Dynamic Appearance-Based Recognition*

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

Abstract

We describe a hierarchical appearance-based method for learning, recognizing, and predicting arbitrary spatiotemporal sequences of images. The method, which implements a robust hierarchical form of the Kalman filter derived from the Minimum Description Length (MDL) principle, includes as a special case several well-known object encoding techniques including eigenspace methods for static recognition. Successive levels of the hierarchical filter implement dynamic models operating over successively larger spatial and temporal scales. Each hierarchical level predicts the recognition state at a lower level and modifies its own recognition state using the residual error between the prediction and the actual lower-level state. Simultaneously, on a longer time scale, the filter learns an internal model of input dynamics by adapting its generative and state transition matrices at each level to minimize prediction errors. The resulting prediction/learning scheme thereby implements an on-line form of the well-known Expectation-Maximization (EM) algorithm from statistics. We present experimental results demonstrating the method's efficacy in mediating robust spatiotemporal recognition in a variety of scenarios containing varying degrees of occlusions and clutter.

1. Introduction

Vision is fundamentally a dynamic process. The images impinging on the retina are seldom comprised of a sequence of unrelated static signals but rather, reflect measurements of a coherent stream of events occurring in the distal environment. The regularity in the structure of the visual input stream stems primarily from the constraints imposed on event outcomes by various physical laws of nature in conjunction with the observer's own choices of actions on the immediate environment. Under such circumstances, the goal of a visual system becomes one of estimating (and predicting) the "hidden" states of an observed dynamic system (the visual environment). Accurate estimation of hidden state then becomes synonymous with accurate recognition of input stimuli. 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.

The notion of hidden state is already implicit in various appearance-based methods for computer vision that have received much attention over the past few years [17, 11, 13, 9, 2, 10]. For example, in eigenspace-based methods, an input image is characterized by the vector of coefficients obtained by projecting the image along the directions given by the dominant eigenvectors of the input covariance matrix. This vector is compared with those of the training objects in order to determine the closest match. Within a more general context, this vector of coefficients can be regarded as a linear estimate of the *hidden state* that generated the given input (by multiplication with the eigenvector matrix - see Section 2). In order to handle occlusions and clutter, alternate "robust" methods of computing the coefficients have been proposed [2, 10], which, in the light of the above discussion, are equivalent to re-estimating the true hidden state. Unfortunately, an eigenspace-based generative model, which uses orthogonal eigenvectors, may deviate significantly from the true generative model that characterizes the image generation process. It is therefore not surprising that recent work on modeling the response properties of neurons in the visual cortex has focused on alternative generative models using goals ranging from maximizing the "sparseness" of the coefficients [12] to making the coefficients statistically independent (for example, independent component analysis (ICA) [1]).

In this paper, we propose a hierarchical appearance-based method for learning, recognizing, and predicting spatiotemporal sequences of input images. As a special case of the method, one obtains the standard eigenspace and related methods for recognition of static images. The method, which essentially implements a robust hierarchical form of

*This work was supported by NIH/PHS research grant no. 1 P41 RRO9283. The author is grateful to Dana Ballard for many useful discussions and suggestions during the course of this work.

1063-6919/97 $10.00 © 1997 IEEE

the Kalman filter, generalizes eigenspace-based methods by allowing hidden state transitions from one time step to the next, thereby allowing prediction and modeling of time-varying processes. In addition, the basis vectors are no longer constrained to be the mutually orthogonal eigenvectors of the input covariance matrix but rather, are *learned* in an unsupervised manner by minimizing an optimization function based on the Minimum Description Length (MDL) principle. The MDL formulation computes penalized maximum a posteriori (MAP) estimates of the parameters (the basis vectors) and the hidden state. Penalizing the cost of the model helps prevent overfitting and encourages generalization to novel situations, unlike eigenspace-based methods which attempt to minimize the gross reconstruction error. Furthermore, the hierarchical architecture prescribed by the method allows dynamic modeling of multiscale phenomena, which are ubiquitous in nature, and perhaps more closely approximate the image formation process than single level generative models such as those embodied in the eigenspace approach.

2. Stochastic Image Generation Models and Optimal Estimation

We begin by first relating the proposed method to principal component/eigenspace methods. Consider the problem of encoding a collection of $n \times 1$ input vectors $\mathbf{I}^1, \mathbf{I}^2, \ldots$ (for example, images) 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 $E(\mathbf{I}\mathbf{I}^T)$ computed from zero-mean samples of input data. This is essentially the eigenspace technique of Turk and Pentland [17] and Murase and Nayar [11]. In this case, the columns of U are orthogonal to each other and the "hidden" state vector corresponding to a given input $\mathbf{I}$ is a $k \times 1$ coefficient (or response) vector:

$$\mathbf{r} = U^T\mathbf{I} \tag{1}$$

Since k is generally much smaller than n, the response vector $\mathbf{r}$ is an efficient compressed representation of the input image. A reconstruction of the input image $\widehat{\mathbf{I}}$ can be generated from $\mathbf{r}$ by using the following relation which simply inverts the transformation in Equation 1:

$$\widehat{\mathbf{I}} = U\mathbf{r} \tag{2}$$

It is well-known (see, for example, [14]) that the eigenvector matrix U minimizes the pixel-wise expected reconstruction error function:

$$J(U) = \sum_{i=1}^{n} (\mathbf{I}_i - U_i\mathbf{r})^2 = (\mathbf{I} - U\mathbf{r})^T(\mathbf{I} - U\mathbf{r}) \tag{3}$$

(where $\mathbf{I}_i$ denotes the ith pixel of $\mathbf{I}$ and U_i denotes the ith row of U) over all inputs subject to the constraint that the columns of U are orthogonal, $\mathbf{r}$ being specified as in Equation 1.

Unfortunately, optimal compression via an eigenvector expansion does not guarantee optimal recognition because: (a) the mechanisms underlying the generation of inputs $\mathbf{I}$ (assuming they can be modeled by Equation 2) do not need to use mutually orthogonal column vectors in U; (b) $\mathbf{r}$ need not be specified as a purely one-shot feedforward function of U^T and $\mathbf{I}$ as in Equation 1; (c) eigenspace or principal component methods are suitable only when the data is well-described by a Gaussian cloud. Recent work by Field [7] and others strongly suggest that natural images form a highly non-Gaussian distribution that cannot be described satisfactorily by orthogonal basis vectors. (d) Perhaps most importantly, eigenvector-based methods 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 [12].

Our approach to appearance-based recognition is inspired by the following *stochastic and linear generative model* that is already implicit in Equation 2 above:

$$\mathbf{I} = U\mathbf{r} + \mathbf{n}_{bu} \tag{4}$$

where $\mathbf{I}$ denotes an input image and $\mathbf{n}_{bu}$ is a "bottom-up" stochastic noise process that accounts for the differences between the reconstruction $U\mathbf{r}$ and the image $\mathbf{I}$. We assume $\mathbf{n}_{bu}$ is Gaussian and $E(\mathbf{n}_{bu}) = 0$ with a covariance matrix specified by $E[\mathbf{n}_{bu}\mathbf{n}_{bu}^T] = \Sigma_{bu}$.

2.1. The Minimum Description Length Principle

Given the generative model in Equation 4, one can obtain estimates of the parameters U and $\mathbf{r}$ by minimizing the least squares error function in Equation 3. However, a potentially serious problem with standard least squares estimation is the possibility of overfitting the model parameters to the observed data. In the extreme case, the parameters may become virtually identical with the training data ("rote memorization") and therefore will fail to generalize to new data. On the other hand, not fitting the parameters accurately to the observed data introduces bias into the representations. We must therefore walk the thin line between fitting but not overfitting our parameters to data in order to ensure proper generalization to novel situations. One promising way out of this "bias-variance dilemma" is to use the Minimum Description Length (MDL) Principle [15, 18]. Simply put, the MDL principle advocates balancing the cost of encoding the data given the use of a model with the cost of specifying the model itself (cost is defined in terms of the length of the encoding in bits).

Given a description language $\mathcal{L}$, data $\mathcal{D}$, and model parameters $\mathcal{M}$, the MDL principle advocates minimizing the following cost function [18]:

$$J(\mathcal{M},\mathcal{D}) = |\mathcal{L}(\mathcal{M},\mathcal{D})| = |\mathcal{L}(\mathcal{D}|\mathcal{M})| + |\mathcal{L}(\mathcal{M})| \quad (5)$$

$|\cdot|$ denotes length of the description. In our case, $\mathcal{D}$ consists of the current input image $\mathbf{I}$ and $\mathcal{M}$ consists of the parameters U and $\mathbf{r}$.

Given the true probability distribution (over discrete events) of the various terms in the above equations, the expected length of the optimal code for each term is given by Shannon's *optimal coding theorem* [16]:

$$|\mathcal{L}(x)| = -\log P(X = x) \quad (6)$$

where $P(X = x)$ denotes the probability of the discrete event x. Thus, in a Bayesian framework, $|\mathcal{L}(\mathcal{D}|\mathcal{M})|$ is simply the negative log-likelihood of the data given the model parameters and $|\mathcal{L}(\mathcal{M})|$ is the negative log of the prior model parameter distributions. Minimizing the description length function J is thus equivalent to computing the *maximum a posteriori* (MAP) estimates of the model parameters $\mathcal{M}$ given the data $\mathcal{D}$, additionally taking into account the cost of the model.

We now formulate an MDL-based optimization function for estimating U and $\mathbf{r}$. In order to do so, it is convenient to view the $n \times k$ generative weight matrix U as an $nk \times 1$ vector $\mathbf{u} = [U_1 U_2 \ldots U_n]^T$ where U_i denotes the ith row of U. Suppose that we have already computed a prediction $\overline{\mathbf{r}}$ of the current state $\mathbf{r}$ based on prior data. In particular, let $\overline{\mathbf{r}}$ be the mean of the current state vector *before* measurement of the input data at the current time instant. The corresponding covariance matrix is given by $E[(\mathbf{r}-\overline{\mathbf{r}})(\mathbf{r}-\overline{\mathbf{r}})^T] = M$. Similarly, let $\overline{\mathbf{u}}$ be the current estimate of the $\mathbf{u}$ calculated from prior data with covariance $E[(\mathbf{u}-\overline{\mathbf{u}})(\mathbf{u}-\overline{\mathbf{u}})^T] = S$.

Shannon's coding theorem relates code lengths to discrete probability distributions. Since we will be using continuous (Gaussian) distributions, we need to calculate the probability *mass* of a particular small interval of values around a given value x [18]. Using a trapezoidal approximation, we may estimate the mass under a continuous (in our case, Gaussian) density p in an interval of width w around a value x to be $P(X = x) \cong p(x)w$. For encoding the data given the model (corresponding to $|\mathcal{L}(\mathcal{D}|\mathcal{M})|$) using the Gaussian associated with this term, we assume w to be a constant infinitesimal width which yields (using Equation 6 and ignoring the constant terms due to the coefficients of the multivariate Gaussian):

$$|\mathcal{L}(\mathcal{D}|\mathcal{M})| = (\mathbf{I} - U\mathbf{r})^T \Sigma_{bu}^{-1} (\mathbf{I} - U\mathbf{r}) \quad (7)$$

For encoding the model parameters, a constant infinitesimal width w may be inappropriate since some values of the parameters may need to be encoded more accurately than others. For example, one could allow w to be a non-linear function of the model parameters in order to seek higher-order statistical structure than just linear, pairwise correlations [12]. The model cost then reduces to:

$$\begin{aligned} |\mathcal{L}(\mathcal{M})| &= (\mathbf{r}-\overline{\mathbf{r}})^T M^{-1} (\mathbf{r}-\overline{\mathbf{r}}) + (\mathbf{u}-\overline{\mathbf{u}})^T S^{-1} (\mathbf{u}-\overline{\mathbf{u}}) \\ &\quad + f(\mathbf{r}) + g(\mathbf{u}) \end{aligned}$$

The first two terms in the sum above arise from the prior Gaussian densities for $\mathbf{r}$ and $\mathbf{u}$ as given by $G(\overline{\mathbf{r}}, M)$ and $G(\overline{\mathbf{u}}, S)$, while the latter two terms are non-linear functions of $\mathbf{r}$ and $\mathbf{u}$ associated with w. For example, one could use $f(x) = g(x) = \alpha x^2$ to allow regularization and avoid overfitting [14]. Using a function such as $f(x) = \alpha \log(1 + x^2)$ causes higher-order correlations to be sought [12]. These functions are applied to all components x of a given vector $\mathbf{x}$ and the results are summed in the optimization function.

2.2. MDL-Based Kalman Filters

In this section, we use the MDL-based optimization function from the previous section to derive iterative Kalman filters for estimating $\mathbf{r}$ and U for static images. We extend this to the case of time-varying imagery in Section 3.

Given an input $\mathbf{I}$, the optimal estimate of current state $\mathbf{r}$ (with respect to a given U) after making a new measurement is a stochastic code whose mean $\widehat{\mathbf{r}}$ and covariance P can be obtained by setting $\frac{\partial J(\mathcal{M},\mathcal{D})}{\partial \mathbf{r}} = 0$ and solving for $\mathbf{r}$ $(= \widehat{\mathbf{r}})$:

$$\begin{aligned} \widehat{\mathbf{r}}(t) &= \overline{\mathbf{r}}(t) + PU^T \Sigma_{bu}^{-1} (\mathbf{I} - U\overline{\mathbf{r}}(t)) - Pf'(\overline{\mathbf{r}}(t)) \quad (8) \\ P(t) &= (M(t)^{-1} + U^T \Sigma_{bu}(t)^{-1} U + f''(\overline{\mathbf{r}}(t)))^{-1} \quad (9) \end{aligned}$$

where f' and f'' denote the first and second partial derivatives of f with respect to $\mathbf{r}$. These are in general *non-linear* functions of $\mathbf{r}$. Note that to obtain the above closed-form equations, we used a first-order Taylor series expansion of f' about $\overline{\mathbf{r}}(t)$. Equations 8 and 9 together implement an MDL-based *Kalman filter* [4] for updating the state estimate given prior estimates $\overline{\mathbf{r}}$ and M. For static images, we may use $\overline{\mathbf{r}}(t) = \widehat{\mathbf{r}}(t-1)$ and $M(t) = P(t-1)$ to obtain an iterative Kalman filter for estimating the optimal state corresponding to a given static input.

The optimal estimate of the generative matrix U (in its vector form $\mathbf{u}$) is obtained in a similar manner, given a particular state $\mathbf{r}$. First, note that $(\mathbf{I} - U\mathbf{r}) = (\mathbf{I} - R\mathbf{u})$ where R is the $n \times nk$ matrix given by:

$$R = \begin{bmatrix} \mathbf{r}^T & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{r}^T & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{r}^T \end{bmatrix} \quad (10)$$

Setting $\frac{\partial J(\mathcal{M},\mathcal{D})}{\partial \mathbf{u}} = 0$ and solving for $\mathbf{u}$ $(= \widehat{\mathbf{u}})$, we obtain the following Kalman filter-based "learning" rule for the mean and covariance of the optimal generative vector:

$$\begin{aligned} \widehat{\mathbf{u}}(t) &= \overline{\mathbf{u}}(t) + K_u(\mathbf{I} - R(t)\overline{\mathbf{u}}(t)) - P_u g'(\overline{\mathbf{u}}(t)) \quad (11) \\ P_u(t) &= (S(t)^{-1} + R(t)^T \Sigma_{bu}(t)^{-1} R(t) + g''(\overline{\mathbf{u}}(t)))^{-1} \end{aligned}$$

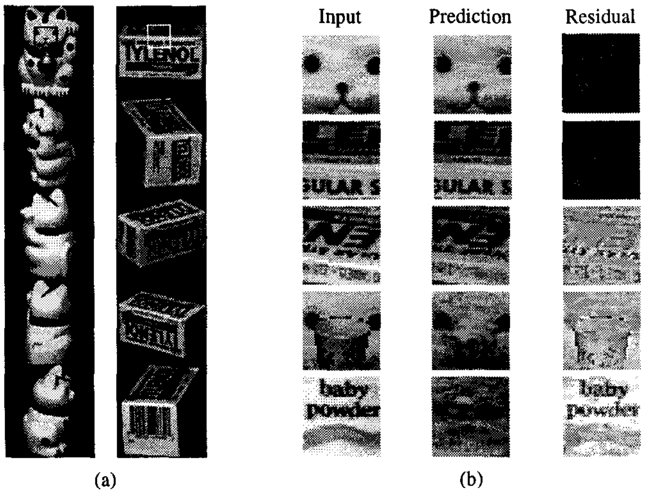

Figure 1. **Static Recognition Results.** (a) shows 5 of the 36 views of two different $3D$ objects from the Columbia database [11] used for learning the generative matrix U. Each view was 10° apart from the next. Only the 32 x 32 image region demarcated by the box was used for training to preserve computational efficiency; other regions are assumed to be analyzed by neighboring modules (see Section 4). U was initialized to a 1024×50 random matrix. (b) shows some examples of the responses generated by the trained filter. Training images produce accurate predictions (reconstructions) with low residual errors (top two rows). An intermediate view that is 5° from nearest training view generates 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 view 5° being from the nearest training view. The second to last row depicts how the effect of occlusions spreads globally [10], as seen in the mediocre prediction and relatively large residuals at most locations. This is handled via robust estimation (Section 2.3). Finally, a completely novel object generates an "average" image, and large residuals as shown in the last row. These residuals can be used to drive learning as in Equation 11, if the object is deemed to be important.

where $\overline{\mathbf{u}}(t) = \widehat{\mathbf{u}}(t-1)$, $K_u = P_u R(t)^T \Sigma_{bu}^{-1}$, $S(t) = P_u(t-1)$, and g' and g'' denote the first and second partial derivatives of g with respect to $\mathbf{u}$.

An interesting question is the issue of convergence of the overall filtering/learning scheme involving $\mathbf{r}$ and U. Fortunately, one can appeal to the well-known Expectation-Maximization (EM) algorithm from statistics [6] and allow the overall scheme to converge by using $U = \widehat{U}(t-1)$ in Equation 8 and $\mathbf{r}(t) = \widehat{\mathbf{r}}$ in the matrix $R(t)$ above, where $\widehat{\mathbf{r}}$ is the converged state estimate for the given static input. Note that the Kalman filter estimation of the state $\mathbf{r}$ can be related to the E-step in the EM algorithm while the adaptation of U using this estimate of $\mathbf{r}$ can be regarded as part of the M-step. Figure 1 illustrates the performance of a filter trained using the above estimation and learning rules. The magnitude of the residual error between the input image $\mathbf{I}$ and the filter's prediction $U\overline{\mathbf{r}}$ indicates the relative accuracy of a recognition hypothesis (low residuals imply correct recognition, high residuals implies novelty). This allows the filter to counter the pervasive problem of false positives common in most purely feedforward systems which lack the ability to "invert" their recognition estimates and verify their hypotheses.

2.3. Robust Kalman Filters and Outlier Rejection

In the previous section, we did not specify how the covariance Σ_{bu} is to be calculated. One possibility is to make it a constant matrix or even a constant scalar. Making Σ_{bu} constant however reduces the Kalman filter estimates to be essentially ordinary least-squares estimates, and it is well-known that least-squares estimation is highly susceptible to outliers or gross errors i.e. data points that lie far away from the bulk of the observed data [8]. The problem of outliers can be tackled using *robust estimation procedures* [8]. One commonly used procedure is *M-estimation* (Maximum likelihood type estimation), which involves minimizing a function of the form:

$$J'(U,\mathbf{r}) = \sum_{i=1}^{n} \rho(\mathbf{I}_i - U_i\mathbf{r}) \tag{12}$$

where ρ is normally taken to be a less rapidly increasing function than the square. This ensures that large residual errors (which correspond to outliers) do not influence the optimization of J, thereby "rejecting" the outliers. Note that when ρ equals the square operation, we obtain the standard least squares function. More interestingly, we obtain the following *weighted* least squares criterion also as a special case:

$$J'(U,\mathbf{r}) = (\mathbf{I} - U\mathbf{r})^T A(\mathbf{I} - U\mathbf{r}) \tag{13}$$

where A is an $n \times n$ diagonal matrix whose diagonal entries $A_{i,i}$ specify how to weight each residual $(\mathbf{I}_i - U_i\mathbf{r})$. An attractive choice for these weights is:

$$A_{i,i} = \min\left\{1, c/(\mathbf{I}_i - U_i\mathbf{r})^2\right\} \tag{14}$$

where c is a threshold parameter. Note that A clips the ith summand of optimization function J (Equation 13) to a constant value c whenever the ith residual exceeds the threshold c; otherwise, it sets it to the squared residual.

For the experiments in this paper, we substituted $\Sigma_{bu}^{-1} = A$ in Equation 7 and rederived the update equations. The corresponding *robust Kalman filter* for the state estimate is given by:

$$\begin{aligned}\widehat{\mathbf{r}}(t) &= \overline{\mathbf{r}}(t) + P(t)U^T G(t)(\mathbf{I} - U\overline{\mathbf{r}}(t)) - P(t)f'(\overline{\mathbf{r}}(t)) \\ P(t) &= (M(t)^{-1} + U^T G(t) U + f''(\overline{\mathbf{r}}(t)))^{-1}\end{aligned}$$

where $G(t)$ is an $n \times n$ diagonal "gating" matrix, whose diagonal entries at time instant t are given by:

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

The gating matrix G effectively filters out any high residuals, thereby allowing the Kalman filter to ignore the corresponding outliers in the input $\mathbf{I}$ and enabling it to robustly

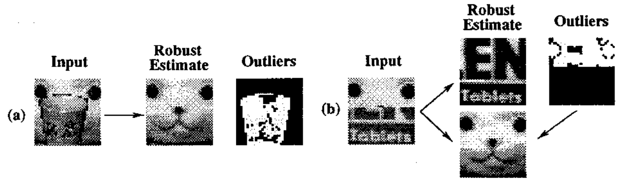

Figure 2. Recognition using Robust Kalman Filters. (a) depicts the estimation of object identity in the presence of an occlusion (compare with Figure 1 (b), fourth row). Portions of the input treated as outliers (diagonal of the gating matrix G) are shown in white on the right. (b) demonstrates the case where the input contains a combination of the training objects (same objects as in Figure 1). The identity of second object is retrieved by using the outlier mask produced by the first estimated object and repeating the estimation process.

estimate the state $\mathbf{r}$. In the experiments, the outlier threshold c was set equal to the sum of the mean plus k standard deviations of the current distribution of squared residual errors $(\mathbf{I}_i - U_i\mathbf{r})^2$. The parameter k was initialized to an appropriately large value (e.g. $k = 5$) and gradually decreased during each iteration to allow the filter to converge to a robust optimal estimate. Figure 2 provides examples of robust estimation in the presence of occlusions and structured noise in the input stream. As seen in the figure, the outliers (white) produce a crude *segmentation* of the occluder(s), which can subsequently be used to ascertain their identity. This is depicted in Figure 2 (b), where the image is a combination of the two training objects. The outlier mask, as given by the complement $1 - G$ of the gating matrix (after estimation of the first object), is subsequently used as the new gating matrix for extracting the identity of the second object (lower arrows in the figure).

3. Dynamic Generative Models and Learning Image Sequences

The generative model in Equation 4 describes how a given hidden state $\mathbf{r}$ is related to the observed image $\mathbf{I}$ via the generative matrix U plus additive noise. In order to model time-varying processes, we need to describe how the state $\mathbf{r}$ itself varies in time. One way this can be achieved is by assuming that $\mathbf{r}$ is a *Gauss-Markov random process* [4].

Given the state $\mathbf{r}(t-1)$, the transition to the state $\mathbf{r}(t)$ at the next time instant is modeled as:[1]

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

where V is the *state transition (or prediction) matrix* and $\mathbf{n}$ is Gaussian noise with mean $E[\mathbf{n}(t)] = \bar{\mathbf{n}}(t)$ and covariance $E[(\mathbf{n}(t) - \bar{\mathbf{n}}(t))(\mathbf{n}(s) - \bar{\mathbf{n}}(s))^T] = \Sigma(t)\delta(t,s)$ where δ is the Kronecker delta function equaling 1 if $t = s$ and 0 otherwise.

In the static recognition case, we used $\bar{\mathbf{r}}(t) = \hat{\mathbf{r}}(t-1)$ and $M(t) = P(t-1)$. It follows from Equation 15 above that in the dynamic case:

$$\begin{aligned} \bar{\mathbf{r}}(t) &= V\hat{\mathbf{r}}(t-1)) + \bar{\mathbf{n}}(t-1) \qquad (16) \\ M(t) &= VP(t-1)V^T + \Sigma(t-1) \qquad (17) \end{aligned}$$

In this dynamic form, the Kalman filter predicts one step into the future using Equation 16, corrects its prediction $\bar{\mathbf{r}}$ using Equation 8 to obtain $\hat{\mathbf{r}}$, and uses this corrected estimate $\hat{\mathbf{r}}$ to make its next state prediction. Note that the filter can predict an arbitrary number of steps into the future, although without new data, the uncertainty in prediction (as given by M) increases with each time step as suggested by Equation 17.

A final issue is the estimation of the state transition matrix V. Fortunately, one can derive an estimation procedure for V in manner analogous to that for U. Let $\mathbf{v}$ be the $k^2 \times 1$ vector obtained by collapsing the rows of the V. Suppose we have computed prior estimates $\bar{\mathbf{v}}$ and T of the mean and covariance of $\mathbf{v}$ (just as we did for $\mathbf{u}$ and $\mathbf{r}$). Augment the MDL-based optimization function in Equation 5 with the additional terms $(\mathbf{v} - \bar{\mathbf{v}})^T T^{-1}(\mathbf{v} - \bar{\mathbf{v}})$ and $h(\mathbf{v})$, where h is a non-linear function similar to f and g. Also, define the $k \times k^2$ matrix $\widehat{R}$ to be:

$$\widehat{R} = \begin{bmatrix} \hat{\mathbf{r}}^T & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{r}}^T & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \hat{\mathbf{r}}^T \end{bmatrix} \tag{18}$$

Notice that the state transition step can then be stated as $\bar{\mathbf{r}}(t+1) = V(t)\hat{\mathbf{r}}(t)) + \bar{\mathbf{n}}(t) = \widehat{R}(t)\mathbf{v}(t) + \bar{\mathbf{n}}(t)$. Differentiating $J(\mathcal{M}, \mathcal{D})$ with respect to $\mathbf{v}$ and setting the result to zero, we obtain the following update equations for the mean and covariance of $\mathbf{v}$:

$$\begin{aligned} \hat{\mathbf{v}}(t) &= \bar{\mathbf{v}}(t) + P_v\widehat{R}(t)^T M^{-1}\left[\mathbf{r}(t+1) - \bar{\mathbf{r}}(t+1)\right] \\ &\quad -P_v h'(\bar{\mathbf{v}}(t)) \qquad (19) \\ P_v(t) &= (T(t)^{-1} + \widehat{R}(t)^T M(t)^{-1}\widehat{R}(t) + h''(\bar{\mathbf{v}}(t)))^{-1} \end{aligned}$$

where $\bar{\mathbf{v}}(t) = \hat{\mathbf{v}}(t-1)$, $T(t) = P_v(t-1)$, and h' and h'' denote the first and second partial derivatives of h with respect to $\mathbf{v}$.

For the experiments in the paper, we used $\mathbf{r}(t+1) = \hat{\mathbf{r}}(t+1)$ in Equation 19 above, although the EM algorithm prescribes the use of $\hat{\mathbf{r}}(t+1|N)$, which is the optimal temporally *smoothed* state estimate [4] for time $t+1$ $(\leq N)$, given input data for each of the time instants $1, \dots, N$. The smoothed estimates are however computationally much more expensive and our preliminary experimental results indicate that the on-line estimates $\hat{\mathbf{r}}(t+1)$ may be used in their place in many cases. Figure 3 illustrates the recognition performance of a robust dynamic filter trained on an image sequence of hand gestures.

[1] We describe the linear case here for simplicity but the technique readily generalizes (via Taylor series approximations) to non-linear dynamic models [14], which yield *extended Kalman filters*.

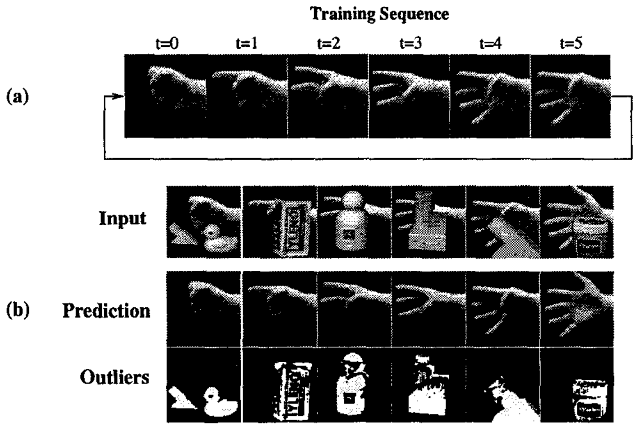

Figure 3. **Robust Segmentation and Recognition of Image Sequences.** (a) Cyclic image sequence of gestures used for training the matrices U and V of a dynamic filter. Each image was of size 75 × 75. The matrices U and V were of size 5625 × 15 and 15 × 15 respectively. (b) Robust prediction and tracking of gestures in the presence of various forms of occlusion and clutter. The outlier threshold c was computed at each time instant as the sum of the mean plus 0.3 standard deviations of the current distribution of squared residual errors. Results shown are those obtained after five cycles of exposure to the occluded gesture images.

4. Hierarchical Dynamic Models

Most natural phenomena manifest themselves over a multitude of spatial and temporal scales. For example, the rich class of stochastic processes possessing a $1/f^{\beta}$ power spectra exhibit statistical and fractal self-similarities that can be satisfactorily captured only in a multiscale framework [5]. There has consequently been much recent interest in multiscale signal processing methods, and techniques such as image pyramids, wavelets, and scale-space theory have found wide application in computer vision and image processing.

In the spirit of these multiscale methods, we propose, in this section, a method for learning hierarchical dynamic models where: (a) each hierarchical level uses the output state of its immediate predecessor as input, with only the lowest level operating directly on the image, 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. A further 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.

Consider the first hierarchical-level. Recall that we used a generative model of the form (Equation 4): $\mathbf{I} = U\mathbf{r} + \mathbf{n}_{bu}$. The higher level (in this case, the second level) uses an identical generative model except that instead of generating $\mathbf{I}$, it generates a composite vector whose components are the *top-down predictions* of the current states $\mathbf{r}$ of a group of spatially adjacent lower level modules:

$$\mathbf{r} = U^h_{i:j}\mathbf{r}^h + \mathbf{n}^h_{bu} \tag{20}$$

where $U^h_{i:j}$ represent the rows i through j of the higher level generative matrix U^h. In other words, $\mathbf{r}^h$ represents the higher level state that generates a long vector given by $U^h\mathbf{r}^h$; this vector is split into smaller vectors $U^h_{i:j}\mathbf{r}^h$ which act as top-down constraints on the various states $\mathbf{r}$ at the lower level.

To simplify notation, we use $\mathbf{r}_{td} = U^h_{i:j}\mathbf{r}^h$ and $\mathbf{n}_{td} = \mathbf{n}^h_{bu}$, and rewrite Equation 20 as:

$$\mathbf{r}(t) = \mathbf{r}_{td}(t) + \mathbf{n}_{td}(t) \tag{21}$$

We assume $E(\mathbf{n}_{td}(t)) = 0$ and $E[\mathbf{n}_{td}(t)\mathbf{n}_{td}(s)^T] = \Sigma_{td}(t)\delta(t,s)$. We modify the MDL-based optimization function to take the top-down information into account as follows:

$$\begin{aligned}|\mathcal{L}(\mathcal{D}|\mathcal{M})| &= (\mathbf{I}-U\mathbf{r})^T\Sigma_{bu}^{-1}(\mathbf{I}-U\mathbf{r}) + \\ &\quad (\mathbf{r}-\mathbf{r}_{td})^T\Sigma_{td}^{-1}(\mathbf{r}-\mathbf{r}_{td}) \qquad (22)\end{aligned}$$

Setting $\frac{\partial J(\mathcal{M},\mathcal{D})}{\partial \mathbf{r}} = 0$, we obtain the following new update equations for the mean and covariance of the state:

$$\begin{aligned}\widehat{\mathbf{r}}(t) &= \overline{\mathbf{r}}(t) + K_{bu}(\mathbf{I}(t) - U\overline{\mathbf{r}}(t)) + \\ &\quad K_{td}(\mathbf{r}_{td}(t) - \overline{\mathbf{r}}(t)) - P(t)f'(\overline{\mathbf{r}}(t)) \qquad (23)\\ P(t) &= (M(t)^{-1} + U^T\Sigma_{bu}^{-1}U + \Sigma_{td}^{-1} + f''(\overline{\mathbf{r}}(t)))^{-1}\end{aligned}$$

where $K_{bu} = P(t)U^T\Sigma_{bu}^{-1}$ and $K_{td} = P(t)\Sigma_{td}^{-1}$. All other quantities are updated as in Section 3. Note that Equation 23 implements an efficient trade-off between information from three different sources: the state prediction $\overline{\mathbf{r}}(t)$, the bottom-up data $\mathbf{I}$, and the top-down prediction $\mathbf{r}_{td}$. This trade-off is mediated by the bottom-up and top-down *Kalman gain* matrices K_{bu} and K_{td}, which can be intuitively interpreted as *signal-to-noise ratios.*

We can apply the method of Section 2.3 to the covariances Σ_{bu} and Σ_{td}, thereby making the hierarchical estimation procedure robust to outliers at the various levels. In addition, by appealing to a version of the EM algorithm, we may use $U = \widehat{U}$ and $V = \widehat{V}$, which are learned from data as described in Sections 2.2 and 3 respectively. The filter implements both a spatial and a temporal hierarchy in the following manner: (a) **Spatial Hierarchy**: The input vector to the filter at the second level is a single long vector formed by augmenting the state vectors $\overline{\mathbf{r}}$ of a set of adjacent modules at the lower level. Thus, successively higher levels in the filter analyze and predict over (exponentially) larger spatial extents. The highest level then has access to spatial information from the entire image, albeit in an abstract form, after having been processed by the lower levels. It can in turn influence processing at the lower levels via its generative (feedback) connections U. (b) **Temporal Hierarchy**: By decreasing exponentially the decay function f' in the Equation 23 for each successive level, one obtains a temporal hierarchy wherein higher levels decay at a slower rate

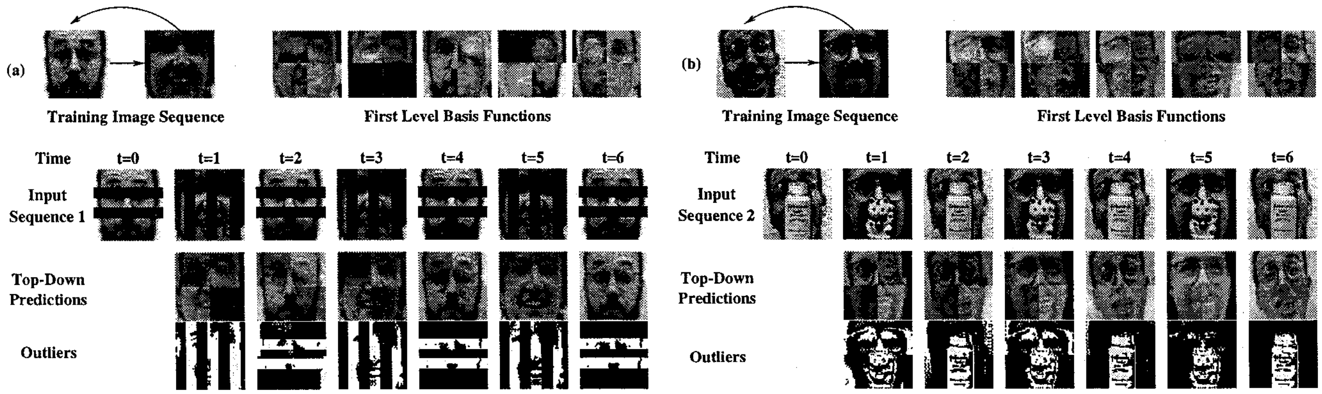

Figure 4. Hierarchical Recognition Results. (a) and (b) show the two training sequences and their corresponding spatial basis vectors (columns of U) at level 1 of a three-level hierarchical filter. At level 0, the 64×64 input image was partitioned into four equal 32×32 sub-images that served as input to four level 1 Kalman filter modules. The level 1 matrices U and V in each module were of size 1024×5 and 5×5 respectively. For each sequence, the five columns of the four first level U matrices are shown on the right as five composite images. A single level 2 module was used to estimate the states of the four level 1 modules. The level 2 matrix U was of size 20×5 and the matrix V was 5×5. (Below) Robust tracking of the two sequences in the presence of occlusions. As shown in the figure, outliers were detected and discounted for within the first four or five frames of the image sequence.

than the lower levels. Thus, the lowest level modules possess the shortest "memories" while the higher levels predict based on longer historical traces, taking into account events that occurred progressively further back in time. Figure 4 illustrates the performance of a three-level hierarchical architecture trained on image sequences depicting alternating facial expressions from four different persons.

5. Summary and Conclusions

This paper presents a new hierarchical appearance-based method for learning, recognizing, and predicting image sequences. The proposed method is based firmly on the information-theoretic MDL principle [15] and utilizes ideas from robust statistics [8] for deriving hierarchical Kalman filter estimators that can tolerate significant occlusion and clutter. Kalman filters have previously been used extensively in computer vision [3] and image processing [5]. However, many of these approaches employ hard-wired dynamic models inferred from a priori knowledge of the task at hand. This paper describes how hierarchical dynamic models can be *learned* from input data, thereby avoiding the need for explicit hand-built physical models of dynamic systems. Preliminary experiments using the described method have been promising. More detailed experiments are currently underway to rigorously evaluate the proposed method and further delineate its strengths and weaknesses.

References

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

[2] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV'96*, pages 329–342, 1996.

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

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

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

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

[7] D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

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

[9] D. Huttenlocher, R. Lilien, and C. Olson. Object recognition using subspace methods. In *ECCV'96*, pages 536–545, 1996.

[10] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *CVPR'96*, pages 453–458, 1996.

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

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

[13] R. Rao and D. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78:461–505, 1995.

[14] R. Rao and D. Ballard. Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, 9:805–847, 1997.

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

[16] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

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

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