.

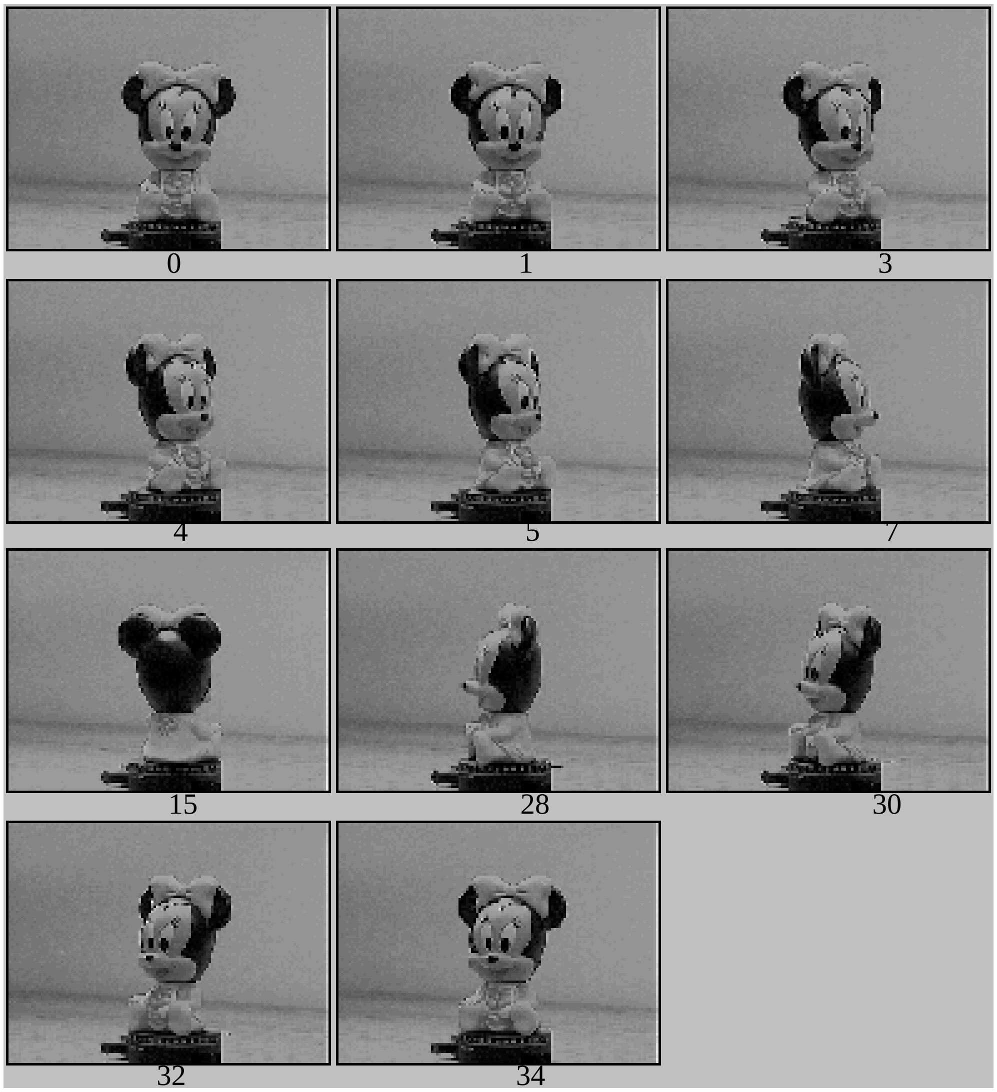

Figure 4: Salient views of TOY1

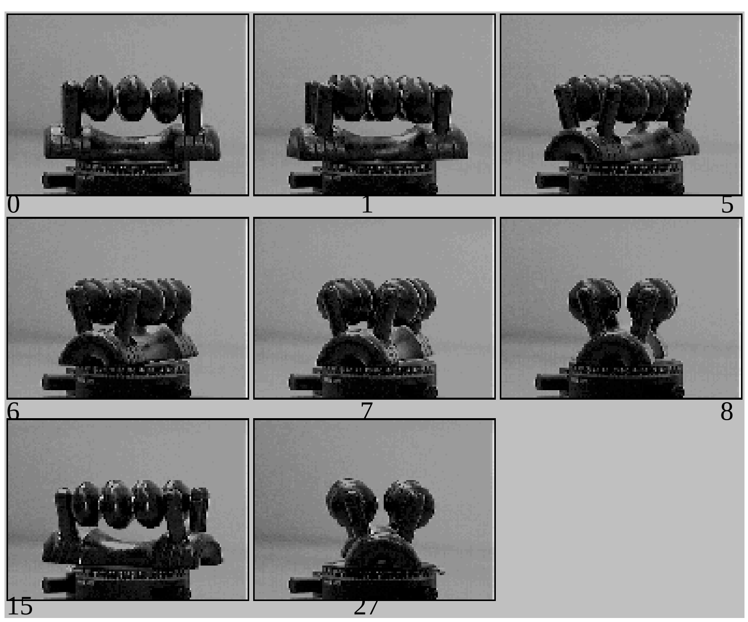

Figure 5: Salient views of TOY5.

rithm for Updating the Singular Value Decomposition," Research Report YALEU/DCS/RR-966, 1994, Yale University, New Haven, CT.
[5] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychology* Vol. 24, pp. 417-441 and pp. 448-520, 1933.
[6] A. Jain, Fundamentals of Digital Image Processing, Prentice Hall 1989.
[7] B. S. Manjunath, S. Chandrasekaran, and Y. F. Wang, "Learning in eigenspace: theory and application," CIPR-TR-94-17, ECE Dept., UCSB, November 1994.
[8] H. Murakami and B. V. K. V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE T-PAMI*, Vol. 4, No. 5, September 1982, pp. 511-515.
[9] H. Murase and S. K. Nayar, "Visual learning of object modules from appearance," *Proc. Image Understanding Workshop 1993*, (San Diego, CA), pp. 547-555, 1993.
[10] H. Murase and M. Lindenbaum, "Partial eigenvalue decomposition of large images using the spatial temporal adaptive method," *IEEE T-IP*, Vol. 4 (5), May 1995, pp. 620-629..
[11] E. Oja, *Subspace methods of pattern recognition*, John Wiley, 1983.
[12] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Proc. IEEE Conf. CVPR '94*, (Seattle, Washington), pp. 84-91, June 1994.
[13] L. Sirovich and M. Kirby, "Low dimensional procedure for the characterization of human faces," J*ournal of Optical Society of America*, Vol. 4, No. 3, pp. 519-524, 1987.
[14] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71- 86, March 1991.
[15] S. Ullman and R. Basri, "Recognition by linear combination of models," IEEE T-PAMI, Vol. 13, No. 10, pp. 992-1006, October 1991.

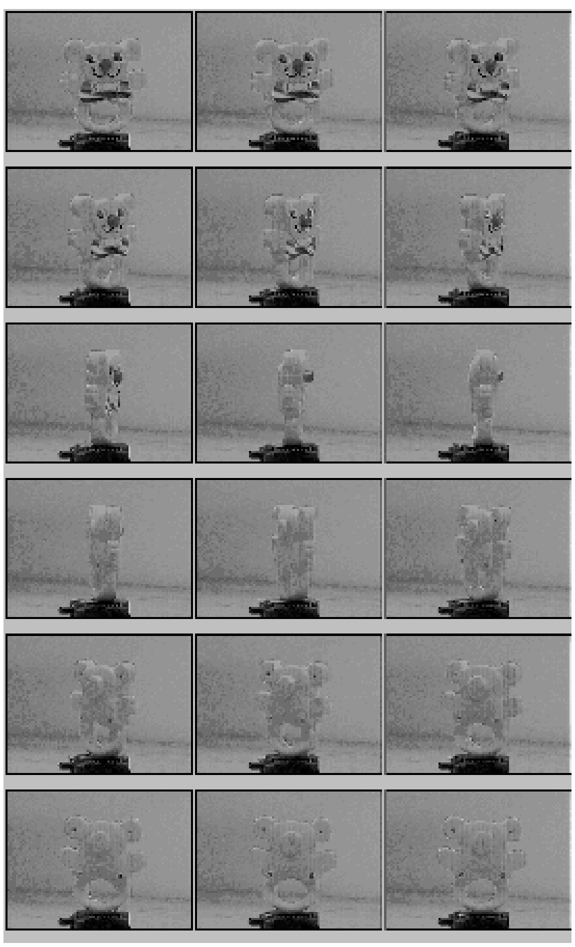

Figure 2: 18 of the 36 views of the TOY2 object.

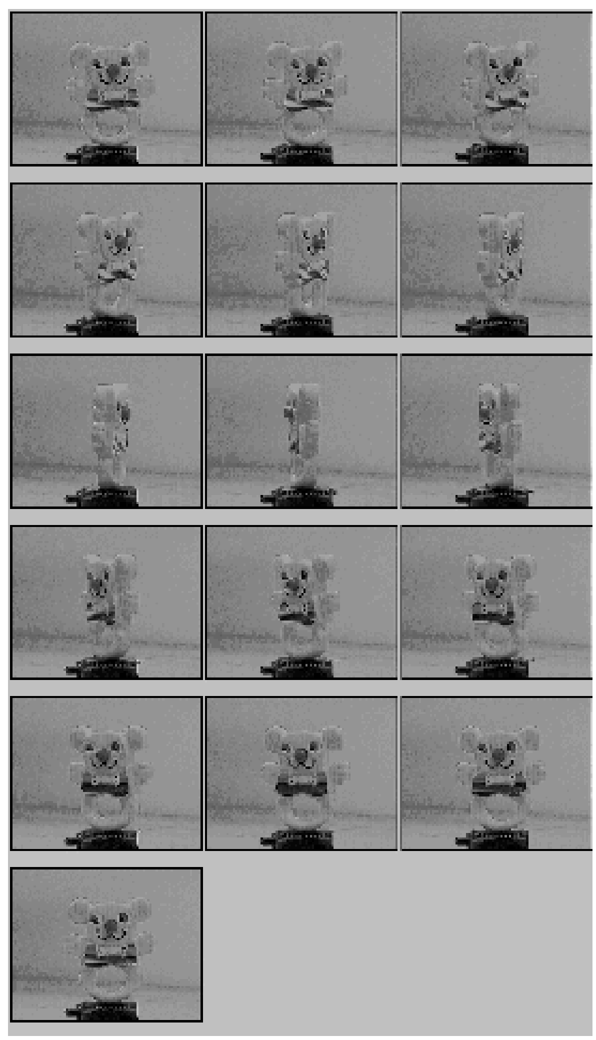

Figure 3: Salient views selected (left-right, top-bottom: 0, 1, 2, 3, 4, 5, 6, 27, 28, 29, 30, 31, 32, 33, 34, 35).

3 Applications

Previous researchers have demonstrated that the eigenspace approach is a powerful tool in recognition and pose estimation of objects from image projections. The objective of our experiments is to illustrate that the proposed learning algorithm selects intuitively meaningful images from the various viewpoints for representing the objects, and thus substantiates our claim that this approach can be used in a dynamic environment.

When a new image is encountered during the learning process, the representation changes under the following three conditions: (a) When the dimensionality of the current basis image space is not sufficient to encode the new image. In this case the number of basis images increases by one; (b) the singular vectors are rotated, and (c) the addition of the new image affects the singular values only. Our experimental studies indicate that when a new representation is required, usually the dimensionality also goes up. This suggests the following simplification in the learning algorithm: check to see if an update is necessary (this can be done by considering the reconstruction error); if necessary, update the representation[1]. The following experiments are conducted using this simplified version implemented in Matlab.

We placed test objects on a rotation stage and took pictures for every $10°$ of rotation. A total of six objects were digitized with 36 images per object. One of the toy objects (TOY2) and the first 18 views (0 - 180 degree) of it are shown in Figure 2. Eigenspace representations are constructed for all six objects by examining the 36 images one-by-one. Only those images that provided significant new information are used in constructing a new, larger representation (see Figure 3). We call these images as salient views of the corresponding object. Figure 4 and Figure 5 show the salient views of two other objects. In both these cases, the number of salient views is about one third the total number of views. Increasing the number of views (acquiring them at a finer angular spacing) does not increase the number of salient views. In our implementation, as new images are acquired, the representation is updated only if the error in reconstruction exceeded the threshold δ (see Figure 1). A good choice of δ appears to be in the range 0.06 - 0.1 per pixel for the case of zero-mean images. We subtracted the running average of the images to approximate zero-mean images. Subtracting the mean simplifies the choice of δ for a wide range of images, but is not necessary either for the analysis of the algorithm or for its implementation.

We observe that the number of salient views is essentially invariant to initial view-point. Our current experiments focus on active recognition wherein the camera acquires images in an adaptive way. The incremental positioning of the camera is adaptively controlled by considering the changes in the eigenspace representation. The system starts acquiring objects initially at one degree rotation. If the new image is close to the current eigenspace representation, the step-size for the rotation is doubled. On the other hand, if the distance of the current image from the eigenspace representation is large, then the step-size is halved. We expect to have the results of these experiments presented at the conference.

3.1 Conclusions

The eigenspace updating technique is suitable for use in an active environment where images are acquired continuously and a representation is incrementally constructed and refined. We have shown its effectiveness in 3D object representation using 2D images which is useful in active recognition and environment exploration. Further research is needed to quantify image saliency in a more objective way, perhaps requiring higher level visual cues to be incorporated into the scheme. The methodology presented here is also applicable to image features such as wavelet transformed data.

Our current research seeks to extend these concepts to video coding and image database problems.

4 References

[1] E. Anderson et al., *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, 1992.

[2] P.E. Gill, G.H. Golub, W. Murray, and M.A. Saunders (1974). "Methods for Modifying Matrix Factorizations," Math. Comp. 28, 505-35

[3] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins Press, 1989.

[4] M. Gu and S. C. Eisenstat, "A Stable and Fast Algo-

1. However, note that in our formulation the cost of checking (projecting onto the current eigenspace) is the same as updating the representation except for a constant factor (strictly less than 1000), and in the case of recognition from large databases it is beneficial to just update the representation using the learning scheme shown in Figure 1.

Nayar [9], Oja's book on subspace methods [11]), but are not very relevant to our discussion here.

Table 1: Comparison of algorithms

Authors	Method	Update
Murakami and Kumar (1982)	$B^T B$	yes
Kirby and Sirkovich (1990)	$B B^T$	no
Turk and Pentland (1991)	$B^T B$	no
Murase and Lindenbaum (1995)	iterative ($B B^T$)	no
This paper	GES (1994)	yes

2 Updating an Eigenspace Representation

If all the N images of the data set are not available at the outset (as in an active sensing scenario), we would need to compute the SVD every time a significant new image is obtained. If we did the naive thing and saved all the images and computed a full-fledged SVD from scratch every time, it would cost $O(mN^3)$ time, which is too slow.

2.1 Adaptive eigenspace computation

We now discuss a more efficient way. Let the left-singular vectors computed by the following incremental updating algorithm after obtaining i images be denoted by $\boldsymbol{U}_{k_i}$, and let Σ_{k_i} denote the corresponding matrix of singular values, where k_i is the number of columns of $\boldsymbol{U}_{k_i}$. Note that U_i can be different from $\boldsymbol{U}_{k_i}$ and the effect of this approximation is studied in this section. When we acquire the new image $\boldsymbol{A}_{i+1}$ we compute a new SVD

$$\left[\boldsymbol{U}_{k_i}\Sigma_{k_i}V_{k_i}^T \quad \boldsymbol{A}_{i+1}\right] = \boldsymbol{U}'\Sigma'\boldsymbol{V}'^T. \qquad (4)$$

We now choose the integer k_{i+1} such that the k_{i+1}th singular value of Σ' is the smallest singular value bigger than δ, where δ depends on ε (the relation between ε and δ will be developed further in Section 2.2). We then pick the first k_{i+1} left singular vectors to form $\boldsymbol{U}_{k_{i+1}}$ and the corresponding singular values to form $\Sigma_{k_{i+1}}$. Since δ does not depend on i it follows from elementary properties of singular values that $k_{i+1} \geq k_i$, which is intuitively obvious.

We now state the algorithm more formally in Figure 1. In practice there is no need to update the SVD for every new image. Only those images which are significantly outside the current object eigenspace, or those that cause a large change in the singular values need be updated.

> $U = A_1/\|A_1\|$, $V = 1$, $\Sigma = \|A_1\|$
> For $i = 2$ to N
> $\left[\boldsymbol{U}\Sigma V^T \quad A_i\right] = \boldsymbol{U}'\Sigma'\boldsymbol{V}'^T$;
>   Find k such that $\sigma_k' > \delta \geq \sigma'_{k+1}$;
>   Let $\boldsymbol{U}$ equal the first k columns of $\boldsymbol{U}'$;
>   Let V equal the first k columns of V' ;
>   Let Σ equal the leading $k \times k$ principal submatrix of Σ ;
> End

Figure 1: Adaptive eigenspace computation

2.2 Accuracy and Efficiency of the Algorithm

Note that $\boldsymbol{U}_{k_{i+1}}$ approximates $\boldsymbol{A}_{i+1}$ to an accuracy of δ. Similarly $\boldsymbol{U}_{k_i}$ approximates A_i to an accuracy of δ. From this it follows that $\boldsymbol{U}_{k_{i+1}}$ approximates A_i to an accuracy of 2δ. In general, $j \leq i$, $\boldsymbol{U}_{k_i}$ approximates $\boldsymbol{A}_j$ to an accuracy of $(i-j+1)\delta$. Therefore if we choose δ to be ε/N we can guarantee that $\boldsymbol{U}_{k_N}$ will approximate all the images to an accuracy of ε.

Let L be the ε rank of B_N. One possibility to guard against is whether k_N may be much larger than L. This would make the algorithm very slow. This can happen only if there are singular values of B_N clustered around ε. This problem should be solved on a case by case basis. More details can be found in [7].

Time Complexity. We now consider the time complexity of the algorithm. To compute the SVD of

$$\left[\boldsymbol{U}_{k_i}\Sigma_{k_i}V_{k_i}^T \quad \boldsymbol{A}_{i+1}\right],$$

we use the fast SVD update algorithm of Gu and Eisenstat [4]. This algorithm computes the new SVD in $O(mk_i)$ time. Therefore, the total time spent on computing SVD's will be $O(mk_N N)$. Note that if $k_N \ll N$ (which is not unreasonable), then computing the SVD one image at a time is *faster* than computing the SVD of B_N directly!

mation is embedded within or how different the image is from the current eigenspace representation), and the current representation is updated accordingly. Since the updating algorithm is based on computing the singular values of a matrix composed of images, we begin with a brief review of the eigenimage representation.

1.1 Eigenimage Representation

In the following discussion, we shall use the standard Euclidean 2-norm denoted by $\|\,.\,\|$:

$$\|x\| = \sqrt{\sum x_i^2} \quad \text{for } x \in \Re^n . \tag{1}$$

Then, for a matrix B ,

$$\|B\| = \max_{\|x\| = 1}\|Bx\| . \tag{2}$$

Let $\{A_i\}$ denote a sequence of image vectors, obtained by row-scanning the two-dimensional images with m pixels in each image. Let B_i denote the matrix $[A_1\ A_2\ \ldots\ A_i]$. Let ε be a given tolerance and define the ε -rank of B_i to be the number of singular values of B_i greater than ε . Denote the ε -rank of B_i by k . Therefore, if σ_j 's are the singular values of B_i in non-increasing order, then $\sigma_k > \varepsilon \geq \sigma_{k+1}$. In many image processing applications, $k \ll i$. Therefore, B_i can be represented efficiently by its first k singular vectors and singular values. Denote the SVD of B_i by

$$\begin{bmatrix} U_i & * \end{bmatrix} \begin{bmatrix} \Sigma_i & 0 \\ 0 & [\varepsilon] \end{bmatrix} \begin{bmatrix} V_i^T \\ * \end{bmatrix} , \tag{3}$$

where U_i is an $m \times k$ matrix, Σ_i is $k \times k$ diagonal matrix, and V_i is an $i \times k$ matrix. Note that U_i and V_i are matrices whose columns are the first k left- and right-singular vectors, respectively.

Note that B_i can be reconstructed to ε accuracy by $U_i \Sigma_i V_i^T$. That is $\|B - \boldsymbol{U}_i \Sigma_i \boldsymbol{V}_i^T\| \leq \varepsilon$. The algorithmic requirement in many applications is to compute $\{U_i, \Sigma_i, V_i\}$ efficiently. This can be directly computed from B_i by using standard SVD algorithms (e.g., Golub-Reinsch [3]). This has a complexity of $O(mi^2)$. Some researchers ([8],[14]) have suggested computing SVD by computing the eigendecomposition of $B_i^T B_i$. While this has the same complexity, its numerical properties are not as good [3]. Nevertheless, in applications involving a large number of images, computing the SVD of B_i can be too slow.

In many situations (as in face recognition, database browsing, video coding, and active recognition) the $\{U_{i-1}, \Sigma_{i-1}, V_{i-1}\}$ is available and this can be used to speed-up the computations. We can approximately compute $\{U_i, \Sigma_i, V_i\}$ by computing the SVD of $\begin{bmatrix} U_{i-1}\Sigma_{i-1}V_{i-1} & A_i \end{bmatrix} = \hat{B}_i$. This is the approach taken in [8] but they compute the SVD of $\hat{B}_i$ by computing the eigendecomposition of $\hat{B}_i^T \hat{B}_i$. This costs $O(mk + k^3)$. While more efficient than computing the SVD of B_i , this still suffers from potential numerical instability [3]. Murase [10] advocates the use of iterative methods for computing the SVD/eigendecomposition. But as is well known in numerical linear algebra [3] it is difficult to get robust implementations of such iterative methods.

In this paper, we propose instead the use of a direct update GES algorithm to compute the SVD of $\hat{B}_i$. This algorithm has good numerical properties, and is as efficient as the approach of [8]. Moreover, for data sets with large k , Gu and Eisentat have a fast version of the algorithm with time complexity $O(mk)$. A brief description of the GES algorithm is given below.

The GES Algorithm: The Gu and Eisenstat algorithm is used for computing the SVD of a matrix which can be represented as a diagonal matrix plus a rank-1 update. That is, if $B = D + uv^T$, where D is a diagonal matrix and u and v are vectors, then the GES algorithm computes the SVD of B in $O(n^2)$ where n is the dimension of D . If the SVD of B_i is known and we append another image A_{i+1} to get the new matrix B_{i+1} , then its SVD is related to the SVD of B_i by a rank-1 update involving A_{i+1} . The algorithm is based on a divide-and-conquer strategy and we refer the reader to [4] for the details.

Table 1 summarizes the algorithmic differences among some previous work in vision and ours. Note that these papers address different applications and it is not our intention here to compare those other aspects. From this table, an important conclusion to be drawn is that there exists a powerful technique from numerical linear algebra (fast and stable SVD update) that has important bearings on many vision applications. This is born out by the fact these techniques have been rediscovered several times in the vision literature. There are several other papers concerning various pattern recognition applications of eigenspace representations (for example, Murase and

An Eigenspace Update Algorithm for Image Analysis

B.S. Manjunath[†], S. Chandrasekaran[†] and Y.F. Wang[‡]
[‡]Department of Computer Science
[†]Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
manj@ece.ucsb.edu, shiv@ece.ucsb.edu, yfwang@cs.ucsb.edu

Abstract

During the past few years several interesting applications of eigenspace representation of the images have been proposed. These include face recognition, video coding, pose estimation, etc. However, the vision research community has largely overlooked parallel developments in signal processing and numerical linear algebra concerning efficient eigenspace updating algorithms. These new developments are significant for two reasons: Adopting them will make some of the current vision algorithms more robust and efficient. More important is the fact that incremental updating of eigenspace representations will open up new and interesting research applications in vision such as active recognition and learning. The main objective of this paper is to put these in perspective and discuss a recently introduced updating scheme that has been shown to be numerically stable and optimal. We will provide an example of one particular application to 3D object representation projections and give an error analysis of the algorithm. Preliminary experimental results are shown.

1 Introduction

The eigenspace representation of images has attracted much attention recently among vision researchers [8]-[14]. The basic idea is to represent images or image features in a transformed space where the individual features are uncorrelated. For a given set of (deterministic) images this can be achieved by performing the Singular Value Decomposition (SVD). The statistical equivalent of this is the Karhunen-Loeve Transform (KLT) which is computed by diagonalizing the autocorrelation matrix of the image ensemble. Note that when the complete data are given, SVD converges to the KLT, if it exists. Both are well known techniques in image processing. However, they are computationally expensive.

Since computing SVD is expensive, there is a need for efficient algorithms for SVD updating. In the updating problem, one is interested in computing the new SVD when a row (or a column) is added to a given matrix whose SVD we already know. The idea of SVD updating has been prevalent in signal processing for about two decades. One of the first papers on the *numerical* issues of updating matrix factorizations appeared in 1974 [2]. However, till recently there was no fast and stable updating algorithm for the SVD [4].

In the context of image analysis in eigenspace, this paper makes the following contributions:

- We provide a comparison of some of the popular techniques existing in the vision literature for SVD/KLT computations, and point out the problems associated with those techniques.
- A brief summary of Gu and Eisenstat's [4] SVD updating algorithm, GES, is given. This algorithm has been proved to be stable and optimal. Using the GES, we suggest a technique for adaptively modifying the number of basis vectors and provide an error analysis.
- We provide preliminary experimental results for the case of 3D object representation using image projections. Other interesting applications in vision are identified.

Although SVD updating techniques have been used by several researchers in the past, to the best of our knowledge this is the first time that a scheme is suggested for adaptively modifying the number of basis vectors.

Let us consider the following scenario: A camera is mounted on a robot which explores a 3D object by viewing it from different angles, and builds an internal representation in terms of image projections. This is a slightly different formulation from the face recognition problem introduced in [13] and later made popular by [14]. In all these cases, we need to be able to recognize an object from its projections only. We assume that image data are directly used in building a representation, but the formulation is valid for any set of image features extracted from the image data. As the sensor acquires each new image, the image is analyzed to determine if it is a salient image (the image saliency is measured by how much new infor-