Subset Selection for Active Object Recognition

Jay Winkeler, B.S. Manjunath, and S. Chandrasekaran
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
jay@iplab.ece.ucsb.edu, manj@ece.ucsb.edu, shiv@ece.ucsb.edu

Abstract

This paper presents an algorithm for constructing object representations suitable for recognition. The system automatically selects a representative subset of the views of the object while constructing the eigenspace basis. These views are actively located for object identification and pose determination. All processing is performed on-line. The camera is actively positioned during both representation and recognition. When tested with 240 views for each of seven objects, the system achieves 100% accurate object recognition and pose determination. These results are shown to degrade gracefully as conditions deteriorate.

1 Introduction

Most vision tasks require some level of object recognition. Recognition requires that each object be represented in some way that facilitates identification. The representation is integral to recognition.

This paper addresses the problem of constructing an eigenspace representation quickly, even on-line. An exact representation is costly to construct, so it is approximated from a subset of the object views. Images are included in the subset based on saliency, a measure of information not already contained in the representation. Saliency is defined in section 2 of this paper. The system stores the coefficients for the selected images only; it actively searches for views corresponding to these images during recognition.

1.1 Previous work

Methods for recognizing objects from appearance have been studied for over a decade. Early object recognition methods use geometric models [1], [3], which are impractical for a large object set. Recent methods learn the object representations automatically [7], [4], [2], using the eigenspace techniques developed for face recognition [9], [10].

Murase and Nayar propose a parametric representation for object recognition [7]. They build object representations by sampling the eigenspace coefficients as the object rotates or the lighting changes. A cubic spline interpolates the coefficient hypersurface from these discrete points.

Borotschnig et al. [2] propose an active framework to augment the parametric eigenspace method. They use probability distributions to fuse the information gathered in multiple views. The system plans the camera motion to quickly distinguish between the most likely candidate identities.

Chandrasekaran et al. [4] propose an algorithm for iteratively updating an eigenspace representation. They present a saliency measure and a method of selecting a subset of the ensemble as part of representation building.

The *salient image search* method presented in this work uses the eigenspace update algorithm from [4]. Their subset selection method can be seen as an approximation to the greedy algorithm developed in this paper.

The paper is organized as follows. Section 2 defines terminology. Section 3 discusses subset selection. Section 4 tests approximations to the greedy algorithm and Section 5 tests each step of the salient image search. Section 6 tests the complete system. Section 7 concludes the paper.

2 Terminology

Eigenspace representation: A low-dimensional representation of an ensemble of images. Given a set of N images stored as vectors, $\{x_i\}$ for $i = 1, 2, \ldots, N$, principal components analysis (PCA) finds a compact eigenspace representation of the images by calculating a small set of vectors, $U = [u_1, u_2, \ldots, u_M]$ where $M \leq N$ and M is the dimension of the representation, which linearly reconstructs every image. The coefficients describing image y are $a_i = (y - \bar{x})^T u_i$ for $i = 1, 2, \ldots, M$, where $\bar{x}$ is the mean image of the ensemble. The image is reconstructed as $\tilde{y} = \sum a_i u_i$, with a summation over M.

In this research, the singular value decomposition (SVD) $X = U \Sigma V^T$ calculates the eigenvectors. This decomposition can be efficiently incremented [4] when a new image, b, is added to the existing representation $U \Sigma V^T$.

Saliency (S): The information in an image, relative to a representation. For an eigenspace representation, the information is the amount of energy not captured by the basis set, the residual error. During representation building, high saliency indicates the representation needs updating to better capture the training set. During object recognition, low saliency indicates the image is a member of the ensemble represented by the basis set. In the latter use, saliency is related to *distance from face space (DFFS)* proposed in [10] for rejecting "non-face" images with a face identification system.

Batch algorithm: The standard method for computing the eigenspace representation of an image ensemble. This algorithm has one parameter: the amount of basis set truncation. All the column-scanned images are gathered in the training ensemble in one matrix; the basis set is computed by a one-time SVD of this matrix. Finally, this set is truncated.

3 Subset selection

Computing an eigenspace basis with the batch algorithm is expensive. Many researchers approximate the basis using a subset of the ensemble. If the images are ordered (e.g. a series of views of an object), the subset is selected as every n th image from the ensemble (SBM). This approach is used in the parametric eigenspace [7]. For unordered images, the subset is drawn randomly (RBM). This paper presents an algorithm that selects images based on their content.

Greedy algorithm: This algorithm has one parameter: the termination criterion. The initial representation is simply the average image. The saliency of every ensemble image is computed; the one with the highest saliency is added to the subset. The representation is updated as described in [4]. The process is repeated until termination is reached.

3.1 Theoretical foundation

Subset selection finds the M columns from which the best representation of matrix X can be constructed. The problem is to rearrange the columns of X when the first M columns will be used to represent the entire matrix. Mathematically, the goal is to find a permutation matrix, P_M, minimizing the residual error, $\|Y - \tilde{Y}_M\|$, across all of the images. $XP_M = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix}$ and Y_1 has M columns. The eigenspace representation is found from the SVD: $Y_1 = U_1 \Sigma_1 V_1^T$. Then the reconstruction is found from $\tilde{Y}_M = \begin{bmatrix} \tilde{Y}_1 & \tilde{Y}_2 \end{bmatrix}$, $\tilde{Y}_1 = U_1 U_1^T Y_1$, and $\tilde{Y}_2 = U_1 U_1^T Y_2$.

The key step of the subset selection algorithm of Golub et al. [5] calculates P_M such that the columns of Y_1 are "sufficiently independent." With a good estimate of matrix rank, they report, "any reasonably independent subset" produces essentially the same size residual. Without such an estimate, the problem is more difficult. Since the SVD is the best method of estimating matrix rank, this algorithm cannot be used to reduce the cost of the SVD by subset selection.

The greedy algorithm selects a subset by iteratively selecting the $y \in Y_2$ maximizing $\|(I - U_1 U_1^T) y\|$. At each step, the algorithm selects the column "most independent" from the representation. Experiments show that no method of iteratively selection will always achieve the optimal subset. The difference in residual error, however, is small.

3.2 Experiments in subset selection

Four image ensembles are used in an empirical comparison of the batch and greedy algorithms. One consists of 120 cropped and registered, forward-facing face images (from the U.S. ARMY'S FERET database). The other ensembles contain every 3° of view of three objects, two of which are shown in Figure 1; a total of 120 images each. The objects are loosely segmented, so that most of the background is removed. The background is set to zero.

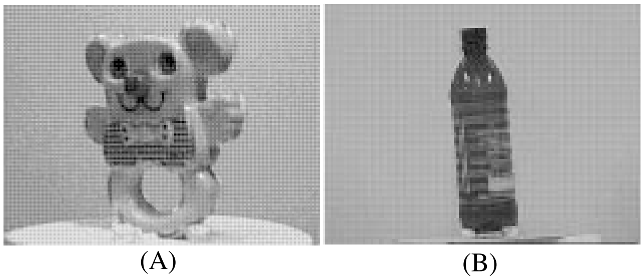

Figure 1: (A) Object 1 at 0°, (B) Object 2 at 0°.

The basis dimension is determined by a threshold for the maximum saliency over the ensemble. The greedy algorithm results are accumulated over all 120 possible initializations. For the SBM and RBM the subset size is chosen prior to PCA: SBM(i) means a subset of i images. The SBM results are calculated over all $120/i$ possible training sets. Ten random image sequences are used to test the RBM algorithm. Tables 1 and 2 show representative results. Threshold and saliency are based on residual error, which is the square root of the sum of the squared pixel errors. These errors are normalized to be per pixel.

Table 1: Representations of FERET face database

Thresh	Alg.	Dim.	Saliency
0.25	Batch	8	0.2398
	RBM(80)	80	0.4143
	Greedy	10	0.2412

The batch algorithm uses the complete image ensemble, so it can be considered optimal. It results in a more compact representation than the greedy algorithm, requiring one-half to two-thirds the basis dimensions for the same threshold.

The RBM performs surprisingly poorly in terms of maximum saliency; it never achieves the threshold saliency even when using two-thirds of the image ensemble. The average image is well-represented, but at least one image has saliency nearly double the threshold.

Using the same number of images as the greedy algorithm, the SBM fails to meet the threshold saliency. Table 2 shows that even using twice as many images, the SBM usually results in a less compact representation than the greedy algorithm. Only when the SBM uses more than one-third of the ensemble, when the threshold is 0.07, is its representation more compact.

Figure 2 shows the histogram of salient image selection when the initial image is chosen manually instead of automatically. The results are accumulated over all 120 initial images. *The peaks in the histogram show that the algorithm selects the same images regardless of starting position.* A stable salient set implies the greedy algorithm selects the images that best describe the complete training ensemble.

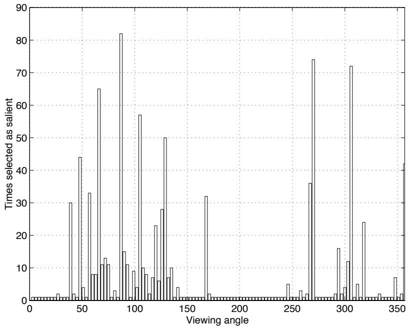

Figure 2: Histogram of image selection for Object 3.

4 Greedy algorithm approximations

The greedy algorithm provides a good trade-off between computational cost and representational quality for an eigenspace representation. The algorithm, however, requires $O(n^2)$ saliency computations for an ensemble of n images. The proposed *peak algorithm* requires $O(n)$ computations.

The peak algorithm examines the ensemble sequentially and computes a running average of the images, so it is amenable to active acquisition of images. The running average causes small distortions in the representation, resulting in a maximum saliency slightly above the threshold.

4.1 Algorithms

Peak algorithm(PA): This algorithm has three parameters: the initial image, the saliency threshold, and the neighborhood size. The ensemble is examined sequentially. Any image that exhibits the highest saliency over a small neighborhood and violates the threshold is added to the subset. The representation is immediately updated. During the search, the non-peaks that violate the threshold are stored. Once a peak is located, the peak search is repeated through the stored images. In theory, this recursive search could degenerate into an $O(n^2)$ search, but such degeneration has never been observed in our experiments.

The neighborhood size parameter determines how accurately peaks are located. A loss of stability of the subset and a decrease in the compactness of the representation are evident when the peaks are located incorrectly. The greedy algorithm represents the largest possible neighborhood, while the subset selection method proposed in [4] represents the opposite extreme: no neighborhood, any image with saliency over the threshold is selected. The latter case results in representations twice the dimension of those constructed with the greedy algorithm.

For uniformity in these experiments, the saliency threshold is determined off-line to produce the desired salient set size. In practice, the threshold would be held constant over all the objects, resulting in saliency set sizes proportional to object complexity [4].

4.2 Experiments with the approximations

Table 2 includes details of representations constructed with the peak algorithm, which selects approximately one more image than the greedy algorithm for the same threshold. The peak algorithm has a stable subset, which is useful for locating images during pose determination. This algorithm is a good approximation to the greedy algorithm for ordered ensembles.

Table 2: Representations of Object 1

Thresh	Alg.	Dim.	Saliency
0.11	Batch	3	0.1051
	SBM(14)	11.5	0.1120
	Greedy	6	0.1099
	Peak	6.625	0.1339
0.09	Batch	7	0.0825
	SBM(26)	26	0.0898
	Greedy	12	0.0892
	Peak	13.608	0.1073
0.07	Batch	11	0.0666
	SBM(48)	14	0.0690
	Greedy	23	0.0692
	Peak	24.283	0.0739

5 Object recognition and pose determination

These experiments use image databases to simulate a camera moving in a circle around the objects. The peak algorithm constructs a representation, storing only the projection coefficients for the salient views, plus the basis images. This saves the computation and storage costs for the hypersurfaces used in [7]. The recognition algorithm proceeds:

1. **Initialization**: The first peak is located with the PA.
2. **Identification**: If identification fails, the object is considered unknown.
3. **Pose calculation**: If pose determination fails, the object identity is rejected.
4. **Verification**: Move the camera and determine the new pose. If verification fails, the previous pose is rejected.

For these experiments, the PA builds representations of every $3°$ of the three objects already described. Interpolated views of the objects are used in testing. Saliency thresholds are chosen so the PA selects approximately eight images.

5.1 Initialization

Object recognition begins with the peak algorithm, allowing the system to proceed efficiently. If the object is unknown, representation building continues. If the object is recognized, the pose is determined next. The peak search does not require knowledge of the object identity.

Since the system knows only the stored images, views corresponding to these images must be found for pose determination. For an unknown object, the first peak will be a salient view. The stability of the salient images to starting point implies this peak is near a stored image for a known object.

Locating the first peak costs $O(P)$ floating point operations for each image examined, where P is the number of pixels in the image. An alternative to locating the first peak searches with the eigenspace coefficients. This search depends on object identity and costs $O(PM)$ operations per image, where M is the basis dimension.

Figure 3 (A) shows a histogram of experimentally determined distances from the first peak to the nearest stored image. The first peak is occasionally distant from a stored image, particularly where the object has symmetries. By comparison, the histogram for a subset chosen by equally spaced sampling of the ensemble is level from $-22°$ to $22°$.

5.2 Identification

For object identification, the image saliency under each of the object representations is calculated. Images with low saliency lie near the space represented by the eigenspace basis set. All representations giving saliency below a threshold are considered candidate identities, with the lowest saliency measure corresponding to the most likely candidate. If no candidates exist, the object is unknown. The threshold is set based on the saliency threshold used during training.

With the salient image search algorithm, the representation of the object library is merely a collection of each object's eigenspace representation. Adding a new object costs $O(PM^3)$; compared to $O(PN^2B^2)$ with the parametric eigenspace method (not counting the spline update), where N is the number of images of each object and B is the number of objects. The representation of a large set of objects is inexpensive to maintain with the salient image algorithm.

As the object library becomes large, however, identification may become expensive. Identification costs $O(PMB)$, compared to $O(PR)$ with the parametric eigenspace method. R is the basis dimension of the representation of the complete library, and $MB > R > M$. For a large object library, the salient search will require an indexing scheme.

For Objects 1, 2 and 3, saliency is sufficient for object identification. The maximum saliency over all images of self is separated from the minimum saliency of images of any other object by more than three times the training threshold.

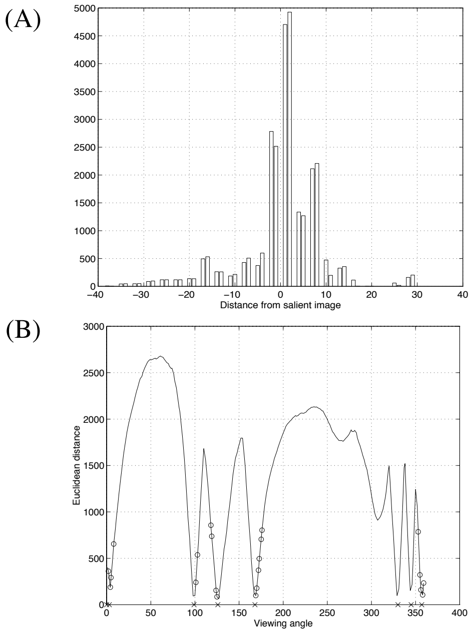

Figure 3: (A) Histogram of distances from nearest salient image for Object 3, Euclidean distances to the nearest vector of salient image coefficients for Object 1. x: stored images, o: views chosen by peak search.

5.3 Pose computation

For pose determination, the camera is maneuvered until it reaches a locally minimal distance between the coefficients of the observed view, a_i, and those of its nearest neighbor in the stored image set. The distance is calculated as $d = min\|a_i - c_j\|$, where the minimum is calculated over $j \in S$, the set of stored images, and c_j is the stored vector of coefficients for view angle j. When the local minimum of d is located, the pose is reported as j. The algorithm rejects distances above a threshold.

This search costs $O(PM)$ operations per image. When the starting point is not locally minimal, the direction of the search around the object is set as downhill on the observed distance curve. Crossing too many peaks indicates an unknown object and is used as a signal to cut the search short.

Figure 3 (B) shows the distance to the nearest vector of stored coefficients for a set of test views. The local minima not corresponding to stored images occur when the nearest stored image in coefficient space differs from the nearest in view space, usually due to object symmetries. Near the stored images, the measures refer to the same image.

For Object 1, the first peak is a local minimum in 13% of the trials. On the average, 4.2 images are examined to locate a local minimum. Since the test sets include only interpolation views, a reported pose corresponding to the nearest stored view to either side of the actual pose is considered correct. For this object, pose determination is 100% correct.

5.4 Verification

Pose is verified by moving the camera to the expected location of a stored view and finding another valid estimate. A more stringent test checks whether the approximate change in position around the object (sensed) corresponds to the change in pose. Verification is open to further heuristics.

As the object library grows larger, false alarms in identity become more likely. As the differences between the training and testing images increase, the coefficient threshold have to rise, causing more false alarms in pose determination. Verification reduces these errors by requiring more information before the identity and pose are accepted.

6 Experimental results for pose determination and object recognition

This section presents results of experiments with the proposed object recognition and pose determination algorithm. As in the previous section, camera motion is simulated. Table 3 shows the details of the three test sets.

1. Seven objects, including Objects 1, 2 and 3.
2. Twenty objects from the Columbia Object Image Library [8].
3. Five objects, including those shown in Figure 4.

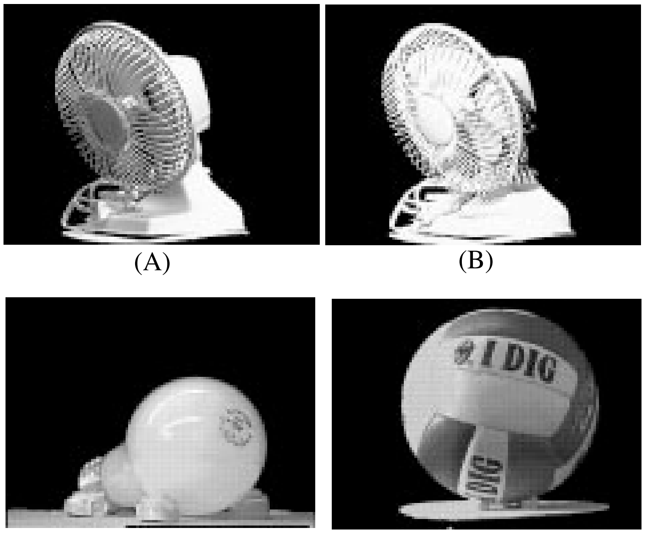
(A) (B)
(C) (D)

Figure 4: (A) and (B) Object L1 under two different lighting conditions (C) Object L2 and (D) Object L5.

The algorithm is tested on interpolation views in pose; test offset indicates the smallest difference between test and training images. The reported pose is considered correct if it is the nearest training view to either side of the actual pose. If the precise pose is required, a simple spline approximation between stored coefficients--the parametric method described in [7]--can be used.

6.1 Favorable conditions

Experiments with the first object set show the quality of results under favorable conditions. A saliency threshold of 1.5 times the training saliency threshold completely determines the object candidate set. Both the initial identification and the reported pose are correct for 100% of the views.

6.2 Graceful degradation

Experiments with object set two show the graceful degradation of the results as the task becomes more difficult.

The initial identification is correct for 87% of the views. Inability to estimate pose forces the rejection of incorrect identities; identification is 92% correct after this step. The simple verification scheme raises identification to 95%, and the more stringent scheme raises it to 99.9%.

After verification, the pose is correctly determined for 84% of the views. Another 11% of the reported poses error by approximately 180° due to object symmetries. After the stringent verification, the pose is correctly determined for 98.8% of the views.

Table 3: Object set information

Set	No. of obj.	Train every	Subset size	Test offset	Light varies
1	7	3°	8	1°	no
2	20	10°	7	5°	no
3	5	4°	8	2°	yes

6.3 Lighting variations

The experiments with the third object set show the pose determination results when lighting varies. The 8 lighting conditions are made up of ambient light, plus a point source near enough to have a significant effect on object appearance. In terms of the average distance between the coefficients, the lighting changes are equivalent to pose changes of 4.8° . When the training set contains more than one lighting condition, the camera completes an entire circuit of the object the object before the lighting is changed.

Table 4 shows the correct pose determination rate for object set 3, given the number of lighting conditions trained and tested. Pose determination is difficult for two of the tested objects; the pre-verification information of these objects are tabled separately from the other three. Object L2 has a specular surface; changes in lighting conditions have a large effect on the appearance of the object. Object L5 has no shape information; the edge of the ball does not move over all viewing positions. Verification slightly increases the pose determination rate for the simple objects and significantly increases the rate for the more difficult objects.

The correct pose determination rate decreases as more lighting conditions are trained. The system is allocating some of its limited resources to representing the lighting conditions. The verification system suppresses much of the error caused by the loss of representational quality.

Table 4: Correct pose determination rate

Train	Test	L2	L5	Others	Post-verify
1	5	70%	91%	97%	99%
2	5	70%	90%	95%	98%
5	3	67%	93%	93%	98%

6.4 Comparison to existing techniques

The proposed salient image search method addresses the same object recognition problem as the parametric method [7], [2]. With the salient search, both representation building and recognition are performed on-line, while the representation is built off-line in the parametric method.

Object recognition performance is similar for the two methods. The salient search recognizes a set of seven objects perfectly. A set of 20 objects is recognized with a 0.1% misidentification rate. Like the parametric method, the salient search method offers nearly perfect object recognition.

Comparing pose estimation between the two methods is more difficult. The reported pose estimation error is $0.5°$ for the parametric method. The correct pose determination rate is approximately 98% for the salient image search method, which corresponds to a normalized estimation error of $0.09°$. Normalization accounts for the artificially imposed minimum distance of $2°$ between the actual and estimated poses. An experiment with a test set differing from the training set only in lighting shows a pose error of $0.46°$. The salient image search method produces results similar to those of the parametric method, at a lower cost.

Training with the salient image search method is much less costly than training with Borotschnig's algorithm [2], which gathers a large number of images similar to each representation-building image to approximate the eigenspace probability densities. Their work shows the advantages of an active framework: the basis dimension required to reach a specific level of recognition is lower than with traditional eigenspace methods, and objects with common appearance in many views can be disambiguated.

7 Conclusions

The greedy algorithm provides a new method of selecting a subset of an image ensemble when constructing an eigenspace basis. Each image is selected based on its content. The greedy algorithm results in a much more compact representation for a given saliency threshold than the current standards of subsampling and selecting at random.

With a linear saliency calculation cost in ensemble size, the peak algorithm represents ordered ensembles nearly as compactly as the quadratically expensive greedy algorithm does. The selected subset is stable to initial conditions, making it appropriate for the proposed recognition system.

For an active camera system with the ability to loosely segment images, the peak algorithm makes for efficient representation and recognition of objects. For ordinary image distortions over a set of seven objects, the system correctly identifies every object.

These results degrade gracefully as the differences between the training and testing images grow larger. The verification system is key to the slow degradation. Heuristics can be added to decrease the errors further.

Future work includes testing face identification with multiple views of the face and designing an image-skipping algorithm for more efficient representation building.

Acknowledgments: Portions of the research in this paper use the FERET database of facial images collected under the ARPA/ARL FERET program.

8 Bibliography

[1] Besl, P. and Jain, R.; “Three-Dimensional Object Recognition,” *ACM Computing Surveys* 17(1), pp. 75-145, Mar. 1985.

[2] Borotschnig, H.; Paletta, L.; Prantl, M.; and Pinz, A.; “Active Object Recognition in Parametric Eigenspace,” *Proc. Brit Machine Vision Conf.*, (Southampton, UK), (2) pp. 629-638, 1998.

[3] Chin, R. and Dyer, C.; “Model-Based Recognition in Robot Vision,” *ACM Computing Surveys* 18(1) pp. 67-108, Mar. 1986.

[4] Chandrasekaran, S.; Manjunath, B.S.; Wang, Y.F.; Winkeler, J. and Zhang, H.; “An Eigenspace Update Algorithm for Image Analysis,” *CVGIP: Graphical Models and Image Processing* 59(5), pp. 321-332, Sept. 1997.

[5] Golub, G.; Klema, V.; and Stewart, G.; “Rank Degeneracy and Least Squares Problems,” Technical Report TR-456, Department of Computer Science, University of Maryland, 1976.

[6] Moghaddam, B. and Pentland, A.; “Probabilistic Visual Learning for Object Representation,” *IEEE Trans. PAMI* 19(7), pp. 696-710, July 1997.

[7] Murase, H. and Nayar, S.K.; “Visual learning of object modules from appearance,” *Proc. Image Understanding Workshop 1993*, (San Diego, CA), pp. 547-555, 1993.

[8] Nene, S.; Nayar, S.; and Murase, H.; “Columbia Object Image Library (COIL-20),” Department of Computer Science, Columbia University, Technical Report CUCS-006-96.

[9] Pentland, A., Moghaddam, B. and Starner, T.; “View-based and modular eigenspaces for face recognition,” *Proc. IEEE Conf. CVPR ‘94*, (Seattle, Washington), pp. 84-91, June 1994.

[10] Turk, M. and Pentland, A.; “Eigenfaces for Recognition,” *Journal of Cognitive Neuroscience* 3(1), pp. 71- 86, Mar. 1991.