Using the Low-Resolution Properties of Correlated Images to Improve the Computational Efficiency of Eigenspace Decomposition

Kishor Saitwal, *Student Member, IEEE*, Anthony A. Maciejewski, *Fellow, IEEE*, Rodney G. Roberts, *Senior Member, IEEE*, and Bruce A. Draper

***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 well on arbitrary video sequences.**

***Index Terms*—Computational complexity, computer vision, correlation, data compression, eigenspace, image resolution, image sampling, image sequences, singular value decomposition (SVD), video coding.**

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 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], [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 take advantage of the fact that a set of highly correlated images can be approximately represented by a small set of eigenimages [24]–[32]. 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 determines the computational cost of 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 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 improve the computational efficiency of one of the fastest known eigendecomposition algorithms, proposed by Chang *et al.* [32], 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 and reviews the previous works that address the problem of calculating the partial SVD of large matrices. This section also defines comparison criteria to quantify errors in eigendecompositions. Section III illustrates two different techniques of using a low-resolution SVD to approximate a high-resolution SVD. Section IV gives an overview of Chang's algorithm [32] and points out the limitation of its computational efficiency. A mathematical analysis along with some empirical results is provided in Section V that explains why reduction by random sampling can be more effective than any low-pass filtering technique. This analysis along with the low-resolution properties of correlated images from Section III motivated several changes to Chang's algorithm, outlined in Section VI, to quickly compute the desired portion of the eigendecomposition

Manuscript received December 17, 2004; revised July 21, 2005. This work was supported by the National Imagery and Mapping Agency under Contract 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. This work was presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, Sept. 28–Oct. 2, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bruno Carpentieri.

K. Saitwal and A. A. Maciejewski are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373 USA (e-mail: kishor.saitwal@colostate.edu; aam@colostate.edu).

B. A. Draper is with the Department of Computer Science, Colorado State University, Fort Collins, CO 80523-1373 USA (e-mail: draper@cs.colostate.edu).

R. G. Roberts is with the Department of Electrical and Computer Engineering, Florida A&M–Florida State University, Tallahassee, FL 32310-6046 USA (e-mail: rroberts@eng.fsu.edu).

Digital Object Identifier 10.1109/TIP.2006.875231

based on a user-specified measure of accuracy. In Section VII, the performance of the proposed algorithm is evaluated on arbitrary video sequences. Finally, some concluding remarks are given in Section VII.

II. Preliminaries

A. Singular Value Decomposition of Correlated Images

In this work, a grayscale 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 sets of related images are considered here, 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., $\mathbf{x} = \mathrm{vec}(\mathcal{X}^T)$. The *image data matrix* of a set of images $\mathcal{X}_1, \ldots, \mathcal{X}_n$ is an $m \times n$ matrix, denoted X, and defined as $X = [\mathbf{x}_1 \cdots \mathbf{x}_n]$, where typically $m \gg n$. The case with fixed n is considered in this study, as opposed to cases where X is constantly updated with new images.

The SVD of X is given by

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

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 $\Sigma_d = diag(\sigma_1, \ldots, \sigma_n)$ 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 $\hat{\mathbf{u}}_i, i = 1, \ldots, m$, are referred to as the left singular vectors or eigenimages of X, while the columns of V, denoted $\hat{\mathbf{v}}_i, i = 1, \ldots, n$, 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 values and the corresponding singular vectors are not known or computed exactly, and instead their estimates are used. Hence, it is important to define appropriate comparison criteria that can measure the errors between the true and approximated eigenspaces. The next section defines four such error measures that are relevant to a user's motivation for preforming an eigendecomposition.

B. Difference Measures for SVD

The simplest error measures considered in this paper are: 1) the difference between the true and the approximated singular values and 2) the angles between the corresponding singular vectors calculated for a set of correlated images. However, the ith approximated singular vector may not be aligned with the ith true singular vector even though the subspaces containing the first k vectors may span the same vector space. Hence, two more error measures are defined in this section that will compare the subspaces consisting of the singular vectors rather than the individual vectors.

1) Energy Recovery Ratio: True and approximated eigenimages of X can be compared in terms of their capability of recovering the amount of the total energy in X. This "energy recovery ratio" can be given by

$$\rho(X, \tilde{\hat{\mathbf{u}}}_1, \tilde{\hat{\mathbf{u}}}_2, \ldots, \tilde{\hat{\mathbf{u}}}_k) = \frac{\sum_{i=1}^{k} ||\tilde{\hat{\mathbf{u}}}_i^T X||^2}{||X||_F^2} \leq 1 \tag{2}$$

where $\| \cdot \|_F$ represents the Frobenius norm and $\tilde{\hat{\mathbf{u}}}_i$ is the ith approximated eigenimage. The true eigenimages $\{\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_k\}$ yield the maximum energy recovery ratio.

2) Residue Between Subspaces: The possibility that the data matrix $B \in \Re^{m \times k}$ can be rotated into the data matrix $A \in \Re^{m \times k}$ is explored [33] by solving the problem

$$\Delta = \min_{Q} ||A - BQ||_F \tag{3}$$

where $Q \in \Re^{k \times k}$ is an orthogonal matrix, and Δ is the residue. The $Q_{\min}$ that minimizes $||A - BQ||_F$ can be calculated as follows:

- form the matrix $C = B^T A$;
- compute the SVD of C, i.e., $C = U_c \Sigma_c V_c^T$;
- find the orthogonal matrix $Q_{\min} = U_c V_c^T$

where U_c, Σ_c, and V_c are the U, Σ, and V matrices for C, respectively. If A and B are orthogonal matrices, it can be shown that the residue becomes

$$\begin{aligned} \Delta^2 &= ||A - BQ_{\min}||_F^2 \\ &= \mathrm{tr}(A^T A) + \mathrm{tr}(B^T B) - 2\mathrm{tr}\left(Q_{\min}^T C\right) \\ &= 2\left(k - \sum_{i=1}^{k} \sigma_{ci}\right) \end{aligned} \tag{4}$$

with the Σ_c matrix containing the singular values, σ_{ci} of C providing the *principal angles* between the subspaces, i.e.,

$$\mathrm{diag}(\cos(\theta_1), \ldots, \cos(\theta_k)) = \Sigma_c \tag{5}$$

where θ_i is the ith *principal angle*. The smaller the residue Δ, the closer A and B are to representing the same subspace.

The above two error measures provide slightly different information regarding the "quality" of the estimated eigenimages. The energy recovery ratio ρ, implicitly includes the effect of the singular values and thus weights the estimated eigenimages differently based on their importance. In contrast, the residue between the subspaces Δ, is purely a subspace measure. The principal angles that constitute Δ provide detailed information about how the estimated eigenspace is oriented relative to the true eigenspace.

C. Previous Work

The precomputation of estimates of the left singular vectors $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k$ of the matrix X can be a computationally expensive operation when m and n are very large. Reducing this computational expense by taking advantage of the fact that only the principle singular vectors are of interest has been the subject of previous work.

One class of techniques relies on finding the eigenimages iteratively. The variants of the SVD power method, which calculates the dominant singular values and vectors one at a time, are discussed in [24] and [25]. In [28], Vogel *et al.* considered the block power method and the Lanczos method to solve the SVD of ill-posed problems. These methods, unlike the power method, iterate with k pairs of singular vectors at a time instead of one. The gradient-type algorithms [26], [27] recast the search for the dominant singular vectors into an optimization problem that is solved using gradient or conjugate gradient methods. There are other iterative methods that work on symmetric matrices [27], [34] that have been applied to either X^TX or XX^T.

Another class of techniques relies on updating a small set of eigenimages by recursively adding one image at a time. Murakami *et al.* [29] illustrated a method for updating a fixed number of eigenimages. Chandrasekaran's method [30], on the other hand, adaptively changed the number of eigenimages calculated.

Other techniques include Murase *et al.* [31] that compute the eigenimages of the approximated matrix X^TX in the discrete cosine transform domain. Chang *et al.* [32] use a fundamentally different algorithm (refer to Section IV) that was motivated by the fact that for a set of planar rotated images, the matrix X^TX is a circulant matrix and the (unordered) SVD of X for this case is known in closed form. However, the major drawback of all these approaches is that their computation time is highly dependent on the resolution of the original images.

III. Effect of Resolution on The Eigendecomposition

This section illustrates two possible ways of using low-resolution SVD to approximate eigenimages of the original high-resolution images. Section III-A explains the required modifications of a low-resolution SVD for a meaningful comparison with its high-resolution counterpart. Sections III-A1 and III-A2 illustrate two techniques that can be used to approximate high-resolution eigenimages using a low-resolution SVD. These two techniques are then evaluated using some empirical results in Sections III-B.

A. Comparison of SVDs at Different Resolutions

The SVD of correlated images at different resolutions give different U, Σ, and V matrices. To perform a meaningful comparison, the singular values and singular vectors of the low-resolution image data matrix must be modified. To distinguish both the resolution and the size of a singular vector, the notation ${}^{(q)}\mathbf{e}_{i(m)}$ is used, where the preceding superscript q denotes the fact that the vector is associated with q-dimensional image vectors and the subscript m denotes the actual dimension of the vector $\mathbf{e}_i$. The matrix consisting of ${}^{(q)}\mathbf{e}_{i(m)}$ column is represented as ${}^{(q)}E_{(m\times n)}$.

The size of the right singular vectors is not affected by the resolution of the images and hence these vectors at different resolutions can be directly compared with each other. On the other hand, low-resolution eigenimages need to be enlarged properly before they can be compared with high-resolution eigenimages. Finally, low-resolution singular values must be scaled up properly before they can be compared with those at high resolution. (This is due to the change in the norm of an image vector when an image is reduced.)

1) Interpolation of Low-Resolution Eigenimages: One obvious way to compare eigenimages at different resolutions is to enlarge the eigenimages at lower resolutions to match in size with those at higher resolution. This can be performed by using a number of different interpolation techniques. Due to the enlargement and interpolation, the resulting eigenimages are typically no longer orthonormal. If ${}^{(q)}U_{(m\times n)}$ consists of all the interpolated eigenimages, ${}^{(q)}\mathbf{u}_{i(m)}$, as its columns, then QR decomposition can be carried out on this matrix, i.e.,

$$ {}^{(q)}U_{(m\times n)} = {}^{(q)}\hat{U}_{(m\times n)}R \tag{6}$$

where R is an upper triangular matrix and the columns of ${}^{(q)}\hat{U}_{(m\times n)}$ are an orthonormal basis for the interpolated eigenimages. These approximated eigenimages can now be compared with the true eigenimages in ${}^{(m)}\hat{U}_{(m\times n)}$.

Because the low-resolution eigenimages are enlarged to the size of high-resolution eigenimages, each low-resolution singular value should be scaled using

$$ {}^{(q)}\sigma_i = {}^{(q)}\tilde{\sigma}_i \left\| {}^{(q)}\mathbf{u}_{i(m)} \right\| \tag{7}$$

where ${}^{(q)}\tilde{\sigma}_i$ represents the ith low-resolution singular value that is to be compared with a higher resolution singular value.

2) Using Low-Resolution Right Singular Vectors: The high-resolution image data matrix X can also be multiplied by ${}^{(q)}V$ to obtain the basis for the corresponding approximate high-resolution eigenimages, i.e.,

$$ {}^{(q)}U_{(m\times n)} = X\ {}^{(q)}V. \tag{8}$$

Again, the QR decomposition can be carried out to find an orthonormal basis for ${}^{(q)}U_{(m\times n)}$, i.e.,

$$ {}^{(q)}U_{(m\times n)} = {}^{(q)}\hat{U}_{(m\times n)}R. \tag{9}$$

The orthonormal eigenimages in ${}^{(q)}\hat{U}_{(m\times n)}$ can now be compared with the true eigenimages in ${}^{(m)}\hat{U}_{(m\times n)}$.

The norms of the ${}^{(q)}\mathbf{u}_{i(m)}$s can be used as an approximation of the corresponding singular values of X. This approximation can also be obtained from the matrix R, i.e.,

$$ {}^{(q)}\sigma_i = \left\| {}^{(q)}\mathbf{u}_{i(m)} \right\| = \sqrt{\sum_{j=1}^{i} R_{j,i}^2}. \tag{10}$$

B. Empirical Results To Evaluate Two Techniques

The two techniques used to approximate high-resolution eigenimages were evaluated using a data set consisting of images from successive frames of arbitrary video sequences. Specifically, there are 18 such video sequences that were used.

Fig. 1. First, middle, and last frames of the 18 video sequences used in this paper. Each video sequence consists of 150 images and the corresponding image sizes are given in Table I in the row labeled "Index (0)."

TABLE I
IMAGE SIZES AT DIFFERENT RESOLUTIONS FOR THE VIDEO SEQUENCES

	Videos						
Index	$(1)-(4)$	$(5)-(8)$	$(9)-(13)$	$(14),(15)$	(16)	(17)	(18)
(0)	352×240	320×240	160×120	304×228	240×180	320×180	192×144
(1)	176×120	160×120	80×60	152×114	120×90	160×90	96×72
(2)	88×60	80×60	40×30	76×57	60×45	80×45	48×36
(3)	44×30	40×30	20×15	38×28	30×22	40×22	24×18
(4)	22×15	20×15	10×8	19×14	15×11	20×11	12×9
(5)	12×8	10×8	5×4	10×7	8×6	10×6	6×4
(6)	6×4	5×4	3×2	5×4	4×3	5×3	3×2
(7)	3×2	3×2		3×2			

Each video sequence consists of 150 images; the first, middle, and last frames from each set are shown in Fig 1.

The original image sizes for all the video sequences are given in Table I in the second row labeled "Index (0)." This "Index" gives the reduction factor for the original images (the larger the index, the larger the reduction of the original images). The high-resolution image data matrices X are formed for all 18 video sequences using the original images and the corresponding "true" SVDs are computed. The images are then reduced to some lower resolutions using a number of common low-pass filtering tech-

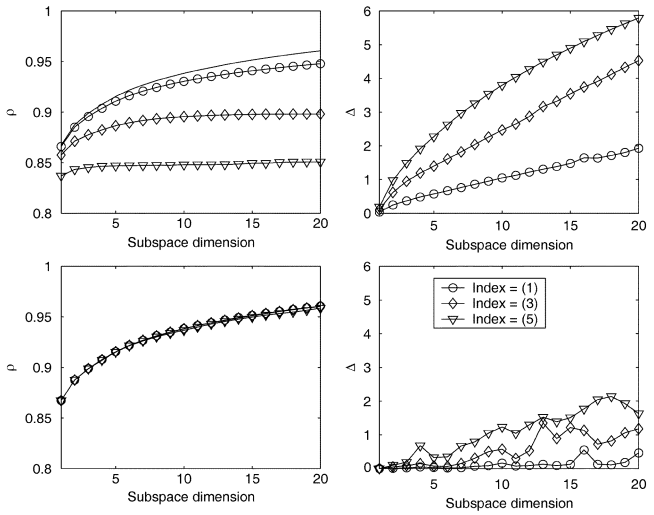


Fig. 2. Comparison of the true and approximated high-resolution eigenimages for video #1 in Fig 1. In particular, the first row shows the comparison plots between the true eigenimages and bilinear interpolated low-resolution eigenimages for the two comparison metrics, i.e., energy recovery ratios (ρ) and rotation indices (Δ), when the subspace dimension k is varied from 1 to 20. The second row shows the same comparison plots between the true eigenimages and the approximated eigenimages computed using the low-resolution right singular vectors. The plots with $\circ$, $\Diamond$, and $\triangledown$ give the comparison between true and approximated eigenimages using images with reduction index (1), (3), and (5), respectively (refer to Table I). Solid lines in the ρ plots give the "true" energy recovery ratio plot.

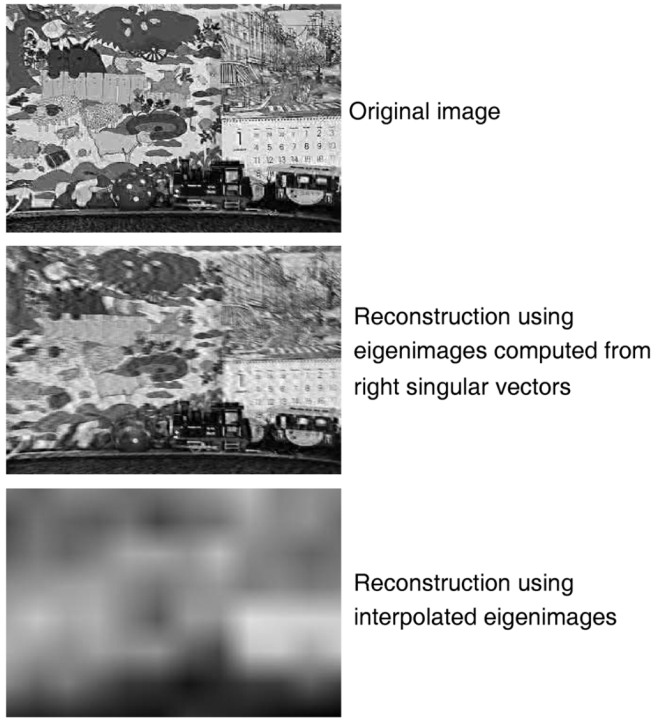


Fig. 3. Comparison of image reconstruction using approximated eigenimages computed by both techniques presented in Section III (the first image in the video sequence #1 in Fig. 1 is used here as an illustrative example). The middle image gives the reconstruction using eigenimages obtained from low-resolution right singular vectors, while the image on the bottom gives the reconstruction using interpolated eigenimages [in both cases, index = (5) and $k = 20$].

niques[1] and the corresponding low-resolution SVDs are computed. These low-resolution SVDs are then used to approximate the high-resolution SVD using the two techniques described earlier.

Fig. 2 shows that using right singular vectors to compute estimates of high-resolution eigenimages is much more accurate than using the interpolation of low-resolution eigenimages. This is also illustrated in Fig. 3 when using the resultant eigenimages for image reconstruction. The mathematical analysis in [35] further supports these results.[2]

IV. Overview of Chang's Algorithm

The previous section has shown that reduction in spatial resolution has less of an effect on right singular vectors as compared to left singular vectors (assuming $m \gg n$). Therefore, eigendecomposition algorithms that operate on right singular vectors are less likely to be affected by reduction in spatial resolution. One such algorithm is Chang's eigendecomposition algorithm [32], which is one of the fastest known algorithms for computing the first k approximate eigenimages of correlated images to the user-specified accuracy. This section gives an overview of that algorithm, along with its computational efficiency. For this purpose, consider X where each $\mathbf{x}_{i+1}$ is obtained from $\mathbf{x}_i$ by a planar rotation of $\theta = 2\pi/n$. The correlation matrix $X^T X$ is given by

$$X^T X = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 & \mathbf{x}_1^T\mathbf{x}_2 & \cdots & \mathbf{x}_1^T\mathbf{x}_n \\ \mathbf{x}_2^T\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{x}_2 & \cdots & \mathbf{x}_2^T\mathbf{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n^T\mathbf{x}_1 & \mathbf{x}_n^T\mathbf{x}_2 & \cdots & \mathbf{x}_n^T\mathbf{x}_n \end{bmatrix}. \tag{11}$$

It is shown in [32] that $X^T X$ is a *circulant matrix* with *circularly symmetric* rows. Hence, its eigendecomposition [36] is given by

$$X^T X = HDH^T \tag{12}$$

where D is an $n \times n$ matrix given by

$$D = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_3, \ldots) \tag{13}$$

and H is an $n \times n$ matrix consisting of the successively higher frequencies, starting from zero frequency, as its columns, which is given by

$$H = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & c_0 & -s_0 & c_0 & -s_0 & \cdots \\ \frac{1}{\sqrt{2}} & c_1 & -s_1 & c_2 & -s_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \frac{1}{\sqrt{2}} & c_{n-1} & -s_{n-1} & c_{2(n-1)} & -s_{2(n-1)} & \cdots \end{bmatrix} \tag{14}$$

[1]The low-pass filtering techniques implemented were box filtering, Gaussian filtering, and finite impulse response filtering with no appreciable differences between any of the results.

[2]A mathematically tractable analysis was performed on all 4×2 image data matrices. It was shown that there always exists a family of such matrices where the interpolation of low-resolution eigenimages gives the worst possible approximations of the high-resolution eigenimages. However, using the right singular vectors always gives good approximations of the high-resolution eigenimages.

where $c_k = \cos(k(2\pi/n))$ and $s_k = \sin(k(2\pi/n))$. If n, the dimension of H, is odd, then the eigenvalues and eigenvectors of X^TX come in pairs except for the first one. If n is even, then the eigenvalue $\lambda_{(n+2/2)}$ does not repeat and its corresponding eigenvector is

$$\mathbf{h}_n = \frac{1}{\sqrt{n}}[1 \quad -1 \quad 1 \quad \cdots \quad -1]^T. \tag{15}$$

The above development means that an unordered SVD for planar rotated image sequence can be given in closed form. In particular, $V = H$, i.e., the right singular vectors are pure sinusoids of frequencies that are multiples of $2\pi/n$ radians and the dominant frequencies of the power spectra of the (ordered) right singular vectors increase linearly with their index. To compute U, observe that $U\Sigma = XH$, which can be computed efficiently using fast Fourier transform (FFT) techniques [32]. Although the above eigendecomposition analysis does not hold true for an arbitrary image sequence, it is shown in [32] that the analytical expressions for planar rotation can serve as good approximations for general 3-D cases. In particular, the following two properties are observed in arbitrary video sequences.

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 two properties indicate that the right singular vectors are approximately spanned by the first few harmonics. 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.

Chang's algorithm makes use of the above two properties 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 restricted to the band $[0, 2\pi p/n]$. Owing to the properties of the singular vectors discussed earlier, p is typically not much larger than k. Let H_p denote the matrix comprising the first p columns of H. Then the first k singular values $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_k$ and the corresponding left singular vectors $\tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k$ of XH_p serve as excellent estimates to those of X.

It was shown in [32] that when p is chosen so as to satisfy $\rho(X^T, \mathbf{h}_1, \ldots, \mathbf{h}_p) \geq \mu$, the quantity $\rho(X, \tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k)$ turns out to exceed μ for some $k \leq p$, with $\tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k$ being very good estimates for $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k$, and $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_k$ being very good estimates for $\sigma_1, \ldots, \sigma_k$. The energy recovery ratio $\rho(X, \tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k)$ can be efficiently approximated by $\sum_{i=1}^{k} \tilde{\sigma}_i^2 / \|X\|_F^2$.

The entire algorithm in [32] for the fast computation of a partial SVD of X is 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 $\rho(X^T, \mathbf{h}_1, \ldots, \mathbf{h}_p) > \mu$, 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 $\sqrt{(2/n)}[(1/\sqrt{2})\mathbf{y}_0 \quad \Re\mathbf{y}_1 \quad \Im\mathbf{y}_1 \quad \Re\mathbf{y}_2 \quad \Im\mathbf{y}_2 \cdots]$, where $\mathbf{y}_i$ denotes the ith column of Y, while $\Re$ and $\Im$ give real and imaginary parts of the vector, respectively.
3) With Z_p denoting the first p columns of the matrix $[(1/\sqrt{2})\mathbf{y}_0 \quad \Re\mathbf{y}_1 \quad \Im\mathbf{y}_1 \quad \Re\mathbf{y}_2 \quad \Im\mathbf{y}_2 \cdots]$, compute the SVD $Z_p = \sum_{i=1}^{p} \tilde{\sigma}_i \tilde{\hat{\mathbf{u}}}_i \tilde{\hat{\mathbf{v}}}_i^T$.
4) Return $\tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k$ such that $\rho(X, \tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k) \geq \mu$.

If $p \ll n$, then the total computation required for Chang's algorithm 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 [29] as well, which requires $O(mnk^2)$ flops.

V. Effect of Spatial Reduction on Temporal Properties

Chang's eigendecomposition algorithm reduces X in the temporal dimension and thus is more computationally efficient than directly computing the SVD of correlated images. However, the first two steps in Chang's algorithm, i.e., the calculation of the value p (referred to here as the "true" p) and the computation of the SVD of XH_p, still requires a significant amount of time because the algorithm works with the full spatial resolution of the images. Hence, it is desirable to reduce the images in the spatial dimension first. It was shown in Section III that the low-resolution SVD can be used to approximate high-resolution eigenimages. Hence these properties are used here to modify Chang's algorithm so that its computational efficiency is improved without sacrificing the accuracy of the resulting eigendecomposition.

As shown earlier, the low-resolution right singular vectors can be effectively used to determine approximate high-resolution eigenimages, when images are reduced using any filtering technique. Therefore, in Chang's algorithm, images can be reduced using a low-pass filtering technique and the corresponding right singular vectors can be used to approximate the eigenimages of XH_p. However, a reduced resolution version of an image can also be determined by random sampling of the pixels from X. The following sections present an illustrative example, empirical evaluation, and a mathematical analysis to demonstrate why this random sampling is typically more effective than any low-pass filtering technique.

A. Illustrative Example

This section presents two artificial examples to illustrate the behavior of low-pass filtering as compared to random sampling. For these examples, box filtering was used to simplify the illustration. Fig. 4 shows the two "high-resolution" image sequences with images of size 4×4 each. The corresponding box-filtered image frames (of size 2×2 each) are shown in Fig. 5.

The sequence in Fig. 4(a) was selected to illustrate the worst case for box filtering, while also being the best case for random sampling. It consists of image frames corresponding to a checkerboard with all pixels changing their intensity between the two extreme values from one frame to the next. Hence all four temporal frequencies are required to attain any user-specified reconstruction ratio for this sequence ($p = 4$). However,

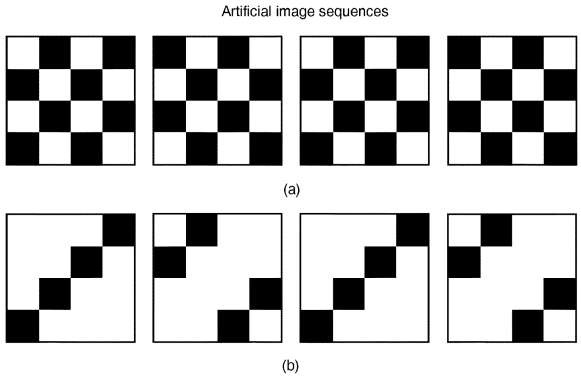


Fig. 4. Two artificial image sequences are used to compare box filtering with random sampling. (a) Sequence #1. (b) Sequence #2.

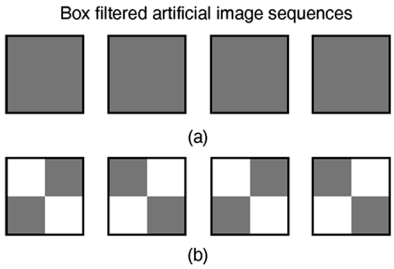


Fig. 5. This figure shows the box filtered image sequences in Fig. 4 with a reduction factor of 2 in both directions. (a) Sequence #1. (b) Sequence 2.

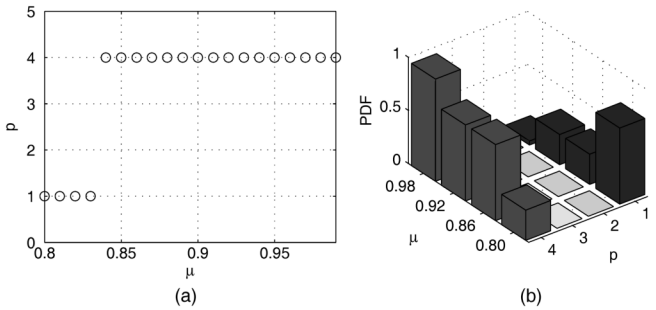


Fig. 6. Results for the video sequence shown in Fig. 4(b). Part (a) plots the true p values against μ, while part (b) plots the values of p_r for different values of μ for all $\binom{16}{4} = 1820$ realizations using random sampling (the value of p_b for the corresponding box filtered sequence in Fig. 5(b) always remains 4 for all values of μ).

for the corresponding box-filtered sequence in Fig. 5(a), only the zero frequency is required to attain any reconstruction ratio $(p_b = 1)$,[3] thus giving the worst possible approximation of the original temporal properties. Now consider reducing the original sequence by randomly sampling 4 out of 16 pixels in all the images. Because the same permutation of four pixels is used over all four image frames, the reduced sequence will always give $p_r = 4$.

Now consider the sequence in Fig. 4(b) which was selected to illustrate what is arguably the best case for box filtering. Fig. 6(a) shows the corresponding values of the true p when μ is varied from 0.8 to 0.99 in steps of 0.01. It is easy to see that the corresponding box-filtered image frames in Fig. 5(b)

[3]The values of p for box-filtered, Gaussian-filtered and randomly sampled sequences are referred to as p_b, p_g, and p_r, respectively.

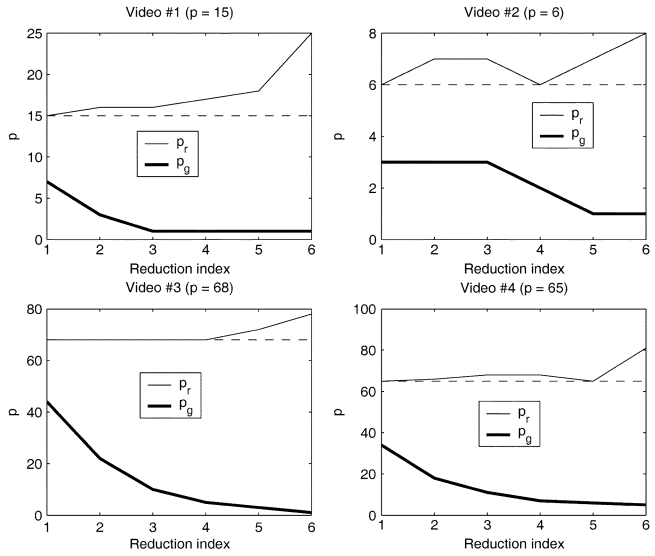


Fig. 7. Plots for the first four video sequences in Fig. 1. The image data matrices at different resolutions are formed after reducing the original images from $m = 240 \times 352$ to the sizes corresponding to indices (1) through (6) given in Table I. Each subplot title gives the video number and its corresponding p value (plotted with a horizontal dashed line) at the highest resolution, where p is the smallest number of frequency harmonics required to obtain $\rho > 0.95$ in (2). The plots p_r and p_g give the p values at the lower resolutions when the images are reduced using random sampling and Gaussian filtering, respectively. For the random sampling technique, the images are reduced four different times to calculate four different p values and the maximum p value is assigned to p_r.

give $p_b = 4$ for any reconstruction ratio. Now consider reducing this sequence using random sampling. There are $\binom{16}{4} = 1820$ such realizations and the corresponding p_r values are plotted in Fig. 6(b). It is easy to see that the median p_r equals the true p for all values of μ. Thus random sampling performs very well even in this case. As the following section will show, this is true in general, even when applied to real video sequences.

B. Empirical Results

The empirical results shown in Fig. 7 for arbitrary video sequences depict similar behavior to those seen in the illustrative example. For all the video sequences, the p_g values decrease rapidly even for small reduction factors, indicating that Gaussian 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 always above the true p values for all the video sequences, indicating that random sampling does not significantly alter the temporal properties of high-resolution images. The mathematical analysis given in the following section explains why spatial reduction of image frames using any low-pass filtering technique will have an undesired effect on the temporal frequencies of the original video sequence.

C. Mathematical Analysis

Objects in a video sequence typically experience relatively simple motions, e.g., translations, rotations, scaling, etc. Consequently, low-pass filtering the images of a video sequence tends to attenuate temporal as well as spatial high frequency information content. This observation will be illustrated with a

mathematical example of a video sequence of an object experiencing a constant linear motion. For simplicity, assume that a video sequence consists of a single object in a black background moving in the negative x direction. To further simplify the equations, assume without loss of generality that the object is moving with a velocity of one pixel per frame and that the video runs at a rate of one frame per second. If the object is represented by a function $f(x,y)$, the image at time t is given by $f(x,y,t) = f(x+t,y)$. The three-dimensional (3–D) discrete Fourier transform (DFT) of the video sequence $f(x,y,t)$ can be determined directly from the two-dimensional (2–D) DFT representation of $f(x,y)$, which is given by

$$f(x,y) = \frac{1}{N_x N_y} \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} F(k_x,k_y) e^{j2\pi\left(x\frac{k_x}{N_x}+y\frac{k_y}{N_y}\right)} \tag{16}$$

where N_x and N_y represent the numbers of rows and columns in an image, respectively and the $F(k_x,k_y)$ terms are the corresponding Fourier coefficients. More specifically

$$\begin{aligned} f(x,y,t) &= \frac{1}{N_x N_y} \\ &\quad \times \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} F(k_x,k_y) e^{j2\pi\left((x+t)\frac{k_x}{N_x}+y\frac{k_y}{N_y}\right)} \\ &= \frac{1}{N_x N_y N_t} \\ &\quad \times \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \sum_{k_t=0}^{N_t-1} N_t F(k_x,k_y)\, \delta_{k_x k_t} e^{j2\pi\left(x\frac{k_x}{N_x}+y\frac{k_y}{N_y}+t\frac{k_t}{N_t}\right)} \end{aligned} \tag{17}$$

where the number of frames $N_t = N_x$ and δ_{jk} is the Kronecker delta function with the property that $a_j = \sum_{k=1}^{n} \delta_{jk} a_k$. Thus the 3-D DFT representation of the video sequence is

$$\begin{aligned} f(x,y,t) = \frac{1}{N_x N_y N_t} &\sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \sum_{k_t=0}^{N_t-1} \\ &F(k_x,k_y,k_t) e^{j2\pi\left(x\frac{k_x}{N_x}+y\frac{k_y}{N_y}+t\frac{k_t}{N_t}\right)} \end{aligned} \tag{18}$$

where

$$F(k_x,k_y,k_t) = \begin{cases} N_t F(k_x,k_y), & \text{if } k_t = k_x \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Applying an ideal low-pass spatial filter to eliminate higher spatial frequencies in the individual frames, (18) becomes

$$\begin{aligned} \tilde{f}(x,y,t) &= \frac{1}{N_x N_y N_t} \sum_{k_x=0}^{N_1} \sum_{k_y=0}^{N_2} \sum_{k_t=0}^{N_t-1} \\ &\quad F(k_x,k_y,k_t) e^{j2\pi\left(x\frac{k_x}{N_x}+y\frac{k_y}{N_t}+t\frac{k_t}{N_t}\right)} \\ &= \frac{1}{N_x N_y} \sum_{k_x=0}^{N_1} \sum_{k_y=0}^{N_2} F(k_x,k_y) e^{j2\pi\left(x\frac{k_x}{N_x}+y\frac{k_y}{N_x}+t\frac{k_x}{N_t}\right)} \end{aligned} \tag{20}$$

where N_1 and N_2 are smaller than $N_x - 1, N_y - 1$, and $N_t - 1$. Note that (19) was used to reduce the number of summations from three to two in the representation of $\tilde{f}(x,y,t)$ so that the label k_t was replaced with k_x. From (20), it is clear that not only are the higher spatial frequency components corresponding to $k_x > N_1$ and $k_y > N_2$ in the individual frames lost, but so are the higher temporal frequency components $k_t > N_1$ in the video sequence. This is because the linear motion coupled the temporal and spatial information as exhibited in this case by the $\delta_{k_x k_t}$ term in (17) [cf., (19)]. This argument also holds for video sequences of an object experiencing an arbitrary constant linear motion. Similar results hold for objects experiencing planar rotations or scaling, although the equations are somewhat more complicated. This inherent loss of important high frequency temporal information makes low-pass filtering an undesirable option for reducing the resolution of a video sequence. Random sampling on the other hand does not suffer from this phenomenon as it does not tend to selectively eliminate temporal frequencies. This motivates a modified version of Chang's algorithm, which is the topic of the next section.

VI. Fast Eigenspace Decomposition

The objective here is to determine the first few left singular vectors of X [37]. Using the analysis of the resolution reduction techniques in the previous section, one can now make the appropriate modifications to Chang's algorithm to improve its computational efficiency. Random sampling is used to reduce X in the spatial dimension and then Chang's method is used to reduce it further in the temporal dimension. An overview of the algorithm is presented first with the details following.

1) Generate the matrix $H_{(n\times n)}$ for $X_{(m\times n)}$.
2) Randomly sample n pixels from each image in X to obtain the $n \times n$ reduced image data matrix X_r.[4]
3) Determine the smallest number p such that

$$\rho\left(X_r^T, \mathbf{h}_1, \ldots, \mathbf{h}_p\right) = \frac{\sum_{i=1}^{p} \|X_r \mathbf{h}_i\|_2^2}{\|X_r\|_F^2} > \mu \tag{21}$$

where μ is the userspecified reconstruction ratio.

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

4) Compute the *thin*[5] SVD of

$$(X_r H_p)^{6}_{(n\times p)} = (U_r)_{(n\times p)}(S_r)_{(p\times p)}(V_r)^T_{(p\times p)}.$$

5) Repeat Steps 2)–4) for three more times and concatenate all $S_r V_r^T$ matrices to form

$$A^T_{(P\times s)} = \begin{bmatrix} S_{r_1}V^T_{r_1} \\ S_{r_2}V^T_{r_2} \\ S_{r_3}V^T_{r_3} \\ S_{r_4}V^T_{r_4} \end{bmatrix}$$

where s is the maximum of the four values of p and P is the sum of all values of p.[7]

6) Compute the SVD of

$$A_{(s\times P)} = (U_s)_{(s\times s)}(S_s)_{(s\times P)}(V_s)^T_{(P\times P)}.$$

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.[8]

8) Perform Steps 2) and 3). If $p > s$, perform Step 4) and compute $Z^{\text{new}} = (H_p)(V_r)$. Update Z using:

```
for i = 1, 2, ..., p
    w = [I − ZZ^T] ẑ_i^new;
    if ||w|| > ε
        Z = [Z, ŵ];
        s = s + 1;
    end
end
```

where I is an $n \times n$ identity matrix and ϵ is a user-specified threshold.

9) Repeat Step 8) until $p \leq s$ for four consecutive times.

10) Compute $\tilde{U}_{(m\times s)} = X_{(m\times n)}Z_{(n\times s)}$ that gives an approximate basis for the left singular vectors of X.

11) Find the orthonormal basis for $\tilde{U}_{(m\times s)}$ using the *thin* QR decomposition, i.e., $\tilde{U}_{(m\times s)} = \hat{\tilde{U}}_{(m\times s)}R_{(s\times s)}$ and return $\hat{\tilde{U}}_{(m\times s)}$.

The above steps will now be explained in more detail. Steps 2)-4) compute the p value for X_r and the SVD of the $X_r H_p$ matrix. Here, it is described 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 [32]; therefore, more than n pixels in each column in the reduced image data matrix are never needed to preserve the rank of X_r 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. However, if n is large, then one may want to incrementally determine an appropriate number of rows for X_r.

Once the SVD of $X_r H_p$ is calculated, $(H_p)(V_r)$ gives the right singular vectors of X_r. These right singular vectors can be considered a "good" approximation of their high-resolution counterparts [35]. However, different X_rs require different numbers of harmonics to satisfy the user-specified reconstruction ratio, because the random pixels used to create a specific X_r may not accurately represent the temporal properties of the entire X. Hence, Steps 2)–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.[9]

Step 5) concatenates all $S_r V_r^T$ matrices to form the matrix A whose range will approximately span the dominant right singular vectors of X [35]. 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_rs are used instead of $S_r V_r$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_r are scaled by their corresponding singular values before being concatenated in Step 5).

Fig. 7 shows that even after performing Steps 2) and 3) four different times, the maximum p values for videos 3 and 4 may fall 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 $\hat{\mathbf{z}}_i^{\text{new}}$ 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.[10] In short, Steps 2)–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. Optionally, one can also compute the minimum subspace that will satisfy the user-specified reconstruction ratio by checking if $k < s$ such that $\rho(X, \hat{\tilde{\mathbf{u}}}_1, \ldots, \hat{\tilde{\mathbf{u}}}_k) > \mu$.

The computational expense of the proposed algorithm is now briefly analyzed. The cost incurred in Step 2), i.e., constructing X_r from X requires 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 $X_r H_p$ requires $O(np^2)$ flops. Step 5) performs Steps 2)–4) four times. 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 ns^2 flops. Steps 8) and 9) that check if any new information should be added to Z require $(n^2 + n^2 p + np^2)$ flops and are repeated an unknown, but typically small, number of times. In Step 10), multiplication of X with Z requires ms^2 flops and the QR decomposition of

[5]A *thin* SVD refers to a decomposition that returns only the first n left singular vectors for an $m \times n$ image data matrix.

[6]$X_r H_p$ is readily available after Step 3).

[7]All $S_r V_r^T$ matrices should be padded with the appropriate number of columns of zeros if necessary, so that each of them has s columns.

[8]H_s consists of the first s columns of H in (14).

[9]Using more than four iterations may be unnecessary and using fewer than four may require more iterations of the more computationally expensive Step 8).

[10]The 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.

TABLE II
TIME REQUIRED FOR THE PROPOSED ALGORITHM (ALL TIMES ARE IN SECONDS AVERAGED OVER TEN TRIALS)

		Time required for different steps					
Video	s	$2-5$	$6,7$	$8,9$	10	11	Total
1	17.1	0.08	0.00	0.07	0.68	1.34	2.18
2	6.9	0.05	0.00	0.05	0.44	0.27	0.81
3	70.5	0.33	0.21	0.24	2.33	19.40	22.51
4	71.9	0.31	0.18	0.22	2.35	20.04	23.11
5	7.7	0.05	0.00	0.04	0.40	0.31	0.80
6	2.3	0.04	0.00	0.04	0.29	0.05	0.41
7	12.2	0.06	0.00	0.05	0.50	0.67	1.28
8	7.7	0.05	0.00	0.05	0.40	0.31	0.80
9	47.9	0.18	0.02	0.14	0.31	2.00	2.65
10	17.4	0.07	0.00	0.06	0.15	0.29	0.57
11	5.9	0.05	0.00	0.04	0.09	0.04	0.22
12	16.7	0.07	0.00	0.06	0.15	0.27	0.54
13	3.0	0.04	0.00	0.04	0.07	0.01	0.16
14	11.2	0.06	0.00	0.05	0.40	0.51	1.02
15	19.1	0.08	0.00	0.07	0.55	1.32	2.03
16	8.0	0.05	0.00	0.05	0.20	0.18	0.48
17	8.4	0.05	0.00	0.05	0.31	0.26	0.67
18	1.0	0.04	0.00	0.03	0.10	0.00	0.17
Avg. percentage		10.63	0.22	8.90	36.66	43.58	100

TABLE III
COMPARISON OF DIFFERENT ALGORITHMS

	Speedup factor of proposed algorithm as compared to		Required subspace dimension (k) for different algorithms		
Video	Chang	Matlab	Proposed	Chang	Matlab (k^*)
1	6.73	34.92	15.0	15	15
2	15.52	88.78	4.0	4	4
3	2.55	3.22	66.1	66	63
4	2.04	3.13	63.0	63	60
5	15.31	80.53	4.0	4	4
6	28.69	155.01	1.0	1	1
7	10.27	50.93	10.0	10	9
8	15.38	79.71	5.0	5	5
9	2.30	5.81	39.2	39	36
10	5.27	26.74	10.0	10	9
11	11.49	71.01	4.0	4	4
12	5.25	28.53	5.0	5	4
13	14.91	95.48	2.0	2	2
14	4.47	54.86	8.0	8	7
15	3.16	27.60	15.0	15	15
16	5.24	73.66	5.2	5	5
17	11.40	70.16	5.0	5	5
18	25.14	139.82	1.0	1	1
Avg	10.28	60.55			

$\tilde{U}$ in Step 11) requires $O(ms^2)$ flops. If $s \ll n \ll m$, then the total computation required is $O(ms^2)$, which is essentially the cost of the QR decomposition.

VII. EXPERIMENTAL RESULTS

The proposed eigendecomposition algorithm was evaluated using all 18 video sequences in Fig 1. In particular, the algorithm was used to calculate the partial SVD of X for each set, with $\mu = 0.95$ and $\epsilon = 10^{-6}$. Table II shows a breakdown of the average time required for the different steps in the proposed algorithm over ten trials for each video sequence. The last row shows the average percentage over all 18 video sequences, i.e., over 180 cases. It shows that in the typical cases, when $s \ll n$, the algorithm is computationally very efficient with Steps 10) and 11) sharing the major computational burden. However, as the value of s increases, the computational time required for Step 11) that computes the orthonormal basis for $\tilde{U}_{(m\times s)}$ using QR decomposition dominates the first ten steps combined. This agrees with the computational expense analysis of the proposed algorithm provided in the previous section. The total average time required for the algorithm is given in the column labeled "Total."

Table III summarizes the performance of the proposed algorithm, showing k^*, k, and the speedup factors for the video sequences. Compared to the direct SVD, the speedup factors with the proposed algorithm are in the range of 3.13–155.01. The original image frames for these video sequences do not have the same resolution, hence the speedup factor depends both on the value of s and the original image resolution. However, one can observe a strong correlation between the value of s and the speedup factor. This is evident from the table results, as the minimum speedup factor is obtained for video 3. Hence, in the typical cases when $s \ll n$, the speedup factors will generally be more than 25.

The difference between $\rho(X, \hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{k^*})$ and $\rho(X, \tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k)$ for each set was less than 0.13%, with an average of 0.07%, which reveals that $\{\tilde{\hat{\mathbf{u}}}_1, \ldots, \tilde{\hat{\mathbf{u}}}_k\}$ provides a very good approximate basis for the first k^* eigenimages $\{\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{k^*}\}$. Table III also shows that the average k values obtained with the proposed algorithm are virtually the same as those obtained with Chang's algorithm, while the speedup factors are in the range of 2.04–28.69. Thus, the proposed algorithm results in almost the same subspace with much better computational efficiency than Chang's algorithm.

The quality of the resulting eigendecomposition was also evaluated using the error measures described in Section II-B with all video sequences. The first video sequence is used as a representative example, the results of which are shown in Fig. 8. These results reveal that the subspaces obtained using the proposed algorithm are as good as those obtained using Chang's algorithm when a k-dimensional eigenspace is used.

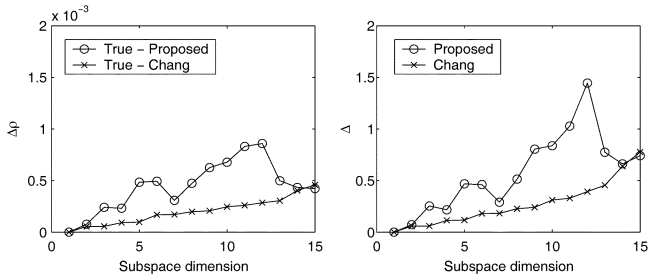


Fig. 8. Comparison of the approximated SVDs with the true SVD for video #1 from Fig 1. The plots with ∘ and × give the results of the proposed and Chang's algorithm, respectively. The plot on the left shows the difference between the energy recovery ratio using true and approximated eigenimages, while the plot on the right shows the residue between the subspaces consisting of true and approximated eigenimages, when the subspace dimension is varied from 1 to k.

In particular, the residue between subspaces show that both the algorithms give nearly the same quality eigenspaces when the same dimension of k is used. The difference between the energy recovery ratio $\Delta\rho$ plots, on the other hand, reveal that both the algorithms give virtually perfect image reconstruction for the k-dimensional subspaces used.

VIII. Conclusion

In this paper, a framework was presented for quantifying the tradeoff associated with performing eigendecomposition of correlated images at lower resolutions in order to mediate the high computational expense of performing these calculations at high resolutions. While doing so, it was shown that the low-resolution right singular vectors can be effectively used to approximate the high-resolution eigenimages without introducing significant error. It was also shown that the low-resolution right singular vectors can be accurately approximated by using a random sampling of pixels from the high-resolution videos. Finally, the similarity between the right singular vectors of correlated images at different resolutions was used to improve the computational efficiency of one of the fastest known eigenspace decomposition algorithms. Examples showed that the modified algorithm performed very well on arbitrary video sequences.

References

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

[2] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, 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, Mar. 1987.

[4] M. Kirby and L. Sirovich, "Application of the Karhunen–Loeve procedure for the characterization of human faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, 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, Mar. 1991.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[7] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, 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 Computer Society Conf. Computer Vision Pattern Recognition*, Seattle, WA, 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 Anal. Mach. Intell.*, 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 Process.*, 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 Anal. Mach. Intell.*, 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 Computer Society Conf. Computer Vision and Pattern Recognition*, San Juan, PR, 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 Computer Society Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, Jun. 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, SC, Jun. 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, TX, Nov. 11–12, 1994, pp. 83–88.

[17] J. Winkeler, B. S. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition," in *Proc. IEEE Comp. Society Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, Jun. 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. Robotics and Automation*, San Diego, CA, May. 8–13, 1994, pp. 3237–3246.

[19] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vis.*, 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. Comput. Vis.*, vol. 14, no. 1, pp. 5–24, Jan. 1995.

[21] ——, "Detection of 3D objects in cluttered scenes using hierarchical eigenspace," *Pattern Recognit. Lett.*, vol. 18, no. 4, pp. 375–384, Apr. 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 Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.

[24] G. W. Stewart, *Introduction to Matrix Computation*. New York: Academic, 1973.

[25] S. Shlien, "A method for computing the partial singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 4, no. 6, pp. 671–676, Nov. 1982.

[26] R. Haimi-Cohen and A. Cohen, "Gradient-type algorithms for partial singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 1, pp. 137–142, Jan. 1987.

[27] 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 Process.*, vol. 37, no. 10, pp. 1550–1556, Oct. 1989.

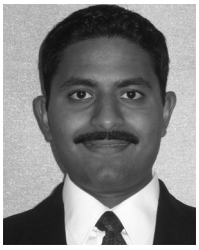
[28] C. R. Vogel and J. G. Wade, "Iterative SVD-based methods for ill-posed problems," *SIAM J. Sci. Comput.*, vol. 15, no. 3, pp. 736–754, May 1994.

[29] H. Murakami and V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 4, no. 5, pp. 511–515, Sep. 1982.

[30] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler, and H. Zhang, "An eigenspace update algorithm for image analysis," *CVGIP: Graphic Models and Image Process.*, vol. 59, no. 5, pp. 321–332, Sep. 1997.

[31] H. Murase and M. Lindenbaum, "Partial eigenvalue decomposition of large images using the spatial temporal adaptive method," *IEEE Trans. Image Process.*, vol. 4, no. 5, pp. 620–629, May 1995.

[32] C. Y. Chang, A. A. Maciejewski, and V. Balakrishnan, "Fast eigenspace decomposition of correlated images," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1937–1949, Nov. 2000.
[33] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins, 1996.
[34] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Philadelphia, PA: Soc. Ind. Appl. Math., 1997.
[35] 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 Robotics and Automation (ICRA)*, New Orleans, LA, Apr. 26-May 1, 2004, pp. 1393–1398.
[36] P. J. Davis, *Circulant Matrices*. New York: Wiley, 1979.
[37] K. Saitwal, A. A. Maciejewski, and R. G. Roberts, "Fast eigenspace decomposition of correlated images using their lowresolution properties," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, Sep. 28–Oct. 2, 2004, pp. 2707–2712.



Kishor Saitwal (S'01) was born in Pune, India, on Jan. 18, 1977. He received the B.E. degree in instrumentation and controls from Vishwakarma Institute of Technology, Pune University, in 1998. He received the National Talent Search scholarship and was ranked in the top 2% in the University. He received the M.S. and Ph.D. degrees from the Electrical and Computer Engineering Department, Colorado State University, Fort Collins, in 2001 and 2006, respectively.

He is currently with Behavioral Recognition Systems, Inc., performing research in video surveillance. His research interests include image/video processing, computer vision, and computer graphics.

Anthony A. Maciejewski (M'87–SM'00–F'05) received the B.S.E.E., M.S., and Ph.D. degrees from The Ohio State University, Columbus, in 1982, 1984, and 1987, respectively.

From 1988 to 2001, he was a Professor of electrical and computer engineering at Purdue University, West Lafayette, IN. He is currently the Department Head of Electrical and Computer Engineering, Colorado State University, Fort Collins.

Rodney G. Roberts (M'92–SM'02) received the B.S. degrees in electrical engineering and mathematics from the Rose-Hulman Institute of Technology, Terre Haute, IN, in 1987 and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1988 and 1992, respectively.

From 1992 to 1994, he was a National Research Council Fellow at Wright Patterson Air Force Base, Dayton, OH. Since 1994, he has been with Florida A & M University–Florida State University College of Engineering, Tallahassee, where he is currently a Professor of electrical and computer engineering. His research interests are in the areas of robotics and image processing.

Bruce A. Draper received the B.S. degree in computer science from Yale University, New Haven, CT, in 1984, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts, Amherst, in 1987 and 1993, respectively.

From 1993 to 1996, he served as a Senior Postdoctoral Scientist in the Computer Vision Laboratory, University of Massachusetts. He is currently an Associate Professor of Computer Science at Colorado State University, Fort Collins. His research interests are in biomimetic object recognition, including color vision, and the control of computer vision.

Dr. Draper was General Co-Chair of the 1999 IEEE Conference on Computer Vision and Pattern Recognition and is the Program Committee Co-Chair for the 2006 International Conference on Vision Systems. He has previously served as an Area Chair for the International Conference on Machine Learning (three times).