Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Analysis of Eigendecomposition for Sets of Correlated Images at Different Resolutions

Kishor Saitwal and Anthony A. Maciejewski
Dept. of Electrical and Computer Eng.
Colorado State University
Fort Collins, CO 80523-1373, USA
Email: {Kishor.Saitwal, aam}@colostate.edu

Rodney G. Roberts
Dept. of Electrical and Computer Eng.
Florida A & M - Florida State University
Tallahassee, FL 32310-6046, USA
Email: rroberts@eng.fsu.edu

***Abstract*— Eigendecomposition is a common technique that is performed on sets of correlated images in a number of computer vision and robotics applications. Unfortunately, the computation of an eigendecomposition can become prohibitively expensive when dealing with very high resolution images. While reducing the resolution of the images will reduce the computational expense, it is not known how this will affect the quality of the resulting eigendecomposition. The work presented here gives the theoretical background for quantifying the effects of varying the resolution of images on the eigendecomposition that is computed from those images. A computationally efficient algorithm for this eigendecomposition is proposed using derived analytical expressions. Examples show that this algorithm performs very well on arbitrary video sequences.**[1]

I. Introduction

Eigendecomposition-based techniques play an important role in numerous image processing and computer vision applications. The advantage of these techniques, also referred to as subspace methods, is that they are purely appearance based and that they require few online computations. Variously referred to as eigenspace methods, singular value decomposition (SVD) methods, principal component analysis methods, and Karhunun-Loeve transformation methods [1], they have been used extensively in a variety of applications such as face characterization [2], [3] and recognition [4]-[8], lip-reading [9], [10], object recognition [11]-[14], pose detection [15], [16], visual tracking [17], [18], and inspection [19]-[22]. 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 [23]. 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.

The resolution of the given correlated images, in terms of the number of pixels, is one of the factors that greatly affects the amount of offline calculation required to compute an eigendecomposition. In particular, many common algorithms that compute the complete SVD of a general matrix require on the order of mn^2 flops, where m is the total number of pixels in a single image and n is the number of images. Most users of eigendecomposition techniques would like to use as large a resolution as is available for the original images in order to maintain as much information as possible; however, this frequently results in an impractical computational burden. Thus users are typically forced to downsample their images to a lower resolution using a "rule of thumb" or some *ad hoc* criterion to obtain a manageable level of computation. The purpose of the work described here is to develop a theoretical background that will quantify the tradeoff between the resolution of correlated images and the "quality" of their resulting eigendecomposition, in terms of measures that are relevant to the user's motivation for preforming an eigendecomposition.

The paper is organized as follows. In Section II, we explain the fundamentals of applying eigendecomposition to related images. We develop a mathematical background in Section III for quantifying the amount of error introduced into an eigendecomposition as a function of resolution. This information is used to develop a fast SVD algorithm, outlined in Section IV, to quickly compute the desired portion of the eigendecomposition based on a user-specified measure of accuracy. In Section V, we evaluate the performance of our algorithm on a set of arbitrary video sequences. Lastly, we give the concluding remarks in Section VI.

II. Preliminaries

In this work, a grey-scale image is described by 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 $\mathcal{X} \in [0,1]^{h \times v}$. Because sets of related images are considered in this paper, the *image vector* $\mathbf{x}$ of length $m = h \times v$ is obtained by "row-scanning" an image into a column vector, i.e., $\mathbf{x} = \mathrm{vec}(\mathcal{X}^T)$. 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 $X = [\mathbf{x}_1 \cdots \mathbf{x}_n]$, where typically $m \gg n$ with fixed n.

The SVD of X is given by

$$X = U \Sigma V^T, \tag{1}$$

[1]This work was supported by the National Imagery and Mapping Agency under contract no. NMA201-00-1-1003 and through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

where $U \in \Re^{m\times m}$ and $V \in \Re^{n\times n}$ are orthogonal, and $\Sigma = [\Sigma_d \ \mathbf{0}]^T \in \Re^{m\times n}$ where Σ_d = diag$(\sigma_1, \cdots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ and $\mathbf{0}$ is an n by $m-n$ zero matrix. The SVD of X plays a central role in several important imaging applications such as image compression and pattern recognition. The columns of U, denoted $\hat{\mathbf{u}}_i$, $i = 1, \cdots, m$, are referred to as the left singular vectors or eigenimages of X, while the columns of V, denoted $\hat{\mathbf{v}}_i$, $i = 1, \cdots, n$, are referred to as the right singular vectors of X.

In practice, the singular vectors $\hat{\mathbf{u}}_i$ are not known or computed exactly, and instead estimates $\mathbf{e}_1, \cdots, \mathbf{e}_k$ which form a k-dimensional basis are used. For quantifying the accuracy of a practical implementation of subspace methods, one of the measures we will use is the "energy recovery ratio" [23], denoted ρ, and defined as

$$\rho(X, \mathbf{e}_1, \cdots, \mathbf{e}_k) = \frac{\sum_{i=1}^{k} \|\mathbf{e}_i^T X\|_2^2}{\|X\|_F^2}, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. To determine the degree to which the first k_2 approximated eigenimages, i.e., the $\tilde{\hat{\mathbf{u}}}_i$'s, span the subspace spanned by the first k_1 true eigenimages, i.e., the $\hat{\mathbf{u}}_i$'s, we will also use the subspace criterion, s, which is given by

$$s = \sqrt{\frac{1}{k_1} \sum_{i=1}^{k_2} \sum_{j=1}^{k_1} (\tilde{\hat{\mathbf{u}}}_i \cdot \hat{\mathbf{u}}_j)^2}, \tag{3}$$

which is 1 if the entire subspace is spanned.

III. Analysis of SVD at different resolutions

This section gives the mathematical background that explains how to approximate high-resolution eigenimages using the SVD computed from low-resolution images, with the assumption that the number of columns in the image data matrices at different resolutions remains the same.

A. A Special Case with a Closed Form SVD

We will start with an image data matrix that has a closed form solution for the SVD at both high and low resolutions. This closed form solution along with its properties can then be used as a basis for the further analysis of arbitrary image data matrix.

Consider two images with m pixels that have been row-scanned and normalized to unit norm. The $m \times 2$ high-resolution image data matrix, X_h, is given by

$$X_h = \begin{bmatrix} \vdots & \vdots \\ \hat{\mathbf{x}}_{h1} & \hat{\mathbf{x}}_{h2} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{m1} & x_{m2} \end{bmatrix}, \tag{4}$$

where the $\hat{}$ notation indicates that the corresponding vectors are normalized to unit norm. The pixels in $\hat{\mathbf{x}}_{h1}$ and $\hat{\mathbf{x}}_{h2}$ are lexicographically ordered so that a pixel in the low-resolution image vectors, $\mathbf{x}_{l1}$ and $\mathbf{x}_{l2}$, can be obtained by box-filtering the consecutive pixels in $\hat{\mathbf{x}}_{h1}$ and $\hat{\mathbf{x}}_{h2}$, respectively. Thus, with the integer reduction factor r, the low-resolution image data matrix X_l is given by

$$X_l = \frac{1}{r} \begin{bmatrix} x_{11} + \cdots + x_{r1} & x_{12} + \cdots + x_{r2} \\ \vdots & \vdots \\ x_{d1} + \cdots + x_{m1} & x_{d2} + \cdots + x_{m2} \end{bmatrix}, \tag{5}$$

where $d = m - r + 1$. The critical step in calculating the SVD of X is to determine an orthogonal matrix V that will orthogonalize the columns of X. This matrix can be formed as a Givens rotation that is designed to orthogonalize two columns and results in the following V and U matrices for X_h:

$$\begin{aligned} V_h &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -\text{sgn}(\hat{\mathbf{x}}_{h1}^T \hat{\mathbf{x}}_{h2}) \\ \text{sgn}(\hat{\mathbf{x}}_{h1}^T \hat{\mathbf{x}}_{h2}) & 1 \end{bmatrix}, \\ U_h &= \begin{bmatrix} \vdots & \vdots \\ \frac{\hat{\mathbf{x}}_{h1} \pm \hat{\mathbf{x}}_{h2}}{\|\hat{\mathbf{x}}_{h1} \pm \hat{\mathbf{x}}_{h2}\|} & \frac{\mp\hat{\mathbf{x}}_{h1} + \hat{\mathbf{x}}_{h2}}{\|\mp\hat{\mathbf{x}}_{h1} + \hat{\mathbf{x}}_{h2}\|} \\ \vdots & \vdots \end{bmatrix}, \end{aligned} \tag{6}$$

where the subscript h denotes that these matrices correspond to X_h.[2] The upper and lower signs in $\pm$ and $\mp$ notations in U_h matrix correspond to the positive and the negative dot product between the high-resolution image vectors, respectively. Note that such closed form solutions can be obtained for these matrices because the image vectors in X_h have equal norms. The column vectors in X_l, however, do not necessarily have equal norms, hence U and V matrices for X_l do not have the same closed form solution. To find these matrices, we have to make use of the formulas for the SVD algorithm that relies on Givens rotations [24]. These formulas are based on the quantities,

$$y_l = \mathbf{x}_{l1}^T \mathbf{x}_{l2}, \tag{7}$$

$$z_l = \mathbf{x}_{l1}^T \mathbf{x}_{l1} - \mathbf{x}_{l2}^T \mathbf{x}_{l2}, \tag{8}$$

$$w_l = \sqrt{4y_l^2 + z_l^2}, \tag{9}$$

so that

$$\cos\theta = \sqrt{\frac{w_l + z_l}{2w_l}}, \ \ \sin\theta = \frac{y_l}{w_l \cos\theta} \tag{10}$$

if $z_l \geq 0$ and

$$\sin\theta = \text{sgn}(y_l)\sqrt{\frac{w_l - z_l}{2w_l}}, \ \ \cos\theta = \frac{y_l}{w_l \sin\theta} \tag{11}$$

if $z_l < 0$. Then the V and U matrices for X_l can be given by

$$\begin{aligned} V_l &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \\ U_l &= \begin{bmatrix} \vdots & \vdots \\ \frac{\cos\theta \mathbf{x}_{l1} + \sin\theta \mathbf{x}_{l2}}{\|\cos\theta \mathbf{x}_{l1} + \sin\theta \mathbf{x}_{l2}\|} & \frac{-\sin\theta \mathbf{x}_{l1} + \cos\theta \mathbf{x}_{l2}}{\|-\sin\theta \mathbf{x}_{l1} + \cos\theta \mathbf{x}_{l2}\|} \\ \vdots & \vdots \end{bmatrix} \end{aligned} \tag{12}$$

[2]Norms in all the equations in this paper always represent the 2-norm unless otherwise stated.

where the subscript l denotes that these matrices correspond to X_l. The left and right singular vectors at high and low resolutions can be compared against each other by using the method suggested in [25]. However, the dot product of the difference between the corresponding vectors indicate that the bound of this error is in the range of [0, 2] for both the right singular vectors and the interpolated eigenimages. Hence we need to study the approximations of the high-resolution eigenimages using the low-resolution SVD in more detail, which is the topic of the next subsection.

B. Limitations of Interpolated Low-Resolution Eigenimages

In our previous work [25], we have shown how the interpolated low-resolution eigenimages can be used as an approximation of their high-resolution counterparts. The results were acceptable as long as the reduction in resolution is not too great. Here we show why this is true using a simple example.

Consider X_h with $m = 4$. The first (unnormalized) eigenimage of X_h is given by

$$\mathbf{u}_{h1} = \frac{1}{2}\begin{bmatrix} x_{11}+x_{12} \\ x_{21}+x_{22} \\ x_{31}+x_{32} \\ x_{41}+x_{42} \end{bmatrix}, \tag{13}$$

where we have assumed the positive dot product between the columns of X_h. For $r = 2$, the matrix V_l for the corresponding X_l can be generated using the quantities in (7), (8), and (9). Then the first (unnormalized) eigenimage of X_l is given by

$$\begin{aligned} \mathbf{u}_{l1} &= X_l \mathbf{v}_{l1} \\ &= \frac{1}{2}\left(\cos\theta \begin{bmatrix} x_{11}+x_{21} \\ x_{31}+x_{41} \end{bmatrix} + \sin\theta \begin{bmatrix} x_{12}+x_{22} \\ x_{32}+x_{42} \end{bmatrix}\right). \end{aligned}$$

The linear interpolation[3] of $\mathbf{u}_{l1}$ to the size of $\mathbf{u}_{h1}$ gives

$$\tilde{\mathbf{u}}_{h1} = L\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 5 & -1 \\ 3 & 1 \\ 1 & 3 \\ -1 & 5 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \tag{14}$$

where L gives the linear interpolation model, $r_1 = 0.5(\cos\theta(x_{11}+x_{21}) + \sin\theta(x_{12}+x_{22}))$, and $r_2 = 0.5(\cos\theta(x_{31}+x_{41})+\sin\theta(x_{32}+x_{42}))$. Note that the columns of the following matrix form an orthonormal basis for the space perpendicular to the column space of L:

$$N_L = \begin{bmatrix} 0.5 & 0.2236 \\ -0.5 & -0.6708 \\ -0.5 & 0.6708 \\ 0.5 & -0.2236 \end{bmatrix}. \tag{15}$$

Now consider the family of all 4×2 matrix X_h with the following properties:

1) The column space of X_h is perpendicular to the column space of L.
2) The columns of X_h have unit norm.
3) The angle between the columns of X_h is α.

[3]Bicubic interpolation is used in [25] for better accuracy, while linear interpolation is used here for mathematical simplicity.

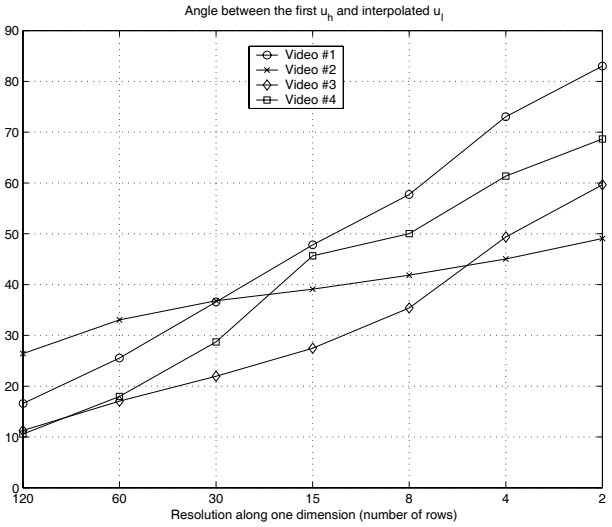

Fig. 1. This figure shows the plots for the four video sequences used in [23], viz., 5, 6, 7, and 17 (referred here as 1, 2, 3, and 4 respectively). The average is subtracted from the original image data matrices and the image data matrices at different resolution are formed after reducing the original images from $m = 240 \times 352$ to the lower resolutions of $120 \times 176, 60 \times 88, 30 \times 44, 15 \times 22, 8 \times 12, 4 \times 6$, and 2×3. The SVD is calculated at all the resolutions and the angle (in degrees) between the first true eigenimages at high resolution and the first interpolated low-resolution eigenimages to the size of the high-resolution eigenimages is plotted in this figure for all four video sequences.

This family, denoted by $X_h(\phi)$, can be parameterized by ϕ in the following way:

$$X_h(\phi) = N_L \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}\begin{bmatrix} 1 & \cos\alpha \\ 0 & \pm\sin\alpha \end{bmatrix}, \tag{16}$$

where ϕ is any angle. For a particular case of X_h, when $\phi = 60^{\circ}$ and $\alpha = 10^{\circ}$, we can see that,

$$\hat{\mathbf{u}}_{hi}^T \tilde{\hat{\mathbf{u}}}_{hj} = 0, \quad \text{for } i, j = 1, 2. \tag{17}$$

Thus the column space of the approximated eigenimages is orthogonal to the column space of the true eigenimages, giving the worst possible approximation even for simple X_h. Fig. 1 also shows that the interpolated eigenimages give a very bad approximation of the true eigenimages at high resolution, when the image data matrices consist of images at very low resolution.

If we reconsider (16) with $\phi = 60^{\circ}$ and $\alpha = 10^{\circ}$, we can observe that the high-resolution right singular vectors and low-resolution right singular vectors are only 2.33° apart. Hence the approximation of the high-resolution eigenimages using the low-resolution right singular vectors is a much more promising approach.

C. Using Low-Resolution Right Singular Vectors

Note that V_h and V_l represent V matrices at two different resolutions, but for the same images. Also both these matrices are of the same size. Hence we can use V_l to approximate U_h

(denoted $\tilde{U}_h$) and Σ_h (denoted $\tilde{\Sigma}_h$), i.e.,

$$\begin{aligned}\tilde{U}_h\tilde{\Sigma}_h &= X_h V_l \\ &= \begin{bmatrix} \vdots & \vdots \\ \hat{\mathbf{x}}_{h1} & \hat{\mathbf{x}}_{h2} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (18)\end{aligned}$$

If we denote the first and the second column of $\tilde{U}_h\tilde{\Sigma}_h$ by $\mathbf{a}_1$ and $\mathbf{a}_2$, respectively, the first column after Gram-Schmidt orthogonalization will be

$$\tilde{\hat{\mathbf{u}}}_{h1} = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} = \frac{\cos\theta\hat{\mathbf{x}}_{h1} + \sin\theta\hat{\mathbf{x}}_{h2}}{\sqrt{1+2y_h\sin\theta\cos\theta}}, \quad (19)$$

where $y_h = \hat{\mathbf{x}}_{h1}^T\hat{\mathbf{x}}_{h2}$. Here we start with column $\mathbf{a}_1$, because it is likely to be of larger norm than $\mathbf{a}_2$.

To check as to how much the approximated eigenimages differ from the correct ones, consider the difference[4],

$$\begin{aligned}\Delta\mathbf{u}_{h1} &= \hat{\mathbf{u}}_{h1} - \tilde{\hat{\mathbf{u}}}_{h1} \\ &= \frac{\hat{\mathbf{x}}_{h1}+\hat{\mathbf{x}}_{h2}}{\sqrt{2(1+y_h)}} - \frac{\cos\theta\hat{\mathbf{x}}_{h1}+\sin\theta\hat{\mathbf{x}}_{h2}}{\sqrt{1+2y_h\sin\theta\cos\theta}}. \quad (20)\end{aligned}$$

The square of the norm of $\Delta\mathbf{u}_{h1}$ is given by

$$\|\Delta\mathbf{u}_{h1}\|^2 = 2 - \sqrt{2(1+y_h)}\frac{\cos\theta+\sin\theta}{\sqrt{1+2y_h\sin\theta\cos\theta}}, \quad (21)$$

whose extremal θ values (denoted θ^*) will give the best case and the worst case conditions on X_h for the approximation of $\hat{\mathbf{u}}_{h1}$. The problem of finding the θ^* values of (21) is equivalent to finding the θ^* values of

$$f(\theta) = \mathrm{sgn}(\cos\theta+\sin\theta)\sqrt{\frac{1+\sin 2\theta}{1+y_h\sin 2\theta}}. \quad (22)$$

Differentiating (22) with respect to θ (for $\cos\theta+\sin\theta \neq 0$, i.e., for $\theta \neq \frac{3\pi}{4}+n\pi$) gives

$$f'(\theta) = \frac{\mathrm{sgn}(\cos\theta+\sin\theta)(1-y_h)\cos 2\theta}{\sqrt{1+\sin 2\theta}(1+y_h\sin 2\theta)^{3/2}}. \quad (23)$$

We thus have the following candidate θ^* values:

1) $\cos 2\theta^* = 0 \Rightarrow 2\theta^* = \frac{(2n+1)\pi}{2} \Rightarrow \theta^* = \frac{(2n+1)\pi}{4}$
2) $\sin 2\theta^* = -1 \Rightarrow 2\theta^* = \frac{(4n+3)\pi}{2} \Rightarrow \theta^* = \frac{(4n+3)\pi}{4}$
3) $\cos\theta^* + \sin\theta^* = 0 \Rightarrow \theta^* = \frac{(4n+3)\pi}{4}$

Hence we can conclude that the candidate θ^* values are $\theta^* = \frac{(2n+1)\pi}{4}$ (odd multiples of 45°). With all non-negative entries in X_h, the θ values will always be bounded by 0 and $\frac{\pi}{2}$. Hence the only θ^* value in this case is 45°, which gives the best-case scenario as shown in Table I. We must also check the boundary condition θ values (0 and $\frac{\pi}{2}$), which give the worst-case scenarios (refer to Table I) for degree of error depending on the value of y_h, with the results being worst when $y_h = 0$. However, note that, as $y_h \to 0$, the problem of finding the high-resolution eigenimages becomes ill-defined. For image data matrices with only two images, the θ^* values for the approximation of the correct second eigenimage, $\tilde{\hat{\mathbf{u}}}_{h2}$ remains the same.

[4]The components of X_h are considered to be all non-negative here. The analysis of the case when X_h contains negative components will be similar to the one presented in this section.

TABLE I
$\hat{\mathbf{u}}_{h1}\cdot\tilde{\hat{\mathbf{u}}}_{h1}$ AND $\|\Delta\mathbf{u}_{h1}\|$ FOR DIFFERENT θ^* VALUES

Case #	θ^*	$\hat{\mathbf{u}}_{h1}\cdot\tilde{\hat{\mathbf{u}}}_{h1}$	$\|\Delta\mathbf{u}_{h1}\|$	Comments
1	0	$\sqrt{\frac{1+y_h}{2}}$	$2-\sqrt{2(1+y_h)}$	worst case
2	$\frac{\pi}{4}$	1	0	best case ($\mathbf{u}$'s line up)
3	$\frac{\pi}{2}$	$\sqrt{\frac{1+y_h}{2}}$	$2-\sqrt{2(1+y_h)}$	worst case

It is instructive to consider the worst case θ^* values in Table I. When $\theta^* = 0$,

$$U_l\Sigma_l = \begin{bmatrix} \vdots & \vdots \\ \hat{\mathbf{u}}_{l1} & \hat{\mathbf{u}}_{l2} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \|\mathbf{u}_{l1}\| & 0 \\ 0 & \|\mathbf{u}_{l2}\| \end{bmatrix}, \quad (24)$$

where $\mathbf{u}_{l1} = \mathbf{x}_{l1}$ and $\mathbf{u}_{l2} = \mathbf{x}_{l2}$. By definition,

$$\hat{\mathbf{u}}_{l1}\cdot\hat{\mathbf{u}}_{l2} = 0 \;\Rightarrow\; \frac{\mathbf{x}_{l1}^T\mathbf{x}_{l2}}{\|\mathbf{x}_{l1}\|\cdot\|\mathbf{x}_{l2}\|} = 0 \;\Rightarrow\; \mathbf{x}_{l1}^T\mathbf{x}_{l2} = 0. \quad (25)$$

When $\theta^* = \pi/2$, we obtain the same condition as in (25). This condition indicates that we will get the worst-case scenarios only when there is no overlap between the reduced image vectors at low resolution. To check for the conditions on the corresponding high-resolution image vectors, let $m = 4$ and $r = 2$. Then, $\mathbf{x}_{l1}^T\mathbf{x}_{l2} = 0 \Rightarrow$

$$\hat{\mathbf{x}}_{h1}^T\hat{\mathbf{x}}_{h2} + x_{11}x_{22} + x_{21}x_{12} + x_{31}x_{42} + x_{41}x_{32} = 0. \quad (26)$$

With all the components of X_h non-negative, all the individual products on the LHS must be 0 to satisfy the condition in (26), which is highly unlikely for most images.

While the above analysis cannot be easily extended to arbitrary X_h, one can experimentally evaluate the quality of the eigenimage approximation. The approximation of the i^{th} eigenimage of X_h (denoted $\tilde{\mathbf{u}}_{hi}$) can be given by $\tilde{\mathbf{u}}_{hi} = X_h\cdot\hat{\mathbf{v}}_{li}$, where $\hat{\mathbf{v}}_{li}$ denotes the i^{th} right singular vector for X_l with low-resolution images. These approximated eigenimages can be decomposed to obtain the orthonormal basis using the QR decomposition of $U = QR$, giving $\hat{U} = Q$, where Q is an orthogonal matrix whose columns give the orthonormal basis $\tilde{\hat{\mathbf{u}}}_{hi}$'s for $\tilde{\mathbf{u}}_{hi}$'s and R is an upper triangular matrix. The norms of the $\tilde{\mathbf{u}}_{hi}$'s can be used as an approximation of the corresponding singular values of X_h.

Fig. 2 shows that using the low-resolution right singular vectors give a much better approximation of the high-resolution eigenimages than those obtained using the interpolated low-resolution eigenimages (shown in Fig. 1). This motivates a computationally efficient technique for the fast eigendecomposition of a set of correlated images that takes advantage of the similarity between the right singular vectors at different resolutions. Our proposed algorithm is presented in the next subsection.

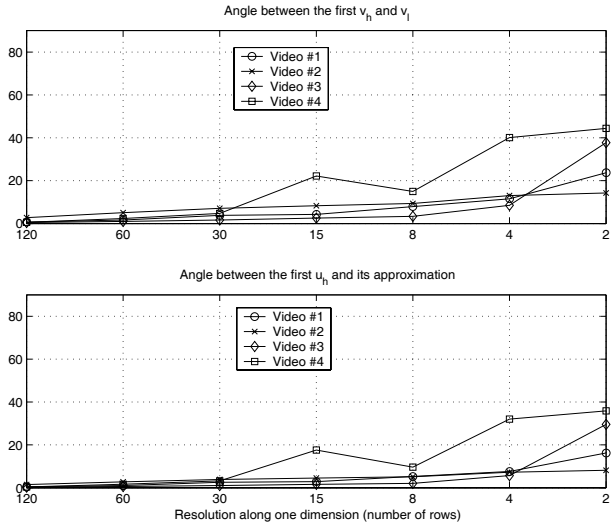

Fig. 2. This figure shows the plots for same four video sequences used in Fig. 1 with same image reduction procedure. The first plot shows the angle (in degrees) between the first right singular vectors at high resolution and at the lower resolutions, while the second plot shows the angles between the first true eigenimage and its approximation using our proposed method for all four video sequences at the lower resolutions.

IV. Fast Eigendecomposition Algorithm

Our objective is to determine the first k left singular vectors of X. Chang et al. [23] proposed a computationally efficient algorithm for the eigendecomposition of correlated images. We will use this algorithm as a benchmark for the accuracy and the computational efficiency while finding the first k eigenimages of X, because this is the fastest algorithm known to us. The approach in [23] was motivated by the fact that for a planar rotation of a 2-D image, analytical expressions can be given for the eigendecomposition, based on the theory of circulant matrices. These analytical expressions turned out to be good approximations of the eigendecomposition of arbitrary video sequences with better computational efficiency. Our algorithm uses this algorithm to reduce the images in the temporal dimension and then uses the theoretical background given in Section III to reduce the images in the spatial dimension. The following steps summarize the proposed algorithm:

1) Generate the Fourier matrix, F, and its *real* part, H, for X and determine the smallest number p such that
$$\rho(X^T, \mathbf{h}_1, \cdots, \mathbf{h}_p) = \frac{\sum_{i=1}^{p} \|X\mathbf{h}_i\|^2}{\|X\|_F^2} > \mu, \qquad (27)$$
where μ is the user-specified reconstruction ratio.
2) Reduce XH_p spatially to X_qH_p such that each of its columns has q pixels with $q > p$. (The matrix XH_p is readily available after Step 1.)
3) Compute the SVD of $X_qH_p = \hat{U}_q\hat{\Sigma}_qV_q^T = \sum_{i=1}^{p} \tilde{\sigma}_i\tilde{\mathbf{u}}_i\tilde{\mathbf{v}}_i^T$.
4) Find the product $(XH_p)V_q$ and apply the QR decomposition to obtain the approximation of $\hat{\mathbf{u}}_i$'s.
5) Return $\tilde{\hat{\mathbf{u}}}_1, \cdots, \tilde{\hat{\mathbf{u}}}_k$ such that $\rho(X, \tilde{\hat{\mathbf{u}}}_1, \cdots, \tilde{\hat{\mathbf{u}}}_k) > \mu$.

We now briefly analyze the computational expense of our algorithm. The cost incurred in Step 1, i.e., estimation of the smallest number p, requires $O(mnp)$ flops. Step 2 involves the reduction of the columns of XH_p to get X_qH_p, which requires $O(mp)$ flops. In Step 3, the cost of computing the SVD of the $q \times p$ matrix X_qH_p requires $O(qp^2)$ flops. In Step 4, multiplication of XH_p with V_q requires $O(mp^2)$ flops and the QR decomposition of $(XH_p)V_q$ requires $O(2mp^2 - \frac{2}{3}p^3)$ flops. Finally, in Step 5, determination of the dimension k requires $O(mnk)$ flops. If $p \ll n$, then the total computation required is $O(mnp)$.

V. Experimental Results

We consider the problem of eigendecomposition of images representing successive frames of arbitrary video sequences. Specifically, we consider eight video sequences that are used in [23], viz., 5, 6, 7, 17, 9, 8, 15, and 20 (referred to here as videos 1 through 8, respectively). Images in the first four sequences and the last four sequences have resolution of 240×352 and 240×320, respectively.

Our algorithm was used to calculate the partial SVD of X for each set, with an energy recovery ratio threshold of 0.95. The matrix XH_p was reduced so that its columns contained the row-scanned images of size 8×12, thus fixing the value of q to 96. Table II summarizes the performance of the algorithm, showing k_1, k_2, p, and the computation times. Compared to the direct SVD, the speedup factors with our algorithm are in the range of $0.92 - 47.06$, depending on the value of p. The difference between $\rho(X, \hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_{k_1})$ and $\rho(X, \tilde{\hat{\mathbf{u}}}_1, \cdots, \tilde{\hat{\mathbf{u}}}_{k_2})$ for each set was less than 0.30%, with an average of 0.13%, which reveals that $\{\tilde{\hat{\mathbf{u}}}_1, \cdots, \tilde{\hat{\mathbf{u}}}_{k_2}\}$ provides a very good approximate basis for the first k_1 eigenimages $\{\hat{\mathbf{u}}_1, \cdots, \hat{\mathbf{u}}_{k_1}\}$. Compared to Chang's algorithm, the value of k_2 remains the same except for the third sequence, where one more eigenimage is required to satisfy the given energy recovery ratio threshold. At the same time, because the SVD computation is performed on a much smaller matrix X_qH_p, our algorithm is computationally more efficient, which is evident from the table entries.

The resultant eigenimages for all the video sequences were also compared using the difference measures defined in [25]. Fig. 3 shows the general behavior for most of the video sequences when comparing their SVD's. The plots for the singular values show that the relative error between the true and the approximated singular values is almost negligible. In particular, the relative error between these values varies from 0% to 11.70% indicating a good approximation of the true singular values at high resolution. The maximum principal angles between the subspaces containing the true and the approximated eigenimages show that the maximum principal angle is 43.21° for $k_1 = k_2 = 15$. The measure s between these two subspaces exhibits similar behavior to that of the maximum principal angles and its value is 0.9822 for $k_1 = k_2 = 15$. Both these plots indicate that the true and the approximated eigenimages span the same vector space when the full dimension is used.

TABLE II

RESULTS FOR DIFFERENT ALGORITHMS

	Proposed algorithm			Chang's algorithm			MATLAB SVD	
Video	Time (s)	k_2	p	Time (s)	k_2	p	Time (s)	k_1
1	11.2070	15	15	18.8590	15	15	67.2910	15
2	3.7950	4	6	13.4600	4	6	67.2920	4
3	73.9310	67	68	88.1430	66	68	67.3320	63
4	68.9240	63	65	78.6590	63	65	67.3320	60
5	3.6460	4	6	13.1600	4	6	62.2240	4
6	1.3220	1	2	11.8480	1	2	62.2240	1
7	6.9210	10	11	15.6930	10	11	62.2440	9
8	4.0670	5	7	13.5900	5	7	62.1840	5

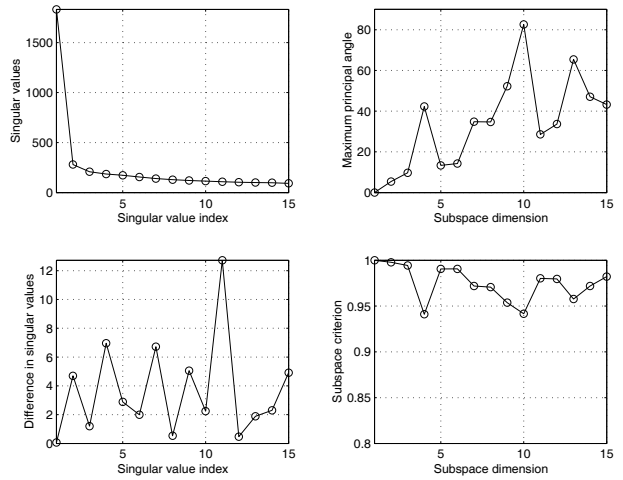

Fig. 3. This figure shows different plots of the error measures calculated for video #5 used in [23] that is representative of the general behavior for most of the video sequences when the approximated SVD was compared with the true SVD. The first column shows the plot of the singular values for high-resolution images and its difference with the approximated singular values using the modified algorithm. The first plot in the second column shows the maximum principal angles (in degrees) between the respective eigenspaces, when the subspace dimension was varied from 1 to 15, while the remaining plot shows the subspace criterion measure, when the subspace dimension was varied from 1 to 15.

VI. CONCLUSION

We have presented a theoretical background for quantifying the tradeoff associated with performing eigendecomposition on correlated images at lower resolutions in order to mediate the high computational expense of performing these calculations at high resolutions. Using this background, we have modified the fastest known algorithm for computing the eigenspace decomposition of correlated images to obtain a more computationally efficient algorithm. The proposed algorithm enjoys the advantage of making use of the similarity within the images along with the similarity between the images. Examples show that the algorithm performs very well even on arbitrary video sequences.

REFERENCES

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. London: Academic Press, 1990.

[2] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer.*, vol. 4, no. 3, pp. 519–524, March 1987.

[3] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *IEEE Trans. PAMI*, vol. 12, no. 1, pp. 103–108, Jan. 1990.

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

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 711–720, July 1997.

[6] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. PAMI*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

[7] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Jun 21-23 1994, pp. 84–91.

[8] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. PAMI*, vol. 24, no. 1, pp. 34–58, Jan. 2002.

[9] H. Murase and R. Sakai, "Moving object recognition in eigenspace representation: Gait analysis and lip reading," *Pattern Recognit. Lett.*, vol. 17, no. 2, pp. 155–162, Feb. 1996.

[10] G. Chiou and J. N. Hwang, "Lipreading from color video," *IEEE Trans. on Image Processing*, vol. 6, no. 8, pp. 1192–1195, Aug. 1997.

[11] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspaces," *IEEE Trans. PAMI*, vol. 16, no. 12, pp. 1219–1227, Dec. 1994.

[12] C. Y. Huang, O. I. Camps, and T. Kanungo, "Object recognition using appearance-based parts and relations," in *Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, San Juan, PR, USA, June 17-19 1997, pp. 877–883.

[13] R. J. Campbell and P. J. Flynn, "Eigenshapes for 3D object recognition in range data," in *Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, June 23-25 1999, pp. 505–510.

[14] M. Jogan and A. Leonardis, "Robust localization using eigenspace of spinning-images," in *Proc. of the IEEE Workshop on Omnidirectional Vision*, Hilton Head Island, South Carolina, USA, June 2000, pp. 37–44.

[15] S. Yoshimura and T. Kanade, "Fast template matching based on the normalized correlation by using multiresolution eigenimages," in *1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, Texas, Nov. 11-12 1994, pp. 83–88.

[16] J. Winkeler, B. S. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition," in *Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, June 23-25 1999, pp. 511–516.

[17] S. K. Nayar, H. Murase, and S. A. Nene, "Learning, positioning, and tracking visual appearance," in *Proc. of the IEEE Int. Conf. on Robot. Automat.*, San Diego, CA, USA, May 8-13 1994, pp. 3237–3246.

[18] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[19] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Computer Vision*, vol. 14, no. 1, pp. 5–24, Jan. 1995.

[20] H. Murase and S. K. Nayar, "Detection of 3D objects in cluttered scenes using hierarchical eigenspace," *Pattern Recognit. Lett.*, vol. 18, no. 4, pp. 375–384, April 1997.

[21] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace method for robot vision," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 750–758, Oct. 1996.

[22] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 696–710, July 1997.

[23] C. Y. Chang, A. A. Maciejewski, and V. Balakrishnan, "Fast eigenspace decomposition of correlated images," *IEEE Trans. on Image Processing*, vol. 9, no. 11, pp. 1937–1949, Nov. 2000.

[24] J. C. Nash, "A one-sided transformation method for the singular value decomposition and algebraic eigenproblem," *Comp. J.*, vol. 18, no. 1, pp. 74–76, 1975.

[25] K. Saitwal, A. A. Maciejewski, and R. G. Roberts, "A comparison of eigendecomposition for sets of correlated images at different resolutions," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct. 27-31 2003, pp. 1011–1017.