar, 1995; Rao & Ballard, 1995; Black & Jepson, 1998), and is consistent with some object recognition studies in the monkey by Logothetis, Pauls, Bülthoff and Poggio (1994) and in humans by Bülthoff and others (Bülthoff, Edelman & Tarr, 1995). For computational efficiency, only the $32 \times 32$ image patches from the central image region were used for training (other regions can be analyzed by neighboring modules in a hierarchical estimation scheme—see, for example, Rao & Ballard, 1997a). The matrix $U$ was of the size $1024 \times 50$.

As shown in Fig. 8, after training, the model produced accurate predictions (reconstructions) of the training images 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 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 & Bischof, 1996), as seen in the mediocre prediction and relatively large residuals at many locations. This is handled via robust estimation (Sections 6 and 7.5). Finally, a completely novel object generates an 'average' image and large residuals as in the previous section.

### 7.3. Spatiotemporal recognition results

The next experiment was intended to verify the ability of the model to learn spatiotemporal internal models of possibly articulated stimuli. The model was trained on an image sequence depicting a set of hand gestures (Fig. 9). Each image was grayscale and of size $75 \times 75$ pixels. The matrices $U$ and $V$ (of the size $5625 \times 15$ and $15 \times 15$, respectively) were initialized to small random values and the model was trained using Eqs. (18) and (19). The learning rates $\alpha$ and $\beta$ were initialized to one and decreased gradually by dividing with 1.0025 at each iteration. The constant gain $N_0/\sigma^2$ for the filter in Eq. (20) was set to 0.2. Some of the basis images (columns of $U$) obtained after training are shown in Fig. 9 (bottom row).

Fig. 10 illustrates the prediction and recognition of the gesture sequence using the learned internal model. The trained Kalman filter was initialized to the zero vector, causing large residual errors (third row) at the initial time step. The errors are however corrected

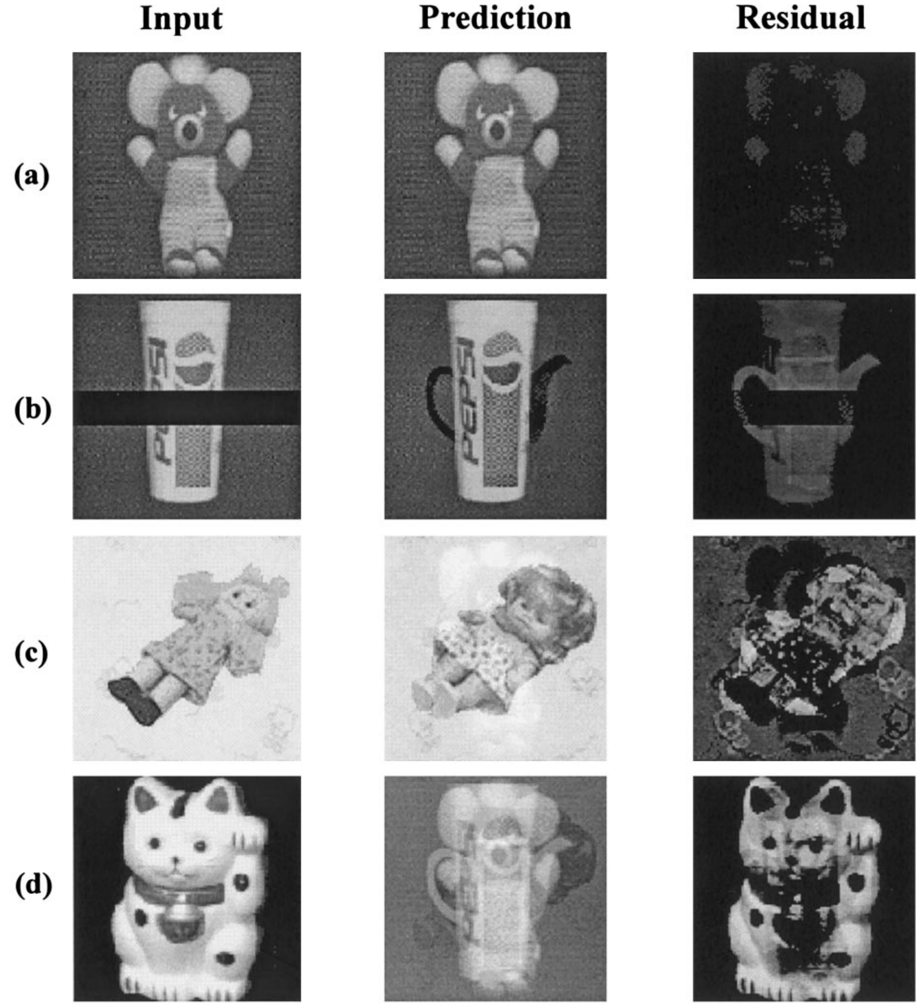

Fig. 7. Using internal models for object recognition, hypothesis verification and novelty detection. The internal models of objects learned in Fig. 5 were tested by using various input images and observing the response of the filter. (a) When an object in the training set is input (left), the prediction 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 6 (see Figs. 12 and 13). (c) A novel object (a doll) that resembles a training object (another doll) causes the filter to predict the closest resembling training object, namely, the doll used during training. 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 (using Eq. (18)) in case the object is deemed relevant to the recognition system.

rapidly due to the Kalman filter dynamics, resulting in relatively accurate predictions at subsequent time steps. An interesting exercise, marked by the arrow in the lower panel of Fig. 10, 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 is evident from the accurate predictions and low residual errors in the subsequent time steps.

### 7.4. Hidden state and perceptual aliasing

An important problem that arises during the estimation of the internal state of an observed system is that of 'hidden state' (McCallum, 1996) or 'perceptual aliasing' (Whitehead & Ballard, 1991; Chrisman, 1992) in partially observable environments. This problem has received much attention in the reinforcement learning literature (for a review, see Kaelbling, Littman & Moore, 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 Fig. 11(a), 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

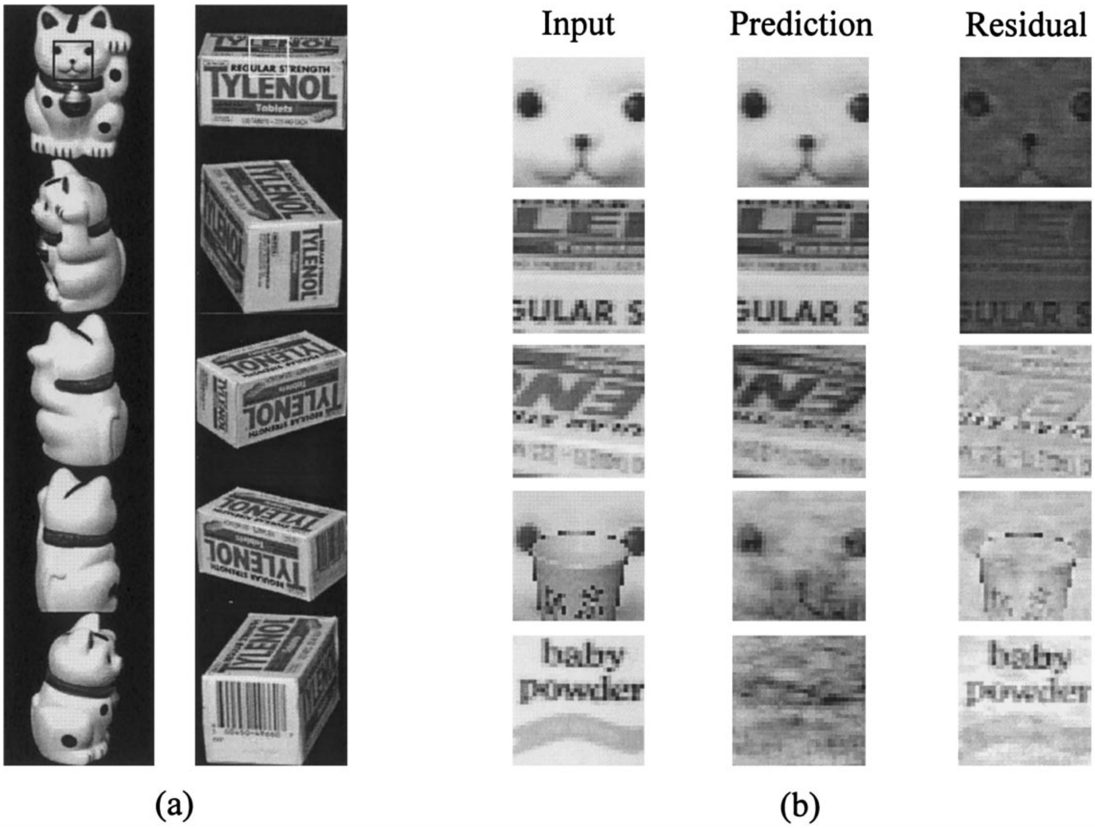

Fig. 8. View-based recognition of 3D objects. (a) shows five of the 36 training views used for learning the generative matrix $U$ for two different 3D objects. The trained filter was then tested on 36 intermediate views for each object. Only the image region demarcated by the box was used for training to preserve computational efficiency; information from other regions can be integrated, for example, using a hierarchical estimation scheme (Rao & Ballard, 1997a). (b) shows some examples of the responses generated by the trained filter.

current state of the input environment and the next prediction to be generated (Fig. 11(b)). One needs to make use of prior contextual information in order to correctly predict the next input.

Fig. 11(c) and (d) show how the model with a five-element state vector can learn to disambiguate the aliased inputs. The five basis images (columns of $U$) learned by the model, after several exposures to the input training sequence, is shown in (c). The learned matrices $U$ and $V$ together allow the model to disambiguate the identical inputs at time steps 2 and 4, as shown in (d). The vertical bars within the dotted boxes represent the model's state predictions $\bar{\mathbf{r}}_1$ and $\bar{\mathbf{r}}_3$ for time steps 2 and 4. Positive values are denoted by bars oriented upwards and negative values by bars oriented downwards. The significant differences between these two vectors show that the model has learned to represent the aliased input as two different states, allowing very different predictions $\bar{\mathbf{r}}_2$ and $\bar{\mathbf{r}}_4$ at the next time steps when multiplied by $V$. However, despite these differences, the representations $\bar{\mathbf{r}}_1$ and $\bar{\mathbf{r}}_3$ were learned by the model in such a manner that they generate the same image when multiplied by the generative matrix $U$.

### 7.5. Robust recognition, attention and segmentation

To evaluate the robust form of the model, we used the objects in Fig. 8 as the training set. 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 model to refine its robust estimate by gradually pruning away the outliers, as the model 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. Fig. 12(a) depicts how the model can reject outliers and produce an accurate prediction of an occluded object (compare with Fig. 8(b)).

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. 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 Eq. (22) and repeating

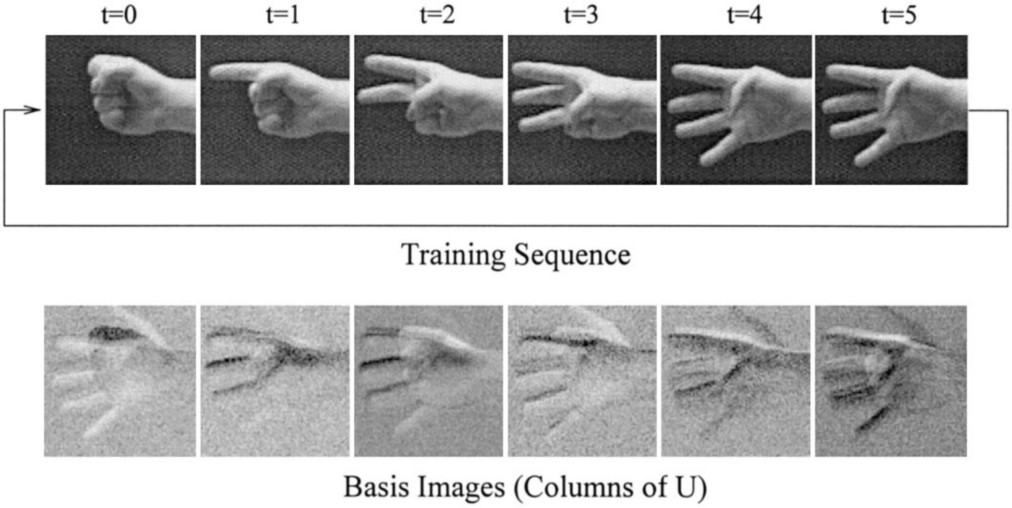

Fig. 9. Learning sequences of gestures. The top row shows a cyclic image sequence of hand gestures used to train a Kalman filter. The bottom row shows six of the 15 basis images (columns of the matrix $U$) learned after exposure to the cyclic training sequence.

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 Fig. 12(b), where the image is a combination of the two training objects in Fig. 8. The model 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).

Results from a second experiment using images with slightly more complex forms of occlusions and clutter are shown in Fig. 13. Static grayscale images of size $65 \times 105$ depicting two 3D objects were used for training the model with the matrix $U$ of size $6825 \times 5$ (Fig. 13(a)). As shown in (b), the model was successful in segmenting and recognizing the training object in spite of occlusion and background clutter. The case where one training object is occluding another is shown in (c). Both objects were successfully recognized by the model whereas the standard least-squares Kalman filter was unable to resolve either of the two objects as shown in the image at the extreme right.

To illustrate attention and segmentation during spatiotemporal recognition, we trained the model on three image sequences: (1) a horizontal bar moving downwards; (2) a vertical bar moving to the right, and (3) an expanding circle. Each sequence consisted of four $38 \times 38$ images. The generative matrix $U$ and the prediction matrix $V$ were initialized to random $1444 \times 15$ and $15 \times 15$ matrices, respectively. These matrices were adapted according to Eqs. (18) and (19) during repeated exposures to the training sequences. In the first test after training, 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 Fig. 14(a), the model produced relatively accurate predictions of the noisy images, when it was primed with the first image of the expanding circle sequence.

A more interesting case involving ambiguous stimuli is shown in Fig. 14(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 model was trained on both a horizontal bar moving downwards as well as a vertical bar moving rightwards. Given ambiguous stimuli containing both these stimuli, the model interprets the input differently depending on the initial 'attentional' priming input. As shown in Fig. 14(b), when the initial input is the first image from the horizontal bar sequence, the model 'pays attention' only to the horizontal bar as it moves downwards, ignoring the vertical bars which are treated as outliers. On the other hand, when the initial priming input is a vertical bar as shown in (c), the model interprets the same input sequence as a vertical bar moving rightwards, not 'paying attention' to the extraneous horizontal bars in the image sequence. These results illustrate how a learned internal model can cause the same stimulus to be perceived differently depending on certain priming inputs that can engage and 'lock' the predictive filter to certain aspects of the input. Such a mechanism may help provide explanations for visual illusions such as the Vase-or-Faces illusion and bi-stable percepts such as the Necker's cube phenomenon.

In a second set of experiments involving spatiotemporal recognition, we tested the trained model from Fig. 9 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

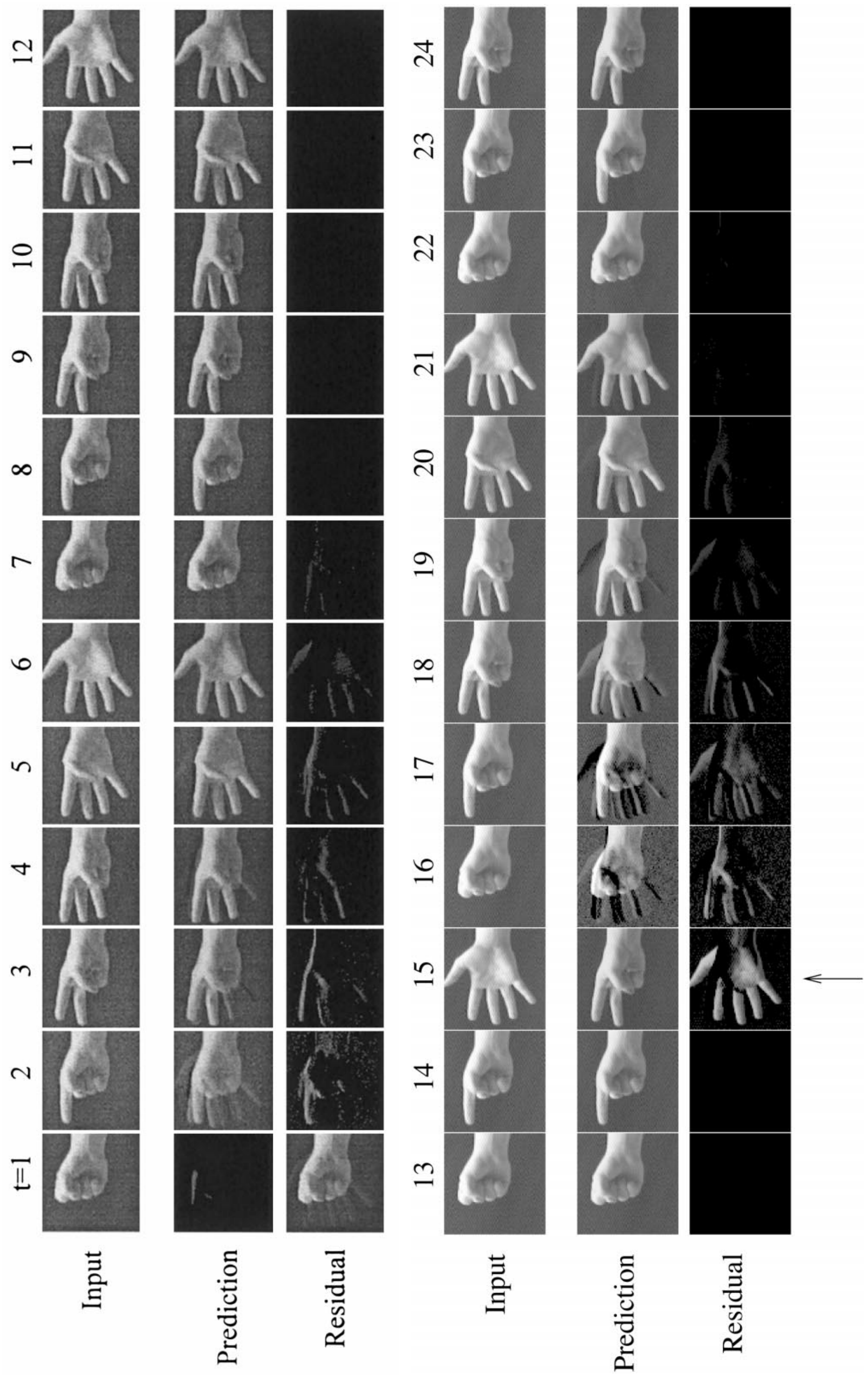

Fig. 10. Tracking and recognizing gestures. (Top panel) After initialization to the zero state vector, the filter rapidly corrects its prediction (second row) such that the initially large residual errors (third row) become appreciably small within the first few time steps. (Bottom panel) If the sequence 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.

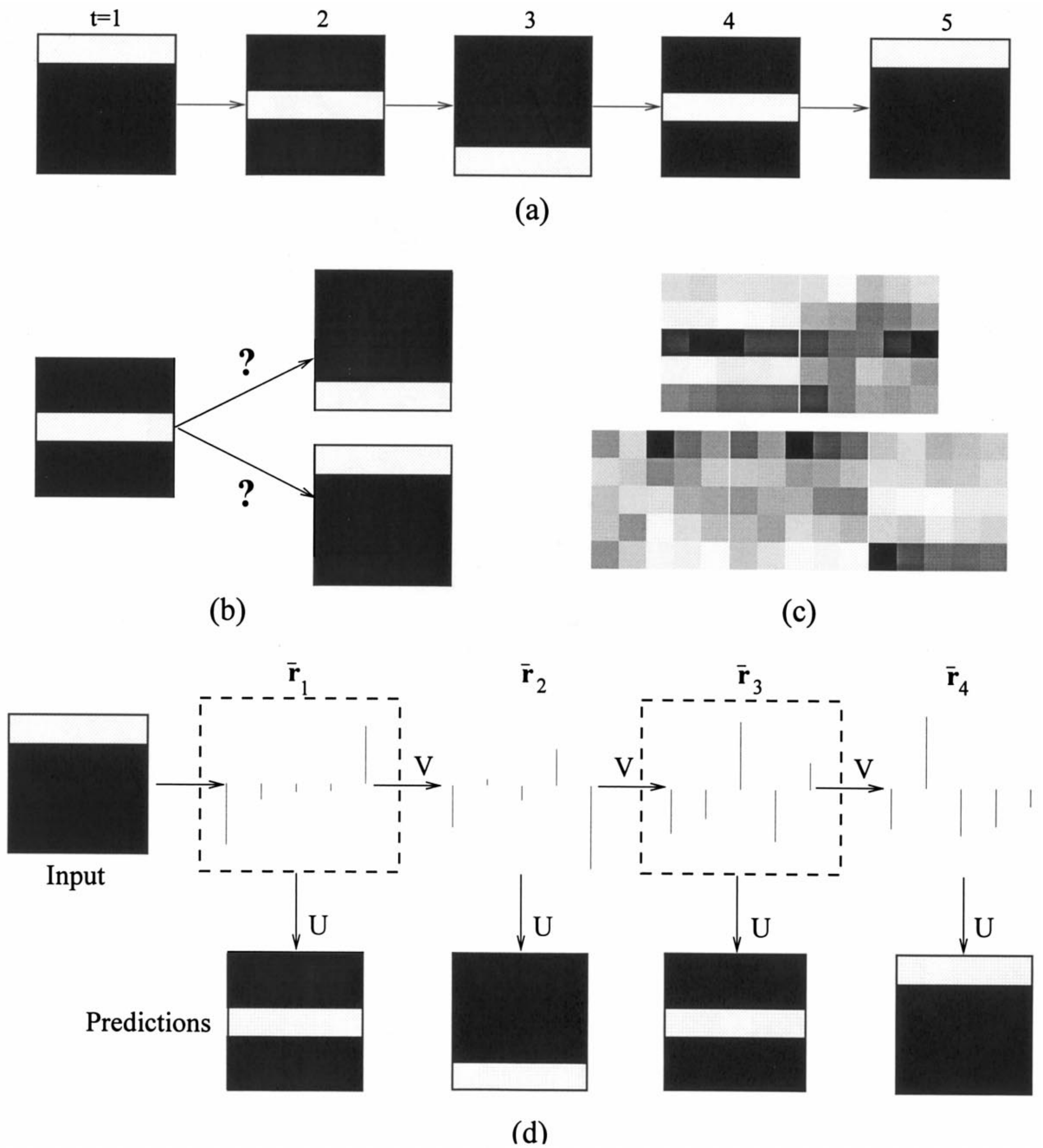

Fig. 11. Learning to disambiguate aliased inputs. (a) shows an image sequence depicting a horizontal bar, first moving down and then up. Note that the same image is encountered at time steps 2 and 4, but different images are to be predicted at the next time step, as shown in (b). This is the problem 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 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 $\bar{\mathbf{r}}_1$ and $\bar{\mathbf{r}}_3$ for 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.

of squared residual errors. The model was initialized with the first occluded gesture image in the sequence. As shown in Fig. 15, the model 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 model converged to stable estimates of the gesture images as shown in the bottom panel of Fig. 15. The occluding objects were also successfully segmented as shown in the last row of images in the figure.

## 8. Discussion

In this paper, we have suggested that the problem of visual perception can be viewed as one of optimally estimating the internal state of the visual environment with the help of an internal model that is learned directly from the input images. We derived a mathematically rigorous model of visual perception and learning using Bayesian principles (Freeman, 1994; Knill & Richards, 1996; Kersten, 1999) and the statistical theory of Kalman filtering. 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. Experimental results were provided to demonstrate the potential usefulness of such a model in understanding specific aspects of visual perception such as:

1. How internal models of objects and dynamic stimuli can be learned given only their input images.
2. How these learned internal models can be used for recognition, categorization, hypothesis verification, novelty detection, and prediction.
3. How the pervasive problem of perceptual aliasing may be resolved by learning unambiguous internal representations.
4. How top–down expectations and bottom–up signals can be integrated to recognize and segment objects in the presence of occlusions and background clutter, and
5. How objects of interest can be attended to in the presence of other objects or noise in the input stream.
6. How ambiguous stimuli may be parsed differently depending on how the recognition system is 'primed.'

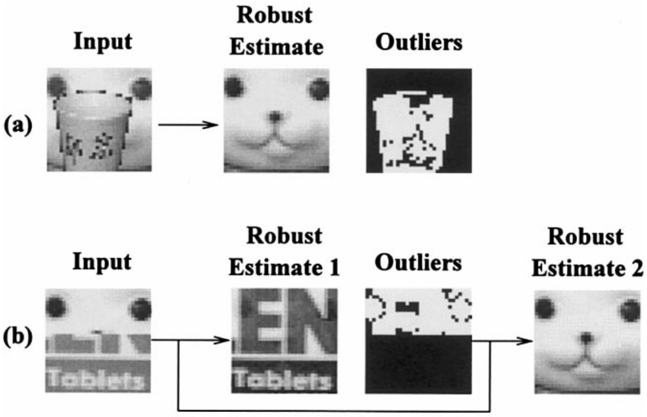

Fig. 12. Robust recognition, attention and segmentation. (a) depicts the robust estimation of object identity in the presence of an occlusion. The portions of the input image treated as outliers (the diagonal of the gating matrix $G$) are shown in white in the rightmost image. (b) demonstrates the case where the input contains combinations of the training objects (same objects as in Fig. 8). The model 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.

Unlike some previous appearance-based approaches such as those relying on PCA or SVD, the basis vectors in the proposed approach can be non-orthogonal and overcomplete. Rather than being restricted to static inputs, the approach allows appearance-based dynamic models of spatiotemporal image sequences to be learned on-line. Also, instead of being a linear purely feedforward function of the input as in previous approaches, the state vector is optimized on-line to suit the choice of the basis vectors. An additional favorable property of the approach is that it allows appropriate prior distributions for the model parameters to be hand-picked or learned so that the basis vectors can capture the higher-order statistics of the data rather than being restricted to pairwise statistical correlations as in the case of PCA. For example, a straightforward extension to the present model is to use a rectified Gaussian prior on the state vector rather than a Gaussian prior (see Rao & Ballard, 1997b). Other alternatives for the prior on the state vector and the basis matrices can be found in Harpur and Prager (1996); Lewicki and Sejnowski (1998); Olshausen and Field (1996) and Rao and Ballard (1997a).

### 8.1. Some weaknesses and limitations of the approach

An obvious shortcoming of the approach 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 non-linear extended Kalman filter case (see for e.g. Rao & Ballard, 1997a). 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 discussed 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 using appropriate prior distributions on model parameters such as a rectified Gaussian prior. 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

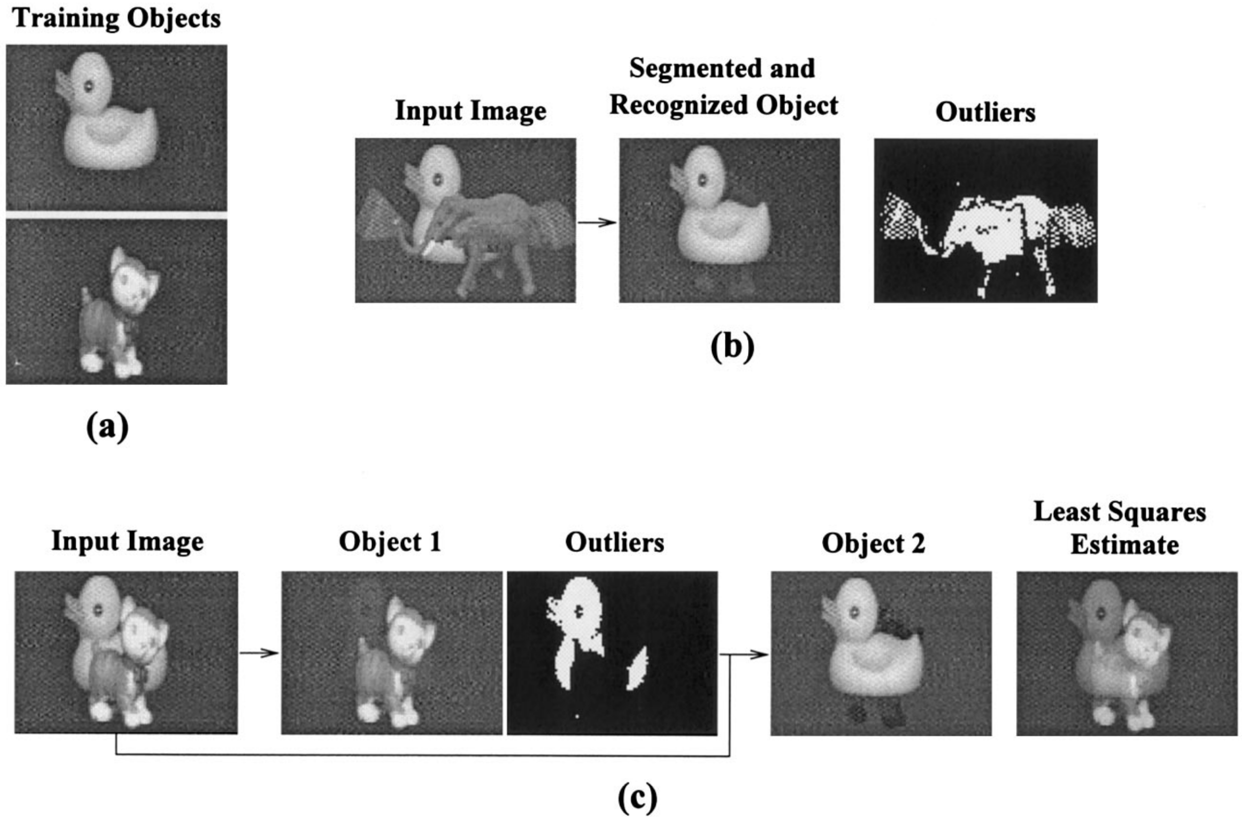

Fig. 13. Robust recognition: Experiment 2. (a) Images used to train the model. (b) Occlusions, background clutter, and other forms of noise are treated as outliers (white regions in the third image, depicting the diagonal of the gating matrix $G$). This allows the model to simultaneously segment and 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 model 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 estimation 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.

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. In addition, some of the complex neural responses that appear nonlinear when a neuron is viewed in isolation can be explained within a hierarchical model as occurring due to feedback inhibition and other limited forms of local nonlinear interactions (Rao & Ballard, 1997a, 1999).

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 (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 & Blake, 1996). However, this limitation is handled in the present model by allowing objects other than the primary one to be treated as outliers. These outliers can be subsequently recognized in a sequential fashion as demonstrated in Section 6.

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 (Eq. (5)). 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 & 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 7.4.

### *8.2. Invariance to transformations*

An important problem that we have not addressed in this paper due to space constraints is the issue of

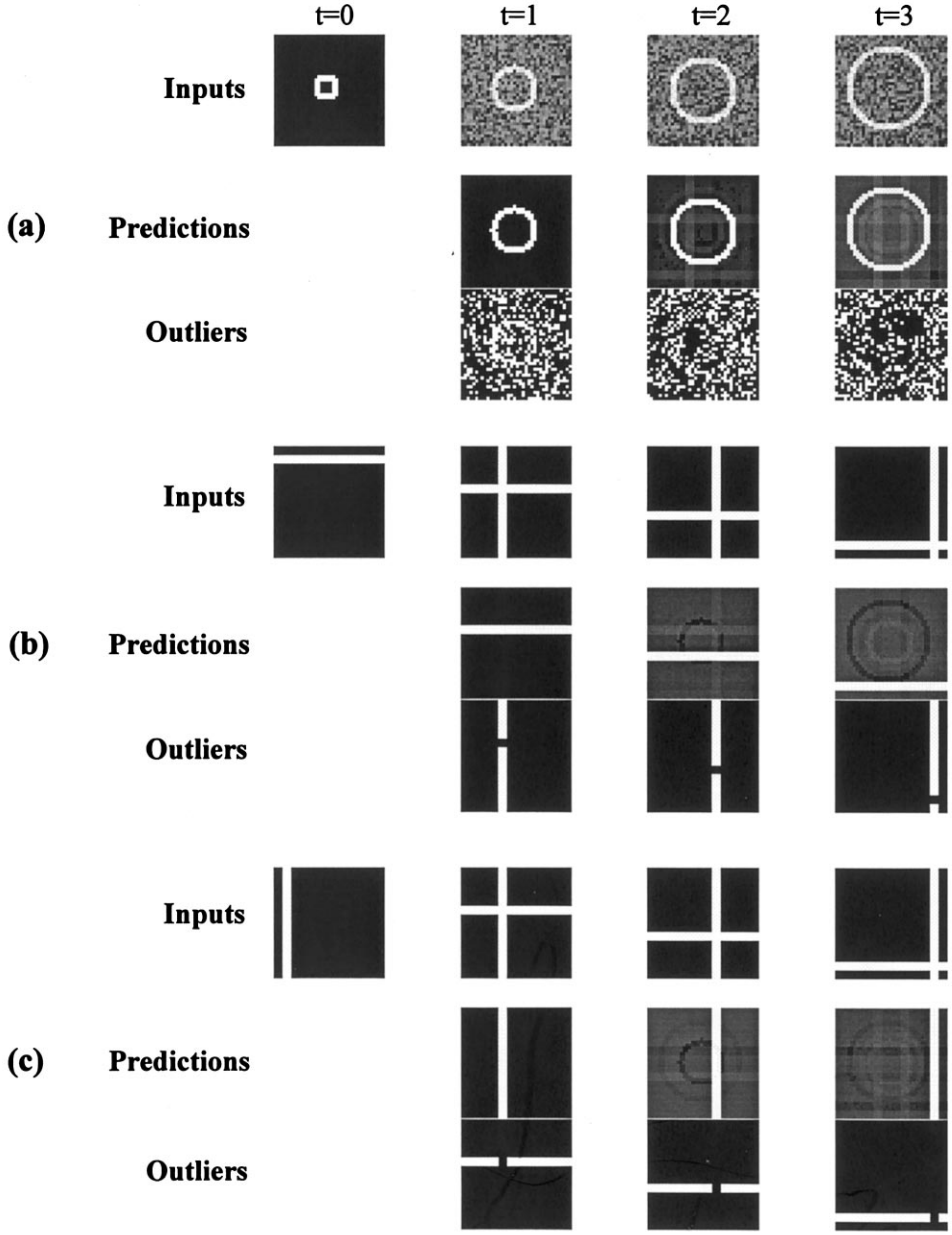

Fig. 14. Robust segmentation and recognition of noisy and ambiguous spatiotemporal stimuli. (a) demonstrates the ability of the robust filter to tolerate additive white noise in the input images by treating noisy pixels as outliers. (b) and (c) show how the same ambiguous stimuli at time steps $t = 1$ 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.

transformation invariance: how can the state estimates calculated by the Kalman filter ('What') be made invariant to object transformations ('Where') such as translations, rotations, and scaling? A simple but attractive solution is to model the transformed image $\mathbf{I}(\mathbf{x})$ as a function of a previously encountered reference image $\mathbf{I}(\mathbf{0})$. Here, $\mathbf{x}$ is a vector denoting a distributed representation of the relative transformation ('Where') with respect to the reference image. In particular, one can expand the transformed image $\mathbf{I}(\mathbf{x})$ in a Taylor series about an original reference point $\mathbf{0}$:

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

For small transformations, the higher order terms can be ignored and their effect can be modeled as stochastic noise:

$$\Delta\mathbf{I}(t) = \frac{\partial \mathbf{I}(\mathbf{0})}{\partial \mathbf{x}}\mathbf{x}(t) + \mathbf{n}(t) \qquad (24)$$

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 $J = \partial\mathbf{I}(\mathbf{0})/\partial\mathbf{x}$ can be approximated as a linear function of the

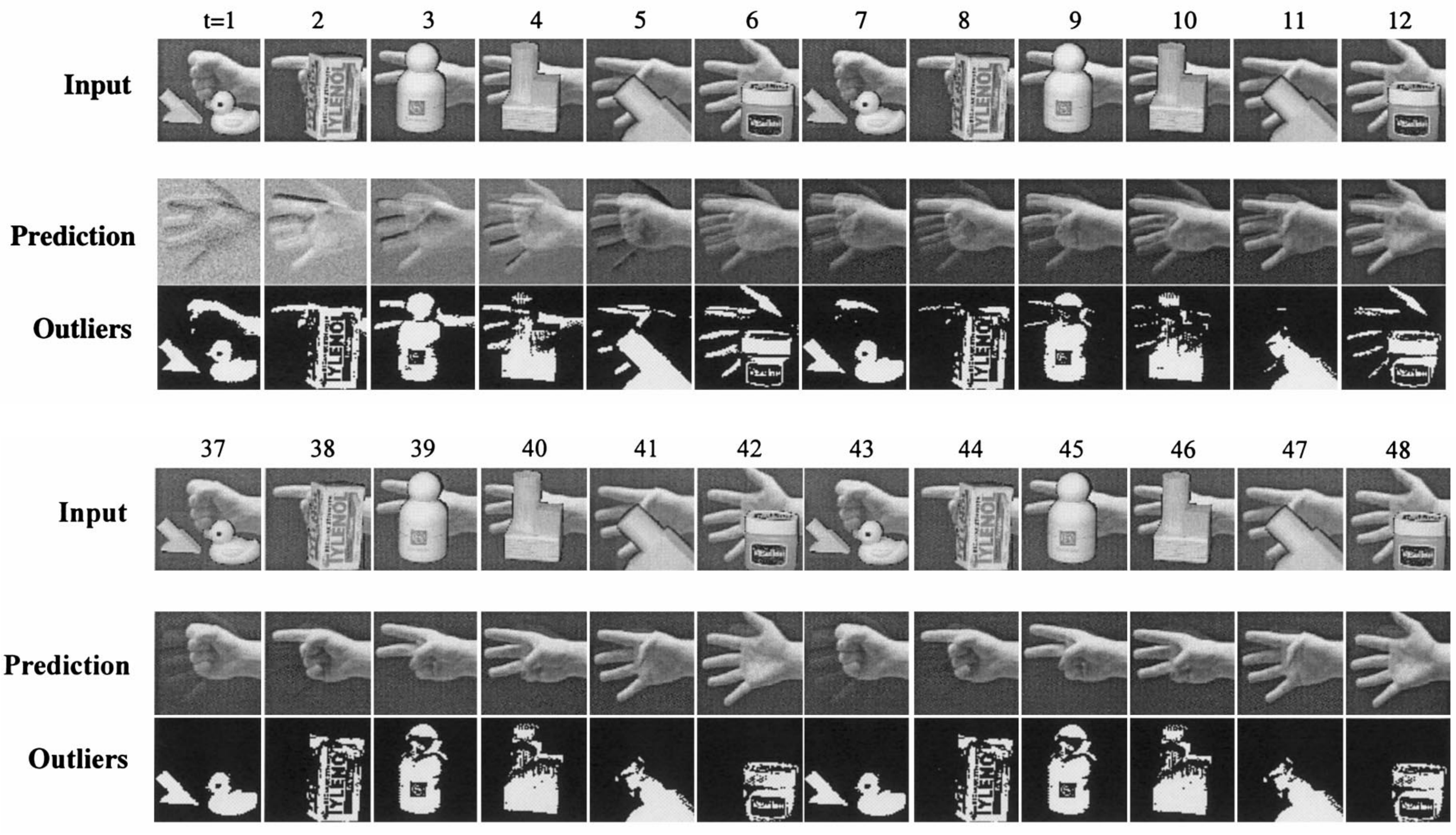

Fig. 15. Robust segmentation and recognition of occluded gestures. The filter from Fig. 9 that was trained on a cyclic image sequence of hand gestures was tested for robustness in the presence of occlusions and clutter. The top panel shows the initial transient phase of the robust filtering process (after 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.

reference image $\mathbf{I}(\mathbf{0})$. In particular, if $J_i$ is the $i$th column of $J$ we may use $J \cong D_i\mathbf{I}(\mathbf{0})$ where $D_i$ is an $n \times n$ matrix whose rows are differential operators.

It is a straightforward exercise to formulate an optimization function based on the generative model in Eq. (24) and derive an optimal estimation rule for the current transformation state $\mathbf{x}$ and a learning rule for the matrices $D_i$ (see Rao & Ballard, 1998a). 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 without regard to object specific features such as visual markings or color of the cup that are generally irrelevant to motor programming. Finally, the problem of large image transformations can be handled in at least two ways: (a) one may view the differential operators $D_i$ as generators of *lie transformation groups* and use a generative model based on a matrix exponential (see Rao & Ruderman, 1999), or (b) one may use a hierarchical estimation framework involving cooperative estimation of object state and relative transformation at multiple scales (Black & Jepson, 1998), similar to the hierarchical scheme suggested for image features in (Rao & Ballard, 1997a, 1999). Interestingly, the computational dichotomy between the estimation of 'What' and 'Where' parameters in such a model parallels the well-known segregation between the ventral occipitotemporal and the dorsal occipitoparietal pathways observed in the primate neocortex (Ungerleider & Mishkin, 1982; Mishkin, Ungerleider & Macko, 1983; Van Essen & Maunsell, 1983; Van Essen, 1985).

### *8.3. Hierarchical estimation and prediction*

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, Willsky & Benveniste, 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 & Levialdi, 1986), wavelets (Daubechies, 1992), and scale-space theory (Lindeberg, 1994) have found wide applications in computer vision and image processing.

The Kalman filter-based model studied herein can be extended to the hierarchical case where (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). We refer the interested reader to Rao and Ballard (1997a, 1999) for more details.

### *8.4. The optimal estimation model and the visual cortex*

The visual cortex has been previously characterized as a roughly hierarchical network composed of many distinct interconnected areas (Van Essen & Maunsell, 1983; Felleman & 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, makes the visual cortex especially well-suited to implement a hierarchical Kalman filter-like prediction and estimation mechanism. For instance, the feedback connections from a higher area may carry the predictions $U\bar{\mathbf{r}}$ of lower level neural activities $\mathbf{I}$, while the feed-forward connections may convey to the higher level the differences or *residuals* $(\mathbf{I} - U\bar{\mathbf{r}})$ between the predictions and the actual lower level activities (Rao & Ballard, 1997a, 1999). These residuals would allow the visual cortex to compute robust optimal estimates of events occurring in the visual environment based on a hierarchical and distributed internal model. The internal model, as encoded by the parameters $U$ and $V$, could be instantiated within the synaptic weights of neurons located in specific cortical laminae (see for instance Rao & Ballard, 1997a) and could be learned or 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, Unnikrishnan & Pandya, 1987; Albus, 1991; Mumford, 1992; Pece, 1992; Pentland, 1992; Hinton, Dayan, Frey & Neal, 1995; Kawato, Hayakawa & Inui, 1993; Dayan, Hinton, Neal & Zemel, 1995; Softky, 1996).

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, Seltzer & Barbas, 1988), a reasonable question to ask is whether a given approach to cortical function is general enough to be uniformly applicable to different cortical areas without regard to input modality. A reassuring feature of the optimal estimation model is that it is independent of the type of input signals being estimated. Thus, the possibility exists that in addition to visual signals, the approach can also be applied to the estimation and prediction of, for example, auditory, olfactory or motor signals as well. Exploring this possibility remains a promising subject for future investigations.

## Acknowledgements

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

## References

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

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

Ballard, D. H. (1997). *Introduction to natural computation*. Cambridge, MA: MIT Press.

Barlow, H. B. (1985). Cerebral cortex as model builder. In *Models of the visual cortex* (pp. 37–46). New York: Wiley.

Barlow, H. B. (1994). What is the computational goal of the neocortex? In C. Koch, & J. L. Davis, *Large-scale neuronal theories of the brain* (pp. 1–22). Cambridge, MA: MIT Press.

Bell, A. J., & Sejnowski, T. J. (1997). The 'independent components' of natural scenes are edge filters. *Vision Research*, *37*(23), 3327–3338.

Black, M. J., & Jepson, A. D. (1998). Eigentracking: robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, *26*(1), 63–84.

Blake, A., & Yuille, A. (1992). *Active vision*. Cambridge, MA: MIT Press.

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

Brunelli, R., & Poggio, T. (1993). Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(10), 1042–1052.

Bryson, A. E., & Ho, Y.-C. (1975). *Applied optimal control*. New York: Wiley.

Buhmann, J. M., Lades, M., & Malsburg, C. v. d. (1990). Size and distortion invariant object recognition by hierarchical graph matching. In *Proceedings of the IEEE IJCNN, vol. II* (pp. 411–416). New York: IEEE Neural Networks Council San Diego.

Bülthoff, H. H., Edelman, S. Y., & Tarr, M. J. (1995). How are three-dimensional objects represented in the brain? *Cerebral Cortex*, *5*(3), 247–260.

Cantoni, V., & Levialdi, S. (1986). Pyramidal systems for computer vision, *Proceedings of a NATO advanced research workshop*, Maratea, Italy, May 5–9, 1986. Berlin: Springer.

Chatfield, C., & Collins, A. J. (1980). *Introduction to multivariate analysis*. New York: Chapman and Hall.

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

Chrisman, L. (1992). Reinforcement learning with perceptual aliasing. *Proceedings of the eleventh national conference on artificial intelligence* (pp. 183–188). Menlo Park, CA: AAAI Press.

Creutzfeldt, O. D. (1977). Generality of the functional structure of the neocortex. *Naturwissenschaften*, *64*, 507–517.

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

Daugman, J. G. (1988). Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Procedure*, *36*(7), 1169–1179.

Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(11), 1148–1161.

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The Helmholtz machine. *Neural Computation*, *7*, 889–904.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, *39*, 1–38.

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

Edelman, S., & Poggio, T. (1991). Models of object recognition. *Current Opinion in Neurobiology*, *1*(2), 270–273.

Felleman, D. J., & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, *1*, 1–47.

Feller, W. (1968). *An introduction to probability theory and its applications, vol. 1*. New York: Wiley.

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

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

Freeman, W. T. (1994). The generic viewpoint assumption in a framework for visual perception. *Nature*, *368*, 542–545.

Grimson, W. E. L. (1990). *Object recognition by computer: The role of geometric constraints*. Cambridge, MA: MIT Press.

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

Hackbusch, W. (1985). *Multi-grid methods and applications*. Springer-Verlag.

Hallam, J. (1983). Resolving observer motion by object tracking. In *Proceedings of the 8th international joint conference on artificial intelligence Vol. 2* (pp. 792–798). Los Altos, CA: William Kaufmann.

Harpur, G. F., & Prager, R. W. (1996). Development of low-entropy coding in a recurrent network. *Network*, *7*, 277–284.

Harth, E., Unnikrishnan, K. P., & Pandya, A. S. (1987). The inversion of sensory processing by feedback pathways: a model of visual cognitive functions. *Science*, *237*, 184–187.

Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science*, *268*, 1158–1161.

Huber, P. J. (1981). *Robust statistics*. New York: Wiley.

Humphrey, N. (1992). *A history of the mind*. New York: Simon and Schuster.

Huttenlocher, D. P., & Ullman, S. (1987). Recognizing solid objects by alignment. In International Conference on Computer Vision.

Isard, M., & Blake, A. (1996). Contour tracking by stochastic propogation of conditional density. In *Proceedings of the fourth european conference on computer vision (ECCV)* (pp. 343–356). New York: Springer.

Jolliffe, I. T. (1986). *Principal component analysis*. New York: Springer-Verlag.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Transactions of the ASME Journal of Basic Engineering*, *83*, 95–108.

Kalman, R. E. (1960). A new approach to linear filtering and prediction theory. *Transactions of the ASME Journal of Basic Engineering*, *82*, 35–45.

Kawato, M., Hayakawa, H., & Inui, T. (1993). A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network*, *4*, 415–422.

Kersten, D. (1999). High level vision as statistical inference. In M. Gazzaniga, *The cognitive neurosciences* (2nd ed.). Cambridge, MA: MIT Press (to appear).

Knill, D. C., & Richards, W. (1996). *Perception as Bayesian inference*. Cambridge, UK: Cambridge University Press.

Kohonen, T. (1988). *Self-organization and associative memory* (2nd ed.). Springer-Verlag, Berlin.

Lamdan, Y., & Wolfson, H. J. (1988). Geometric hashing: a general and efficient model-based recognition scheme. In *International conference on computer vision* (pp. 238–249). Washington, DC: IEEE Computer Society Press.

Leonardis, A., & Bischof, H. (1996). Dealing with occlusions in the eigenspace approach. In *Proceedings of the CVPR* (pp. 453–458). Los Alamitos, CA: IEEE Computer Society Press.

Lewicki, M. S., & Sejnowski, T. J. (1998). Learning nonlinear overcomplete representations for efficient coding. In M. I. Jordan, M. J. Kearns, & S. A. Solla, *Advances in neural information processing systems 10* (pp. 556–562). Cambridge, MA: MIT Press.

Lindeberg, T. (1994). *Scale-space theory in computer vision*. Netherlands: Kluwer Academic Publishers.

Logothetis, N. K., Pauls, J., Bülthoff, H. H., & Poggio, T. (1994). View-dependent object recognition by monkeys. *Current Biology*, *4*, 401–414.

Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, *31*, 355–395.

MacKay, D. M. (1956). The epistemological problem for automata. In *Automata studies* (pp. 235–251). Princeton, NJ: Princeton University Press.

Matthies, L., Kanade, T., & Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, *3*, 209–236.

Maybeck, P. S. (1979). *Stochastic models, Estimation, and Control, vol. I and II*. New York: Academic Press.

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

Mel, B. (1996). SEEMORE: a view-based approach to 3-D object recognition using multiple visual cues. In D. Touretzky, M. Mozer, & M. Hasselmo, *Advances in neural information processing systems 8* (pp. 865–871). Cambridge, MA: MIT Press.

Mishkin, M., Ungerleider, L. G., & Macko, K. A. (1983). Object vision and spatial vision: two cortical pathways. *Trends in Neuroscience*, *6*, 414–417.

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

Murase, H., & Nayar, S. K. (1995). Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision*, *14*, 5–24.

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Nelson, R. C., & Selinger, A. (1998). A Cubist approach to object recognition. *International conference on computer vision* (pp. 614–621). New Dehli, India: Narosa Publishing House.

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

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, *37*(23), 3311–3325.

O'Reilly, R. C. (1996). The LEABRA model of neural interactions and learning in the neocortex. Ph.D thesis, Department of Psychology, Carnegie Mellon University.

Pandya, D. N., Seltzer, B., & Barbas, H. (1988). Input–output organization of the primate cerebral cortex. In H. D. Steklis, & J. Erwin, *Comparative primate biology, volume 4: neurosciences* (pp. 39–80). New York: Alan R. Liss.

Pece, A. E. C. (1992). Redundancy reduction of a Gabor representation: a possible computational role for feedback from primary visual cortex to lateral geniculate nucleus. In I. Aleksander, & J. Taylor, *Artificial neural networks 2* (pp. 865–868). Amsterdam: Elsevier Science.

Pentland, A. P. (1992). Dynamic vision. In G. A. Carpenter, & S. Grossberg, *Neural networks for vision and image processing* (pp. 133–159). Cambridge, MA: MIT Press.

Picton, T. W., & Stuss, D. T. (1994). Neurobiology of conscious experience. *Current Opinion in Neurobiology*, *4*, 256–265.

Poggio, T., & Edelman, S. (1990). A network that learns to recognize 3D objects. *Nature*, *343*, 263–266.

Rao, R. P. N., & Ballard, D. H. (1995). An active vision architecture based on iconic representations. *Artificial Intelligence* (*Special Issue on Vision*), *78*, 461–505.

Rao, R. P. N., & Ballard, D. H. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, *9*(4), 721–763.

Rao, R. P. N., & Ballard, D. H. (1997). Efficient encoding of natural time varying images produces oriented space-time receptive fields. Technical Report 97.4, National Resource Laboratory for the study of Brain and Behavior, Computer Science Department, University of Rochester.

Rao, R. P. N., & Ballard, D. H. (1998). Development of localized oriented receptive fields by learning a translation-invariant code for natural images. *Network*: *Computation in Neural Systems*, *9*(2), 219–234.

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive field effects. *Nature Neuroscience 2*(1) (in press).

Rao, R. P. N., & Ruderman, D. L. (1999). Learning Lie transformation groups for invariant visual perception. In M. Kearns, S.A. Solla, & D. Cohn, *Advances in neural information processing systems 11*. Cambridge, MA: MIT Press (in press).

Schiele, B., & Crowley, J. L. (1996). Object recognition using multidimensional receptive field histograms. In *Proceedings of the fourth European conference on computer vision (ECCV)* (pp. 610–619). New York: Springer.

Schmid, C., & Mohr, R. (1996). Combining greyvalue invariants with local constraints for object recognition. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 872–877). Los Alamitos, CA: IEEE Computer Society Press.

Simoncelli, E. P., Freeman, W. T., Adelson, E. H., & Heeger, D. J. (1992). Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, *38*(2), 587–607.

Softky, W. R. (1996). Unsupervised pixel-prediction. In D. Touretzky, M. Mozer, & M. Hasselmo, *Advances in neural information processing systems 8* (pp. 809–815). Cambridge, MA: MIT Press.

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*, 11–32.

Tarr, M. J., & Pinker, S. (1989). Mental rotation and orientation-dependence in shape recognition. *Cognitive Psychology*, *21*, 233–282.

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

Ungerleider, L., & Mishkin, M. (1982). Two cortical visual systems. In D. Ingle, M. Goodale, & R. Mansfield, *Analysis of visual behavior* (pp. 549–585). Cambridge, MA: MIT Press.

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

Van Essen, D. C. (1985). Functional organization of primate visual cortex. In A. Peters, & E. G. Jones, *Cerebral cortex*, vol. 3 (pp. 259–329). New York, NY: Plenum.

Viola, P. (1996). Complex feature recognition: A Bayesian approach for learning to recognise objects. Artificial Intelligence Lab Memo 1591, MIT, Cambridge, MA.

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