

Fast Eigenspace Decomposition of Correlated Images Using Their Low-Resolution Properties

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 *a priori* how this will affect the quality of the resulting eigendecomposition. The work presented here provides an analysis of how different resolution reduction techniques affect the eigendecomposition. A computationally efficient algorithm for calculating the eigendecomposition based on this analysis is proposed. 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. Various referred to as eigenspace methods, singular value decomposition (SVD) methods, principal component analysis methods, and Karhunen-Loeve transformation methods [1], [2], they have been used extensively in a variety of applications such as face characterization [3], [4] and recognition [5]-[9], lip-reading [10], [11], object recognition [12]-[15], pose detection [16], [17], visual tracking [18], [19], and inspection [20]-[23]. 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 [24]-[31]. 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 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. Many users of eigendecomposition techniques would like to use as high 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 provide an analysis of how different resolution reduction techniques affect the resulting eigendecomposition. This analysis is then used to modify the fastest known eigendecomposition algorithm, proposed by Chang et al. [31], to improve its computational efficiency without sacrificing the quality of the resulting eigenimages.

The remainder of this paper is organized as follows. Section II provides a review of the fundamentals of applying eigendecomposition to related images. An overview of Chang’s algorithm is given in Section III, while the limitation of its computational efficiency, due to working with the highest resolution, is pointed out in Section IV. An analysis of a simple example is also provided in Section IV that explains why downsampling by selecting random pixels can be more effective than using simple filtering techniques. This analysis motivated a fast SVD 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 on a set of arbitrary video sequences. Finally, some concluding remarks are given in Section VII.

II. PRELIMINARIES

A grey-scale 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 $\mathcal{X} \in [0, 1]^{h \times v}$. Because we will be considering sets of related images, the *image vector* \mathbf{x} of length $m = h \times v$ can be obtained by “row-scanning” an image into a column vector, i.e.,

¹This 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.

$\mathbf{x} = \text{vec}(\mathcal{X}^T)$. The *image data matrix* of a set of images $\mathcal{X}_1, \dots, \mathcal{X}_n$ is an $m \times n$ matrix, denoted X , and defined as $X = [\mathbf{x}_1 \cdots \mathbf{x}_n]$, where typically $m \gg n$. We consider only the case where n is fixed, as opposed to cases where X is constantly updated with new images.

The SVD of X is given by

$$X = U\Sigma V^T, \quad (1)$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal, and $\Sigma = [\Sigma_d \mathbf{0}]^T \in \mathbb{R}^{m \times n}$ where $\Sigma_d = \text{diag}(\sigma_1, \dots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ 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 $\hat{\mathbf{u}}_i$, $i = 1, \dots, m$, are referred to as the left singular vectors or eigenimages of X , while the columns of V , denoted $\hat{\mathbf{v}}_i$, $i = 1, \dots, n$, are referred to as the right singular vectors of X . The corresponding singular values measure how ‘‘aligned’’ the columns of X are with the associated eigenimage.

In practice, the singular vectors $\hat{\mathbf{u}}_i$ are not known or computed exactly, and instead estimates $\hat{\mathbf{e}}_1, \dots, \hat{\mathbf{e}}_k$ which form a k -dimensional basis are used. For quantifying the accuracy of these estimates, one of the measures we will use is the ‘‘energy recovery ratio’’ [31], denoted ρ , and defined as

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

where $\|\cdot\|_F$ denotes the Frobenius norm. Another measure we will use is the degree to which these estimates span the subspace of the first k^* true eigenimages, which will be referred to as the subspace criterion, γ , given by

$$\gamma = \sqrt{\frac{1}{k^*} \sum_{i=1}^k \sum_{j=1}^{k^*} (\hat{\mathbf{e}}_i \cdot \hat{\mathbf{u}}_j)^2}, \quad (3)$$

which is 1 if the entire subspace is spanned.

III. CHANG’S EIGENDECOMPOSITION ALGORITHM

As described earlier, calculation of the estimates of the singular vectors $\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{k^*}$ of X 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 [24]–[31]. The algorithm proposed by Chang et al. [31] is currently the fastest known algorithm for this purpose. This section provides an overview of Chang’s algorithm with the following sections discussing how low-resolution estimates can be used to improve its computational efficiency without sacrificing the quality of the resulting eigenimages.

The technique in [31] is motivated by the observation that the SVD of X can be determined in a closed form when the images are derived by a planar rotation of a single image, thus resulting in $X^T X$ being circulant. The *real* eigendecomposition of such an $X^T X$ is given by

$$X^T X = HDH^T, \quad (4)$$

where D is an $n \times n$ diagonal matrix containing the eigenvalues of $X^T X$ as its diagonal elements, while H consists of n eigenvectors of $X^T X$ as its columns and it is given by

$$\begin{aligned} H &= [\hat{\mathbf{h}}_1 \quad \hat{\mathbf{h}}_2 \quad \hat{\mathbf{h}}_3 \quad \hat{\mathbf{h}}_4 \quad \hat{\mathbf{h}}_5 \quad \cdots] \\ &= \sqrt{2} \left[\begin{array}{cccccc} \frac{1}{\sqrt{2}} \hat{\mathbf{f}}_0 & \Re \hat{\mathbf{f}}_1 & \Im \hat{\mathbf{f}}_1 & \Re \hat{\mathbf{f}}_2 & \Im \hat{\mathbf{f}}_2 & \cdots \end{array} \right] \\ &= \sqrt{\frac{2}{n}} \left[\begin{array}{cccc} \frac{1}{\sqrt{2}} & c_0 & -s_0 & \cdots \\ \frac{1}{\sqrt{2}} & c_1 & -s_1 & \cdots \\ \vdots & \vdots & \vdots & \cdots \\ \frac{1}{\sqrt{2}} & c_{n-1} & -s_{n-1} & \cdots \end{array} \right], \quad (5) \end{aligned}$$

where $\hat{\mathbf{f}}_i$ gives the i^{th} column of the Fourier matrix $F_{(n \times n)}$, $c_k = \cos(k\theta)$, and $s_k = \sin(k\theta)$. Thus 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 $V = H$. Finally, U can be calculated using $U\Sigma = XH$. This analysis indicates that for planar transformations, the right singular vectors of X are pure sinusoids of frequencies that are multiples of $2\pi/n$ radians and the frequencies of the (ordered) right singular vectors increase linearly with their index.

Although the above analysis does not hold true for general 3-D transformations, the empirical results in [31] show that for correlated images, their right singular vectors are approximately spanned by a handful of harmonics that are dominated by low 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 calculating the SVD can be significantly reduced. This forms the basis for Chang’s eigendecomposition algorithm.

Specifically, if p is such that the power spectra of the first k singular vectors are essentially restricted to the band $[0, 2\pi p/n]$, then it is shown in [31] that the first k singular values $\hat{\sigma}_1, \dots, \hat{\sigma}_k$ and the corresponding singular vectors $\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_k$ of XH_p serve as excellent estimates to those of X , where H_p denotes the matrix comprising the first p ($p \ll n$) columns of H . It is also shown that when p is the smallest number such that $\rho(X^T, \hat{\mathbf{h}}_1, \dots, \hat{\mathbf{h}}_p) \geq \mu$, the quantity $\rho(X, \hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_k)$ will exceed μ for some $k \leq p$, thus achieving the user-specified reconstruction ratio, μ .

In summary, Chang’s algorithm makes use of the fact that the analytical expressions for the eigendecomposition, based on the theory of circulant matrices, can serve as a good approximation to the eigendecomposition of arbitrary video sequences. The algorithm shows better computational efficiency, because the SVD is computed on a much smaller matrix XH_p , which is formed by reducing X in the temporal dimension. However, one can still improve this computational efficiency by operating on lower resolution images. The appropriate manner of downsampling to achieve these low-resolution images is the topic of the next section.

IV. EFFECT OF SPATIAL REDUCTION TECHNIQUES

In our recent work [32], a mathematical framework was provided for quantifying the effect of varying the resolution of the images on the eigendecomposition that is computed from those images. Image data matrices at different resolutions were formed and the corresponding SVDs were calculated. The analysis showed that the right singular vectors of correlated images are not appreciably affected by image resolution. Therefore the SVD of high-resolution images can be accurately estimated from the right singular vectors obtained from low-resolution images. These observations motivated several modifications to Chang's algorithm that can improve its computational efficiency.

One can observe that the first two steps in Chang's algorithm, i.e., the calculation of the value p and the computation of the SVD of XH_p , still requires a significant amount of time, because the algorithm always works with the full spatial resolution of the images. Hence it is desirable to reduce the images in the spatial dimension first. In this section, two different image reduction techniques are considered: box filtering and random pixel selection. The effect that these techniques have on the spatial and the temporal properties of X will be illustrated using a simple example.²

Consider an m by 2 image data matrix X . Because the box filtering technique acts like a low pass filter, it is convenient to represent X as

$$\begin{aligned} X &= \begin{bmatrix} H\alpha & | & H\beta \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^m \alpha_i \hat{\mathbf{h}}_i & | & \sum_{i=1}^m \beta_i \hat{\mathbf{h}}_i \end{bmatrix}, \end{aligned} \quad (6)$$

where the $m \times m$ orthogonal matrix H has the form (5) and α, β are $m \times 1$ coefficient vectors. The correlation matrix for X is given by

$$\begin{aligned} X^T X &= \begin{bmatrix} (H\alpha)^T H\alpha & (H\alpha)^T H\beta \\ (H\beta)^T H\alpha & (H\beta)^T H\beta \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^m \alpha_i^2 & \sum_{i=1}^m \alpha_i \beta_i \\ \sum_{i=1}^m \alpha_i \beta_i & \sum_{i=1}^m \beta_i^2 \end{bmatrix}. \end{aligned} \quad (7)$$

Now consider reducing the image vectors in X using simple box filtering with the integer reduction factor r . If the pixels in both the image vectors in X are ordered so that a pixel in the low-resolution image vectors can be obtained by box-filtering the consecutive pixels in the corresponding high-resolution image vectors, then the resulting low-resolution image data matrix is given by

$$X_b = \frac{1}{r} \begin{bmatrix} x_{11} + \cdots + x_{r1} & x_{12} + \cdots + x_{r2} \\ \vdots & \vdots \\ x_{d1} + \cdots + x_{m1} & x_{d2} + \cdots + x_{m2} \end{bmatrix}, \quad (8)$$

where $d = m - r + 1$. For simplicity of presentation, consider the case where $m = 4$ that gives the following

²Other image reduction techniques such as nearest neighbor and bicubic interpolation were also implemented. The performance of these techniques was comparable to that of box filtering.

high-resolution image data matrix:

$$X_{42} = \frac{1}{2} \begin{bmatrix} \alpha_1 + \sqrt{2}\alpha_2 + \alpha_4 & \beta_1 + \sqrt{2}\beta_2 + \beta_4 \\ \alpha_1 - \sqrt{2}\alpha_3 - \alpha_4 & \beta_1 - \sqrt{2}\beta_3 - \beta_4 \\ \alpha_1 - \sqrt{2}\alpha_2 + \alpha_4 & \beta_1 - \sqrt{2}\beta_2 + \beta_4 \\ \alpha_1 + \sqrt{2}\alpha_3 - \alpha_4 & \beta_1 + \sqrt{2}\beta_3 - \beta_4 \end{bmatrix}. \quad (9)$$

The box filtered image data matrix is then given by

$$X_b = \frac{1}{4} \begin{bmatrix} 2\alpha_1 + \sqrt{2}\alpha_2 - \sqrt{2}\alpha_3 & 2\beta_1 + \sqrt{2}\beta_2 - \sqrt{2}\beta_3 \\ 2\alpha_1 - \sqrt{2}\alpha_2 + \sqrt{2}\alpha_3 & 2\beta_1 - \sqrt{2}\beta_2 + \sqrt{2}\beta_3 \end{bmatrix} \quad (10)$$

and the corresponding correlation matrix is

$$X_b^T X_b = \frac{1}{4} \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \quad (11)$$

where

$$\begin{aligned} a &= 2\alpha_1^2 + \alpha_2^2 + \alpha_3^2 - 2\alpha_2\alpha_3, \\ b &= 2\alpha_1\beta_1 + \alpha_2\beta_2 - \alpha_2\beta_3 - \alpha_3\beta_2 + \alpha_3\beta_3, \\ c &= 2\beta_1^2 + \beta_2^2 + \beta_3^2 - 2\beta_2\beta_3. \end{aligned} \quad (12)$$

If we compare (11) with (7), we can observe that the box filtering eliminates the highest frequency component altogether and amplifies the DC component by 2. Thus, it makes the low-frequency components more dominant and reduces the importance of the high-frequency components. This changes the spatial and the temporal properties of the original images resulting in different singular values and singular vectors for X and X_b .

An alternative method for determining a reduced resolution version of an image is to randomly select pixels from X . Thus, with the reduction factor r , any $\frac{m}{r}$ rows from X can be selected to form the low-resolution image data matrix X_r . If we try to reduce X_{42} in (9) using this technique with $r = 2$, then we obtain any of the $\binom{4}{2} = 6$ different possible X_r 's with equal probability, i.e., $P(X_{r_i}) = \frac{1}{6}$, for $i = 1, 2, \dots, 6$. If we denote each $X_r^T X_{r_i}$ by RX_i , then the expectation of these correlation matrices gives

$$\begin{aligned} E[RX] &= \sum_{i=1}^6 (RX_i) P(RX = RX_i) \\ &= \sum_{i=1}^6 \frac{1}{6} RX_i \\ &= \frac{1}{2} \begin{bmatrix} \sum_{i=1}^4 \alpha_i^2 & \sum_{i=1}^4 \alpha_i \beta_i \\ \sum_{i=1}^4 \alpha_i \beta_i & \sum_{i=1}^4 \beta_i^2 \end{bmatrix} \\ &= \frac{1}{2} X_{42}^T X_{42}. \end{aligned} \quad (13)$$

Note that the right singular vectors and their relative importance are the same at both the resolutions (although the corresponding singular values are halved for X_r), indicating that X_r reflects the same properties as X .

The empirical results (refer to Fig. 1) for arbitrary video sequences also depict a similar behavior for both the resolution reduction techniques. For all the video sequences, the p_b values decrease rapidly even for small reduction factors,

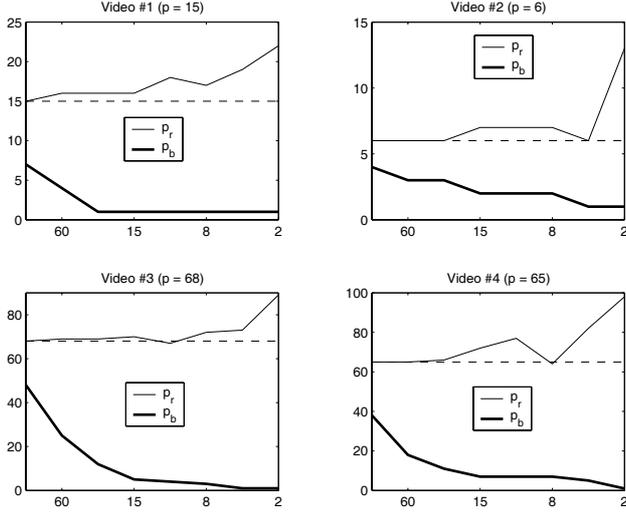


Fig. 1. This figure shows the plots for the first four video sequences used in this paper. The image data matrices at different resolutions are formed after reducing the original images from $m = 240 \times 352$ to the lower resolutions of 120×176 , 60×88 , 30×44 , 15×22 , 10×15 , 8×12 , 4×6 , and 2×3 . Each subplot title gives the video number and its corresponding p value (plotted with a horizontal dashed line) at the highest resolution, where p is the smallest number of frequency harmonics required to get $\rho > 0.95$ in (2). The horizontal axis gives the resolution of low-resolution images in one dimension, while the plots p_r and p_b give the p values at the lower resolutions when the images are reduced using random pixel selection and box filtering, respectively. For the random pixel selection approach, the images are reduced for four different times to calculate four different p values and the maximum p value is assigned to p_r .

indicating that box filtering makes the first few frequency components in the low-resolution images more dominant than they actually are in the high-resolution images. On the other hand, the p_r values are (almost) always above the “true” p values for all the video sequences, indicating that random pixel selection does not significantly alter the temporal properties of high-resolution image sequences. (This behavior is quite intuitive as we are working with the pixels themselves rather than their averages.) The above analysis motivates a modified version of Chang’s algorithm, which is the topic of the next section.

V. FAST EIGENDECOMPOSITION ALGORITHM

Our objective is to determine the first k left singular vectors of X . Using the analysis of the resolution reduction techniques in the previous section, we now make the appropriate modifications to Chang’s algorithm to improve its computational efficiency. Random pixel selection is used to reduce X in the spatial dimension and then Chang’s method is used to reduce it further in the temporal dimension. We first present an overview of our algorithm and then expand on the details.

- 1) Generate the Fourier matrix, $F_{(n \times n)}$, and its *real* counterpart, $H_{(n \times n)}$ for $X_{(m \times n)}$.
- 2) Randomly select n pixels from each image in X to obtain the $n \times n$ reduced image data matrix X_n .³

³The same permutation of n pixels is used over all m images, however, the order of these randomly selected pixels in the reduced images does not matter.

- 3) Determine the smallest number p such that

$$\rho(X_n^T, \hat{\mathbf{h}}_1, \dots, \hat{\mathbf{h}}_p) = \frac{\sum_{i=1}^p \|X_n \hat{\mathbf{h}}_i\|_2^2}{\|X_n\|_F^2} > \mu, \quad (14)$$

where μ is the user-specified reconstruction ratio.

- 4) Compute the reduced SVD of $(X_n H_p)_{(n \times p)} = (U_n)_{(n \times p)} (S_n)_{(p \times p)} (V_n)_{(p \times p)}^T$.⁴
- 5) Repeat Steps 2 through 4 for three more times and concatenate all $S_n V_n^T$ matrices to form

$$A_{(P \times s)}^T = \begin{bmatrix} S_{n_1} V_{n_1}^T \\ S_{n_2} V_{n_2}^T \\ S_{n_3} V_{n_3}^T \\ S_{n_4} V_{n_4}^T \end{bmatrix},$$

where s is the maximum of the four values of p and P is the sum of all values of p .⁵

- 6) Compute the reduced SVD of $A_{(s \times P)} = (U_s)_{(s \times s)} (S_s)_{(s \times P)} (V_s)_{(P \times P)}^T$.
- 7) Compute $Z_{(n \times s)} = (H_s)_{(n \times s)} (U_s)_{(s \times s)}$ to get an initial estimate of right singular vectors of X .⁶
- 8) Perform Steps 2 and 3. If $p > s$, perform Step 4 and compute $Z^{\text{new}} = (H_p)(V_n)$. Then compute the component of Z^{new} that is orthogonal to Z using

$$\mathbf{w} = [I - ZZ^T] \hat{\mathbf{z}}_i^{\text{new}}, \quad \text{for } i = 1, \dots, p, \quad (15)$$

where I is an $n \times n$ identity matrix. If $\|\mathbf{w}\| > \epsilon$ for any $\hat{\mathbf{z}}_i^{\text{new}}$, update $Z = [Z, \hat{\mathbf{w}}]$ and $s = s + 1$, where ϵ is some user-specified threshold.

- 9) Repeat Step 8 until $p \leq s$ for four consecutive times.
- 10) Compute $E_{(m \times s)} = X_{(m \times n)} Z_{(n \times s)}$ that gives an approximate basis for the left singular vectors of X .
- 11) Find the orthonormal basis for E using the reduced QR decomposition, i.e., $E_{(m \times s)} = \hat{E}_{(m \times s)} R_{(s \times s)}$.
- 12) Return $\hat{E}_{(m \times s)}$.
- 13) Optionally, check if $k < s$ by finding $\hat{\mathbf{e}}_1, \dots, \hat{\mathbf{e}}_k$ such that $\rho(X, \hat{\mathbf{e}}_1, \dots, \hat{\mathbf{e}}_k) > \mu$.

We will now explain the above steps in more detail. Steps 2 through 4 compute the p value for X_n and the SVD of the $X_n H_p$ matrix. Here we will describe why n pixels are selected as the resolution of the downsampled version of an image. Recall that it is always true that $p \leq n$ (typically with $p \ll n$) for any image data matrix [31], therefore more than n pixels in each column in the reduced image data matrix are never needed to preserve the rank of X_n at p . However, if the resolution is selected at a value less than n , one always runs the risk of artificially reducing the rank below p .

Once the SVD of $X_n H_p$ is calculated, $(H_p)(V_n)$ gives the right singular vectors of X_n . These right singular vectors can be considered as a “good” approximation of

⁴ $X_n H_p$ is readily available after Step 3.

⁵All $S_n V_n^T$ matrices should be padded with the appropriate number of columns of zeros if necessary, so that each of them has s columns.

⁶ H_s consists of the first s columns of H in (5).

their high-resolution counterparts [32]. However, different X_n 's require different numbers of harmonics to satisfy the user-specified reconstruction ratio, because the random pixels used to create a specific X_n may not accurately represent the temporal properties of the entire X . Hence Steps 2 through 4 are performed four times to improve the probability of accurately representing the high-resolution image data matrix. The number of times to repeat Steps 2 – 4 was empirically determined (on average) to optimize computational efficiency.⁷

Step 5 concatenates all $S_n V_n^T$ matrices to form the matrix A whose range will approximately span the dominant right singular vectors of X [32]. The SVD of A is computed in Step 6 to find its range, given by U_s . Thus, $(H_s)(U_s)$ computed in Step 7 can be considered as a good initial estimate of the right singular vectors of X . Note that if V_n 's are used instead of $S_n V_n$'s to form A , then Step 7 will result in an unordered estimate of the right singular vectors of X . To obtain an ordered estimate, the right singular vectors in each V_n are scaled by their corresponding singular values before being concatenated in Step 5.

Fig. 1 shows that even after performing Steps 2 and 3 four different times, the maximum p values for Videos 3 and 4 were below the “true” p values. Hence Steps 8 and 9 are performed to check if there is any new information available in additional samplings of X . If the new information in any \hat{z}_i^{new} is above a threshold, the Z matrix is updated. When no columns are added to the Z matrix for four consecutive times, the algorithm assumes that the final Z matrix provides a “good” basis for the right singular vectors of X .⁸ In short, Steps 2 through 9 are performed to find the approximate right singular vectors of X .

Step 10 computes the approximate basis for the left singular vectors of X , while Step 11 computes the corresponding orthonormal basis using the QR decomposition. Step 13 optionally computes the minimum subspace that will satisfy the user-specified reconstruction ratio.

We now briefly analyze the computational expense of our algorithm. The cost incurred in Step 2, i.e., constructing X_n from X requires $O(n^2)$ flops, while the estimation of the smallest number p in Step 3 requires $O(n^2 p)$ flops. In Step 4, the cost of computing the SVD of the $n \times p$ matrix $X_n H_p$ requires $O(np^2)$ flops. Step 5 performs Steps 2 through 4 four times requiring $O(n^2) + O(n^2 p) + O(np^2)$ flops. In Step 6, the cost of computing the SVD of the $s \times P$ matrix A requires $O(sP^2)$ flops, while finding the initial estimate of the right singular vectors of X in Step 7 requires $O(ns^2)$ flops. Steps 8 and 9 that check if any new information should be added to Z requires $O(n^2) + O(n^2 p) + O(np^2)$ flops and is repeated an unknown, but typically small, number of times. In Step 10, multiplication of X with Z requires $O(ms^2)$ flops and the

⁷Using more than four iterations may be unnecessary and using fewer than four may require more iterations of the more computationally expensive Step 8.

⁸The value four was empirically determined to make it highly unlikely that the number of columns in Z is far from the true value of p .

QR decomposition of E in Step 11 requires $O(2ms^2 - \frac{2}{3}s^3)$ flops. Finally, determination of the minimum dimension k in Step 13 requires $O(mnk)$ flops. If $s \ll n \ll m$, then the total computation required is $O(2ms^2 - \frac{2}{3}s^3)$, which is the cost of the QR decomposition.

VI. EXPERIMENTAL RESULTS

We consider the problem of computing the eigendecomposition of images representing successive frames of arbitrary video sequences. Specifically, we consider eight video sequences that are used in [31], i.e., 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 $\mu = 0.95$ and $\epsilon = 10^{-6}$. Table I shows a breakdown of the average time required for the different steps in the proposed algorithm. It shows that the first ten steps do not take much time as compared to Step 11 that computes the orthonormal basis for E using QR decomposition. The table also shows that the calculation of the first k eigenimages in Step 13 requires a significant amount of time. Thus a user may prefer to stop after Step 12 and simply use all s eigenimages. The total time for Steps 1 – 12 is given in the column labelled “part”. If one indeed needs to know the minimum subspace, then the total time required for our algorithm is given in the “Total” column.

The proposed algorithm was run ten different times for each video sequence and the mean values for k , “Time”, s , and γ were calculated. (Mean values for s and k values were rounded to the nearest integer.) Table II summarizes the performance of the algorithm, showing k^* , k , p , s , γ , and the computation times.⁹ Compared to the direct SVD, the speedup factors with our algorithm are in the range of 2.99 – 148.14, depending on the value of s . The difference between $\rho(X, \hat{u}_1, \dots, \hat{u}_{k^*})$ and $\rho(X, \hat{e}_1, \dots, \hat{e}_k)$ for each set was less than 0.13%, with an average of 0.07%, which reveals that $\{\hat{e}_1, \dots, \hat{e}_k\}$ provides a very good approximate basis for the first k^* eigenimages $\{\hat{u}_1, \dots, \hat{u}_{k^*}\}$.

VII. CONCLUSION

We have explained how different image resolution reduction techniques affect the eigendecomposition computed using those images and have shown that downsampling using randomly selected pixels is more effective than simple filtering techniques. Using the low-resolution properties of correlated images, we have been able to improve the fastest known eigenspace decomposition algorithm to obtain a more computationally efficient algorithm. The proposed algorithm enjoys the advantage of making use of the similarity within the images as well as the similarity between the images. Examples show that the algorithm performs very well even on arbitrary video sequences.

⁹Note that there is only one column for k in Table II as these values remain the same for both the proposed algorithm and Chang’s algorithm.

TABLE I
TIME REQUIRED FOR THE PROPOSED ALGORITHM
(ALL TIMES ARE IN SECONDS)

Video	Time required for different steps					
	2 - 5	6 - 10	11	Part	13	Total
1	0.07	0.75	1.34	2.16	4.18	6.35
2	0.05	0.46	0.21	0.72	1.48	2.20
3	0.32	2.53	18.40	21.25	16.96	38.21
4	0.32	2.42	17.36	20.10	15.94	36.04
5	0.05	0.45	0.32	0.82	1.39	2.21
6	0.04	0.32	0.04	0.40	0.68	1.08
7	0.06	0.56	0.66	1.28	2.87	4.15
8	0.05	0.40	0.20	0.65	1.63	2.28

TABLE II
COMPARISON OF DIFFERENT ALGORITHMS
(ALL TIMES ARE IN SECONDS)

Video	k	Proposed			Chang			Direct	
		Time	s	γ	Time	p	γ	Time	k^*
1	15	2.1	17	0.99	15.7	15	0.98	67.3	15
2	4	0.8	7	1.00	12.8	6	1.00	67.3	4
3	66	21.8	70	0.96	74.3	68	0.95	67.3	63
4	63	22.5	71	0.94	65.7	65	0.92	67.3	60
5	4	0.8	7	0.95	12.6	6	0.90	62.2	4
6	1	0.4	2	1.00	11.9	2	1.00	62.2	1
7	10	1.3	12	0.99	13.7	11	0.99	62.2	9
8	5	0.8	8	0.98	12.6	7	0.98	62.2	5

REFERENCES

- [1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. London: Academic Press, 1990.
- [2] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, Feb. 2001.
- [3] 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.
- [4] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, Jan. 1990.
- [5] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71-86, March 1991.
- [6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.
- [7] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042-1052, Oct. 1993.
- [8] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Jun 21-23 1994, pp. 84-91.
- [9] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, Jan. 2002.
- [10] 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.
- [11] G. Chiou and J. N. Hwang, "Lipreading from color video," *IEEE Trans. Image Processing*, vol. 6, no. 8, pp. 1192-1195, Aug. 1997.
- [12] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1219-1227, Dec. 1994.
- [13] C. Y. Huang, O. I. Camps, and T. Kanungo, "Object recognition using appearance-based parts and relations," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, San Juan, PR, USA, Jun 17-19 1997, pp. 877-883.
- [14] R. J. Campbell and P. J. Flynn, "Eigenshapes for 3D object recognition in range data," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, June 23-25 1999, pp. 505-510.
- [15] M. Jogan and A. Leonardis, "Robust localization using eigenspace of spinning-images," in *Proc. IEEE Workshop on Omnidirectional Vision*, Hilton Head Island, South Carolina, USA, June 2000, pp. 37-44.
- [16] 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.
- [17] J. Winkler, B. S. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, June 23-25 1999, pp. 511-516.
- [18] S. K. Nayar, H. Murase, and S. A. Nene, "Learning, positioning, and tracking visual appearance," in *Proc. IEEE Int. Conf. on Robot. Automat.*, San Diego, CA, USA, May 8-13 1994, pp. 3237-3246.
- [19] M. J. Black and A. D. Jepson, "Eigenttracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Computer Vision*, vol. 26, no. 1, pp. 63-84, 1998.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696-710, July 1997.
- [24] H. Murase and M. Lindenbaum, "Partial eigenvalue decomposition of large images using the spatial temporal adaptive method," *IEEE Trans. Image Processing*, vol. 4, no. 5, pp. 620-629, May 1995.
- [25] R. Haimi-Cohen and A. Cohen, "Gradient-type algorithms for partial singular value decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 137-142, Jan. 1987.
- [26] 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. Acoustics, Speech, and Signal Processing*, vol. 37, no. 10, pp. 1550-1556, Oct. 1989.
- [27] C. R. Vogel and J. G. Wade, "Iterative SVD-based methods for ill-posed problems," *SIAM J. Sci. Comput.*, vol. 15, no. 3, pp. 736-754, May 1994.
- [28] S. Shlien, "A method for computing the partial singular value decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 6, pp. 671-676, Nov. 1982.
- [29] H. Murakami and V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 5, pp. 511-515, Sept. 1982.
- [30] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkler, and H. Zhang, "An eigenspace update algorithm for image analysis," *CVGIP: Graphic Models and Image Processing*, vol. 59, no. 5, pp. 321-332, Sept. 1997.
- [31] C. Y. Chang, A. A. Maciejewski, and V. Balakrishnan, "Fast eigenspace decomposition of correlated images," *IEEE Trans. Image Processing*, vol. 9, no. 11, pp. 1937-1949, Nov. 2000.
- [32] K. Saitwal, A. A. Maciejewski, and R. G. Roberts, "Analysis of eigendecomposition for sets of correlated images at different resolutions," in *Proc. IEEE Int. Conf. on Robot. Automat.*, New Orleans, LA, USA, Apr. 26-May 1 2004, pp. 1393-1398.