??, ??, 1–29 (??)

EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation

MICHAEL J. BLACK
black@parc.xerox.com
Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304

ALLAN D. JEPSON*
jepson@vis.toronto.edu
Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3H5 Canada

Received ??. Revised ??.

Abstract.
This paper describes a new approach for tracking rigid and articulated objects using a view-based representation. The approach builds on and extends work on eigenspace representations, robust estimation techniques, and parameterized optical flow estimation. First, we note that the least-squares image reconstruction of standard eigenspace techniques has a number of problems and we reformulate the reconstruction problem as one of robust estimation. Second we define a "subspace constancy assumption" that allows us to exploit techniques for parameterized optical flow estimation to simultaneously solve for the view of an object and the affine transformation between the eigenspace and the image. To account for large affine transformations between the eigenspace and the image we define a multi-scale eigenspace representation and a coarse-to-fine matching strategy. Finally, we use these techniques to track objects over long image sequences in which the objects simultaneously undergo both affine image motions and changes of view. In particular we use this "EigenTracking" technique to track and recognize the gestures of a moving hand.

Keywords: Eigenspace methods, Robust estimation, View-based representations, Gesture Recognition, Parametric models of optical flow, Tracking, Object recognition, Motion analysis.

1. Introduction

This paper addresses the problem of tracking a previously viewed object in an image sequence as the view of the object changes due to its motion or the motion of the camera. Traditional optical flow techniques treat an image region simply as moving "stuff" (Adelson and Bergen, 1991) and hence cannot distinguish between changes in viewpoint or configuration of the object and changes in position relative to the camera. Trackers that use templates of one form or another have a notion of the "thing" being tracked, but if the view changes significantly then the "thing" is no longer the same and tracking can fail. Recovering 3D motion or tracking a 3D model of an object are possible alternatives for tracking rigid objects, but for tracking and recognizing moving articulated objects such as human hands these solutions are computationally expensive. We would prefer the computational simplicity of working with 2D image-based models but we need to extend them to account for changing views or changing structure. We would like a view-based representation (or model) of objects with a small set of views and a method that will take an image and find both the view of the object and the transformation that maps the image onto the model. To achieve this,

* Also: Canadian Institute for Advanced Research

we combine lines of research from object recognition using eigenspaces, parameterized optical flow models, and robust estimation techniques into a novel method for tracking objects using a view-based representation.

View-based, or appearance-based, object representations have found a number of expressions in the computer vision literature, in particular in the work on eigenspace representations (Murase and Nayar, 1995), (Turk and Pentland, 1991). Eigenspace representations can provide a compact approximate encoding of a large set of images in terms of a small number of orthogonal basis images. These basis images span a subspace of the training set called the eigenspace and a linear combination of these images can be used to approximately reconstruct any of the training images. Previous work on eigenspace representations has focused on the problem of object recognition and has only peripherally addressed the problem of tracking objects over time. Additionally, these eigenspace reconstruction methods are not invariant to image transformations such as translation, scaling, and rotation. Previous approaches have typically assumed that the object of interest can be located in the scene, segmented, and transformed into a canonical form for matching with the eigenspace. In this paper we generalize and extend the previous work in the area to ameliorate some of these problems.

There are three primary observations underlying this work. First, standard eigenspace techniques rely on a least-squares fit between an image and the eigenspace (Murase and Nayar, 1995), and this can lead to poor results when there is structured noise in the input image. We reformulate the eigenspace matching problem as one of robust estimation and show how it overcomes some of the problems of the least-squares approach. This makes eigenspace methods more practical in that they can cope with problems in which the standard least-squares formulation gives erroneous results.

Second, we observe that rather than try to represent all possible views of an object from all possible viewing positions, it is more practical to represent a smaller set of canonical views and allow a parameterized transformation (eg. affine) between an input image and the eigenspace. What this implies is that matching using an eigenspace representation involves both estimating the view of object as well as the transformation that takes this view into the image. This allows a *multiple-views plus transformation* model of object recognition (Tarr and Pinker, 1989). We formulate this problem in a robust estimation framework and simultaneously solve for the view and the transformation. For a particular view of an object we define a *subspace constancy assumption* between the eigenspace and the image. This is analogous to the "brightness constancy assumption" used in optical flow estimation and it allows us to exploit parameterized optical flow techniques to recover the transformation between the eigenspace and the image. Recovering the view and transformation requires solving a non-linear optimization problem which we minimize using gradient descent with a continuation method. To account for large transformations between model and image we define a multi-scale eigenspace representation (the Eigen-Pyramid) and a coarse-to-fine matching scheme.

Third we note that the above techniques can be used to track previously viewed objects that are undergoing rigid motions with respect to the camera as well as significant changes in viewpoint. Unlike some object recognition tasks, in tracking applications one has a prediction of the object location in a given frame. Given a rough prediction, the optimization technique we define for parameterized eigenspace matching can refine the transformation between the eigenspace and the image, effectively tracking the object. This approach, which we call *EigenTracking*, can be applied to both rigid and simple articulated objects and can be used for object and gesture recognition in video sequences.

The following two sections review related work on object recognition, motion estimation, and tracking. Section 4 develops the robust subspace projection framework and Section 5 extends this framework to allow parameterized transformations between the image and the eigenspace. Section 6 shows how these techniques can be used to solve interesting tracking problems in long image sequences. Examples and results with natural images are provided to illustrate the ideas throughout the paper.

2. Related Work

Various representations and mechanisms have been proposed for object recognition ranging from approaches based on object-centered structural descriptions to those that emphasize visual appearance. Eigenspaces are one promising candidate for an appearance-based object representation. While eigenspace approaches are based on the well known "principle component analysis", there are still some technical problems that need to be solved before these techniques can be widely applied. We consider four of these problems in turn.

First the position, orientation, and scale of the object within the given image must be estimated. It is either assumed that the object can be detected by simple thresholding (Murase and Nayar, 1995), through some other feature detection process (Moghaddam and Pentland, 1995), or through global search (Moghaddam and Pentland, 1995),(Turk and Pentland, 1991). Turk and Pentland (1991) showed that eigenspace matching could be used to perform a global search under translation simply by comparing the eigenspace with the input image at every image location. This amounts to a correlation-style matching. Moghaddam and Pentland (1995) extended this global search idea to include scale in a straightforward way by matching the input at different scales using the standard eigenspace approach. These exhaustive search techniques may be used to provide a coarse initial guess about the transformation between the eigenspace and the image. This can then be refined using our continuous optimization technique.

Second the object must be segmented from the background so that the reconstruction and recognition is based on the object and not the appearance of the background. We present a robust formulation of the eigenspace matching problem that can tolerate structured noise (for example, from the background) and still reconstruct the object.

Third, in addition to locating and segmenting the object, it must also be transformed into some canonical form for matching. For example, face databases typically store representations of people's heads in the standard upright orientation at a particular scale. If a test image contains a head that is tilted and at a different scale, the image must first be transformed into the canonical position (Moghaddam and Pentland, 1995). In previous work this has been viewed as a preprocessing step. We will show how the problem can be formulated and solved using the eigenspace representation itself.

Murase and Nayar (1995) took a different approach. While they still preprocessed their images to segment them and normalize their size, they did not try to "rotate" their objects into some canonical orientation. Rather, they constructed their eigenspace from a training set that contained images of the object from a dense sampling of viewpoints. The linear combination of basis vectors is computed for each image in the training set by projecting the images onto the eigenspace. These coefficients define a manifold that is parameterized by the view of the object (i.e. its pose). The pose of an input image can then be determined by finding the point on the manifold nearest to where its projection lies. This information can be used by a robot to actively track a moving object by moving to maintain a particular view (Nayar *et al.*, 1994). In related work, Bobick and Wilson (1995) represent hand gestures using computed trajectories in the space of the eigenspace coefficients. Similarly, Bregler and Omohundro (1994) learn manifolds in the eigenspace projections of lip sequences.

Our approach is quite different. Rather than try to represent every possible view in the eigenspace, or learn surfaces in the eigenspace that interpolate between views, we represent views from only a few orientations. We can recognize objects in other orientations by recovering a parameterized transformation (or warp) between the image and the eigenspace. This is consistent with the model of human object recognition proposed by Tarr and Pinker (1989) which suggests that objects are represented by a set of views corresponding to familiar orientations and that new views are transformed to one of these stored views for recognition.

Finally, we are interested in tracking objects over time. Turk and Pentland (1991) proposed tracking faces using a simple motion-based tracking algorithm. When a face was localized then it could be checked against the eigenspace to make sure it was actually a face. Murase and Nayar (1995) also mention tracking but assume that tracking has already been achieved through a simple segmentation algorithm. What these previous approaches have failed to exploit is that the eigenspace itself provides a representation (i.e. an image) of the object that can be used for tracking. We exploit our robust parameterized matching scheme to perform tracking of objects undergoing affine image distortions and changes of view.

Traditional tracking approaches often use techniques such as normalized correlation or template matching. Such approaches are typically limited to situations in which the image motion of the object is simple (eg. translation) and the viewpoint of the object is either fixed or changing slowly. Darrell and Pentland (1993) extended these tracking approaches to allow a set of learned views for an object. Unlike eigenspace approaches, they represented these views individually and used correlation hardware to perform a brute-force match between all the stored views and the input images.

Parameterized optical flow techniques (Bergen *et al.*, 1992) represent image motion in terms of some low-order polynomial (eg. an affine transformation) and have proved to be useful for tracking objects undergoing a variety of rigid transformations (Black and Yacoob, 1995). Being purely image based, these techniques cannot cope with situations in which the viewpoint of the object changes over time. Changes in viewpoint will be represented as optical flow and if the initial view completely disappears, tracking will fail.

Recent work by Hager and Belhumeur (1996) extends an affine tracking scheme to deal with changing illumination using a method similar in spirit to the one described here. To track an object under changing illumination they first view the object (in a single pose) under various lighting conditions. They then construct a set of basis images from which they can approximate the object viewed under any illumination. They then simultaneously solve for the affine motion of the object and the illumination. They use a novel motion formulation and can track objects under varying illumination in real time. Unlike the EigenTracking approach described here, they do not track objects that are also changing in viewpoint and it may be more difficult to achieve real-time performance in this case. Their real-time performance is achieved by pre-computing "motion templates" which are the product of the spatial derivatives of the reference image to be tracked and a set of motion fields. If the eigenspace to be tracked contains multiple views of an object, or multiple objects, then it is not clear whether this pre-computation of motion templates is feasible.

A popular set of approaches for tracking objects under changing views employs 3D models of the objects being tracked. Such 3D schemes work well for tracking rigid objects such as cars that are relatively simple to model (for example (Koller *et al.*, 1993)). For articulated objects such as hands these approaches become much more complex (Rehg and Kanade, 1995). For many objects it may not be straightforward to construct 3D models and one would like to be able to build the object models automatically. Furthermore, such models encode the structure of the object but not necessarily its appearance and, for some objects, the markings or texture may be more salient than their 3D shape.

Image-based tracking schemes that emphasize learning of views or motion have focused on region contours (Baumberg and Hogg, 1994),(Blake *et al.*, 1994),(Cootes *et al.*, 1992),(Kervrann and Heitz, 1994). In particular, Baumberg and Hogg (1994) track articulated objects by first computing a silhouette of the object via image differencing. They fit a spline to the object's outline and the knot points of the spline form the representation of the current view. They learn a view-based representation of people walking by computing an eigenspace representation of the knot points over many training images. Tracking an object amounts to projecting the knot points of a particular view onto the eigenspace. As with the other eigenspace approaches the spline-based representation must be normalized for training and recognition. Our work differs from that of Baumberg and Hogg in that we use the brightness values corresponding to image region rather than its outline and we allow parameterized transformations of the input data in place of the standard preprocessing normalization.

Recently a number of authors have combined deformation information with eigenspace approaches (Beymer, 1996), (Hallinan, 1995), (Nastar *et al.*, 1996) for recognition but not tracking. These approaches focus on the problem of recognizing or reconstructing faces from a learned set of face images. The approaches model information about the allowed types of deformations between human faces. To recognize a new face image they simultaneously solve for the optimal combination of basis faces and deformation which reconstructs the input. In particular Hallinan's method, while used in a different context, has much in common with the formulation described here.

Also recently, Leonardis and Bischof (1996) have proposed a robust eigenspace matching method similar to ours. They use a hypothesize and test approach rather than our continuous formulation and do not address parameterized transformations or tracking.

3. Eigenspace Approaches

Given a set of images, eigenspace approaches construct a small set of basis images that characterize the majority of the variation in the training set and can be used to approximate any of the training images. For each $n \times m$ image in a training set of p images we construct a 1D column vector by scanning the image in the standard lexicographic order. Each of these 1D vectors becomes a column in a $nm \times p$ matrix A. We assume that the number of training images, p, is less than the number of pixels, nm and we use Singular Value Decomposition (SVD)[1] to decompose the matrix A as

$$A = U\Sigma V^T. \tag{1}$$

U is an orthogonal matrix of the same size as A representing the principle component directions in the training set. Σ is a diagonal matrix with singular values $\sigma_1, \sigma_2, \ldots, \sigma_p$ sorted in decreasing order along the diagonal. The $p \times p$ orthogonal matrix V^T encodes the coefficients to be used in expanding each column of A in terms of the principle component directions.

If the singular values σ_k, for $k \geq t$ for some t, are small then, since the columns of U are orthonormal, we can approximate some new $nm \times 1$ vector $\mathbf{e}$ as

$$\mathbf{e}^* = \sum_{i=1}^{t} c_i U_i, \tag{2}$$

where the c_i are scalar values that can be computed by taking the dot product of $\mathbf{e}$ and the column U_i. This amounts to a projection of the input image, $\mathbf{e}$, onto the subspace defined by the t basis vectors.

For illustration we constructed an eigenspace representation for soda cans. Figure 1 (top row) shows some example soda can images in the training set. The training set contained 200 images of Coke and 7UP cans viewed from the side. These training images are 121×227 pixels in size. The eigenspace was constructed as described above and the first few principle components are shown in the bottom row of Figure 1.

Due to the high frequency structure in the texture on the soda cans, the training images are not particularly well modeled using a linear subspace. To reconstruct the images accurately enough for tracking requires the use of 50 principle components (a relatively high number). Faces of different people, on the other hand, have much more in common than the different views than the soda cans used in our experiments. Eigenspace approaches work in either case, but the compactness of the encoding will depend on the structure of the training images.

4. Robust Matching

The eigenspace defined in the previous section can be thought of as a compact view-based object representation that is learned from a set of input images. Previously observed views of an object can be approximated by a linear combination of the basis vectors. This can be thought of as "matching" between the eigenspace and the image. This section describes how this matching process can be made robust.

Let $\mathbf{e}$ be an input image, written as a $nm \times 1$ vector, that we wish to match to the eigenspace. Recall that traditional eigenspace methods construct an approximation, $\mathbf{e}^*$, to the input image $\mathbf{e}$ as

$$\mathbf{e}^* = \sum_{i=1}^{t} c_i U_i,$$

where each c_i is computed by taking the dot product of $\mathbf{e}$ with U_i. This approximation corresponds

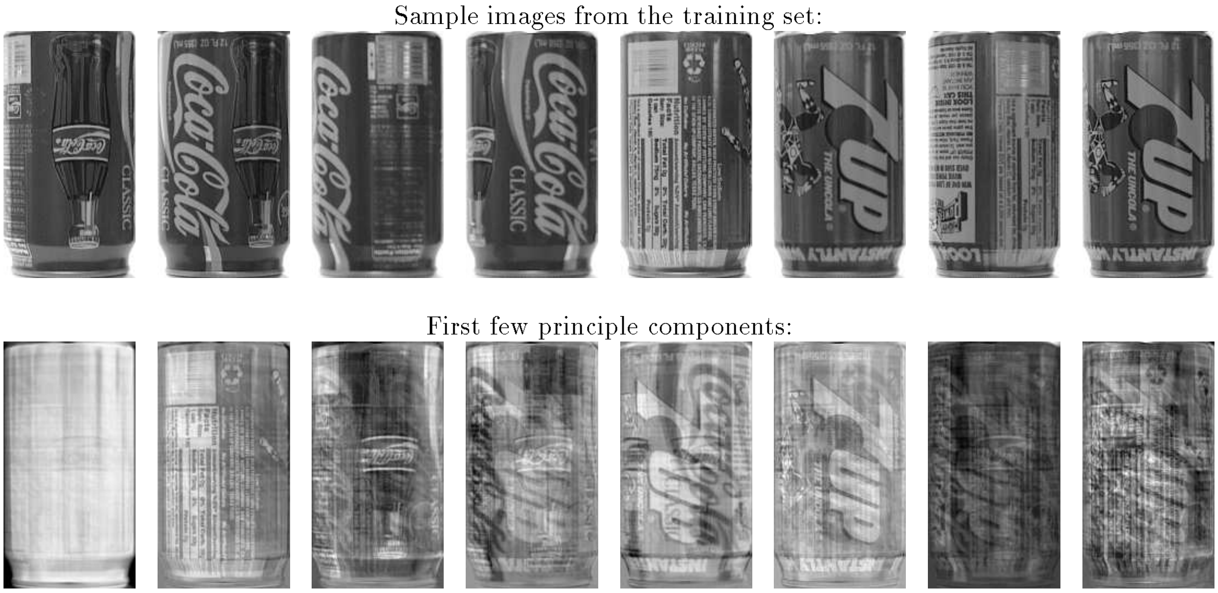


Fig. 1. Sample images from the soda-can training set are shown along with the first eight principle components.

to the least-squares estimate of the c_i (Murase and Nayar, 1995),(Strang, 1976). In other words, the c_i are those that give a reconstructed image that minimizes the squared error $E(\mathbf{c})$ between $\mathbf{e}$ and $\mathbf{e}^*$ summed over the entire image:

$$\begin{aligned} E(\mathbf{c}) &= \sum_{j=1}^{n\times m} (\mathbf{e}_j - \mathbf{e}^*_j)^2, \\ &= \sum_{j=1}^{n\times m} \left(\mathbf{e}_j - \left(\sum_{i=1}^{t} c_i U_{i,j} \right) \right)^2 . \quad (3) \end{aligned}$$

This least-squares approximation works well when the input images have clearly segmented objects that look roughly like those used to build the eigenspace. But it is commonly known that least-squares is sensitive to gross errors, or "outliers" (Hampel *et al.*, 1986), and it is easy to construct situations in which the standard eigenspace reconstruction is a poor approximation to the input data. In particular, if the input image contains structured noise (eg. from the background) that can be represented by the eigenspace then there may be multiple possible matches between the image and the eigenspace and the least-squares solution will return some combination of these views. Typically this will result in a blurry or noisy reconstruction.

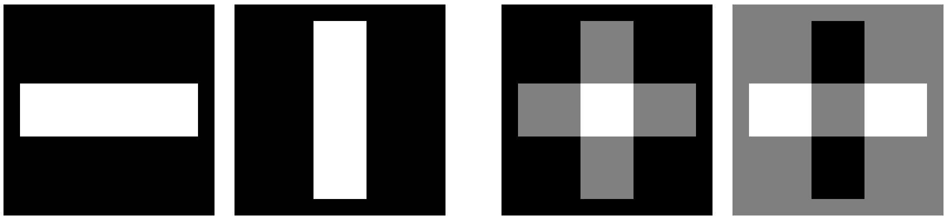
For example consider the very simple training set in Figure 2 (*a* and *b*). The basis vectors in the eigenspace are shown in Figure 2 (*c, d*).[2] Now, consider the test image in Figure 3*a* which does not look the same as either of the training images. The least-squares reconstruction shown in Figure 3*b* attempts to account for all the data and hence partially recovers the horizontal bar to account for the data on the right of the vertical bar. In doing so, there is no way to fully account for the vertical bar using a linear combination of the basis images. The robust formulation described below recovers the dominant feature which is the vertical bar (as shown in Figure 3*c*) and to do so, treats the data to the right as outliers (shown in black in Figure 3*d*).

To robustly estimate the coefficients $\mathbf{c}$ we replace the quadratic error norm in Equation (3) with a robust error norm, ρ, and minimize

$$E(\mathbf{c}) = \sum_{j=1}^{n\times m} \rho \left(\left(\mathbf{e}_j - \left(\sum_{i=1}^{t} c_i U_{i,j} \right) \right), \sigma \right) . \quad (4)$$

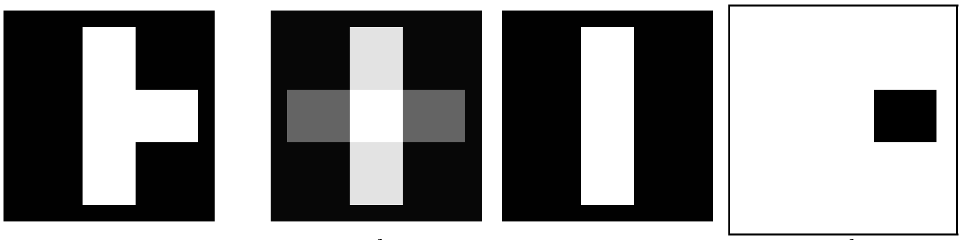
where σ is a scale parameter. For the experiments in this paper we take ρ to be

$$\rho(x, \sigma) = \frac{x^2}{\sigma^2 + x^2},$$



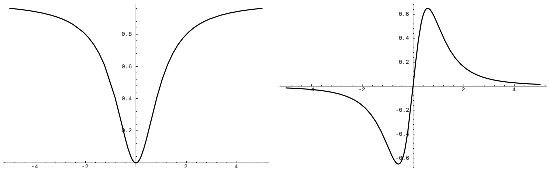
a *b* *c* *d*

Fig. 2. A simple example. *(a, b):* Training images. *(c, d):* Eigenspace basis images.



a *b* *c* *d*

Fig. 3. Reconstruction. *(a):* New test image. *(b):* Least-squares reconstruction. *(c):* Robust reconstruction. *(d):* Outliers (shown as black pixels).



$\rho(x,\sigma)$ $\psi(x,\sigma)$

Fig. 4. Robust error norm (ρ) and its derivative (ψ).

$$\frac{\partial}{\partial x}\rho(x,\sigma) \;=\; \psi(x,\sigma) = \frac{2x\sigma^2}{(\sigma^2+x^2)^2},$$

which is a robust error norm that has been used extensively for optical flow estimation (Black and Anandan, 1993a), (Black and Anandan, 1993b). The shape of the function, as shown in Figure 4, is such that it "rejects", or downweights, large residual errors. The function $\psi(x,\sigma)$, also shown in Figure 4, is the derivative of ρ and characterizes the influence of the residuals (Hampel *et al.*, 1986). As the magnitudes of residuals $(\mathbf{e}_j - \mathbf{e}_j^*)$ grow beyond a point their influence on the solution begins to decrease and the value of $\rho(\cdot)$ approaches a constant.

The value σ is a scale parameter that affects the point at which the influence of outliers begins to decrease. By examining the ψ-function we see that this "outlier rejection" begins where the second derivative of ρ is zero. For the error norm used here, this means that those residuals where

$$|(\mathbf{e}_j - \mathbf{e}_j^*)| > \sigma/\sqrt{3}$$

have reduced influence on the solution and can be viewed as outliers. The value of σ can also be estimated from the data (Rousseeuw and Leroy, 1987).

The computation of the coefficients $\mathbf{c}$ now involves the minimization of the non-linear function in Equation (4). We perform this minimization using a simple gradient descent scheme with a continuation method that begins with a high value for σ and lowers it during the minimization (see (Black and Anandan, 1993a), (Black and Anandan, 1993b) and the Appendix for details). The effect of this procedure is that initially no data are rejected as outliers then gradually the influence of outliers is reduced.

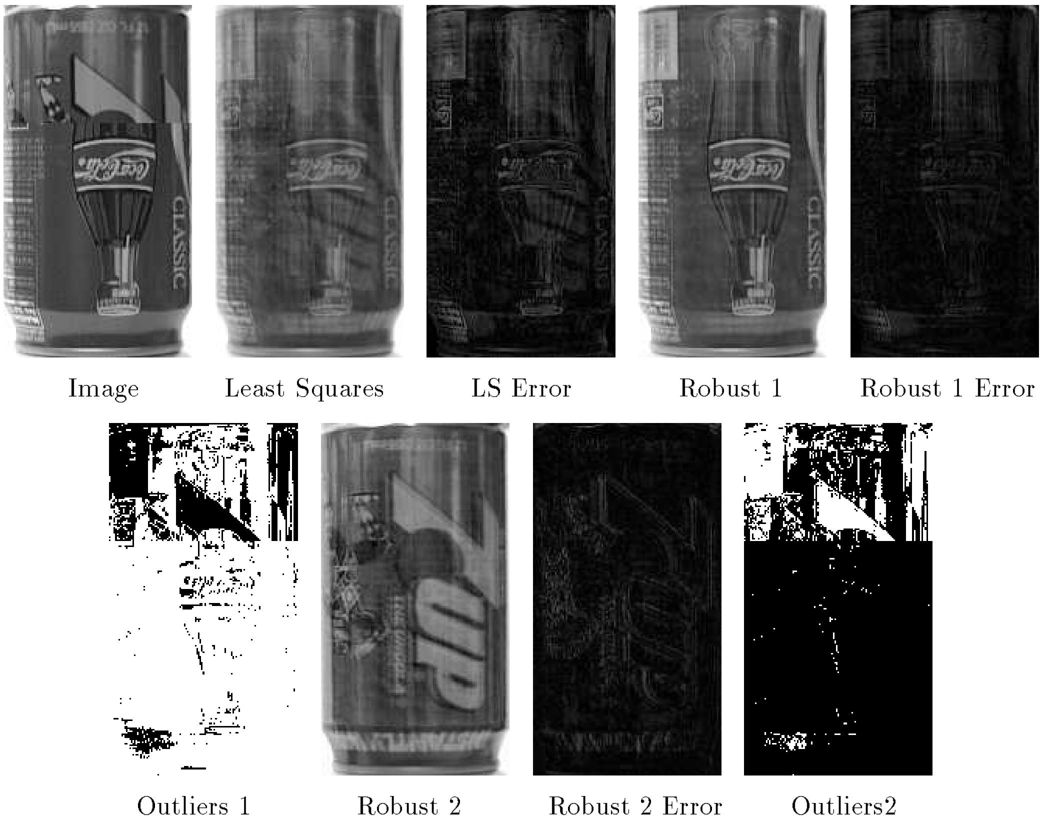


Fig. 5. Robust matching with structured noise (see text).

Robust estimation approaches such as this can be characterized by their "breakdown point" which is the percentage of outliers that they can tolerate before the solution can be made arbitrarily bad. One fact about M-estimation, such as the one here, is that the breakdown point is less than $1/(t+1)$ where t is the number of parameters to be estimated (Li, 1985). So as the number of parameters increases, the robustness of the method decreases. In our experiments we used up to 50 basis vectors and six affine parameters (as described in the next section) and therefore one might expect that the approach will not be very robust. However in our experiments we have observed breakdown points of roughly $35-45\%$ which is in line with the robust estimation of a single parameter rather than 56 parameters. This discrepancy deserves further study but we suspect that it is due to the large amount of data available for fitting within a single subimage.

4.1. Outliers and Multiple Matches

As we saw in Figure 3 it is possible for an input image to contain a brightness pattern that is not well represented by any single "view". Given a robust match that recovers the "dominant" structure in the input image, we can detect those points that were treated as outliers. We define an outlier vector, or "mask", $\mathbf{m}$ as

$$m_j = \begin{cases} 0 & |(\mathbf{e}_j - \mathbf{e}_j^*)| \leq \sigma/\sqrt{3} \\ 1 & \text{otherwise.} \end{cases}$$

If a robust match results in a significant number of outliers, then additional matches can be found by minimizing

$$E(\mathbf{c}) = \sum_{j=1}^{n\times m} m_j\ \rho\left(\left(\mathbf{e}_j - \left(\sum_{i=1}^{t} c_i U_{i,j}\right)\right), \sigma\right). \quad (5)$$

Alternatively one could adopt a mixture-model formulation (Jepson and Black, 1993), (McLach-

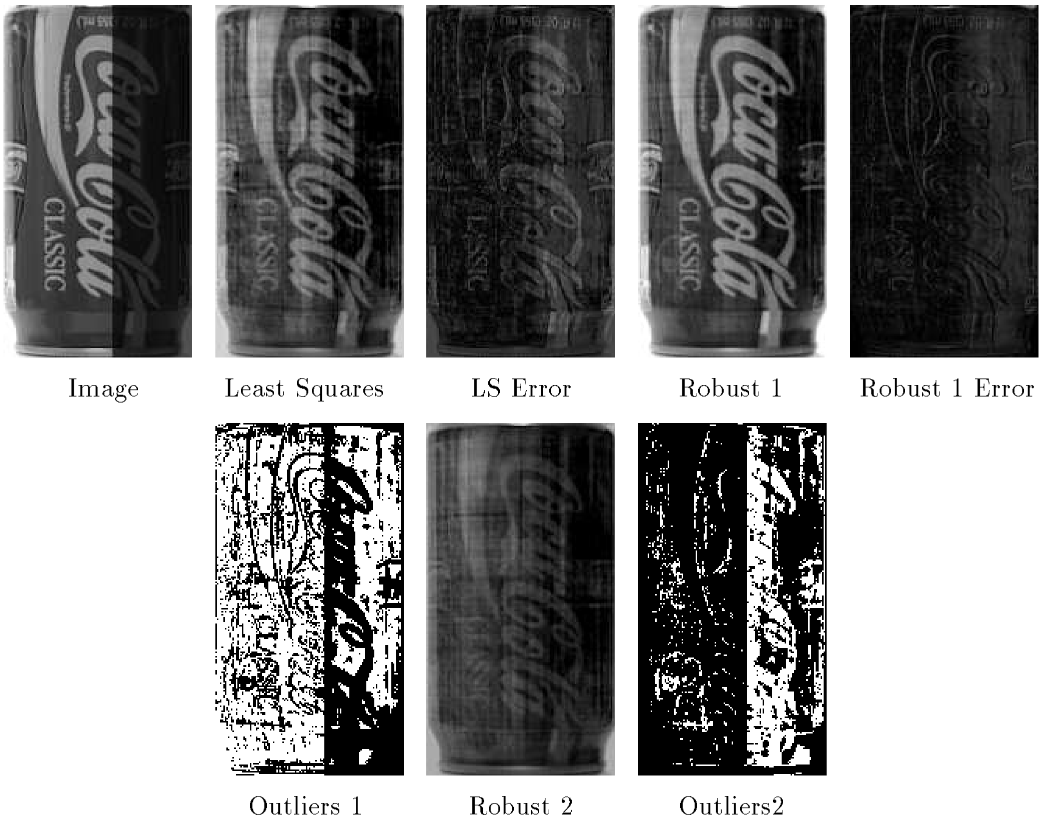


Fig. 6. Robust matching with shadow (see text).

lan and Basford, 1988), (Saund, 1995) and recover multiple sets of coefficients simultaneously. Note that this use of mixture models for matching differs from that of Moghaddam and Pentland (1995) who use mixture models to group, or classify, training data in the space of the coefficients.

4.2. Robust Matching Examples

Two examples will help illustrate the problems with the least-squares solution and the effect of robust estimation. Figure 5 shows an artificial image constructed from two images that were present in the training data; the bottom two thirds of the image is that of a Coke can while the top third is from an image of a 7UP can. It is impossible to reconstruct the entire input image accurately with the eigenspace despite the fact that both parts of the image can be represented independently. The least-squares solution shown in Figure 5 recovers a single view that contains elements of both possible views. The Coke can, which occupies the majority of the input image, is faint and blurred in the reconstruction and a "ghost" of the characters "7UP" can be seen combined with it. The absolute pixel error between the reconstruction and the actual view of the Coke can be seen in Figure 5 (LS Error). The mean of the squared residual errors (χ^2) for this reconstruction was $\chi^2 = 632$.

The simple robust estimation of the linear coefficients results in a much more accurate reconstruction of the dominant view (Figure 5, Robust 1). Compare the least-squares error image with the absolute pixel error in (Robust 1 Error) for which $\chi^2 = 247$ (roughly 61% less than the error for the least-squares fit).

Moreover, with the robust fit, we can detect those points in the image that did not match the reconstruction very well and were treated as outliers; these are shown in black in Figure 5 (Outliers 1). We can now take just those points that were treated as outliers and recover the view that best fits them using Equation (5). This robust reconstruction using the outliers from the first reconstruction is shown in Figure 5 (Robust 2); even with very little data, the reconstructed image reasonably approximates the correct view of the 7UP can (see the absolute pixel error (Robust 2 Error), $\chi^2 = 134$). The points that were treated as "inliers" for the second robust reconstruction are shown as white in Figure 5 (Outliers 2).

Another example is shown in Figure 6 (Image) in which a Coke can has had a strong artificial shadow placed over the right 43% of the image. The least-squares estimate is affected by this non-uniform illumination change and the resulting reconstruction is both noisy and has an overall brightness that falls between that of the two regions (Figure 6, Least Squares). The absolute pixel error (LS Error) clearly shows the error in the fit ($\chi^2 = 1021$). The first robust fit (Figure 6, Robust 1) does a much better job of fitting the majority of the can; that is the robust reconstruction is of the left side of the can which is under the bright illumination. The error in the fit (Robust 1 Error) is significantly reduced ($\chi^2 = 336$, or roughly 67% lower than the least squares fit). The outliers (Figure 6, Outliers 1) of the first robust fit are concentrated in the darkened region of the can. A second fit to these outliers results in a second reconstruction of the darkened portion of the can (Figure 6, Robust 2).

5. EigenSpaces and Parametric Transformations

The previous section showed how robust estimation can improve the reconstruction of an image that is already aligned with the eigenspace. In this section we consider how to achieve this alignment in the first place. It is impractical to represent all possible views of an object at all possible scales and all orientations. One must be able to recognize a familiar object in a previously unseen pose and hence we would like to represent a small set of views and recover a transformation that maps an image into the eigenspace. In the previous section we formulated the matching problem as an explicit non-linear parameter estimation problem. In this section we will simply extend this problem formulation with the addition of a few more parameters representing the transformation between the image and the eigenspace.

To extend eigenspace methods to allow matching under some parametric transformation we first need to generalize the notion of brightness constancy typically used to define optical flow. Recall that brightness constancy states that the brightness of a pixel remains constant between frames, but that its location may change. For eigenspaces we wish to say that there is a view of the object, as represented by some linear combination of the basis vectors and some parametric spatial distortion, such that pixels in the reconstruction have the same brightness as the corresponding pixels in the image. We call this the *subspace constancy assumption*, and we formulate it precisely below.

Let I be an $n \times m$ sub-image of some larger input image and let

$$U = [U_1, U_2, \ldots U_t], \tag{6}$$

$$\mathbf{c} = [c_1, c_2, \ldots c_t]^T, \tag{7}$$

$$U\mathbf{c} = \sum_{i=1}^{t} c_i U_i, \tag{8}$$

where $U\mathbf{c}$ is the approximated image for a particular set of coefficients, $\mathbf{c}$. While $U\mathbf{c}$ is a $nm \times 1$ vector we can index into it as though it were an $n \times m$ image. We define $[U\mathbf{c}](\mathbf{x})$ to be the value of $U\mathbf{c}$ at the position associated with pixel location $\mathbf{x} = (x, y)$.

Then the robust matching objective function from the previous section can be written as

$$E(\mathbf{c}) = \sum_{\mathbf{x}} \rho(I(\mathbf{x}) - [U\mathbf{c}](\mathbf{x}), \sigma). \tag{9}$$

Pentland *et al.* (1994) define the distance-from-feature-space (DFFS) to be the root mean square of the residual image $I - U\mathbf{c}$, and note that this error measure could be used for localization and detection. The goal is to find the region I in some larger image that best matches the stored eigenspace. They compute a "saliency" map by computing the DFFS for each possible image position in the large image and then choose the minimum value as the location of the best match. Moghaddam and Pentland (1995) extend this to search over scale by constructing multiple input images at various scales and searching over all of them simultaneously.

We take a different approach in the spirit of parameterized optical flow estimation. First we define the subspace constancy assumption by parameterizing the input image as follows

$$I(\mathbf{x}+\mathbf{u}(\mathbf{x},\mathbf{a})) = [U\mathbf{c}](\mathbf{x}), \ \forall \mathbf{x}, \qquad (10)$$

where $\mathbf{u}(\mathbf{x},\mathbf{a}) = (u(\mathbf{x},\mathbf{a}), v(\mathbf{x},\mathbf{a}))$ represents an image transformation (or motion) where u and v represent the horizontal and vertical displacements at a pixel and the parameters $\mathbf{a}$ are to be estimated. For example we may take $\mathbf{u}$ to be the affine transformation

$$u(\mathbf{x},\mathbf{a}) = a_0 + a_1 x + a_2 y \qquad (11)$$

$$v(\mathbf{x},\mathbf{a}) = a_3 + a_4 x + a_5 y \qquad (12)$$

where x and y are defined with respect to the image center. Equation (10) states that there should be some transformation, $\mathbf{u}(\mathbf{x},\mathbf{a})$, that, when applied to image region I, makes I look like some image reconstructed using the eigenspace. This transformation can be thought of as *warping* the input image into the coordinate frame of the training data.

Our goal is then to simultaneously find the $\mathbf{c}$ and $\mathbf{a}$ that minimize the *robust subspace constancy* objective function,

$$E(\mathbf{c},\mathbf{a}) = \sum_{\mathbf{x}} \rho(I(\mathbf{x}+\mathbf{u}(\mathbf{x},\mathbf{a})) - [U\mathbf{c}](\mathbf{x}), \sigma). \qquad (13)$$

As opposed to the exhaustive search techniques used by previous approaches (Moghaddam and Pentland, 1995), (Turk and Pentland, 1991), we derive and solve a continuous optimization problem.

We find it convenient to use an optimization algorithm which interleaves two previously developed algorithms, each of which applies to simpler sub-problems. In particular, the first sub-problem is to minimize $E(\mathbf{c},\mathbf{a})$ with respect to $\mathbf{c}$ while the warp parameters, $\mathbf{a}$, are held fixed. To do this we apply the robust eigenspace matching algorithm discussed in Section 4, with the only change being that we use the warped image $I(\mathbf{x}+\mathbf{u}(\mathbf{x},\mathbf{a}))$ instead of $I(\mathbf{x})$.

The second sub-problem is to minimize $E(\mathbf{c},\mathbf{a})$ with respect to the warp parameters, $\mathbf{a}$, but now with the expansion coefficients $\mathbf{c}$ held fixed. To do this we use a simple modification of a robust regression approach for optical flow (Black and Anandan, 1993b), (Black and Anandan, 1993a). This optical flow approach is based on the *robust brightness constancy* objective function

$$E(\mathbf{a}) = \sum_{\mathbf{x}} \rho(I(\mathbf{x}+\mathbf{u}(\mathbf{x},\mathbf{a}),t) - I(\mathbf{x},t+1), \sigma).$$

(Here $I(\mathbf{x},t)$ and $I(\mathbf{x},t+1)$ are the images at times t and $t+1$, while $\mathbf{u}(\mathbf{x},\mathbf{a})$ is the displacement map between these frames.) Notice that (13) has the same form as this expression, except the frame at time $t+1$ has been replaced by $[U\mathbf{c}](x)$, the image from the eigenspace. As a result, the algorithms developed for the robust regression of optical flow can be easily adapted to minimize (13) with respect to the warp parameters $\mathbf{a}$.

The overall approach repeatedly applies these two simpler minimization algorithms as σ is gradually reduced. The details are presented in the Appendix.

Note that this optimization scheme will not perform a global search to "find" the image region that matches the stored representation. Rather, given an initial guess, it will refine the pose and reconstruction. While the initial guess can be fairly coarse, the approach described here does not obviate the need for global search techniques but rather compliments them. In particular, as discussed in Section 6 the method will be useful for tracking an object where a reasonable initial guess is typically available.

5.1. *Multi-Scale Eigenspace*

As in the case of optical flow, the recovery of transformations that result in large pixel differences necessitates a coarse-to-fine strategy. To do this here we first construct a multi-scale representation as illustrated in Figure 7. For every image in the training set we construct a pyramid of images by spatial filtering and sub-sampling. The images at each level in the pyramid form distinct training sets and for each level a singular value decomposition is used to construct an eigenspace description of that level.

The input image is similarly smoothed and subsampled. The coarse-level input image is then matched against the coarse-level eigenspace and the values of $\mathbf{c}$ and $\mathbf{a}$ are estimated at this level.

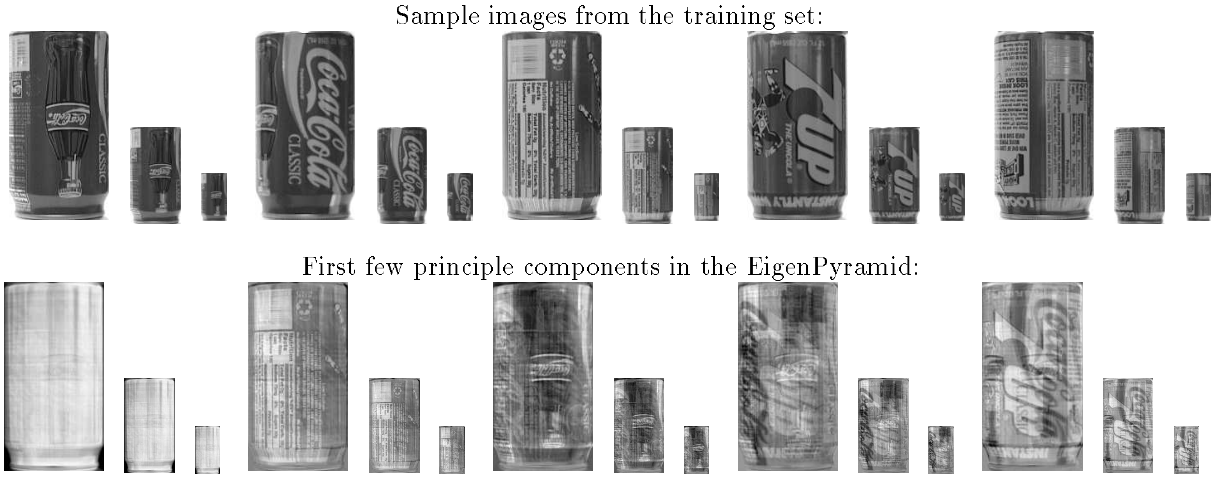


Fig. 7. Example of a multi-scale eigenspace representation.

The idea behind this scheme is that at the coarse levels in the pyramid, the low-frequency information will dominate and the physical distance between the model and the image will be small. The new values of **a** are then projected to the next level (in the case of the affine transformation the values of a_0 and a_3 are multiplied by 2). This **a** is then used to warp the input image towards the eigenspace and the value of **c** is estimated and the a_i are refined. The process continues to the finest level.

In the experiments in Section 6 the motions of the object between frames can be quite large (up to about 15 pixels). While significant displacements can be solved for using the multi-scale approach, the maximum displacement for a given object will be dependent on the size of the object, its spatial frequency structure, and the number of levels used in the pyramid.

5.2. *Experiment*

To test the parameterized matching technique we constructed a simple experiment in which we chose 200 images at random from the training set and, for each image, generated a random affine transformation where the affine parameters a_0 and a_3 were selected uniformly from the interval $[-2.0, 2.0]$ and the remaining affine parameters were selected from the interval $[-0.04, 0.04]$. The images were warped by the inverse affine transformation and then the method described above was used to recover the original affine transformation. For reference, the affine parameters in the experiment distorted the input images by at most 10 pixels. To cope with the large deformations we used a three-level EigenPyramid. Since we have both the original transformation and the recovered transformation we can compare the true and measured displacements of every pixel. The maximum disparity between true and measured displacements gives a useful measure of error for the recovered transformation. This maximum disparity, averaged over the 200 trials, was less than a pixel.

An example is shown in Figure 8. The images labeled "0" show the warped input image and the initial reconstruction. The maximum displacement between the original and the warped images is 9.87 pixels. This amount of distortion means that the initial approximation is not very good. The top row shows the input image after it has been warped by the estimated affine transformation at each iteration of the algorithm. An "iteration" here means one complete coarse-to-fine pass of the algorithm which provides an update of the coefficients **c** and **a** (see the appendix for more details). The bottom row shows the current reconstruction of the transformed image in the eigenspace. Over four iterations of the algorithm (numbered 1 – 4) a marked improvement can be seen in the accuracy of the approximation and in the error in the transformation, which has decreased to 0.23 pixels.

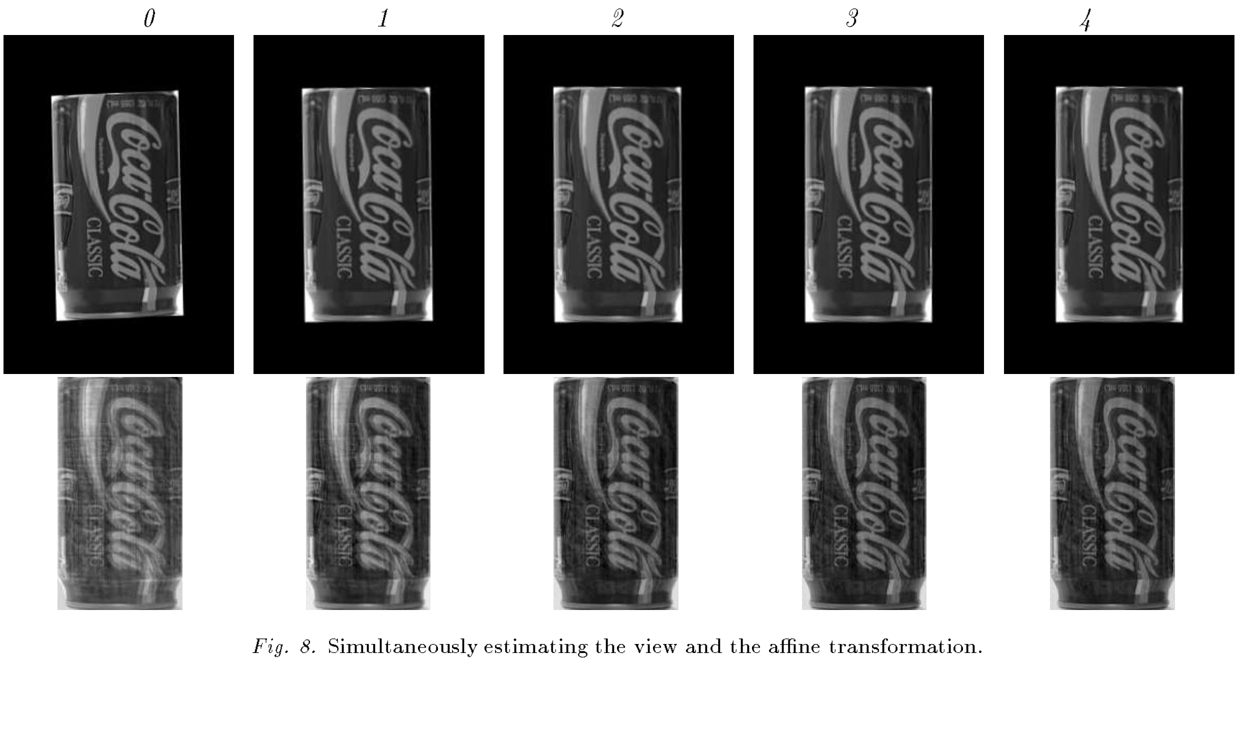


Fig. 8. Simultaneously estimating the view and the affine transformation.

6. EigenTracking

The robust parameterized matching scheme described in the previous section can be used to track objects undergoing changes in viewpoint or changes in structure. As an object moves and the view of the object changes, we would like to separately recover the current view of the object and the parameterized transformation between the current view and the eigenspace. For example, we will consider a number of sequences in which soda cans rotate about their major axis while moving. Template-based trackers would have trouble with the constantly changing view of the object while optical-flow based methods would not be able to separate the motions caused by the changing view and the changing position of the object. We will also consider a simple example of articulated motion and the recogintion of hand gestures.

It is important to note that no "image motion" is being used to "track" the objects in these experiments. The tracking is achieved entirely by the parameterized matching between the eigenspace and the image. We call this *EigenTracking* to emphasize that a view-based representation is being used to track an object over time.

All the experiments in this section used the same parameters unless otherwise noted. The value of σ started at $65\sqrt{3}$ and was lowered to a minimum of $15\sqrt{3}$ by a factor of 0.85 at each of a maximum of 15 stages. Within each stage of the continuation method a three level pyramid was used and a maximum of 15 iterations of the descent scheme were used to update $\mathbf{c}$ and $\mathbf{a}$ at each level. The minimization was terminated if a convergence criterion was met. The input images were 320×240 pixels and the algorithm was given a rough initial guess (to within a few pixels) of the transformation between the first image and the eigenspace. From then on the algorithm automatically tracked the object by estimating $\mathbf{c}$ and $\mathbf{a}$ for each frame. No prediction scheme was used so the initial guess for the transformation of the object in frame k is just the value of $\mathbf{a}$ from the previous frame. The motion in the experiments ranges from 0 to about 4 pixels per frame. For the soda-can sequences, fifty basis vectors were used in the reconstruction. This is more than would be necessary for recognition, but an accurate reconstruction of the image is needed for tracking.[3] Finally, for these experiments the affine transformation model was unnecessary and we restricted the transformation to recover translation, rotation, and scale.

6.1. *A simple example*

First we consider a simple example in which a hand picks up a soda can. The can undergoes translation and rotation in the image plane. Every tenth frame in the sequence is shown in Figure 9. The region corresponding to the eigenspace is displayed as a white box in the image. This box is generated by computing the inverse transformation between the eigenspace and the image and serves to illustrate the accuracy of the recovered transformation. Beside each image is shown the robust reconstruction of the image region within the box. Note that the motion is not purely translation, rotation, and scale. The can tilts in depth resulting is noticeable perspective effects. A quadratic optical flow model (Black and Yacoob, 1995) should improve the fit for this type of motion.

6.2. *Tracking a Rotating Object*

The example in Figure 10 is more challenging. In this example, a soda can translates left and right while moving in depth over 200 frames. This change in depth can be seen by comparing images 000 and 196. What makes this an interesting sequence is that while the can is changing position relative to the camera it is also undergoing rotations about its major axis (compare the views in frames 000 and 028 for example). Note that the traditional brightness constancy assumption of optical flow will not track the "can" but rather the "texture" on the can. This can be seen in the dense flow field shown in Figure 11a which was computed using the method in (Black and Anandan, 1993a). The result shows that the image motion in the scene corresponds to the rotation of the can, resulting in a roughly horizontal flow field.

Using the subspace constancy assumption, on the other hand, means that we will recover the transformation between the eigenspace representation of the can and the image. Hence, it is the

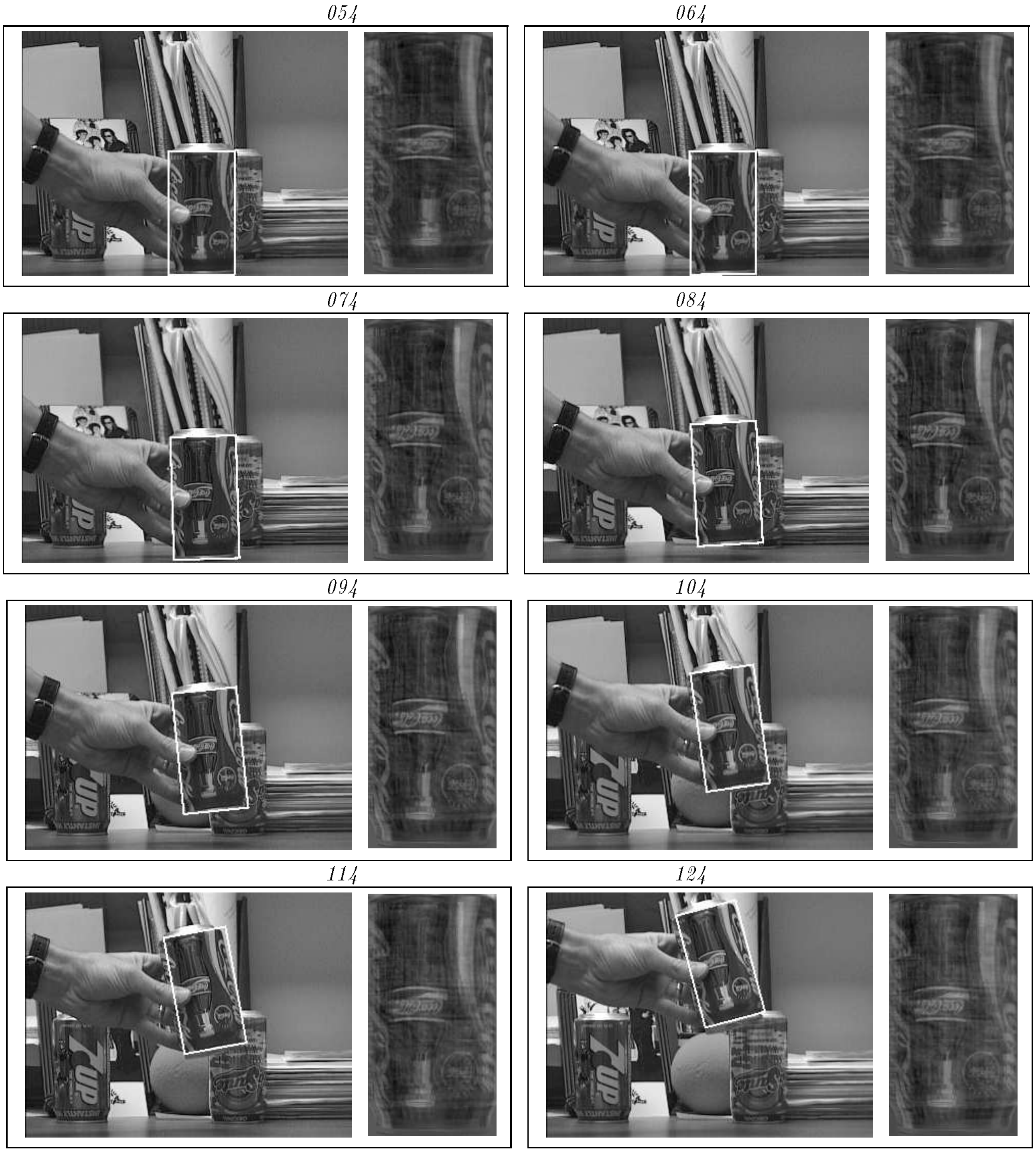


Fig. 9. Pickup Sequence. EigenTracking with translation and rotation in the image plane. Every 20 frames in the 75 frame sequence are shown (frame numbers are above the images). The left image in each box shows the bounding box of the eigenspace model projected into the image – this illustrates the recovery of the transformation. On the right in each box is the image reconstructed using the eigenspace.

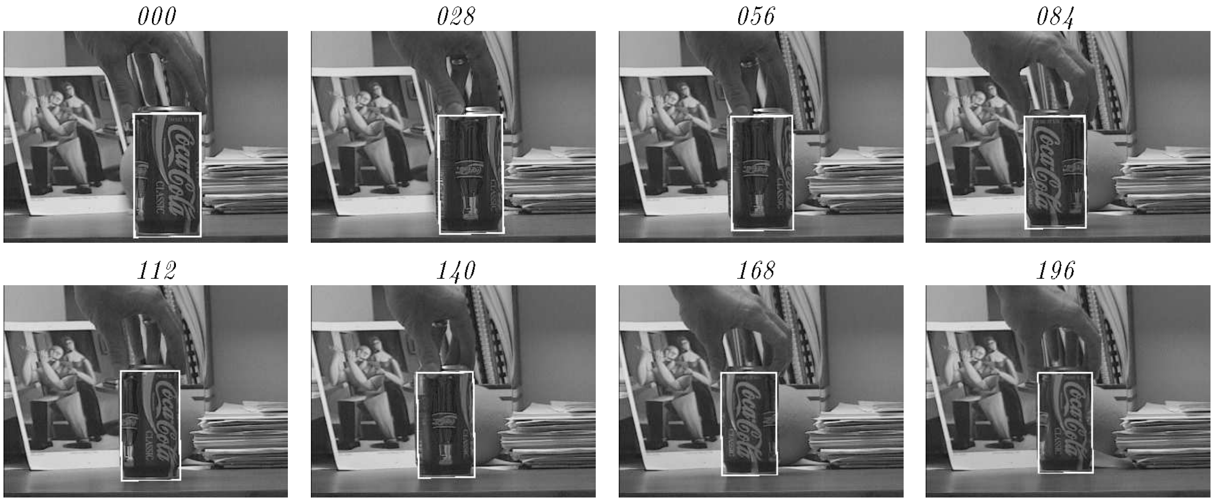


Fig. 10. EigenTracking with translation and divergence over 200 frames. The Coke can rotates about its major axis while moving relative to the camera. The white box is the bounding box of the model backprojected onto the image.

016:

018:

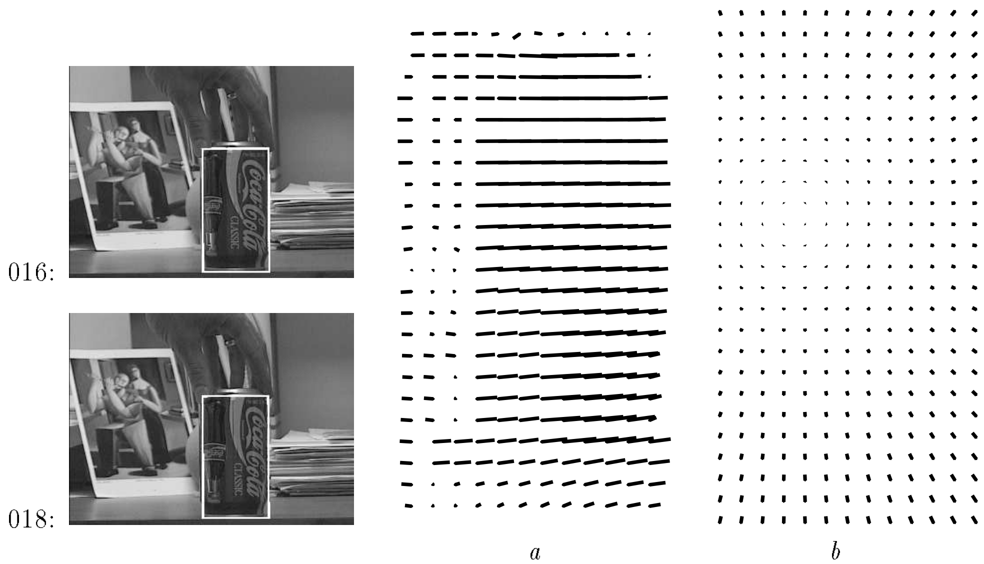


Fig. 11. Brightness versus Subspace constancy. "Motion" between frames 016 and 018 (computed within the white boxed region). *(a)* Dense optical flow for the soda can computed using the brightness constancy assumption. *(b)* "Flow" computed using the subspace constancy assumption for the same frames.

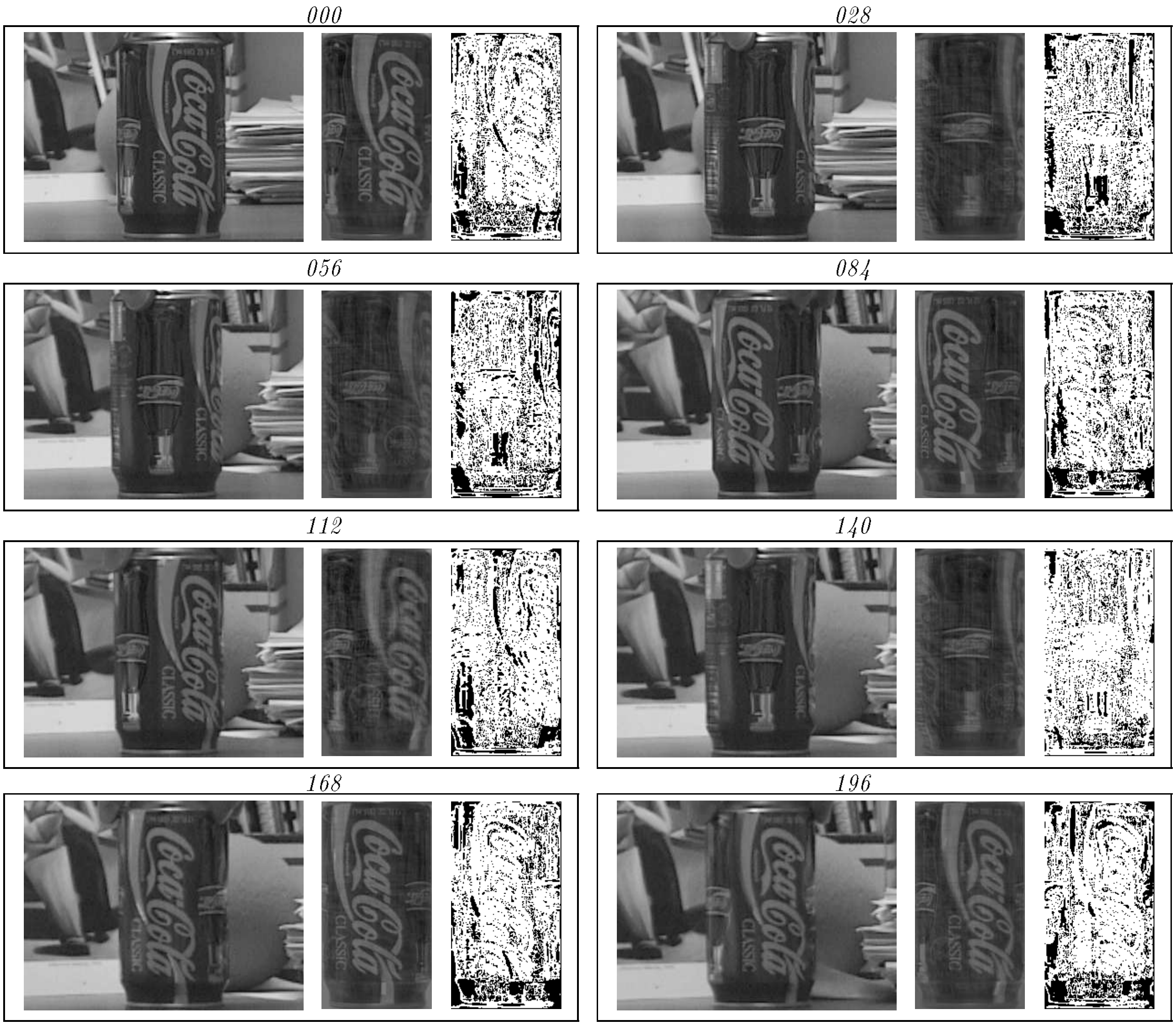


Fig. 12. Within each box, the left image is the original image "stabilized" with respect to the eigenspace; that is, the original image is warped towards the eigenspace using the recovered transformation. The middle image is the reconstructed eigenspace approximation to the image region. The right image shows outliers in black corresponding to points where the image and reconstruction differ.

"can" that is tracked rather than its "texture". The resulting motion of the soda can is shown in Figure 11*b* for the same frames as the dense optical flow. The flow field is primarily converging indicating the the can is moving away from the viewer. The rotation of the can is not present in this motion field but rather is captured as a change of view by the eigenspace.

Figure 12 provides more details. On the left of each box is the "stabilized" image which shows how the original image is "warped" into the coordinate frame of the eigenspace. Notice that the background differs over time as does the view of the can, but that the can itself is in the same position and at the same scale. This stabilized sequence can be played as a movie and one sees the can as remaining stationary while rotating about its axis.

The middle image in each box is the robust reconstruction of the image region being tracked. On the right of each box are the "outliers." The black points in these images are the places where the observed image and the reconstruction differed by more than $\sigma/\sqrt{3}$. Many of these outliers are due to specular reflections.

6.3. Motion without Constancy

The example in Figure 13 is an extreme and unnatural situation that serves to illustrate the difference between EigenTracking and traditional motion-based tracking methods. The *stop-action* image sequence was constructed such that a soda can moves a small amount in each frame, but that the *identity* of the can is different in each frame (it alternates between Coke and 7UP cans). Between pairs of adjacent frames, there is no coherence of the brightness pattern of the can, but there is *subspace coherence*. Since both objects are representable using the subspace, the presence of a known object *is* coherent between frames and, hence, is easily tracked.

6.4. Partially Robust EigenTracking

The tracking experiments presented so far all use the full robust minimization scheme described in the paper. This is a computationally intensive process as the 50 coefficients of the reconstruction must be updated repeatedly for each frame as the input image is brought into registration with the eigenspace. For the images in our experiments, the robust reconstruction of the view was unnecessary since there was no significant structured noise. When this is the case, the reconstruction portion of the algorithm can simply be replaced by least-squares estimation resulting in a speedup of approximately eight times for the complete algorithm. As the input image is incrementally warped towards the eigenspace, this least-squares approximation provides a reasonable reconstruction. Tracking with the least-squares reconstruction and robust parameter estimation took slightly less than 30 seconds per frame on a 75MHz SGI Indy workstation; still well below frame-rate, though no attempts have been made to optimize the algorithm. This hybrid tracking scheme was used for the 400 image sequence in Figure 14 which contains a 7UP can undergoing translation and scale changes while rotating. The recent work of Hager and Belhumeur (1996) may provide a framework for a real-time version of this view-based tracking scheme.

6.5. Articulated Motion and Gesture Recognition

In the final example we consider the problem of recognizing hand gestures in video sequences in which the hand is moving. We defined a simple set of four hand gestures illustrated in Figure 15. A 100 image training sequence was collected by fixing the wrist position and recording a hand as it smoothly moved between these four gestures. The eigenspace was constructed and 25 basis vectors were used for reconstruction. In our preliminary experiments we have found brightness images to provide sufficient information for both recognition and tracking of hand gestures (cf. (Moghaddam and Pentland, 1995)).

Figure 16 shows the tracking algorithm applied to a 100 image test sequence in which a moving hand executed the four gestures. The motion in this sequence was large (as much as 15 pixels per frame) and the hand moved while changing gestures. The figure shows the backprojected box corresponding to the eigenspace model and, on

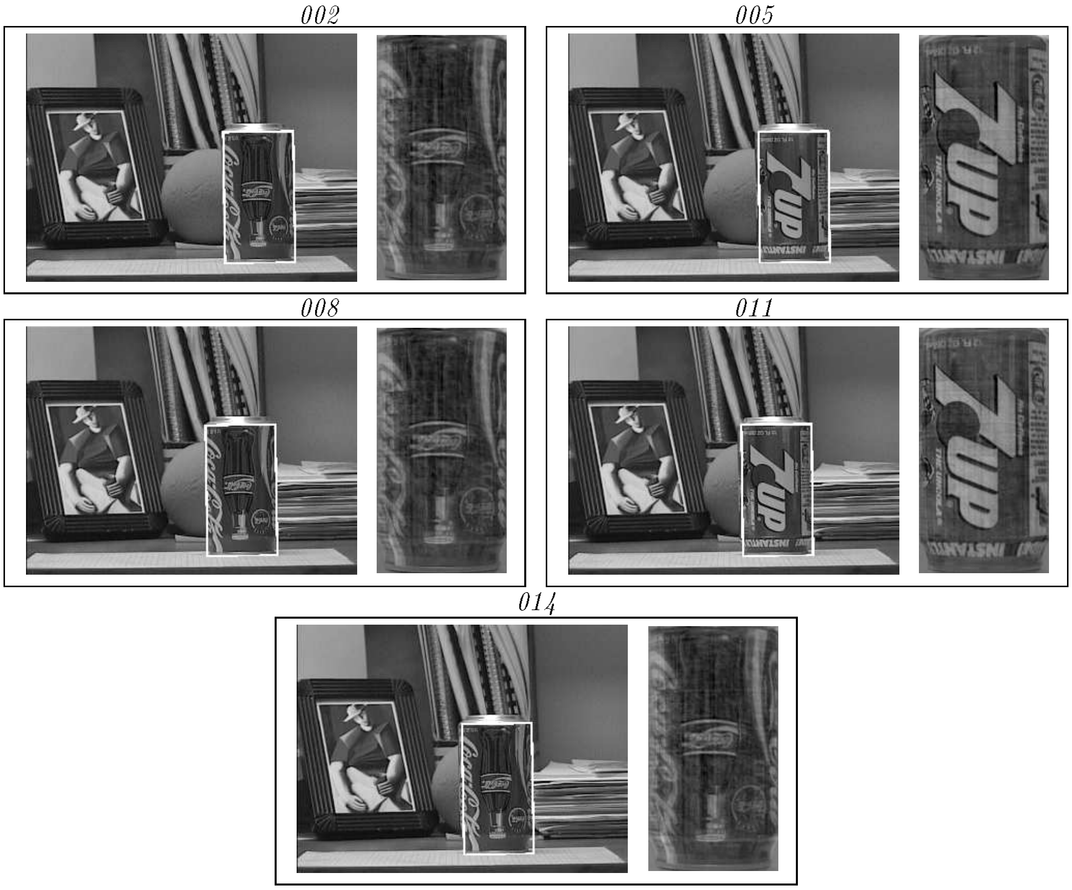


Fig. 13. Motion without brightness constancy.

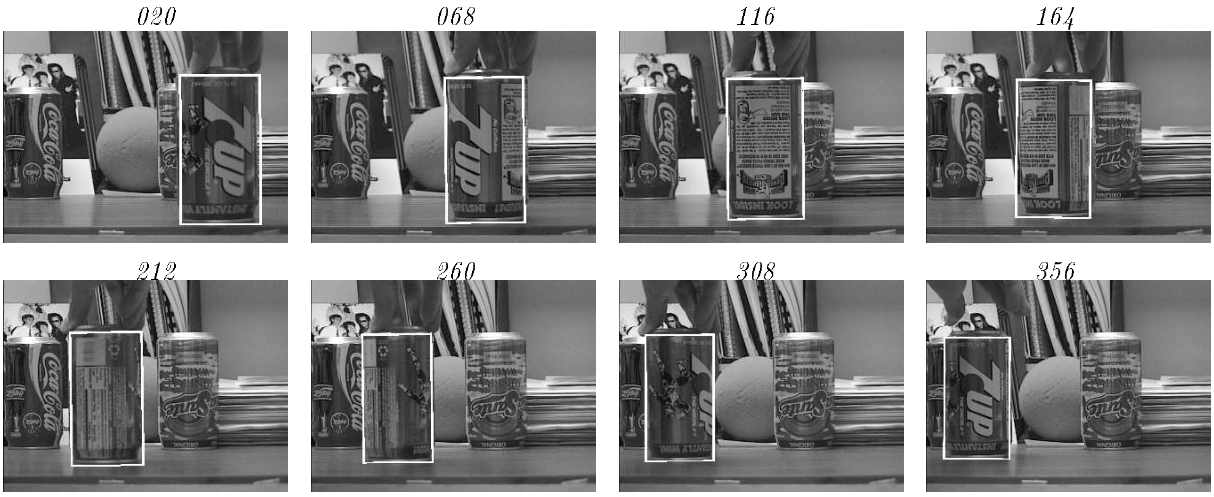


Fig. 14. EigenTracking with translation and divergence over 400 frames (see text).

Fig. 15. Examples of the four hand gestures used to construct the eigenspace.

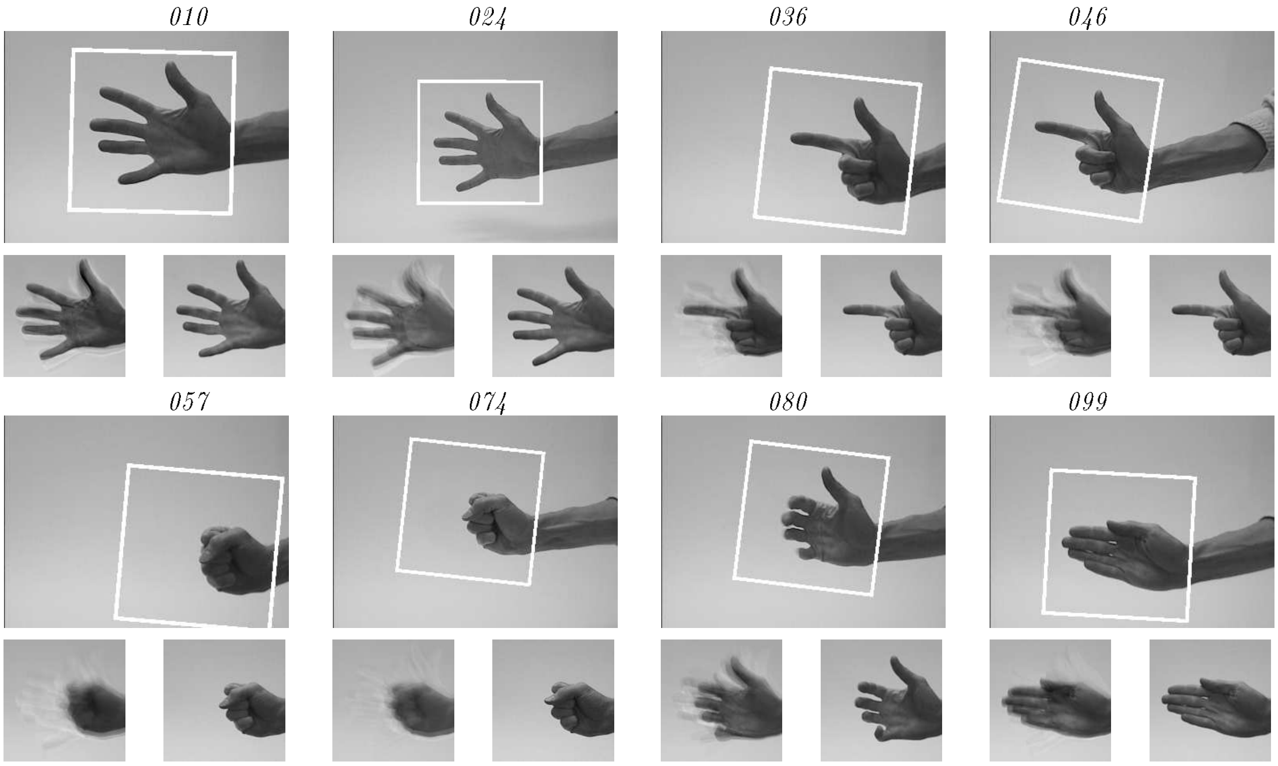


Fig. 16. Tracking and recognizing hand gestures in video.

the left below this, the reconstructed image. To the right of the reconstructed image is the "closest" image in the original training set. The closest image is determined by computing a simple Euclidean distance between the estimated match coefficients for a given image and the coefficients for each of the training images. The closest training images indicate the feasibility of recognizing the gestures from the recovered coefficients. While most of the sequence contained recognizable gestures, frame 080 shows one of the transition frames between gestures. Note that while the frame is well approximated and can be tracked, it is not one of the four defined gestures but rather it is recognized as one of the transition states present in the training sequence.

Out of the 100 frames in the sequence, the gesture was misclassified (the wrong nearest training image was found) in only four frames. Despite these misclassifications, the algorithm maintained tracking and quickly recovered.

Note that a textureless background was used for this experiment. The robust estimation scheme can tolerate situations in which roughly 30-40% of the data are outliers. In the case of the hand the majority of the data modeled in the view-based representation correspond to the background; see, for example, the "fist" gesture in Figure 15. Tracking this model against a textured background is made difficult by the high percentage of outliers. We are exploring the idea of learning a view-based "mask" along with the view based representation. A mask indicates where the object is in each of the training images and an "eigen-mask" is recovered with the robust reconstruction. This mask is used in the optimization scheme to down-weight measurements which are not expected to correspond to the object.

This simple example illustrates how deformable motions such as hand gestures can be split into a view-based component and a motion component. The non-rigid motion of the fingers is modeled using a view-based representation while the scale, translation, and rotation of the entire hand is modeled in terms of its image motion. It may be impractical to model more complex articulated motions involving the independent motion of many parts using such a simple view-based scheme. The extension of these view-based methods to articulated objects with many independent parts (for example the entire human body) is an area of ongoing research.

6.6. Discussion

In constructing our training sets we were careful not to include any images that contained affine transformations of the objects. If we are not so careful in constructing the training set, the solution to Equation (13) may be ambiguous; that is, for some input image it may be possible to accurately reconstruct the image using more than one set of coefficients and transformation parameters. In general, to use the EigenTracking approach one wants to carefully separate those image deformations that are due to changes in the appearance of the object and those that are due to the global transformations of interest.

Even though we were careful in constructing our training sets some examples of this ambiguity were present. For example when tracking the rotating cans, the algorithm would tend to "lock on" to a particular view and account for the changing view over a few frames as image motion. Eventually the the reconstructed fit would become poor and the algorithm would "jump" to a new view. In the case of the soda cans, this behavior might be exaggerated due to the high-frequency texture in the images. As mentioned earlier, the linear subspace does not do a good job of smoothly interpolating intermediate views with highly textured objects such as this. The algorithm may be finding the "closest" good view represented by the eigenspace and tracking that view until the images change enough that the closest view changes. Despite this behavior, the algorithm was able to maintain its tracking of the soda cans over the long image sequences. With images such as faces, this is expected to be less noticeable.

7. Conclusions

This paper has described robust eigenspace matching, the recovery of parameterized transformations between an image region and an eigenspace representation, and the application of these ideas to EigenTracking and gesture recognition. These ideas extend the useful applications of eigenspace approaches and provide a new form of tracking for previously viewed objects. In particular, the robust formulation of the subspace matching problem extends eigenspace methods to situations involving occlusion, background clutter, noise, etc. Currently these problems pose serious limitations to the usefulness of the eigenspace approach. Furthermore, the recovery of parameterized transformations in a continuous optimization framework provides an implementation of a *views+transformation* model for object recognition. In this model a small number of views are represented, and the transformation between the image and the nearest view is recovered. Finally, the experiments in the paper have demonstrated how a view-based representation can be used to track objects, such as human hands, undergoing both changes in viewpoint and changes in pose.

There are a two immediate future directions for this work. First, face recognition is the most common application of eigenspace techniques and is likely to benefit from the robust views+transformation model. The robust framework should provide insensitivity to the background and moderate variations in appearance, while the parameterized transformations would allow recognition of faces in novel poses. Second, the efficiency of the tracking method should be enhanced by the incorporation of standard motion estimation techniques, prediction, filtering, and stochastic optimization techniques. In particular, the stochastic gradient descent algorithm discussed by Viola (1995) has been shown to provide a significant speed-up on roughly similar problems. Also, the work of Hager and Belhumeur (1996) combines fast affine motion estimation with a view-based representation of illumination to perform real-time tracking of objects with changing illumination. It may be possible to incorporate elements such as these into the EigenTracking framework to achieve near real-time tracking of objects with changing views.

Our long-term work is focused on the tracking and recognition of more complex articulated objects that are changing both in position and view. The simple hand tracking example in the previous section does not deal with the problem of recognizing the hand gestures from an arbitrary view but rather deals with affine deformations and articulations from a single viewpoint. Additionally we have not addressed the problem of recognizing gestures, or activities, that have temporal extent (cf. (Bobick and Wilson, 1995)). While there are a number of human-computer interaction applications where our simple gesture tracking and recognition approach would be appropriate, more work needs to be done to recognize and track gestures or poses of complex articulated objects, such as hands or the human body, from an arbitrary viewpoint.

Appendix

Optimization Details and Implementation

We use a coarse-to-fine strategy to minimize the robust subspace constancy objective function given in equation (13). For each level of the multi-scale pyramid, say $l = 0, \ldots, L$, this objective function, $E_l(\mathbf{c}_l, \mathbf{a}_l)$, is

$$\sum_{\mathbf{x}} \rho(I_l(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}_l)) - [U_l \mathbf{c}_l](\mathbf{x}), \sigma). \tag{A1}$$

Here $l = 0$ corresponds to the full resolution level and $l = L$ denotes the coarsest level.

The algorithm we use is based on the application of two simpler algorithms, one for minimizing the objective function with respect to $\mathbf{c}$ alone, the other for variations in $\mathbf{a}$ alone. We consider these two simpler algorithms next, as applied to a particular level. Given these components we then describe the overall multi-scale algorithm.

Eigenspace Coefficients

First consider the minimization with respect to $\mathbf{c}$ at some level l and some fixed value of σ. (For convenience we will drop the level l from the notation.) Suppose we have the initial guess $(\mathbf{c}^0, \mathbf{a}^0)$, which is typically obtained from the previous level of the pyramid. We use a Gauss-Newton optimization scheme (Bergen *et al.*, 1992),(Black and Anandan, 1993a) to update $\mathbf{c}$ according to

$$c_i^{n+1} = c_i^n - \delta c_i,$$

with δc_i given by

$$\delta c_i = \frac{1}{w(c_i)} \frac{\partial}{\partial c_i} E(\mathbf{c}^n, \mathbf{a}^0)$$

$$= \frac{1}{w(c_i)} \sum_{\mathrm{x}} U_i(\mathbf{x}) \psi(I(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0)) - [U\mathbf{c}^n](\mathbf{x}), \sigma). \quad \text{(A2)}$$

The normalizing term $w(c_i)$ is defined as

$$w(c_i) = \sum_{\mathrm{x}} (U \frac{\partial}{\partial c_i} \mathbf{c})^2 \max \psi' = \sum_{\mathrm{x}} U_i^2 \max \psi',$$

where U_i^2 is the square of U_i at pixel $\mathbf{x}$, and

$$\max \psi' = \max_x \frac{\partial^2}{\partial x^2} \rho(x, \sigma) = \frac{2}{\sigma^2}$$

for the robust error norm used in the paper. These updates are computed for k iterations, or until convergence.

Incremental Warp Linearization

Given the resulting value of $\mathbf{c}$ for the some level l and some σ, we then consider updates of the warp parameters, $\mathbf{a}$. We use the general approach developed for the robust regression of optical flow (see (Black and Anandan, 1993a)). In this approach, the need to rewarp the image for each update of $\mathbf{a}$ is avoided by linearizing the variation of $I(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}))$ with respect to $\mathbf{a}$. In particular, setting $\mathbf{a} = \mathbf{a}^0 + \mathbf{b}$ and performing a Taylor's expansion in the incremental warp parameters $\mathbf{b}$, provides

$$I(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0 + \mathbf{b})) = I + \nabla I \frac{\partial \mathbf{u}}{\partial \mathbf{a}} (\mathbf{x}, \mathbf{a}^0) \mathbf{b} + O(||\mathbf{b}||^2).$$

Here both I and $\nabla I = [I_x \; I_y]$ are evaluated at $(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0))$. Notice the affine displacement $\mathbf{u}(\mathbf{x}, \mathbf{a})$ used in this paper is a linear function of the warp parameters $\mathbf{a}$, so that

$$\frac{\partial \mathbf{u}}{\partial \mathbf{a}} (\mathbf{x}, \mathbf{a}^0) \mathbf{b} = \mathbf{u}(\mathbf{x}, \mathbf{b}).$$

Using this in the above Taylor's expansion, and substituting the result into (A1) gives

$$\sum_{\mathrm{x}} \rho(\nabla I(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0)) \cdot \mathbf{u}(\mathbf{x}, \mathbf{b}) + (I(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0)) - [U\mathbf{c}](\mathbf{x})), \sigma), \quad \text{(A3)}$$

which we refer to as the approximate objective function, $\tilde{E}(\mathbf{c}, \mathbf{b})$. Notice that $\tilde{E}(\mathbf{c}, \mathbf{b})$ takes the form of a robust motion constraint objective function (Black and Anandan, 1993a), but here the out-of-subspace projection $I - U\mathbf{c}$ plays the role of the temporal derivative.

The approximate objective function $\tilde{E}$ has three important properties. First notice that $\tilde{E}$ only involves the evaluation of I and ∇I at $(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0))$, which does not depend on the incremental warp $\mathbf{b}$. We therefore obtain the desired result of avoiding the need to recompute warps of I and ∇I during the computation of $\mathbf{b}$; instead these quantities can be precomputed given the initial guess $\mathbf{a}^0$.

Secondly $\tilde{E}$ is a good approximation of the original objective function for small incremental warps $\mathbf{b}$. Indeed, from the above derivation it follows that

$$E(\mathbf{c}, \mathbf{a}^0 + \mathbf{b}) = \tilde{E}(\mathbf{c}, \mathbf{b}) + O(||\mathbf{b}||^2). \quad \text{(A4)}$$

In practice, the coarse-to-fine strategy will lead to estimated incremental warps of no more than a pixel or so (in the subsampled grid), in which case $\tilde{E}$ provides a close approximation of E. Therefore it is reasonable to attempt to minimize $\tilde{E}$ with respect to $\mathbf{b}$ in order to compute an update for $\mathbf{a}^0$.

Finally, the third property of $\tilde{E}(\mathbf{c}, \mathbf{b})$ is that if it has a minimum at $\mathbf{b} = \mathbf{0}$, then the gradient, $E_{\mathbf{a}}(\mathbf{c}, \mathbf{a}^0)$, of the original objective function must also vanish (see (A4)). This is important since, upon convergence our overall algorithm produces a negligible update $\mathbf{b}$, and so this third property ensures that the original objective function $E(\mathbf{c}, \mathbf{a}^0)$ also has a zero gradient with respect to the warp parameters. That is, upon convergence, $(\mathbf{c}, \mathbf{a}^0)$ is a stationary point of the original objective function, typically a local minimum. Thus the error in approximating the original objective function by the computationally convenient, $\tilde{E}$, vanishes upon convergence of our overall algorithm.

Warp Parameters

The minimization of $\tilde{E}(\mathbf{c}, \mathbf{b})$ with respect to $\mathbf{b}$ is done using a similar Gauss-Newton algorithm to the one described above for updating $\mathbf{c}$. That is, the updates for $\mathbf{b}$ are

$$b_i^{n+1} = b_i^n - \delta b_i,$$

for $\mathbf{b}^0 = \mathbf{0}$ and

$$\begin{aligned}\delta b_i &= \frac{1}{w(b_i)} \frac{\partial}{\partial b_i} \tilde{E}(\mathbf{c}, \mathbf{b}) \\ &= \frac{1}{w(b_i)} \sum_{\mathrm{x}} \nabla I \cdot \frac{\partial}{\partial b_i} \mathbf{u}(\mathbf{x}, \mathbf{b}) \psi(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{b}) \\ &\qquad + (I - U\mathbf{c}), \sigma), \quad \text{(A5)}\end{aligned}$$

with I and ∇I evaluated at $(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}^0))$. The normalizing term $w(b_i)$ is defined as

$$w(b_i) = \sum_{\mathrm{x}} \left(\nabla I \cdot \frac{\partial}{\partial b_i} \mathbf{u}(\mathbf{x}, \mathbf{b}) \right)^2 \max \psi',$$

where $\max \psi'$ is as above. Upon convergence of this iteration, or after a fixed number of steps, the new value of $\mathbf{a}$ is set to be $\mathbf{a}^0 + \mathbf{b}$.

Multi-scale Projection Operations

The overall algorithm executes several coarse to fine sweeps, during which estimates for $\mathbf{c}$ and $\mathbf{a}$ obtained at one level used to generate initial guesses at the next finer level. For the warp parameters, the updated $\mathbf{a}$ can be 'projected' to the next finer scale simply by multiplying a_0 and a_3 by 2 (see (11) and (12)).

The projection of the eigenspace coefficients $\mathbf{c}$, however, can be more of a problem. Suppose $\mathbf{c}_{l+1}$ is the vector of eigenspace coefficients computed in the coarser level $l+1$. We seek an initial guess, say $\mathbf{c}_l^0$, for these coefficients at the next finer level. There are several ways to do this, depending on the structure of the basis vectors across scales.

One approach would be to simply use the robust fitting algorithm discussed in Section 4. That is, first obtain a least squares estimate for the new coefficients $\mathbf{c}_l$. Then use this estimate for the starting point of the algorithm described above for updating $\mathbf{c}$ alone, gradually reducing σ back down from a temporarily inflated value. As we saw in the results presented in Section 4, this approach has an empirical breakdown point of 30-50% outliers.

This strategy could be improved by using some information about the spatial distribution of inliers, determined at the previous level $l+1$, to compute the initial estimate for $\mathbf{c}_l$. In particular, for a residual reconstruction error $r_{l+1}(\mathbf{x}) = I_{l+1} - U_{l+1}\mathbf{c}_{l+1}$, define the weight $m_{l+1}(\mathbf{x})$ to be

$$m_{l+1}(\mathbf{x}) = \frac{1}{2} \psi(r_{l+1}(\mathbf{x}), \sigma) / r_{l+1}(\mathbf{x}).$$

These weights can be projected to level l in the pyramid and used to compute a weighted least-squares estimate of $\mathbf{c}_l$. This approach should be able to downweight the majority of pixels at which there are outliers, thereby increasing the breakdown point.

However, in the tests reported in this paper, we used a much simpler strategy for projecting the eigenspace coefficients $\mathbf{c}$. We noted that the multi-scale pyramids for the soda cans and hands exhibited the property that the i^{th} basis function at level $l+1$, namely $U_{l+1,i}$, was well approximated by the filtered and subsampled version of the corresponding basis function at the next finer scale, $U_{l,i}$ (see Figure 7). Presumably this property arises from the correlation of information in the training set across scales. As a consequence, in our implementation of the robust fitting of a multi-scale eigenspace we simply took c_l^0, the initial guess for the expansion coefficients at the next finer scale, to be *equal to* c_{l+1}, the updated expansion coefficients at the coarser scale. It should be noted, though, that such a simple strategy is expected to work only when the eigenspace pyramids have this special structure.

Algorithm Summary

Finally, the coarse-to-fine sweep described above is repeated several times, gradually aligning the given image with the eigenspace as σ is reduced. The complete algorithm can be summerized as follows:

For q iterations or until convergence:

(a) For each level in the pyramid from coarse to fine

   i. perform k iterations of the update for each c_i to produce a new reconstruction,

ii. perform k iterations of the update for each b_i to produce an updated estimate of the transformation $\mathbf{a}$,

iii. project the $\mathbf{a}$ and $\mathbf{c}$ to the next level in the pyramid,

iv. warp the input image by $\mathbf{a}$ to register it with the eigenspace,

v. repeat.

(b) lower the value of σ according to the continuation strategy,

(c) repeat.

Notes

1. While other approaches have been described in the literature (cf. (Murase and Nayar, 1995)) and would work equally well, the basic method used here is conceptually straightforward.
2. We subtracted the mean from each of Figure 2 *a* and *b* and included the constant image in the expansion basis.
3. A variant of the EigenTracking approach described here could use a smaller number of basis vectors and use the coefficients of these to find the closest training image. This training image, rather than the reconstruction, could be used to solve for the transformation between the image and the eigenspace. The advantage of this would largely be efficiency; the high frequency information necessary for accurate tracking could be obtained with a smaller number of basis images.

References

E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, pages 1–20, Boston, MA, 1991. MIT Press.

A. Baumberg and D. Hogg. Learning flexible models from image sequences. In J. Eklundh, editor, *European Conf. on Computer Vision, ECCV-94*, volume 800 of *LNCS-Series*, pages 299–308, Stockholm, Sweden, 1994. Springer-Verlag.

J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, pages 237–252. Springer-Verlag, May 1992.

D. Beymer. Feature correspondence by interleaving shape and texture computations. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 921–928, San Francisco, June 1996.

M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. Int. Conf. on Computer Vision, ICCV-93*, pages 231–236, Berlin, Germany, May 1993.

M. J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. In *Proceedings of the International Conference on Computer Vision*, pages 374–381, Boston, Mass., June 1995.

A. Blake, M. Isard, and D. Reynard. Learning to track curves in motion. In *Proceedings of the IEEE Conf. Decision Theory and Control*, pages 3788–3793, 1994.

A. F. Bobick and A. D. Wilson. A state-based technique for the summarization and recognition of gesture. In *Proceedings of the International Conference on Computer Vision*, pages 382–388, Boston, Mass., June 1995.

C. Bregler and S. M. Omohundro. Surface learning with applications to lip reading. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 43–50, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models for shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 9–18, September 1992.

T. Darrell and A. Pentland. Space-time gestures. In *Proc. Computer Vision and Pattern Recognition, CVPR-93*, pages 335–340, New York, June 1993.

G. Hager and P. Belhumeur. Real-time tracking of image region with changes in geometry and illumination. To appear: *Proc. Computer Vision and Pattern Recognition, CVPR-96*, San Francisco, June 1999.

P. Hallinan. *A deformable model for the recognition of human faces under arbitrary illumination.* PhD thesis, Harvard University, Cambridge, MA, August 1995.

F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions.* John Wiley and Sons, New York, NY, 1986.

A. Jepson and M. J. Black. Mixture models for optical flow computation. In Ingmer Cox, Pierre Hansen, and Bela Julesz, editors, Partitioning Data Sets: With Applications to Psychology, Vision and Target Tracking, pages 271–286, DIMACS Workshop, April 1993. AMS Pub., Providence, RI.

C. Kervrann and F. Heitz. A hierarchical statistical framework for the segmentation of deformable objects in image sequences. In *Proc. Computer Vision and Pattern Recognition, CVPR-94*, pages 724–7228, Seattle, WA, June 1994.

D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 453–458, San Francisco, June 1996.

G. Li. Robust regression. In D. C. Hoaglin, F. Mosteller, and J. W. Tukey, editors, *Exploring Data, Tables, Trends and Shapes*, NY, 1985. John Wiley & Sons.

G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker Inc., N.Y., 1988.

B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings of the International Conference on Computer Vision*, pages 786–793, Boston, Mass., June 1995.

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

C. Nastar, B. Moghaddam, and A. Pentland. Generalized image matching: Statistical learning of physically-based deformations. In B. Buxton and R. Cipolla, editors, *European Conf. on Computer Vision, ECCV-96*, volume 1064 of *LNCS-Series*, pages 589–598, Cambridge, UK, 1996. Springer-Verlag.

S. K. Nayar, H. Murase, and S. Nene. Learning, positioning, and tracking visual appearance. In *IEEE Conf. on Robotics and Automation*, San Diego, May 1994.

A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. Computer Vision and Pattern Recognition, CVPR-94*, pages 84–91, Seattle, WA, June 1994.

J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proceedings of the International Conference on Computer Vision*, pages 612–617, Boston, Mass., June 1995.

P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, New York, 1987.

E. Saund. A multiple cause mixture model for unsupervised learning. In *Neural Computation*, volume 7, pages 51–71, 1995.

G. Strang. *Linear Algebra and its Applications.* Academic Press, New York, 1976.

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

M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. Computer Vision and Pattern Recognition, CVPR-91*, pages 586–591, Maui, June 1991.

P. A. Viola. *Alignment by Maximization of Mutual Information.* Ph.D. Thesis, AI-Lab., M.I.T., Cambridge, AI Technical Report 1548, March 1995.

Michael J. Black received his B.Sc. in 1985 from the University of British Columbia, his M.S. in 1989 from Stanford University, and his Ph.D. in 1992 from Yale University. Between 1990 and 1992, he was a visiting researcher at the NASA Ames Research Center, Aerospace Human Factors Research Division, from 1992 to 1993 he was an Assistant Professor in the Department of Computer Science at the University of Toronto, and since 1995 has been an Adjunct Professor in the same department. In 1993 Dr. Black joined the Xerox Palo Alto Research Center and is currently the head of the image understanding research group. He has won a number of awards including the IEEE Computer Society Outstanding Paper Award (CVPR'91) for his work with P. Anandan on robust motion estimation. His research interests include optical flow estimation, applications of robust statistics in computer vision, and human motion and gesture understanding.

Allan Jepson received the B.Sc. degree in 1976 from the University of British Columbia and the Ph.D. degree in Applied Mathematics in 1980 from the California Institute of Technology. He spent two years as a postdoctoral fellow at Stanford University in the Mathematics Department, and then moved to the Department of Computer Science at the University of Toronto where he is now a Professor. He was appointed a member of the Canadian Institute for Advanced Research from 1986 to 1995. His current research interests include various aspects of computer vision (see http://www.cs.toronto.edu/~jepson/).