2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

The Effect of Spatial Resolution Reduction Techniques on the Temporal Properties of Video Sequences[*]

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*— Singular value decomposition (SVD) is a common technique that is performed on video sequences in a number of computer vision and robotics applications. The left singular vectors represent the eigenimages, while the right singular vectors represent the temporal properties of the video sequence. It is obvious that spatial reduction techniques affect the left singular vectors, however, the extent of their effect on the right singular vectors is not clear. Understanding how the right singular vectors are affected is important because many SVD algorithms rely on computing them as an intermediate step to computing the eigenimages. The work presented here quantifies the effects of different spatial resolution reduction techniques on the right singular vectors that are computed from those video sequences. Examples show that using random sampling for spatial resolution reduction rather than a low-pass filtering technique results in less perturbation of the temporal properties.**

***Index Terms*— Singular value decomposition, right singular vectors, spatial resolution reduction, temporal properties.**

I. Introduction

Eigendecomposition-based techniques play an important role in numerous image processing and computer vision applications. The advantage of these techniques, also referred to as subspace methods, is that they are purely appearance based and that they require few online computations. Variously referred to as eigenspace methods, singular value decomposition (SVD) methods, principal component analysis methods, and Karhunun-Loeve transformation methods [1], they have been used extensively in a variety of applications such as robot vision [2]–[4], robot control [5], [6], face characterization [7], [8] and recognition [9]–[13], lip-reading [14], [15], object recognition [16]–[19], pose detection [20], [21], visual tracking [22], [23], and inspection [24]–[27]. All of these applications are based on taking advantage of the fact that a set of highly correlated image frames in a video sequence is typically spanned by the first few temporal frequencies [28]. Once the set of these frequencies is determined, the corresponding principal eigenimages can be determined and online computation using these eigenimages can be performed very efficiently. However, the offline calculation required to determine the appropriate number of frequencies can be prohibitively expensive.

The spatial resolution of the video frames, in terms of the number of pixels, is one of the factors that greatly affects the amount of offline calculation required to compute an eigendecomposition. In particular, many common algorithms that compute the complete SVD of a general matrix require on the order of mn^2 flops, where m is the total number of pixels in a single frame and n is the number of frames. Most users of eigendecomposition techniques would like to use as large a resolution as is available for the original image frames 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 image frames 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 spatial resolution reduction techniques affect the resulting eigendecomposition, with a particular emphasis on the temporal properties represented by the right singular vectors.

The paper is organized as follows. In Section II, the fundamentals of applying eigendecomposition to video sequences are explained. This section also explains how closely the right singular vectors are associated with the temporal properties of video sequences and how this property can be used to reduce the computational cost of the SVD. The effect of different spatial resolution reduction techniques on these right singular vectors is then discussed in Section III using artificially generated video sequences. In Section IV, the performance of these spatial reduction techniques on a set of arbitrary video sequences is evaluated. Lastly, the concluding remarks are given in Section V.

II. Overview

In this work, a grey-scale image frame is described by an $h \times v$ array of square pixels with intensity values normalized between 0 and 1. Thus, an image frame will be represented by a matrix $\mathcal{X} \in [0,1]^{h\times v}$. Because sets of related image frames are considered in this paper, the *image vector* $\mathbf{x}$ of

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

length $m = h \times v$ is obtained by "row-scanning" a frame into a column vector, i.e., $\mathbf{x} = \mathrm{vec}(\mathcal{X}^T)$. The *video matrix* of a set of image frames $\mathcal{X}_1, \cdots, \mathcal{X}_n$ is an $m \times n$ matrix, denoted X, and defined as $X = [\mathbf{x}_1 \cdots \mathbf{x}_n]$.

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

In practice, the singular vectors $\hat{\mathbf{v}}_i$ are not known or computed exactly, and instead estimates $\mathbf{e}_1, \cdots, \mathbf{e}_k$ which form a k-dimensional basis are used. One measure for quantifying the accuracy of a practical implementation of subspace methods that is used in this paper is the "energy recovery ratio" [28], denoted ρ, and defined as

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

where $\|\cdot\|_F$ denotes the Frobenius norm.

The precomputation of the eigenimages of the matrix X can be a computationally expensive operation when m and n are very large. To improve this computational efficiency, many SVD algorithms have been proposed that take advantage of the fact that only the first k eigenimages are of interest. When $m \gg n$, some of the SVD algorithms rely on computing the eigenvectors of $X^T X$ [29], [30], which are the right singular vectors of X. These right singular vectors can then be used to compute the corresponding eigenimages of X. However, the computation of $X^T X$ itself requires mn^2 flops, which can be prohibitively expensive.

To avoid the computation of $X^T X$, Chang et al. [28] proposed an algorithm that was motivated by the fact that if the matrix $X^T X$ is a circulant matrix then the (unordered) SVD of X is known in closed form. In particular, consider a video matrix X where each $\mathbf{x}_{i+1}$ is obtained from $\mathbf{x}_i$ by a planar rotation of $\theta = 2\pi/n$. It is shown in [28] that $X^T X$ is a *circulant matrix* with *circularly symmetric* rows and its eigendecomposition is given by

$$X^T X = H D H^T \tag{3}$$

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

$$D = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_2, \lambda_3, \lambda_3, ...) \tag{4}$$

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

$$H = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & c_0 & -s_0 & c_0 & -s_0 & \cdots \\ \frac{1}{\sqrt{2}} & c_1 & -s_1 & c_2 & -s_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \frac{1}{\sqrt{2}} & c_{n-1} & -s_{n-1} & c_{2(n-1)} & -s_{2(n-1)} & \cdots \end{bmatrix} \tag{5}$$

where $c_k = \cos(k\frac{2\pi}{n})$ and $s_k = \sin(k\frac{2\pi}{n})$. From the definition of the SVD in (1), it can be seen that $V = H$. This analysis shows that

1) the right singular vectors are pure sinusoids of frequencies that are multiples of $2\pi/n$ radians and
2) 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 [28]. Although the analytical expression in (3) does not hold true for arbitrary video sequences, Fig. 1 illustrates a typical case, where the corresponding right singular vectors approximately satisfy the two properties from the planar rotation case. Thus it can be observed that the right singular vectors effectively track the temporal changes in any video sequence.

In a typical eigendecomposition application, a user will specify a desired energy recovery ratio, denoted here as μ. If the columns of H are used as estimates of $\hat{\mathbf{v}}_i$ in (2), then the number of columns required to satisfy a given μ will be denoted by p.[1] If the matrix X has the properties described above, then p (number of required temporal frequencies) will typically be not much larger than the minimum subspace dimension, which is obtained when the estimates are the true right singular vectors. Let H_p denote the matrix comprising the first p columns of H. Then the first few singular values and the corresponding left singular vectors of XH_p serve as excellent estimates to those of X. However, the computation of the true SVD of XH_p can be computationally very expensive due to the highest spatial resolution of the original images in X. Hence it is desirable to downsample the image frames and use the correlation within them to approximate their right singular vectors. These approximated right singular vectors can then be used to compute the corresponding eigenimages of X. An overview of the effect of some of the different spatial resolution reduction techniques that can be used for this purpose, is provided in the next section.

III. Effect of Spatial Reduction Techniques on Temporal Properties

This section studies the effect of different spatial resolution reduction techniques on the temporal properties of the video sequences. These spatial reduction techniques include traditional low-pass filtering, interpolation techniques, the nearest

[1] The value of p computed for the original video matrix X will be referred to here as "true" p.

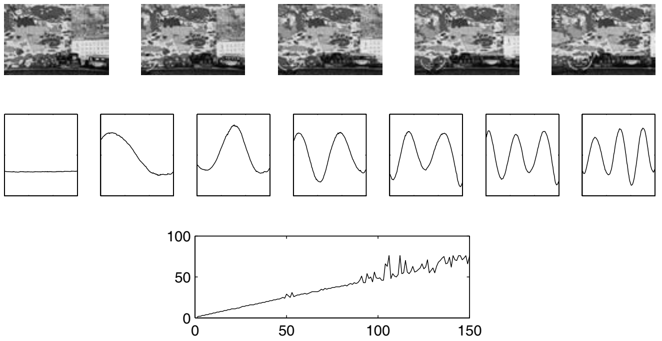

Fig. 1. This figure shows the eigendecomposition of the video matrix X for an arbitrary video sequence. The first row shows five equally spaced frames from the 150 images of the video matrix X. The second row shows the first seven right singular vectors of X, while the plot in the third row shows the frequency at which the power spectra of the corresponding right singular vectors achieves a maximum (i.e., the "dominant" frequencies). It can be seen that the dominant frequencies of the power spectra of the right singular vectors increase approximately linearly with their frequency index.

neighbor method and random sampling. To study the effect of these techniques, two artificially generated sets of video sequences are considered in this section. The first set consists of small image frames to illustrate the fundamental issues involved and the second set consists of larger image frames with the typical spatial properties of real image frames. The simplest low-pass filtering technique, i.e., box filtering, is evaluated with the first set, while Gaussian filtering (which is more commonly applied in image processing applications) is evaluated with the second set. Random sampling is evaluated with both sets of video sequences and it is compared with both the above low-pass filtering techniques.

Fig. 2(a) and 3(a) show two very simple artificial video sequences consisting of 4×4 image frames and four frames each for the purpose of illustration. Consider reducing these image frames down to 2×2 using box filtering and random sampling. The first video sequence in Fig. 2(a) has the 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 (true p = 4). Now consider the box-filtered image frames for this video sequence shown in Fig. 2(b). It is easy to observe that this low-resolution sequence requires only the zero frequency to attain any reconstruction ratio ($p_b = 1$).[2] Hence there is a major loss of information as far as the temporal properties of the first sequence is concerned when the image frames are reduced using box filtering. This is an example of a worst-case sequence for the box filtering approach. Now consider reducing this sequence using random sampling. Because the same permutation of four pixels is used over all four image frames, the reduced video sequence will have the same temporal properties as in the original sequence and will also give $p_r = 4$. Thus the worst-case sequence for

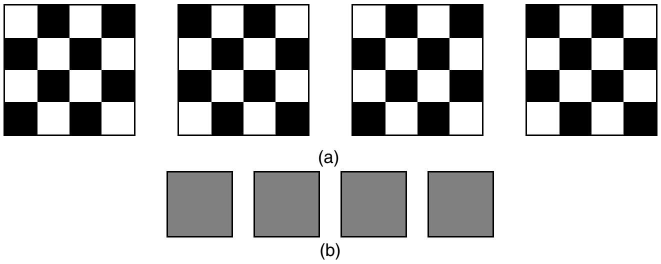

Fig. 2. Part (a) of this figure shows the first artificial video sequence consisting of four frames of size 4×4 each. All the image frames have a checkerboard pattern with each pixel changing its intensity between the two extreme values from one frame to the next. Part (b) shows the image frames of size 2×2, when the original image frames in this sequence are reduced using box filtering.

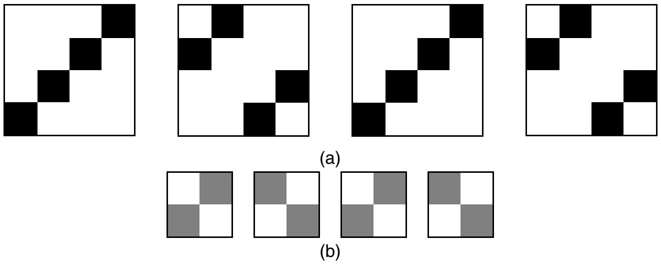

Fig. 3. Part (a) of this figure shows the second artificial video sequence consisting of four frames of size 4×4 each, while part (b) shows the image frames of size 2×2, when the original image frames in this sequence are reduced using box filtering.

box filtering actually turns out to be the best-case sequence for random sampling.

Consider the second artificial video sequence in Fig. 3(a). Fig. 4(a) shows the corresponding values of true p when μ is varied from 0.8 to 0.99 in steps of 0.01. Now consider the box-filtered image frames for this sequence in Fig. 3(b). Because all the pixels in these reduced image frames are changing their intensity from one frame to the next, all four frequencies will be required ($p_b = 4$) to attain any reconstruction ratio. As it overestimates the values of the true p for some values of μ, this sequence is arguably the best-case sequence for box filtering. Now consider reducing this sequence using random sampling, in which any four out of sixteen pixels can be selected for a single realization. There are $\binom{16}{4} = 1820$ such realizations and the corresponding p_r values are plotted in Fig. 4(b). These results show that the median p_r is equal to the true p for all values of μ.

The two artificial example sequences considered in Fig. 2 and 3 do not represent the properties of real video sequences. Hence to generate a video sequence with typical spatial properties, the following procedure was performed:

1) Randomly select different images of the same size.
2) Add these images and intensity normalize the resulting image.
3) Use this normalized image as the first frame in a synthetic video sequence.

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

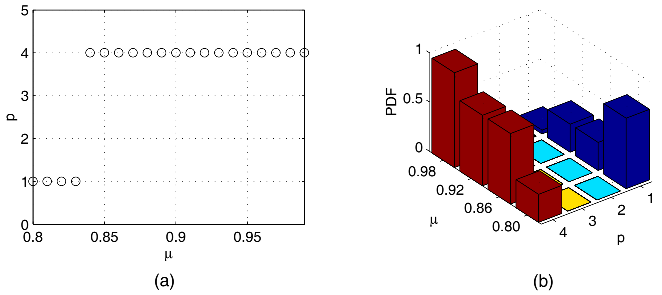

Fig. 4. This figure shows the results for the artificial video sequence shown in Fig. 3(a). Part (a) plots the true p values against μ, while part (b) plots the values of p_r for different values of μ for all $16C_4 = 1820$ realizations using random sampling. (The value of p_b for box filtered sequence in Fig. 3(b) always remains four for all values of μ.)

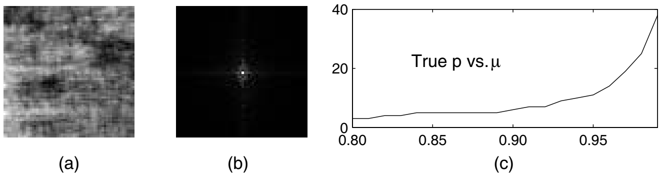

Fig. 5. This figure shows the results for one of the realizations of the synthetically generated video sequences. Part (a) shows the first frame (size 64×64) of the video sequence, while part (b) shows its FFT. An additional 63 image frames are generated by adding Gaussian random noise with zero mean and a standard deviation of one to the previously generated image frames. All the frames are renormalized to have intensity values in the range [-1,1]. The plot in part (c) shows the variation of true p when the reconstruction ratio, μ is varied from 0.8 to 0.99.

4) Add a random perturbation to the current frame and renormalize to generate the next frame.
5) Repeat Step 4 until all the frames are generated in the video sequence.

In particular, 180 different real images of size 64×64 were randomly selected. These images were added together and the resulting image was intensity normalized between -1 and 1. This image was used as the first frame in the video sequence. Steps 4 and 5 in the above procedure were then performed to generate the remaining frames by adding a random perturbation (with fixed statistics) to each previously generated frame. Fig. 5 shows one such realization that is used here as a representative example. The remaining section studies this example in detail to evaluate the effect of different reduction techniques on the temporal properties of the original video sequence. The nearest neighbor method, which can be considered as a special case of the random sampling technique is also evaluated. In this method, instead of random sampling of the pixels, they are selected from pre-specified locations in the original image frames based on the reduction factor.

All low-pass filtering techniques take a weighted average of several pixels, at some stage or the other, to find the value of one pixel in the reduced image frames. Hence the first few temporal frequency components become more dominant than they actually are in the original video sequence and thus a fewer number of frequencies are required to achieve the desired energy recovery ratio. This is evident from Fig. 6, which shows that box filtering and Gaussian filtering both results in a poor approximation of the true p. Hence the loss of important high temporal frequencies makes low pass filtering an undesirable option for spatial reduction. The nearest neighbor method, on the other hand, directly selects the pixels out of the original image frames and hence does not significantly change the temporal properties of the original video sequence. This can also be observed from Fig. 6, as the corresponding values of p track the true p very closely.[3] However, pre-specified pixel selection may result in poor performance for some video sequences, which motivates the random sampling technique.

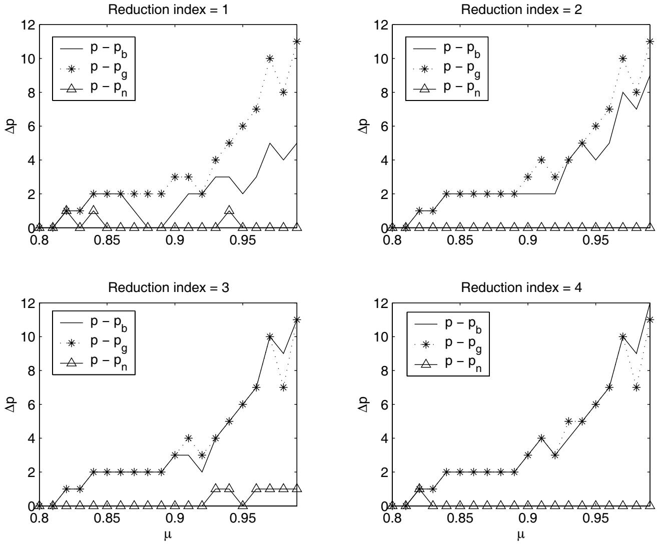

Fig. 6. This figure shows the results for the synthetically generated video sequence in Fig. 5, when the original image frames are reduced using box filtering, Gaussian filtering and nearest neighbor method. Reduction indices 1, 2, 3, and 4 indicate that the resolutions of the reduced frames are 32×32, 16×16, 8×8, and 4×4, respectively. All the plots show the difference between the true p and the values of p for all three reduction techniques, when the user-reconstruction ratio, μ is varied from 0.8 to 0.99. The values p_b, p_g, and p_n indicate the values of p obtained for box filtering, Gaussian filtering, and nearest neighbor method, respectively.

Fig. 7 shows that the random sampling technique tracks the temporal properties of the original video sequence almost perfectly, even with reduced frames of size 4×4, as it does not tend to selectively eliminate temporal frequencies. There are very few realizations out of 1000 that had smaller values of p_r than the true ones for the given values of μ. This property is evident from Fig. 8, which shows that the median of all the values of p_r is always equal to the true value of p.

Thus it can be observed that if the random sampling technique is performed several times, then it is almost guaranteed to result in a value of $p_r > p$. These realizations can be obtained quickly and once the desired low-resolution video sequence is obtained, the SVD can be computed much more efficiently than that for the original video sequence to find an excellent approximation of the high-resolution right singular

[3] p_n (the value of p for nearest neighbor method) is truncated to the true value of p if $p_n > p$ for simplicity of representation.

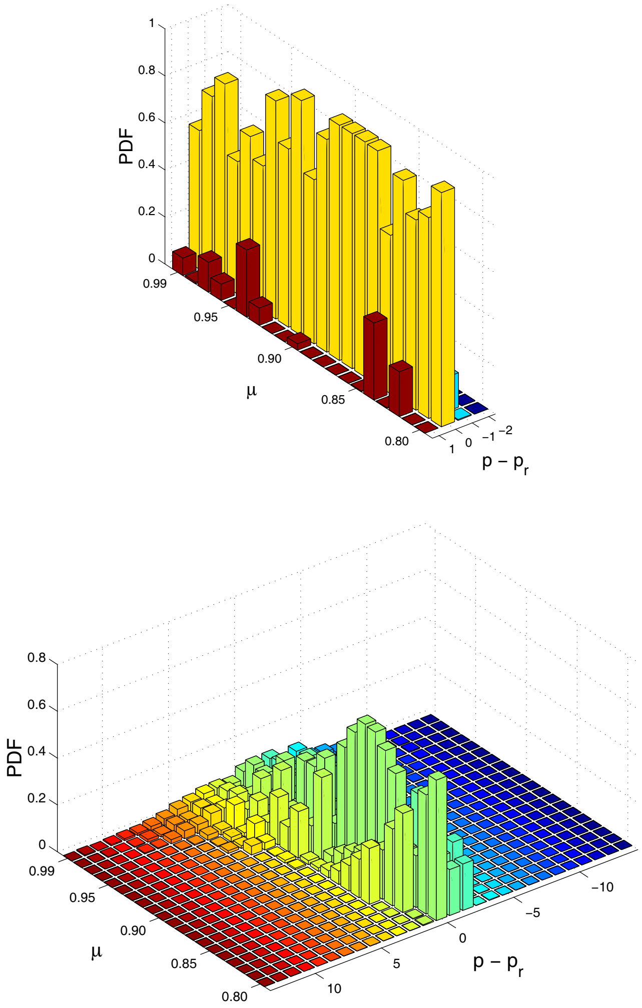

Fig. 7. This figure shows the effect of random sampling on the temporal properties of the original video sequence in terms of the value of p. A total of 1000 different realizations of the reduced video sequences are obtained from the original sequence in Fig. 5. The first graph gives the results for the reduced frames with 1024 (32×32) pixels each, while the second graph gives the results for the reduced frames with 16 (4×4) pixels each. The variable p_r denotes the value of p obtained for one of the realizations of the reduced video sequence. The bar graphs are plotted as the probability distribution functions (PDF's) for $p - p_r$ for different values of μ.

vectors. The next section validates this property by evaluating all the reduction techniques using real video sequences.

IV. Experimental Results

The problem of computing the right singular vectors of images representing successive frames of arbitrary video sequences is considered here. Specifically, 20 video sequences (with $n = 150$) that are used in [28] are evaluated for this purpose. For $\mu = 0.95$, the true p values of these 20 video sequences range from 1 to 68. The two video sequences (7 and 17) with true p values of 68 and 65, respectively, are considered as the representative examples here. Fig. 9 shows that even with the reduction indices 1 and 2,[4] Gaussian filtering gives poor approximation of the temporal properties of the original video sequences. On the other hand, the random sampling technique gives a very good approximation of these temporal properties even when the images are reduced to only 150 out of a total of 84480 pixels.[5]

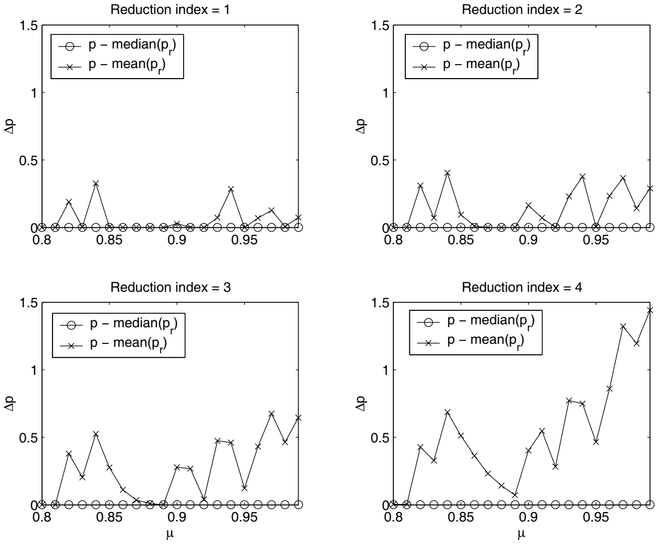

Fig. 8. This figure shows the results for the synthetically generated video sequence in Fig. 5, when the original frames are reduced using random sampling. Reduction indices 1, 2, 3, and 4 indicate that the resolutions of the reduced frames are 32×32, 16×16, 8×8, and 4×4, respectively. All the plots show the difference between the true p and median and mean values of p_r, when the user-reconstruction ratio μ is varied from 0.8 to 0.99.

Because low-pass filtering techniques severely degrade the approximation of high-resolution right singular vectors, they will likewise not produce a good approximation of the high-resolution eigenimages and hence they should not be used to compress the images in the eigenspace. On the other hand, the random sampling technique can be efficiently used to compute the full eigendecomposition as illustrated in [31].

V. Conclusion

Different spatial resolution reduction techniques were evaluated to quantify the error associated with computing the right singular vectors of video sequences at lower resolutions in order to mediate the high computational expense of performing these calculations at high resolutions. It was indicated that none of the traditional low-pass filtering techniques preserved the temporal properties of the original video sequences. The nearest neighbor method outperformed the filtering techniques, but it was shown that the pre-selected pixels may result in errors for certain video sequences. Downsampling using random sampling was shown to be typically much more effective than the other spatial reduction techniques in preserving the

[4] Original images of size 240×352 each are reduced to 120×176 (reduction index = 1) and to 60×88 (reduction index = 2).

[5] It is shown in [28] that $p \leq n$. Hence more than n pixels are not needed in the reduced image frames to preserve the original temporal properties.

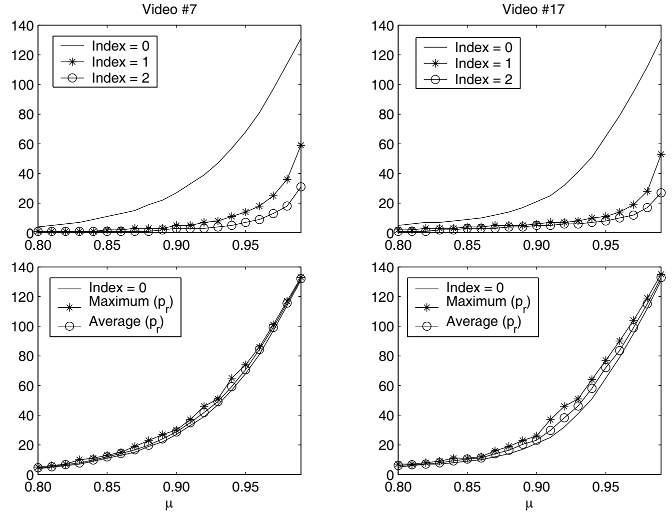

Fig. 9. This figure shows the results for the two video sequences (7 and 17) used in [28]. (The original images in these video sequences are of size 240×352 pixels.) The first row shows the values of p at different resolutions, when the image frames are reduced using Gaussian filtering. (The reduction indices 1 and 2 correspond to the reduced images of size 120×176 and 60×88, respectively.) The second row shows the plots for the values of p for ten different realizations of the reduced frames, when the images are reduced to 150 pixels using random sampling. The solid plot with the legend "Index = 0" gives the true p as μ increases.

temporal properties of the video sequences and hence gave excellent approximations of the high-resolution right singular vectors.

Computing the eigenimages of X using random sampling as illustrated here, is more computationally efficient than computing the eigenvectors of $X^T X$ for several reasons. First, forming $X^T X$ requires an overhead of mn^2 flops. Next, computing the eigenvectors of $X^T X$ is an $O(n^3)$ operation, whereas estimating p is only $O(n^2)$. Finally, multiplying to compute the eigenimages using all the columns of X requires mn^2 flops whereas using our temporally compressed version of X allows us to reduce n significantly.

References

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

[2] J. Barreto, P. Menezes, and J. Dias, "Human-robot interaction based on haar-like features and eigenfaces," in *Proc. IEEE Int. Conf. on Robot. Automat.*, New Orleans, LA, Apr. 26 - May 01 2004, pp. 1888–1893.

[3] C. Belta and V. Kumar, "An svd-based projection method for interpolation on SE(3)," *IEEE Trans. Robot. Automat.*, vol. 18, no. 3, pp. 334–345, June 2002.

[4] Y. Mae, T. Masuda, T. Arai, and K. Inoue, "Error analysis of dead reckoning of multi-legged robots," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, Maui, Hawaii, Oct. 29 - Nov. 03 2001, pp. 1558–1563.

[5] R. G. Roberts and D. W. Repperger, "The kinematics of robotic wrists," in *Proc. IEEE Int. Conf. on Robot. Automat.*, San Diego, CA, May 08 - 13 1994, pp. 3337–3341.

[6] M. V. Kircanski, "Symbolical singular value decomposition for a 7-dof manipulator and its application to robot control," in *Proc. IEEE Int. Conf. on Robot. Automat.*, Atlanta, GA, May 02 - 06 1993, pp. 895–900.

[7] 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.

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

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

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

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

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

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

[14] 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.

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

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

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

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

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

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

[21] J. Winkeler, B. S. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition," in *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognit.*, Fort Collins, CO, June 23-25 1999, pp. 511–516.

[22] 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, May 8-13 1994, pp. 3237–3246.

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

[24] 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.

[25] 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.

[26] 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.

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

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

[29] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[30] 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. ASSP*, vol. 37, no. 10, pp. 1550–1556, Oct. 1989.

[31] K. Saitwal, A. A. Maciejewski, and R. G. Roberts, "Fast eigenspace decomposition of correlated images using their low-resolution properties," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, Sep. 28 - Oct. 02 2004, pp. 2707–2712.