Fast Eigenspace Decomposition of Correlated Images

Chu-Yin Chang, Anthony A. Maciejewski, *Senior Member, IEEE*, and Venkataramanan Balakrishnan, *Member, IEEE*

***Abstract*—We present a computationally efficient algorithm for the eigenspace decomposition of correlated images. Our approach is motivated by the fact that for a planar rotation of a two-dimensional (2-D) image, analytical expressions can be given for the eigendecomposition, based on the theory of circulant matrices. These analytical expressions turn out to be good first approximations of the eigendecomposition, even for three-dimensional (3-D) objects rotated about a single axis. In addition, the theory of circulant matrices yields good approximations to the eigendecomposition for images that result when objects are translated and scaled. We use these observations to automatically determine the dimension of the subspace required to represent an image with a guaranteed user-specified accuracy, as well as to quickly compute a basis for the subspace. Examples show that the algorithm performs very well on a number of test cases ranging from images of 3-D objects rotated about a single axis to arbitrary video sequences.**

I. Introduction

ONE OF the fundamental problems in computer vision is the recognition and localization of three-dimensional (3-D) objects. Subspace methods represent one computationally efficient approach for dealing with this class of problems. Variously referred to as eigenspace methods, principal component analysis methods, and Karhunen–Loeve transformation methods [1], these have been used extensively in a variety of applications such as face characterization [2] and recognition [3], lip-reading [4], [5], object recognition, pose detection, visual tracking, and inspection [6]–[9]. All of these applications are based on taking advantage of the fact that a set of highly correlated images can be approximately represented by a small set of eigenimages. Once the set of principal eigenimages is determined, online computation using these eigenimages can be performed very efficiently. However, the offline calculation required to determine both the appropriate number of eigenimages as well as the eigenimages themselves can be prohibitively expensive. This issue has been previously addressed by three different approaches. One class of techniques find the eigenimages iteratively. For example, the power method [10] and the conjugate gradient algorithm [11], [12] calculate one eigenimage at a time, while the block power method and Lanczos iteration [13] calculate a set of eigenimages together. Another class of techniques relies on updating a small set of eigenimages by recursively adding one image at a time. In [14], the number of eigenimages is fixed, while in [15], this number is adjusted based on the content of the added image. Another approach is based on structuring the computations in order to efficiently perform the matrix calculations involved [16]. The computational complexity of this approach is essentially independent of the number of desired eigenimages.

Our work addresses the computational expense of computing the desired eigenimages in a fundamentally different manner, resulting in considerable computational savings as compared to previous approaches. We present a brief overview of subspace methods in the next section, followed by the problem statement. In Section III, we use the theory of circulant matrices to derive an analytical expression for the eigendecomposition of images resulting from planar rotations. We also show how this theory can be used to derive good approximations for the eigendecomposition when images result from planar translation, and briefly consider scaling. In Section IV, we illustrate through a simple example that these analytical expressions represent a good approximation for the eigendecomposition of 3-D transformations as well. We use this observation as the core of an algorithm, outlined in Section V, to quickly compute the desired portion of the eigendecomposition based on a user-specified measure of accuracy. In Section VI, we evaluate the performance of our algorithm, first on images resulting from a 3-D rotation of test objects, and then on a set of twenty arbitrary video sequences. In all cases, the algorithm is seen to perform well both in terms of accuracy and computational efficacy.

II. Preliminaries

An image is an $h \times v$ array of square pixels with intensity values normalized between 0 and 1. Thus, an image will be represented by a matrix $\mathcal{X} \in [0, 1]^{h \times v}$. Since we will be considering sets of related images, it will be convenient to represent an image equivalently as a vector, obtained simply by "row-scanning," i.e., concatenating the rows to obtain the *image vector* $\mathbf{x}$ of length $m = hv$

$$\mathbf{x} = \mathrm{vec}(\mathcal{X}^T).$$

The *image data matrix* of a set of images $\mathcal{X}_1, \cdots, \mathcal{X}_n$ is an $m \times n$ matrix, denoted X, and defined as

$$X = [\mathbf{x}_1 \cdots \mathbf{x}_n]$$

with typically $m \gg n$. We consider only the case where n is fixed, as opposed to cases where X is constantly updated with new images.

Manuscript received June 30, 1998; revised March 21, 2000. This work was supported by the Sze Tsao Chang Memorial Engineering Fund, the National Imagery and Mapping Agency under Contract NMA201-00-1-1003, and by the Office of Naval Research under Contract N00014-97-1-0640. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Josiane B. Zerubia.

C.-Y. Chang is with Semiconductor Technologies and Instruments, Inc., Plano, TX 75074 USA.

A. A. Maciejewski and V. Balakrishnan are with the School of Electrical and Computer Engineering, West Lafayette, IN 47907–1285 USA (e-mail: maciejew@ecn.purdue.edu).

Publisher Item Identifier S 1057-7149(00)06976-1.

The *average image vector* is denoted $\overline{\mathbf{x}}$ and defined as

$$\overline{\mathbf{x}} = (\mathbf{x}_1 + \cdots + \mathbf{x}_n)/n.$$

The corresponding *average image data matrix*, denoted $\overline{X}$, is

$$\overline{X} = [\overline{\mathbf{x}} \cdots \overline{\mathbf{x}}].$$

The matrix $X - \overline{X}$, which we denote $\hat{X}$, has the interpretation of an "unbiased" image data matrix.

The singular value decomposition (SVD) of $\hat{X}$ is given by

$$\hat{X} = \hat{U}\hat{\Sigma}\hat{V}^T$$

where $\hat{U} \in \mathbb{R}^{m\times m}$ and $\hat{V} \in \mathbb{R}^{n\times n}$ are orthogonal, and $\hat{\Sigma} \in \mathbb{R}^{m\times n}$, with $\hat{\Sigma} = [\hat{\Sigma}_d\ \mathbf{0}]^T$, where $\hat{\Sigma}_d = \text{diag}(\hat{\sigma}_1, \cdots, \hat{\sigma}_n)$, with $\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_n \geq 0$, and $\mathbf{0}$ is an n by $m-n$ zero matrix. (When the singular values $\hat{\sigma}_i$ are not ordered, we will refer to the decomposition as an "unordered" SVD.) The SVD of $\hat{X}$ plays a central role in several important imaging applications such as image compression, pattern recognition and pose detection. The columns of $\hat{U}$, denoted $\hat{\mathbf{u}}_i$, $i = 1, \cdots, m$, are referred to as the eigenimages of $\hat{X}$; these can be interpreted as estimates of the eigenvectors of the covariance matrix of the image vector. The eigenimages provide an orthonormal basis for the columns of $\hat{X}$, ordered in terms of importance; the corresponding singular values measure how "aligned" the columns of $\hat{X}$ are, with the associated eigenimage. The components of the ith column of $\hat{V}$ measure how much each individual image contributes to the ith eigenimage.

In practice, the singular vectors $\hat{\mathbf{u}}_i$ are not known or computed exactly, and instead estimates $\mathbf{q}_1, \cdots, \mathbf{q}_k$ which form a k-dimensional basis are used. The accuracy of a practical implementation of subspace methods then depends on three factors: the properties of X, the dimension k, and the quality of the estimates $\mathbf{q}_i$. The measure we will use for quantifying this accuracy is the "energy recovery ratio," denoted ρ, and defined as

$$\rho(\hat{X}, \mathbf{q}_1, \cdots, \mathbf{q}_k) = \frac{\sum_{i=1}^{k} \|q_i^T \hat{X}\|^2}{\|\hat{X}\|_F^2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Note that if the $\mathbf{q}_i$ are orthonormal, $\rho \leq 1$, and for any given k achieves a maximum value of $(\sum_{i=1}^{k} \hat{\sigma}_i^2)/(\sum_{i=1}^{n} \hat{\sigma}_i^2)$ when $\text{span}(\mathbf{q}_1, \cdots, \mathbf{q}_k) = \text{span}(\hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_k)$.

The principal calculation required with subspace methods is the precomputation of estimates of the singular vectors $\hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_k$ of the $m \times n$ matrix $\hat{X}$. This is a very computationally expensive operation when m and n are very large. Reducing this computational expense by exploiting any correlation between image vectors has been the subject of much previous work [10]–[16]. Our solution to this problem uses a fundamentally different approach that is considerably faster than these methods when the image vectors are "correlated," as in many pose-detection problems. Our technique is motivated by the observation that the SVD of $\hat{X}$ can be determined in a closed form when the images are derived by a planar rotation of a single image about the surface normal, thus resulting in X^TX being circulant.[1] We describe this in the next section.

III. Planar Transformations

A. Planar Rotation

Consider an image data matrix where each $\mathbf{x}_{i+1}$ is obtained from $\mathbf{x}_i$ by a planar rotation[2] of $\theta = 2\pi/n$. Equivalently, $\mathbf{x}_i$ and $\mathbf{x}_j$ are related by a planar rotation of $|i-j|\theta$. Consider

$$X^TX = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 & \mathbf{x}_1^T\mathbf{x}_2 & \cdots & \mathbf{x}_1^T\mathbf{x}_n \\ \mathbf{x}_2^T\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{x}_2 & \cdots & \mathbf{x}_2^T\mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n^T\mathbf{x}_1 & \mathbf{x}_n^T\mathbf{x}_2 & \cdots & \mathbf{x}_n^T\mathbf{x}_n \end{bmatrix}. \quad (1)$$

To within an accuracy imposed by the resolution, $\mathbf{x}_i^T\mathbf{x}_j$ is a function of $|i-j|$. Also, $\mathbf{x}_1^T\mathbf{x}_j = \mathbf{x}_1^T\mathbf{x}_{n+2-j}$, for $2 \leq j \leq n$. Thus, row $i+1$ of X^TX can be obtained by a right-circular shift of row i (the first row is a right-circular shift of the last one). In other words, X^TX is a *circulant* matrix [18]. Much is known about the properties of such matrices; in particular, closed-form expressions can be given for their eigenvalues and eigenvectors: The eigenvalues of X^TX are simply given by the discrete Fourier transform (DFT) of its first row, and the eigenvectors given by the Fourier matrix F. That is

$$X^TX = F\Lambda F^* \quad (2)$$

where, with $\omega = e^{-j2\pi/n}$ and $P(z) = \sum_{i=0}^{n-1} \mathbf{x}_1^T\mathbf{x}_{i+1}z^i$

$$\Lambda = \text{diag}(P(\omega^0), P(\omega^1), \cdots, P(\omega^{n-1})) \quad (3)$$

and

$$F = [\mathbf{f}_0\ \mathbf{f}_1\ \cdots\ \mathbf{f}_{n-1}] = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix}. \quad (4)$$

It is easy to verify from (3) that the eigenvalues of X^TX satisfy $P(\omega^i) = P(\omega^{n-i})$ for $i = 1, 2, \cdots, \lfloor (n-1)/2 \rfloor$, and the corresponding eigenvectors (i.e., columns of F) are complex conjugates of each other. Therefore, a *real* eigendecomposition of X^TX is given by

$$X^TX = HDH^T \quad (5)$$

[1]This observation can be found in [17], which was published while this manuscript was under preparation.

[2]More precisely, the image $i+1$ is obtained by rotating the infinite-resolution image represented by the ith image, and then sampling it.

where D equals the $n \times n$ matrix

$$\text{diag}(P(\omega^0), P(\omega^1), P(\omega^1), P(\omega^2), \cdots) \tag{6}$$

and H consists of the first n columns of

$$\begin{aligned}&[\mathbf{h}_1\, \mathbf{h}_2\, \mathbf{h}_3\, \mathbf{h}_4\, \mathbf{h}_5 \cdots] \\ &= \sqrt{2}\left[\frac{1}{\sqrt{2}}\,\mathbf{f}_0\, \Re\mathbf{f}_1\, \Im\mathbf{f}_1\, \Re\mathbf{f}_2\, \Im\mathbf{f}_2 \cdots\right] \\ &= \sqrt{\frac{2}{n}}\begin{bmatrix} \frac{1}{\sqrt{2}} & c_0 & -s_0 & c_0 & -s_0 & \cdots \\ \frac{1}{\sqrt{2}} & c_1 & -s_1 & c_2 & -s_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \cdots \\ \frac{1}{\sqrt{2}} & c_{n-1} & -s_{n-1} & c_{2(n-1)} & -s_{2(n-1)} & \cdots \end{bmatrix}\end{aligned} \tag{7}$$

where $c_k = \cos(k\theta)$ and $s_k = \sin(k\theta)$. The above development means that Σ and V corresponding to an unordered SVD of X can be computed in a closed form. In particular, the square roots of the diagonal entries of D are the singular values of X, and $V = H$. To compute U, observe that $U\Sigma = XH$, which can be computed efficiently using Fast Fourier Transform (FFT) techniques. In particular, if Y is a matrix whose ith row is the FFT of the ith row of X, then $Y = \sqrt{n}\,XF$. The matrix XH can be formed from the first n columns of Y as

$$XH = \sqrt{\frac{2}{n}}\left[\frac{1}{\sqrt{2}}\,\mathbf{y}_0\, \Re\mathbf{y}_1\, \Im\mathbf{y}_1\, \Re\mathbf{y}_2\, \Im\mathbf{y}_2 \cdots\right]. \tag{8}$$

The above development has focused on obtaining an SVD of X. Note that the (unordered) SVD of $\hat{X}$ can be immediately obtained from the (unordered) SVD of X as follows:

$$\begin{aligned}\hat{X} &= X - \overline{X} \\ &= X - \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T \\ &= \sum_{i=2}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.\end{aligned}$$

In other words, for $i = 1, \cdots, n-1$, we have $\hat{\sigma}_i = \sigma_{i+1}$ (and similarly for $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$), with $\hat{\sigma}_n = 0$, $\hat{\mathbf{u}}_m = \mathbf{u}_1$, and $\hat{\mathbf{v}}_n = \mathbf{v}_1$.

B. Planar Translation

Consider a sequence of images of an object moving at a constant velocity.[3] It is easy to verify that to within an accuracy imposed by the resolution, $\mathbf{x}_i^T\mathbf{x}_j$ is a function of $|i-j|$. Thus, X^TX is a symmetric *Toeplitz* matrix. While no closed-form expressions exist for the eigendecomposition of symmetric Toeplitz matrices, we now show how the theory of circulant matrices can be used to derive good approximations for the eigendecomposition under certain conditions.

[3]We assume that the velocity vector lies in the plane perpendicular to the camera view vector, and the object stays in the field of view. We also assume that the perspective, background and lighting effects are negligible.

We begin by embedding the $n \times n$ Toeplitz matrix T in a $2n \times 2n$ circulant matrix C as follows:

$$C = \begin{bmatrix} T & R \\ R & T \end{bmatrix} \tag{9}$$

where R is a symmetric matrix whose upper triangular part is specified by

$$r_{(i,j)} = \begin{cases} \text{arbitrary} & \text{if } i = j, \\ t_{(1,\,n+1-j+i)} & \text{if } i < j \end{cases} \tag{10}$$

where the $r_{(i,j)}$ and $t_{(i,j)}$ denote the (i, j) elements of R and T, respectively.

Since C is a symmetric circulant matrix, following the development in Section III-A, we can write down an eigendecomposition of C as

$$C = F_c \Lambda_c F_c^* \tag{11}$$

where, with $\omega = e^{-j2\pi/(2n)}$ and $Q(z) = \sum_{i=0}^{n-1} c_{(1,\,i+1)} z^i$

$$\Lambda_c = \text{diag}(Q(\omega^0), Q(\omega^1), \cdots, Q(\omega^{2n-1})) \tag{12}$$

and

$$\begin{aligned}F_c &= [\mathbf{f}_0\, \mathbf{f}_1\, \cdots\, \mathbf{f}_{2n-1}] \\ &= \frac{1}{\sqrt{2n}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{2n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(2n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{2n-1} & \omega^{2(2n-1)} & \cdots & \omega^{(2n-1)(2n-1)} \end{bmatrix}.\end{aligned} \tag{13}$$

Using the properties of the matrix F_c, we may rewrite an SVD of C as

$$\begin{aligned}C &= \begin{bmatrix} T & R \\ R & T \end{bmatrix} \\ &= \begin{bmatrix} U^{(e)} & U^{(o)} \\ U^{(e)} & -U^{(o)} \end{bmatrix}\begin{bmatrix} S^{(e)} & 0 \\ 0 & S^{(o)} \end{bmatrix}\begin{bmatrix} V^{(e)} & V^{(o)} \\ V^{(e)} & -V^{(o)} \end{bmatrix}^T\end{aligned} \tag{14}$$

where

$$\begin{aligned}\begin{bmatrix} U^{(e)} \\ U^{(e)} \end{bmatrix} &= [\mathbf{f}_0, \mathbf{f}_2, \cdots, \mathbf{f}_{2n-2}], \\ \begin{bmatrix} U^{(o)} \\ -U^{(o)} \end{bmatrix} &= [\mathbf{f}_1, \mathbf{f}_3, \cdots, \mathbf{f}_{2n-1}] \\ S^{(e)} &= \text{diag}(|Q(\omega^0)|, |Q(\omega^2)|, \cdots, |Q(\omega^{2n-2})|), \\ S^{(o)} &= \text{diag}(|Q(\omega^1)|, |Q(\omega^3)|, \cdots, |Q(\omega^{2n-1})|)\end{aligned}$$

and $U^{(e)}$, $U^{(o)}$, $V^{(e)}$, and $V^{(o)}$ are $n \times n$ matrices.

We then note the following.

- $U^{(e)}$, $U^{(o)}$, $V^{(e)}$, and $V^{(o)}$ are orthogonal matrices.
- The diagonal entries of $S^{(e)}$ and $S^{(o)}$ are the singular values of C corresponding to "even" and "odd" harmonics respectively.

We then have

$$T = U^{(e)}S^{(e)}V^{(e)^T} + U^{(o)}S^{(o)}V^{(o)^T}. \tag{15}$$

In the special case where either $S^{(e)}$ or $S^{(o)}$ is identically zero, equation (15) provides an unordered SVD of T. (See Appendix A for one such case.) When the motion of the object between frames is not too fast, which is typically the case, only the first few diagonal entries of both $S^{(e)}$ and $S^{(o)}$ will be significant. Then, with $\mathbf{u}_i^{(e)}$, $\mathbf{u}_i^{(o)}$, $\mathbf{v}_i^{(e)}$, and $\mathbf{v}_i^{(o)}$, denoting the ith columns of $U^{(e)}$, $U^{(o)}$, $V^{(e)}$, and $V^{(o)}$ respectively, and with $\mathbf{s}_i^{(e)}$ and $\mathbf{s}_i^{(o)}$ denoting the ith diagonal entry of $S^{(e)}$ and $S^{(o)}$ respectively, we have

$$T \approx \sum_{i=1}^{n_1} \mathbf{s}_i^{(e)}\mathbf{u}_i^{(e)}\mathbf{v}_i^{(e)^T} + \sum_{i=1}^{n_2} \mathbf{s}_i^{(o)}\mathbf{u}_i^{(o)}\mathbf{v}_i^{(o)^T}$$

where $n_1, n_2 \ll n$. Moreover, it is easy to show that for each k, the outer product $\mathbf{u}_k^{(o)}\mathbf{v}_k^{(o)^T}$ can be well approximated by a linear combination of the outer products $\mathbf{u}_1^{(e)}\mathbf{v}_1^{(e)}$, $\mathbf{u}_2^{(e)}\mathbf{v}_2^{(e)}$, $\cdots$, $\mathbf{u}_{k+\delta}^{(e)}\mathbf{v}_{k+\delta}^{(e)}$, where δ is small. Consequently

$$T \approx \sum_{i=1}^{\max(n_1,n_2)+\delta} \sigma_i \mathbf{u}_i^{(e)}\mathbf{v}_i^{(e)^T}.$$

In summary, we have an analytical approximation to the singular vectors of T for the planar translation case.[4]

IV. Extension to Three-Dimensional Transformations

In the previous section, we showed that for pure planar transformations, it is possible to derive analytical expressions (or good approximations) for the eigendecomposition. In this section, we consider how well these analytical expressions serve as approximations to the eigendecomposition for the more general case of 3-D transformations. One approach toward this end is to apply results from a perturbation analysis of eigendecompositions [19], [20] to quantify the changes in the eigendecomposition when the 3-D transformation is regarded as a perturbed version of the ideal case. However, the bounds obtained from such an analysis turn out to be crude, and not very effective in drawing conclusions that are useful in practice. Therefore, we use an alternate approach, where we examine a simple 3-D problem where the effects of the nonplanar nature of the transformation can be parametrized by a single variable. Specifically, we consider the case where the n images in the image data matrix are obtained from 3-D objects rotated about a single axis at increments of $2\pi/n$, with the camera view vector making an angle of α with the axis of rotation. We assume that the entire object is always within the field of view (see Fig. 1). We will see shortly that this study yields considerable insight into when and why the analytical expressions for the eigendecomposition serve as good approximations to the true eigendecomposition for more general 3-D transformations. Perhaps more importantly, the conclusions from this study yield a straightforward technique for using the analytical expressions as a basis for a computationally efficient algorithm for computing the true eigendecomposition.

[4]Some of the development for the planar translation case applies to scaling as well, although not as gracefully. We explore this further in Appendix B.

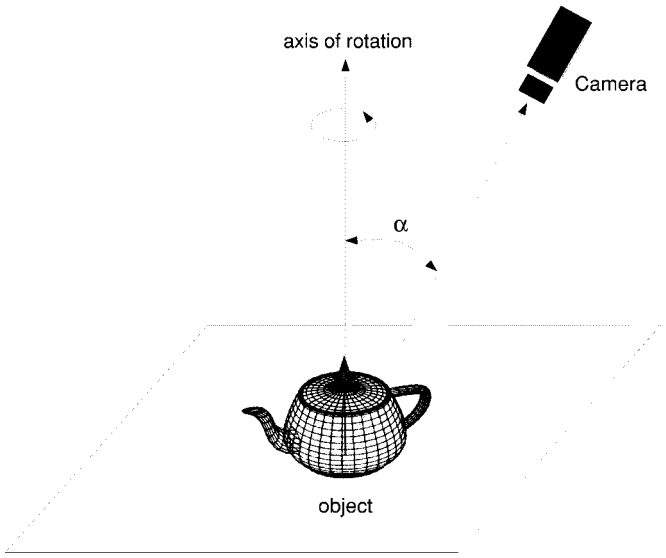

Fig. 1. Framework for obtaining images of a 3-D object, rotated about a single axis through the object. The camera view vector makes an angle of α with the axis of rotation, with the entire object always lying completely within the camera's field of view.

Returning to our example, suppose that the axis of rotation and the camera view vector are aligned (i.e., $\alpha = 0$). Then, all of the results of Section III-A apply directly.[5] If α is nonzero, then in general, the results of Section III-A do not apply. However, consider a *planar* object whose surface normal is aligned with the axis of rotation. Then, the results of Section III-A apply independent of α. To see this, let $\mathbf{x}_i(\alpha)$ denote the image vector of the object with camera view angle α, and note that we have

$$\mathbf{x}_i(\alpha)^T\mathbf{x}_j(\alpha) = \mathbf{x}_i(0)^T\mathbf{x}_j(0)\cos\alpha + c(\alpha) \tag{16}$$

where $c(\alpha)$ represents the contribution due to the background of the image [21]. From the arguments in Section III-A, it follows that $\mathbf{x}_i(0)^T\mathbf{x}_j(0)$ is only a function of $|i-j|$, and consequently, so is $\mathbf{x}_i(\alpha)^T\mathbf{x}_j(\alpha)$.

To explore the consequence of 3-D effects that arise when α is nonzero, we select as the object a cylinder that is half-black and half-white, split along the longitudinal axis. In the first scenario, the cylinder is viewed along the longitudinal axis, so that it appears as a circle. Images are taken at increments of 4° while the cylinder is rotated along the longitudinal axis. Fig. 2 shows nine of the $n = 90$ images that make up the image data matrix X, as well as the singular vectors of X. The results of Section III-A apply here, and it is seen that the right singular vectors are pure sinusoids of frequencies that are multiples of $2\pi/n$ rad. Moreover, an examination of the (ordered) singular vectors reveals that they correspond to harmonics of increasing frequency. Note that this ordering is an artifact of this particular example, and is not true in general.

[5]This assumes that all the light sources rotate with the object, or equivalently the camera rotates and everything else is stationary. We will also assume that the camera is far enough away that perspective effects can be neglected, that the background is uniform, and that resolution effects are negligible.

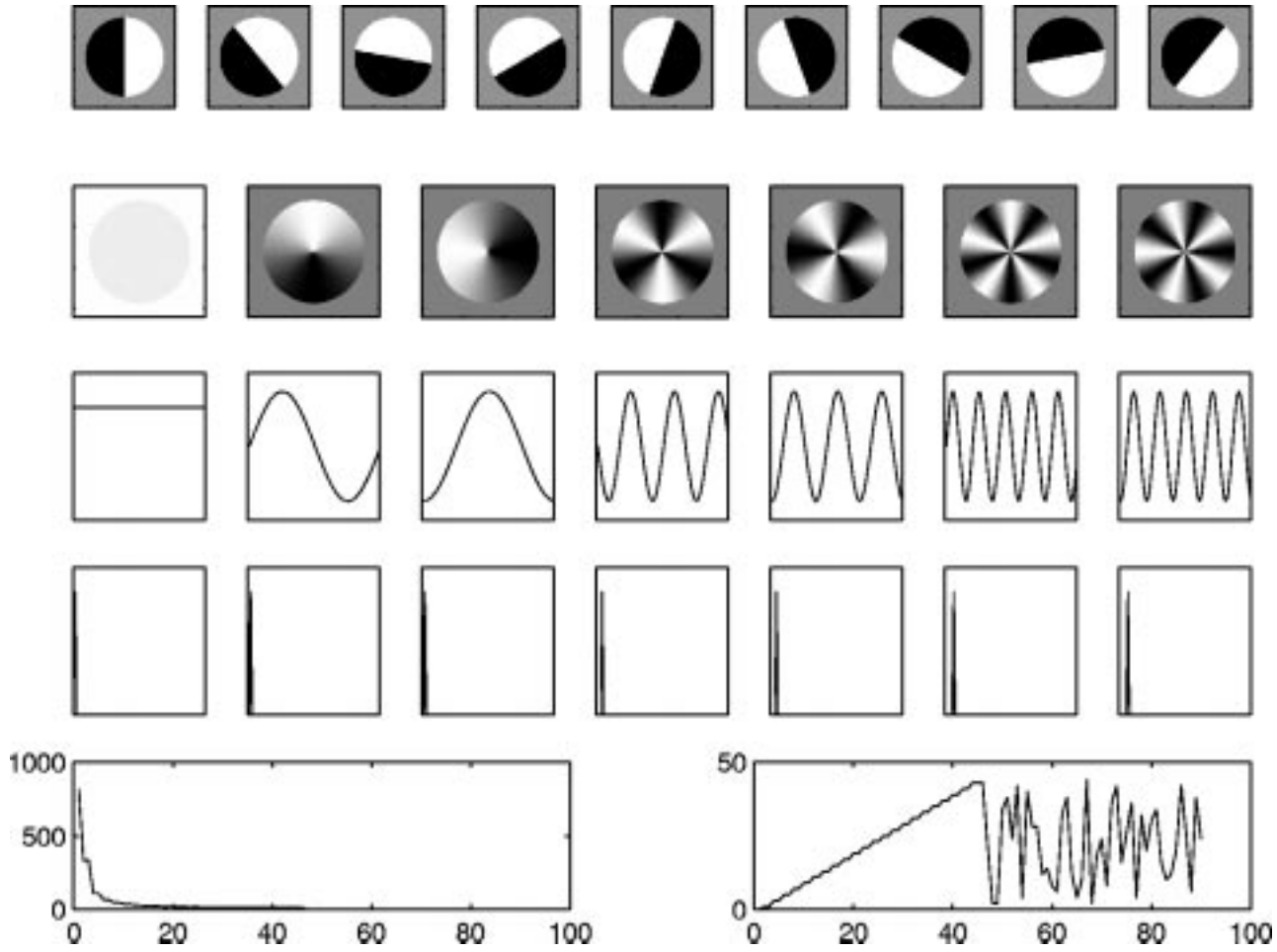

Fig. 2. Eigendecomposition of the image matrix X obtained from rotating a half black, half white cylinder, with a view angle of $\alpha = 0$. The first row shows nine of the 90 images of the image data matrix X. The second row shows the first seven eigenimages (left singular vectors of X) using the same gray scale encoding, with white denoting the maximum positive pixel value and black denoting the maximum negative value. The third row shows the first seven right singular vectors of X. The fact that these are pure sinusoids is illustrated in the fourth row, where the FFT magnitude-squared, i.e., the "power spectra" of these right singular vectors are shown. The plot on the left in the last row row shows the singular values of X. Note that the singular values from indices 45 onwards are identically zero, due to the symmetry of the object. The plot on the right shows the frequency at which the power spectra of the corresponding right singular vectors achieves a maximum (i.e., the "dominant" frequencies). It can be seen that the dominant frequencies of the power spectra of the right singular vectors corresponding to nonzero singular values increase linearly with their harmonic index.

In the second scenario, the camera is placed at a view angle of $\alpha = 60°$. Once again, images are taken at increments of 4° with the cylinder rotated along the longitudinal axis. Fig. 3 shows nine of the $n = 90$ images that make up the image data matrix X, as well as the singular vectors of X. Though the results of Section III-A do not apply here, two properties are again immediately apparent. 1) The right singular vectors are well-approximated by sinusoids of frequencies that are multiples of $2\pi/n$ radians, and the magnitude-squared of the spectra, i.e., the "power spectra" of the right singular vectors consist of a narrow band around the corresponding dominant harmonics. 2) The dominant frequencies of the power spectra of the (ordered) singular vectors increase approximately linearly with their index. These properties (assuming they hold) suggest an approach for reducing the expense in computing the eigendecomposition to within a prespecified accuracy. In particular, the first property means that the singular vectors are approximately spanned by a handful of harmonics. In addition, if the second property holds, then the frequencies of the dominant harmonics can be quickly identified by simply searching from low to high frequencies. Consequently, by projecting the row space of X to a smaller subspace spanned by a few of the harmonics, the computational expense associated with the SVD computation can be significantly reduced. Note that this approach can be used to generate the SVD to within any prespecified accuracy; the deviation of the actual singular vectors from pure harmonics only affects the computational savings that this approach offers. [It is also important to note that circulancy, by itself, only guaranteed property (1); thus circulancy is not sufficient for this approach to work well. However, the discussion on circulant matrices provides a sound theoretical basis for our approach.] Empirical evidence (see Section VI) suggests that the two properties discussed above hold true for sequences of images whose content varies slowly, independent of the underlying transformation (rotation, translation, scaling, etc.). We describe the details of the fast eigenimage computation technique in the next section.

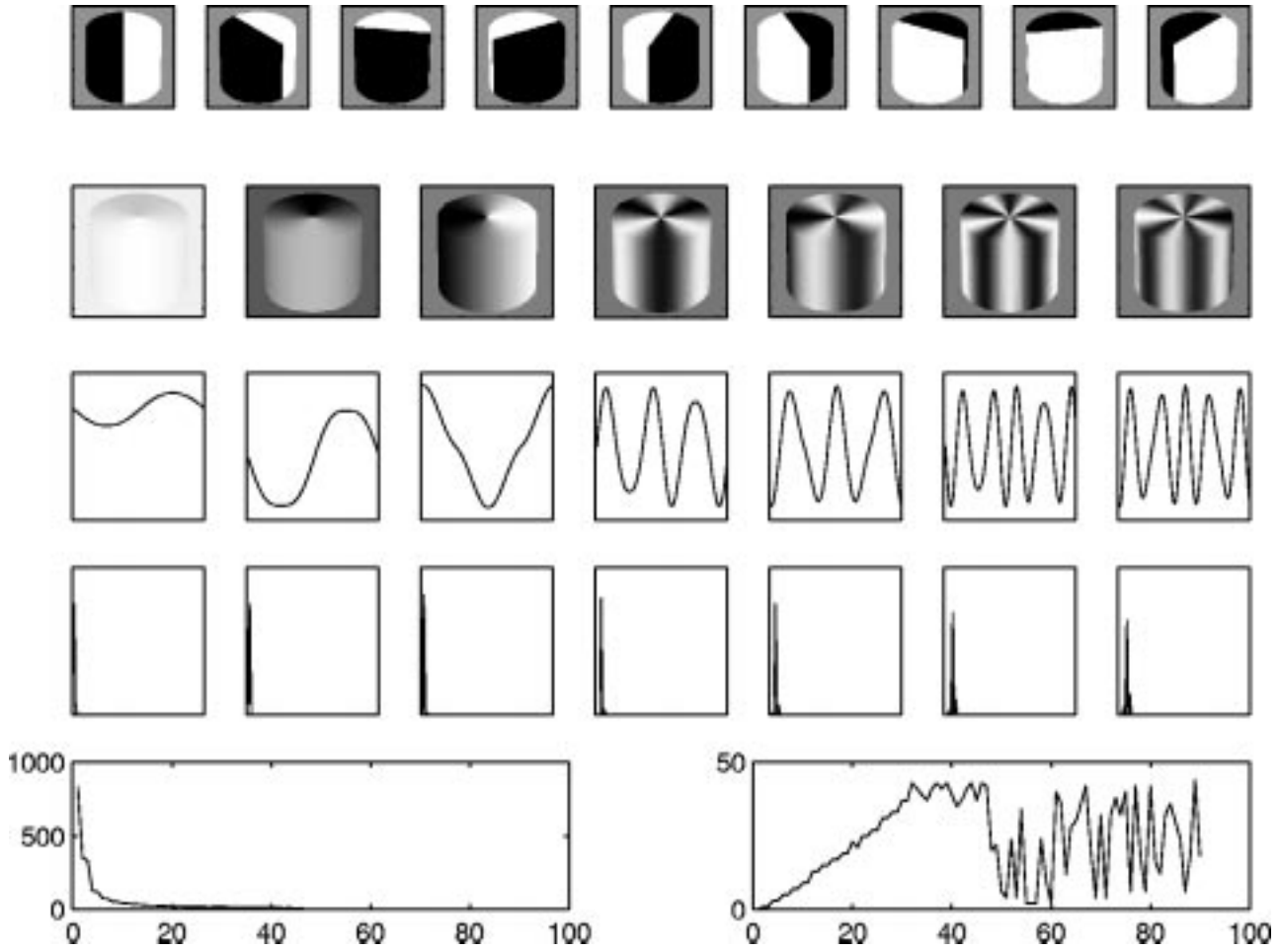

Fig. 3. Eigendecomposition of the image matrix X obtained from rotating a half black, half white cylinder, with a view angle of $\alpha = 60°$, with the data presented exactly as in Fig. 2. From the fourth row, it is apparent that though the right singular vectors of X are not pure sinusoids, their power spectra are concentrated in a narrow band around frequencies that are harmonics of $2\pi/n$. It can also be seen that as with the $\alpha = 0$ case, the dominant frequencies of the power spectra of the right singular vectors corresponding to nonzero singular values increase approximately linearly with their index.

V. Fast Eigendecomposition Algorithm

Our objective is to determine the first k left singular vectors of X. Let p be such that the power spectra of the first k singular vectors are essentially restricted to the band $[0, 2\pi p/n]$. Owing to the properties of the singular vectors discussed in the previous section, p is typically not much larger than k. Let H_p denote the matrix comprising the first p columns of H [i.e., the first p columns of the matrix given in (7)]. Then the first k singular values $\tilde{\sigma}_1, \cdots, \tilde{\sigma}_k$ and the corresponding left singular vectors $\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k$ of XH_p serve as excellent estimates to those of X. (Note that XH_p typically has far fewer columns than X, so that its SVD can be computed much more quickly.) Moreover, the accuracy of the approximated singular vectors with power spectra concentrated around "lower" frequencies will tend to be better, i.e., the smaller i is, the better estimate $\tilde{\mathbf{u}}_i$ is of $\mathbf{u}_i$. This is illustrated in Fig. 4 for a typical image (Object 1 from Fig. 5), where we have used ρ to measure the quality of the estimates of the $\mathbf{u}_i$. The solid line shows $\rho(X, \mathbf{u}_1, \cdots, \mathbf{u}_p)$ as a function of p, while the dashed lines show $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ for $k = 1, 2, \cdots, p$ and $p = 2, 4, 6, 8$. It is evident that for any p, the error $\rho(X, \mathbf{u}_1, \cdots, \mathbf{u}_k) - \rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ increases as k increases from 1 to p.

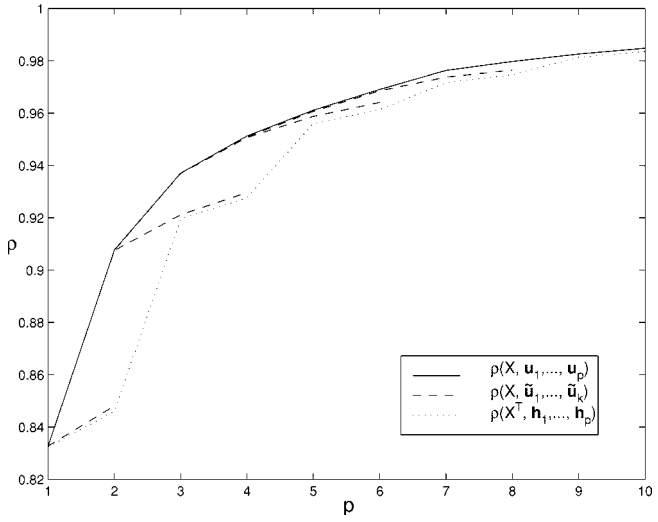

Fig. 4. Typical relationship between several energy recovery ratios as a function of k, $1 \leq k \leq p$, for several fixed values of p. (The plots shown here correspond to Object 1 from Fig. 5.) For fixed p, $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ behaves as a very good lower bound to $\rho(X, \mathbf{u}_1, \cdots, \mathbf{u}_k)$ for small k, and is very well approximated from below by $\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_k)$ for large k.

Our ultimate goal is to guarantee, upon termination, that $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ exceeds a user-specified threshold μ. While

Fig. 5. Objects used to evaluate the proposed algorithm (provided by [22]). The objects are rotated throughout 360° and 72 images were taken for each of them. Each image is of size 128 × 128 and is scale normalized such that the object touches a boundary.

$\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ depends critically on k and $\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k$, neither of which are available a priori, we show below that

$$\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_p) \geq \rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p) \tag{17}$$

where $\mathbf{h}_i$ denotes the ith column of H. The right-hand side of (17) is readily computed; and ensuring that $\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p) \geq \mu$ in turn guarantees that $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_p) \geq \mu$.

We now prove (17). Let the SVD of XH_p be

$$XH_p = \tilde{U}_p \tilde{\Sigma}_p \tilde{V}_p^T.$$

Note that $\tilde{U}_p = [\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_p]$. Then

$$\begin{aligned}\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_p) &= ||\tilde{U}_p^T X||_F^2 / ||X||_F^2 \\ &= ||\tilde{U}_p^T XH||_F^2 / ||X||_F^2 \\ &\geq ||\tilde{U}_p^T XH_p||_F^2 / ||X||_F^2 \\ &= ||\tilde{\Sigma}_p \tilde{V}_p^T||_F^2 / ||X||_F^2 \\ &= \rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p).\end{aligned}$$

From Fig. 4, it can be seen that $\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p)$ is a very conservative lower bound for $\rho(X, \mathbf{u}_1, \cdots, \mathbf{u}_p)$, with the quality of the bound improving uniformly with increasing p. For fixed p, $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ behaves as a very good lower bound

TABLE I
ALGORITHM PERFORMANCE ON ROTATIONALLY CORRELATED IMAGES THE PERFORMANCE OF OUR ALGORITHM IS COMPARED AGAINST THAT OF THE DIRECT SVD FOR THE OBJECTS IN FIG. 5. IN ALL CASES, THE EIGENIMAGES OF $\hat{\mathbf{X}}$ WERE COMPUTED WITH A DESIRED ENERGY RECOVERY RATIO OF 0.90. ALL COMPUTATIONS WERE PERFORMED USING MATLAB ON A HP9000/C110 WORKSTATION

	Dimension		Time (sec)	
Object no.	k	p	Our algorithm	Direct SVD
1	9	11	6.72	40.53
2	12	15	8.46	40.53
3	25	29	16.53	41.12
4	16	19	10.21	40.33
5	31	33	19.57	41.95
6	27	29	16.70	40.67
7	12	17	8.99	40.24
8	15	19	9.94	40.67
9	36	39	24.52	41.30
10	18	19	10.76	40.19
11	15	19	10.14	40.96
12	24	45	27.47	39.13
13	22	25	13.80	40.32
14	18	25	13.41	41.15
15	15	21	11.10	39.86
16	14	31	16.32	39.42
17	18	39	22.83	40.39
18	22	29	16.09	40.53
19	20	23	12.59	41.66
20	31	37	22.19	40.57

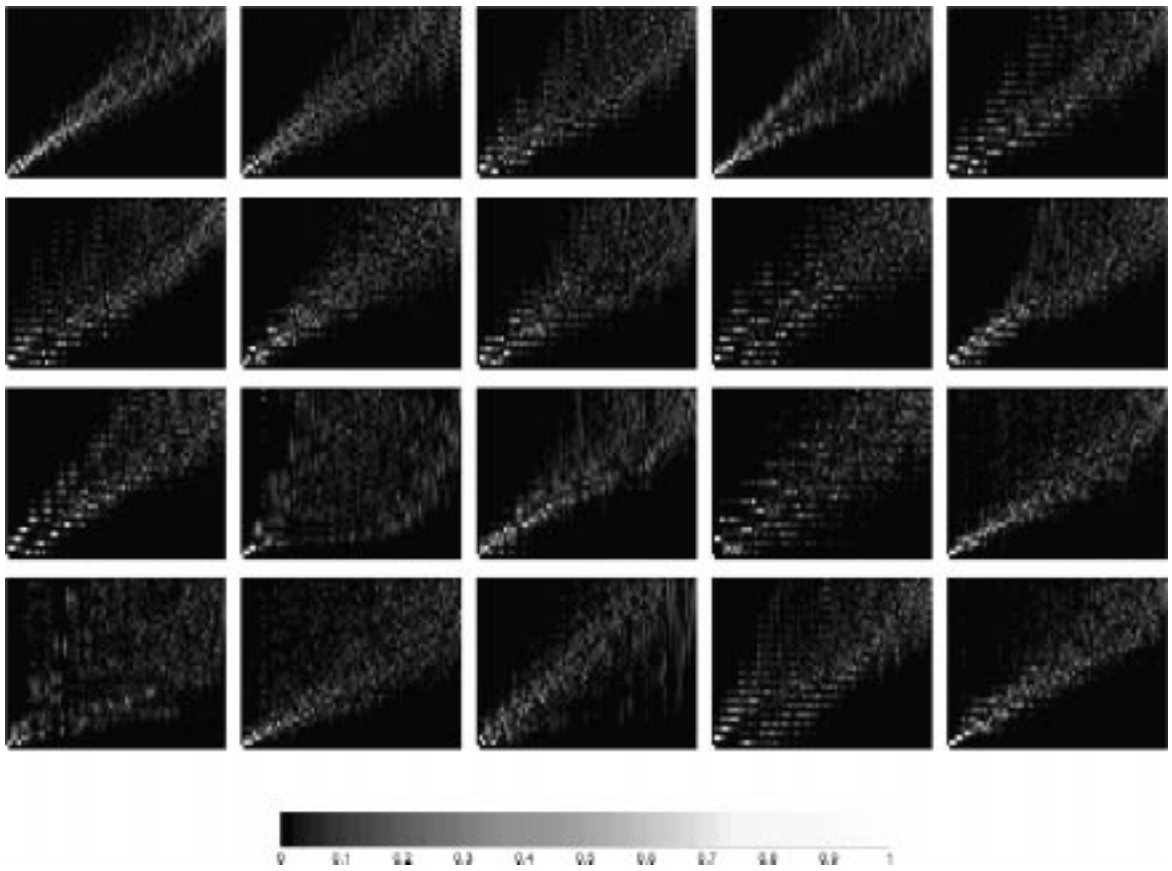

Fig. 6. For each of the objects in Fig. 5, the color-coded magnitude of the power spectra of the ordered right singular vectors are shown as a function of their index (along the x axis) and frequency (along the y axis). It can be seen that for most of the objects, the power spectrum of each right singular vector is approximately band-limited, and that the dominant frequency of each right singular vector increases roughly linearly with increasing index.

to $\rho(X, \mathbf{u}_1, \cdots, \mathbf{u}_k)$ for small k, and is very well approximated from below by $\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_k)$ for large k.

In summary, when p is chosen so as to satisfy $\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p) \geq \mu$, the quantity $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ turns out to exceed μ for some $k \leq p$, with $\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k$ being very good estimates for $\mathbf{u}_1, \cdots, \mathbf{u}_k$, and $\tilde{\sigma}_1, \cdots, \tilde{\sigma}_k$ being very good estimates for $\sigma_1, \cdots, \sigma_k$. The energy recovery ratio $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ can be efficiently approximated by $\sum_{i=1}^{k} \tilde{\sigma}_i^2 / \|X\|_F^2$.

The entire algorithm for the fast computation of a partial SVD of X can be summarized as follows.

1) Form the matrix Y, whose ith row is the FFT of the ith row of X.
2) Determine the smallest number p such that $\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p) > \mu$, where μ is the user-specified reconstruction ratio. The key observation here is that the matrix XH_p can be constructed as the first p columns of the matrix $\sqrt{2/n}[1/\sqrt{2}\,\mathbf{y}_0\, \Re\mathbf{y}_1\, \Im\mathbf{y}_1\, \Re\mathbf{y}_2\, \Im\mathbf{y}_2 \cdots]$, where $\mathbf{y}_i$ denotes the ith column of Y.
3) With Z_p denoting the first p columns of the matrix $[1/\sqrt{2}\,\mathbf{y}_0\, \Re\mathbf{y}_1\, \Im\mathbf{y}_2\, \Re\mathbf{y}_1\, \Im\mathbf{y}_2 \cdots]$, compute the SVD $Z_p = \sum_{i=1}^{p} \tilde{\sigma}_i \tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_i^T$.
4) Return $\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k$ such that $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k) \geq \mu$.

The above algorithm computes the partial SVD of X. If instead the partial SVD of $\hat{X}$ is sought, the algorithm is modified as follows. In Step 2, p is estimated as the smallest number p such that $\sum_{i=2}^{p} \|X\mathbf{h}_i\|^2 > \mu(\|X\|_F^2 - \|X\mathbf{h}_1\|^2)$. In Step 3, the SVD of the matrix comprising the second through p columns of Z is computed.

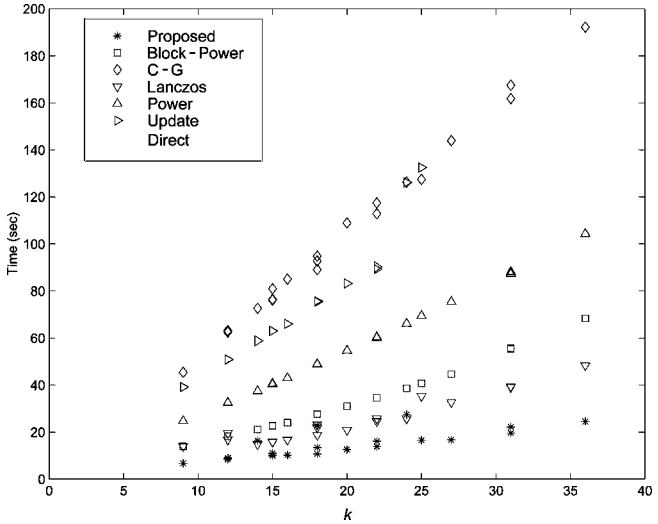

Fig. 7. This figure shows the computation time versus the subspace dimension k. For iterative algorithms, the stopping criterion was set to be such that the energy recovery ratio is comparable to or lower than that of the proposed method; in other words, the comparison with the iterative methods was by design conservative. In order to keep the comparison fair, the updating algorithm [14] was used to compute only the first k singular vectors. Note that all of these algorithms were implemented in MATLAB.

Fig. 8. The first, middle, and last frames of the 20 video sequences used to test our algorithm.

We briefly analyze the computational expense of our algorithm. The cost incurred in Step 1, i.e., performing the FFT of each row of X, requires $O(mn \log_2 n)$ flops. Step 2, that of estimating p, requires $O(mp)$ flops. In Step 3, the cost of computing the SVD of the matrix comprising the first p columns of $\sqrt{n/2}\,XH$ is of order $O(mp^2)$. Step 4, determining the needed dimension k, requires $O(mnk)$ flops. If $p \ll n$, then the total computation required is approximately $O(mn \log_2 n)$. This compares very favorably with the direct SVD approach which requires $O(mn^2)$ flops, and in most cases with the updating SVD method [14] as well, which requires $O(mnk^2)$ flops. The computational savings offered by our algorithm are significant if the condition

$$\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p) = \frac{\sum_{i=1}^{p} ||X\mathbf{h}_i||^2}{||X||_F^2} \geq \mu$$

holds for $p \ll n$. As the vectors $\mathbf{h}_i$ are harmonics consisting of increasing frequencies, this condition simply means that the individual pixel values do not change rapidly across the sequence

of images. This is typically the case, as evidenced by the examples presented in the next section.

VI. Examples

A. Rotationally Correlated Images

We first illustrate our approach on a database of images provided by [22]. There are 20 different objects available, with each image data matrix being of size $128^2 \times 72$. A single image of each object is shown in Fig. 5.

The algorithm outlined in Section V was used to compute the eigendecomposition of $\hat{X}$ corresponding to each of the image data matrices, with an energy recovery ratio threshold of 0.90. Table I summarizes the performance of the algorithm, showing k, p, and computation times. In addition, Table I also shows the data when the direct SVD of MATLAB is used to compute the first n singular values and vectors.

The difference between $\rho(\hat{X}, \hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_k)$ and $\rho(\hat{X}, \tilde{\hat{\mathbf{u}}}_1, \cdots, \tilde{\hat{\mathbf{u}}}_k)$ for each object is less than 2.22%, with an average of 0.85%, which reveals that $\{\tilde{\hat{\mathbf{u}}}_1, \cdots, \tilde{\hat{\mathbf{u}}}_k\}$ provides a very good approximate basis for the span of the first k eigenimages $\{\hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_k\}$. As discussed in Section IV, this is a consequence of the following empirical facts: 1) The power spectra of the right singular vector $\hat{\mathbf{v}}_i$ is approximately band-limited. 2) The frequency at which the power spectrum of $\hat{\mathbf{v}}_i$ achieves a maximum roughly increases with increasing i. (See Fig. 6.) Thus, the span of $\{\mathbf{h}_2, \cdots, \mathbf{h}_p\}$ effectively "covers" the span of $\{\hat{\mathbf{v}}_1, \cdots, \hat{\mathbf{v}}_k\}$.

Fig. 7 shows a comparison of the computation times for different SVD algorithms, as a function of the subspace dimension k. In almost all cases, our algorithm exceeded by far the performance of all of the algorithms that it was tested against. For instance, compared to the direct SVD which took about 40 s for each object, the median speedup factor with our algorithm was approximately three. Similar computational savings accrue when compared with other algorithms as well. Remarkably, it can also be seen that the rate of growth in computational effort with increasing subspace dimension k is the smallest for our algorithm.

We next turn to image-specific conclusions that can be inferred from Fig. 5 and Table I. While Object 1 requires a value of $k = 9$ to achieve an energy recovery ratio of 0.90, the value of k for Object 9 is four times as large. This illustrates that determining *a priori* the dimension of the subspace required to achieve a prespecified quality of reconstruction is difficult. Thus, other algorithms such as the updating SVD which do use a fixed value of k cannot be expected to perform uniformly well over all images. In contrast, our online estimate of k (given by p) can be seen to perform extremely well for most objects. Since the computational expense of our algorithm is directly related to p, this means that for most cases there is no "wasted" computation with our algorithm. In cases when the estimate p of k is poor, it can be seen that the corresponding object is rotationally symmetric; thus the associated pose-detection problem is ill-conditioned (see Object 17, and also 12 and 16).

TABLE II
Algorithm Performance on Video Sequences the Performance of our algorithm is Compared Against that of the Direct SVD Algorithms, for the 20 Sets of Video Sequences Shown in Fig. 8. In All Cases, the Eigenimages of X were Computed with a Desired Energy Recovery Ratio of 0.95. All Computations were Performed using MATLAB on a HP9000/C110 Workstation

	Dimension		Time (sec)	
Sequence no.	k	p	Our algorithm	Direct SVD
1	1	3	8.68	206.19
2	5	7	11.69	207.50
3	16	17	18.84	211.34
4	21	23	22.68	208.41
5	15	15	16.14	192.95
6	3	7	10.51	195.91
7	66	69	83.56	201.14
8	1	3	9.03	201.91
9	4	7	10.78	196.81
10	5	7	10.93	181.36
11	41	49	50.04	194.44
12	11	17	16.43	172.89
13	1	3	9.16	202.84
14	4	5	10.33	185.86
15	11	13	14.50	193.02
16	5	17	15.12	197.98
17	63	65	76.32	198.70
18	5	7	10.96	190.87
19	16	19	18.33	198.71
20	6	7	11.24	191.93

B. Video Sequences

We next consider the performance of our algorithm on images with more general correlation properties, in particular, when images are derived from a combination of 3-D rotation, translation and scaling. To this end, we consider the problem of eigendecomposition of images representing successive frames of arbitrary video sequences. Specifically, we consider six video sequences that have been used to evaluate MPEG standards [23], and an additional 14 video clips obtained from [24]. Each video sequence consists of 150 images, each of size 160×120. The first, middle and last frames from each set are shown in Fig. 8.

Our algorithm was used to calculate the partial SVD of X for each set, with an energy recovery ratio threshold of 0.95. Table II summarizes the performance of the algorithm, showing k, p, and the computation times. Compared to the direct SVD, the speedup factors with our algorithm are in the range of 2.4–23.8, depending on the value of p. The difference between $\rho(X, \mathbf{u}_1, \cdots, \mathbf{u}_k)$ and $\rho(X, \tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k)$ for each set was less than 0.32%, with an average of 0.12%, which again reveals that $\{\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_k\}$ provides a very good approximate basis for the span of the first k eigenimages $\{\mathbf{u}_1, \cdots, \mathbf{u}_k\}$. Fig. 9 shows that the power spectra of the right singular vectors for each video sequence are approximately band-limited, and that the dominant frequencies increase approximately linearly. These are precisely the properties that our algorithm is designed to take advantage of. The extent to which these properties hold (and consequently how well our algorithm works) is directly related to the rate of variation of the content across successive images.

We now illustrate the above general comments by discussing some specific cases. Note that in Table II, p is only slightly

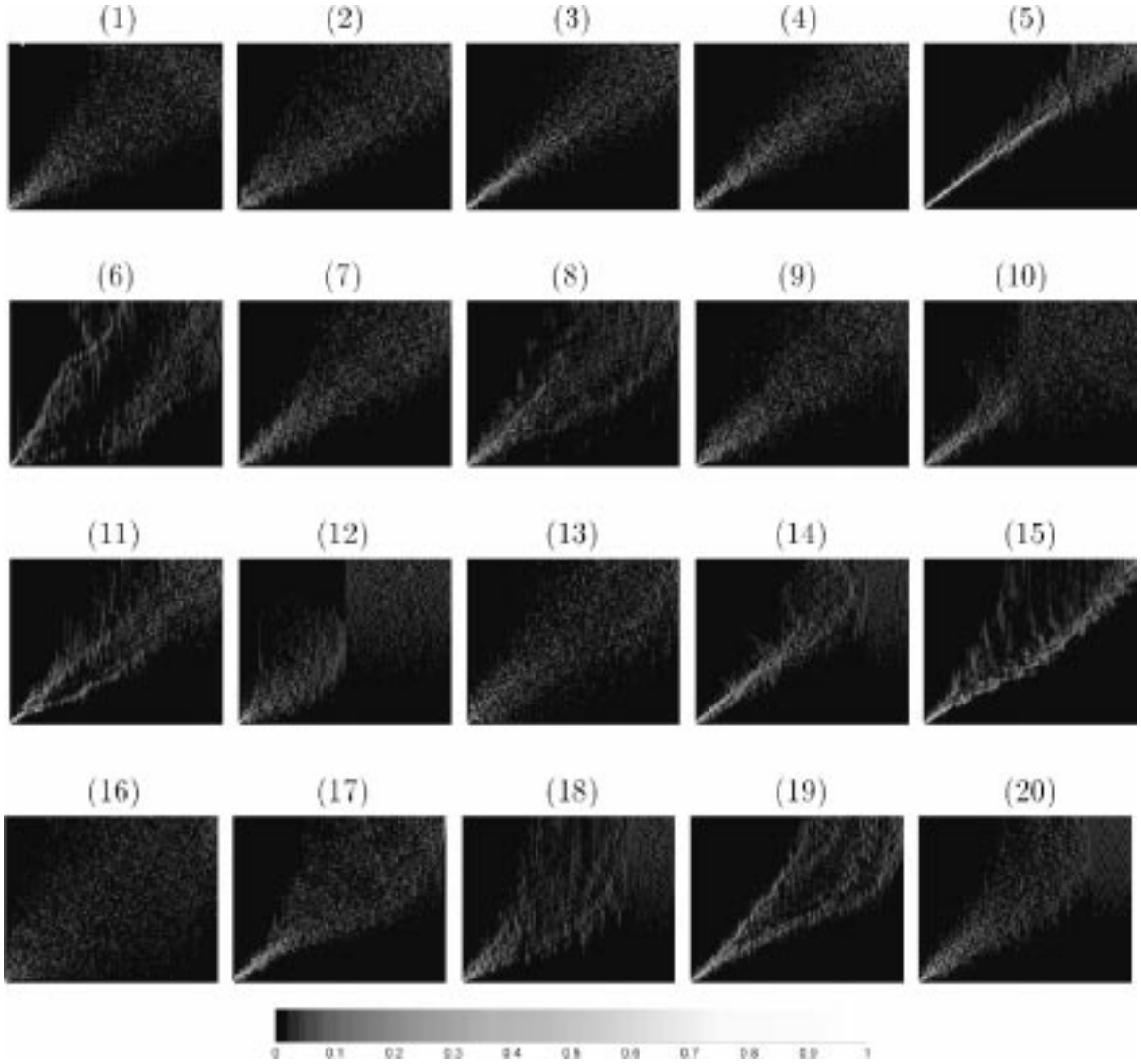

Fig. 9. For each of the video sequences in Fig. 8, the color-coded magnitude of the power spectra of the ordered right singular vectors are shown as a function of their index (along the x axis) and frequency (along the y axis). It can be seen that the power spectrum of each right singular vector stays within a narrow band, and that the dominant frequencies increase roughly linearly.

greater than k for most cases, implying that for a vast majority of cases, the low frequency harmonics indeed provide a good approximation to the actual dominant right singular vectors. Recalling that the computational expense of our algorithm is directly related to p, we note once again that the amount of computation is adapted to the difficulty of the problem. Moreover, it can be seen that for most problems, p is much smaller than n, the number of images; therefore, the computational savings that accrue with our algorithm are significant. When p is large, two scenarios are possible. The first scenario is that k can be large as well, indicating that the underlying eigenspace indeed has a high dimension. This is illustrated by Sequences 7 and 17, where the image content changes considerably between frames due to significant scaling effects. Thus, these two sequences represent high content, i.e., less correlation between images, and it is a strength of our algorithm that it can adapt the amount of computation to suit the difficulty underlying the eigendecomposition problem. In the second scenario, p is much larger than k, indicating that while the underlying eigenspace indeed has a small dimension, a large number of harmonics are required to span it. This is seen with Sequence 16 (and to a lesser extent Sequence 12). Sequence 16 presents rapid "morphing" between human faces. Therefore, even though the dimension of the eigenspace is effectively only 5, the first 17 harmonics are needed to span the corresponding eigen-subspace. This is confirmed by the power spectral plot of the actual right singular vectors corresponding to Sequence 16, in Fig. 9.

VII. Conclusions

We have illustrated a computationally efficient algorithm for computing the eigenspace decomposition of correlated images. In addition to its speed, the algorithm enjoys the advantage that the dimension of the subspace required to achieve a desired fidelity of representation is determined automatically; thus the amount of computation is "adapted" to meet accuracy requirements. Examples show that the algorithm performs very well on a range of test images composed of 3-D objects rotated about a single axis, and even arbitrary video sequences.

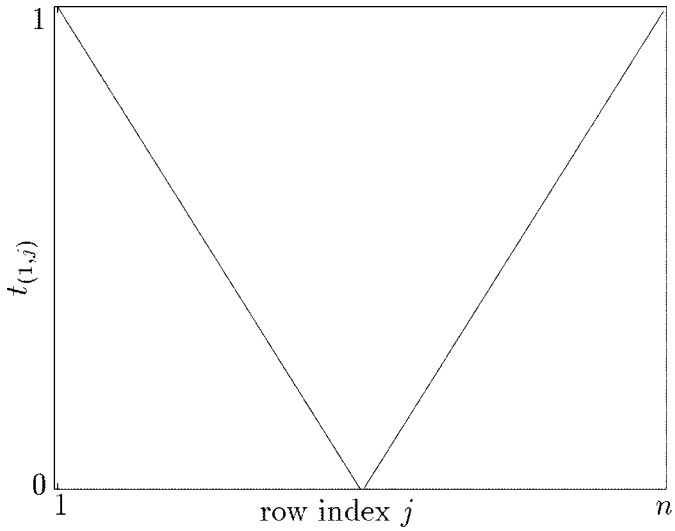

Fig. 10. Plot of the first row of T as a function of the column index. $T = X^T X$ is a symmetric Toeplitz matrix, where X is the image data matrix that results when a constant-intensity planar object translates back and forth at a constant speed.

APPENDIX A
PLANAR TRANSLATION: A SPECIAL CASE

We now present a special case where one of the terms in equation (15) is zero, with the other term therefore providing an unordered SVD of $T = X^T X$. Let n be even. With $t_{(i,j)}$ denoting the (i, j) element of T, let the first row of T be given by

$$t_{(1,j)} = \begin{cases} 1 - \dfrac{2j-2}{n}, & j = 1, 2, \cdots, \dfrac{n}{2} + 1, \\ \dfrac{2j-2}{n} - 1, & j = \dfrac{n}{2} + 2, \cdots, n. \end{cases}$$

Fig. 10 shows a plot of the first row of T as a function of the column index. This situation arises when an object translates back and forth at a constant speed.

It is easy to verify that in this case, when the (i, i) entries of R [recall (10)] are chosen to be one, the quantity $S^{(o)}$ in (15) is identically zero, i.e., the singular values of C corresponding to the odd harmonics are identically zero. Thus, (15) reduces to

$$T = U^{(e)} S^{(e)} V^{(e)^T} \tag{18}$$

i.e., we have an analytical expression for the eigendecomposition of T.

APPENDIX B
PLANAR SCALING

In this subsection, we briefly consider the case of "scaling," when an object in the image expands or contracts. We focus our attention on a special case where the image intensity is constant across the object, and when the image of the expanded object "covers" its nonexpanded image. Under these assumptions, with α_i denoting the difference in area of the object between the $(i-1)$th image and the ith image, it is easy to verify that the corresponding correlation matrix can be written as

$$\begin{aligned} X^T X &= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1+\alpha_1 & \cdots & 1+\alpha_1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1+\alpha_1 & \cdots & 1+\sum_{i=1}^{n-1} \alpha_i \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \alpha_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \alpha_{n-1} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix}. \end{aligned} \tag{19}$$

In the special case when $\alpha_1 = \alpha_2 = \cdots = \alpha_{n-1} = 1$, the singular vectors of $X^T X$ are singular vectors of a Toeplitz matrix. Conclusions similar to the one in Section III-B can be inferred here as well.

REFERENCES

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.

[2] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer.*, vol. 4, pp. 519–524, Mar. 1987.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, Mar. 1991.

[4] H. Murase and R. Sakai, "Moving object recognition in eigenspace representation: Gait analysis and lip reading," *Pattern Recognit. Lett.*, vol. 17, pp. 155–162, Feb. 1996.

[5] G. Chiou and J.-N. Hwang, "Lipreading from color video," *IEEE Trans. Image Processing*, vol. 6, pp. 1192–1195, Aug. 1997.

[6] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 1219–1227, Dec. 1994.

[7] ——, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vis.*, vol. 14, no. 1, pp. 5–24, 1995.

[8] ——, "Detection of 3-D objects in cluttered scenes using hierarchical eigenspace," *Pattern Recognit. Lett.*, vol. 18, no. 4, pp. 375–384, 1997.

[9] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace method for robot vision," *IEEE Trans. Robot. Automat.*, vol. 12, Oct. 1996.

[10] S. Shlien, "A method for computing the partial singular value decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 4, pp. 671–676, Nov. 1982.

[11] R. Haimi-Cohen and A. Cohen, "Gradient-type algorithms for partial singular value decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 137–142, Jan. 1987.

[12] X. Yang, T. K. Sarkar, and E. Arvas, "A survey of conjugate gradient algorithms for solution of extreme eigen-problems for a symmetric matrix," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1550–1556, Oct. 1989.

[13] C. R. Vogel and J. G. Wade, "Iterative SVD-based methods for ill-posed problems," *SIAM J. Sci. Comput.*, vol. 15, pp. 736–754, May 1994.

[14] H. Murakami and V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 4, pp. 511–515, Sept. 1982.

[15] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler, and H. Zhang, "An eigenspace update algorithm for image analysis," *Comput. Vis. Graphics Image Process.: Graph. Models Image Process.*, vol. 59, pp. 321–332, Sept. 1997.

[16] H. Murase and M. Lindenbaum, "Partial eigenvalue decomposition of large images using the spatial temporal adaptive method," *IEEE Trans. Image Processing*, vol. 4, pp. 620–629, May 1995.

[17] M. Uenohara and T. Kanade, "Optimal approximation of uniformly rotated images: Relationship between Karhunen–Loeve expansion and discrete cosine transform," *IEEE Trans. Image Processing*, vol. 7, pp. 116–119, Jan. 1998.
[18] P. J. Davis, *Circulant Matrices*. New York: Wiley, 1979.
[19] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation," *SIAM J. Numer. Anal.*, vol. 7, pp. 1–46, Mar. 1970.
[20] T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed. New York: Springer-Verlag, 1984.
[21] C.-Y. Chang, "Eigenspace Analysis of Correlated Images," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1999.
[22] S. Nene, S. Nayar, and H. Murase. (1996) Columbia Object Image Library (COIL-20). Dept. Comput. Sci., Columbia Univ., New York. [Online]http://www.cs.columbia.edu/CAVE/coil-20.html
[23] F. Periera and T. Alpert, "MPEG-4 video subjective test procedures and results," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 32–51, Jan. 1997.
[24] [Online] Available: http://www.dazzlemm.com.sg/html/samples4.html; http://www.sfdj.com/videos/movie.html; and http://www.vol.it.

Chu-Yin Chang received the B.S. degree in mechanical engineering from National Central University, Chung-Li, Taiwan, R.O.C., in 1988. the M.S. degree in electrical engineering from the University of California, Davis, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, in 1999.

He is currently a Machine Vision Systems Engineer with Semiconductor Technologies and Instruments, Inc., Plano, TX. His research interests include computer vision, computer graphics, image processing, and robotics.

Anthony A. Maciejewski (S'82–M'87–SM'00) received the B.S.E.E., M.S., and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1982, 1984, and 1987 respectively.

In 1985–1986, he was an American Electronics Association Japan Research Fellow with the Hitachi Central Research Laboratory, Tokyo, Japan. He is currently a Professor of electrical and computer engineering at Purdue University, West Lafayette, IN. His primary research interests center on robotics and imaging.

Venkataramanan Balakrishnan (M'94) received the B.Tech degree in electronics and communication from the Indian Institute of Technology, Madras, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA.

He is an Associate Professor of electrical and computer engineering at Purdue University, West Lafayette, IN. He joined Purdue University in 1994 as an Assistant Professor, following post-doctoral positions with Stanford University, California Institute of Technology (Caltech), Pasadena, and the Institute for Systems Research, University of Maryland, College Park. He is a co-author of the monograph *Linear Matrix Inequalities in System and Control Theory* (Philadelphia, PA: SIAM, 1994). His primary research interests are in applying convex optimization to problems in systems and control.

Dr. Balakrishnan received the President of India Gold Medal from the Indian Institute of Technology in 1985, the Young Investigator Award from the Office of Naval Research in 1997, and the Ruth and Joel Spira Outstanding Teacher Award from the School of Electrical and Computer Engineering at Purdue University in 1998.