A Case for the Average-Half-Face in 2D and 3D for Face Recognition

Josh Harguess and J. K. Aggarwal
Computer & Vision Research Center / Department of ECE
The University of Texas at Austin
harguess@ece.utexas.edu, aggarwaljk@mail.utexas.edu

Abstract

We observe that the human face is inherently symmetric and we would like to exploit this symmetry in face recognition. The average-half-face has been previously shown to do just that for a set of 3D faces when using eigenfaces for recognition. We build upon that work and present a comparison of the use of the average-half-face to the use of the original full face with 6 different algorithms applied to two- and three-dimensional (2D and 3D) databases. The average-half-face is constructed from the full frontal face image in two steps; first the face image is centered and divided in half and then the two halves are averaged together (reversing the columns of one of the halves). The resulting average-half-face is then used as the input for face recognition algorithms. Previous work has shown that the accuracy of 3D face recognition using eigenfaces with the average-half-face is significantly better than using the full face. We compare the results using the average-half-face and the full face using six face recognition methods; eigenfaces, multi-linear principal components analysis (MPCA), MPCA with linear discriminant analysis (MPCA-LDA), Fisherfaces (LDA), independent component analysis (ICA), and support vector machines (SVM). We utilize two well-known 2D face database as well as a 3D face database for the comparison. Our results show that in most cases it is superior to employ the average-half-face for frontal face recognition. The consequences of this discovery may result in substantial savings in storage and computation time.

1. Introduction

Face recognition has been extensively researched and has recently received much interest. This is partly due to recent efforts in improving security, such as automatic surveillance and the use of biometrics in identification. As a biometric, face image is the least intrusive, but several challenges remain in improving the accuracy of face recognition under illumination changes, variations in pose, occlusions (including self-occlusion), image resolution and other such difficulties. Many face recognition algorithms have been developed and each has its strengths. Each share a common element which is that they all input a full face image into the algorithm. However, none of the methods currently exploit the inherent symmetry of the face for recognition. In this paper we demonstrate the effectiveness of using the average-half-face as an input to face recognition algorithms for an increase in accuracy and potential decrease in storage and computation time.

A previous analysis has been performed using the average-half-face for face recognition [7]. The average-half-face is a transformation method that attempts to preserve the bilateral symmetry that is present in the face. It has been shown to increase the accuracy of face recognition with eigenfaces applied to 3D face images, but is this increase simply an artifact of the database and/or the recognition method chosen? We attempt to answer that question with a comparison of the accuracy using both the average-half-face and the original full face image of the following 6 face recognition algorithms:

1. Eigenfaces

2. Multilinear Principal Components Analysis (MPCA)

3. MPCA with Linear Discriminant Analysis (MPCA-LDA)

4. Fisherfaces or Linear Discriminant Analysis (LDA)

5. Independent Component Analysis (ICA)

6. Support Vector Machine (SVM)

Eigenfaces, based on principal components analysis (PCA), is arguably the best known face recognition method and is used extensively as a benchmark for other methods. MPCA and MPCA-LDA extend PCA to tensor objects (multilinear arrays). This is done to preserve relationships between neighboring pixels that are lost when forming one-dimensional (1D) vectors in traditional PCA. Based on LDA, Fisherfaces attempts to use class information to compute a classifier that maximizes between class scatter

7

while minimizing within class scatter and has shown success in face recognition. Most notably known as a method for solving a problem known as the *cocktail-party problem*, ICA can also be applied to face recognition by modeling each image as a linear combination of non-Gaussian random vectors where the weights of the linear combinations of the training and testing images are used for identification. The SVM, a recent and popular data mining and machine learning method, finds the hyperplanes that maximize the margins between training data classes.

We employ each of the above algorithms on three datasets; two well-known 2D datasets (The Yale Face Database [1] and The AR Face Database [9]) and a previously cited 3D (range image) dataset [6]. The results of each algorithm on the different datasets for both the average-half-face and full face are shown.

This paper is organized as follows. First, we introduce the concept of the average-half-face. Then, we briefly describe each of the algorithms used for comparison in this paper. We then perform 18 total experiments, 3 datasets with 6 face recognition methods, to compare the accuracy of using the average-half-face to using the full face. We present a discussion of our results and conclude with future work.

2. Average-Half-Face

Several previous attempts to utilize the symmetry of the face for face recognition and face detection have been made. It is well-known that the "face is roughly symmetrical" [14]. Zhao and Chellappa [17] use the idea of face symmetry to solve the problem of illumination in face recognition using Symmetric Shape-from-Shading, while Ramanathan *et al*. [11] introduce the notion of 'Half-faces' (in the sense of exactly one half of the face) to assist in computing a similarity measure between faces using images that have non-uniform illumination. In face recognition, the use of the bilateral symmetry of the face has been limited to extracting facial profiles for recognition [10, 16]. Additionally, one successful attempt has been made in using the average-half-face for face recognition, but the method was only applied to a single database with the eigenfaces method [7]. Therefore, we believe the discoveries found in our work to be largely novel.

The average-half-face [7] is inspired by the symmetry preserving singular value decomposition (SPSVD) [13]. The SPSVD is used to reduce the dimensionality of data while simultaneously preserving symmetry that is present in the data. When applied to face images (2D or 3D) this amounts to two steps. First, the image is centered about the nose of the (properly oriented) face to represent the data as symmetric as possible. When we speak of the data being symmetric, we mean that two spatial halves of the data are similar, not that the matrix of the data itself is symmetric.

(a) Full Face   (b) Average-Half-Face
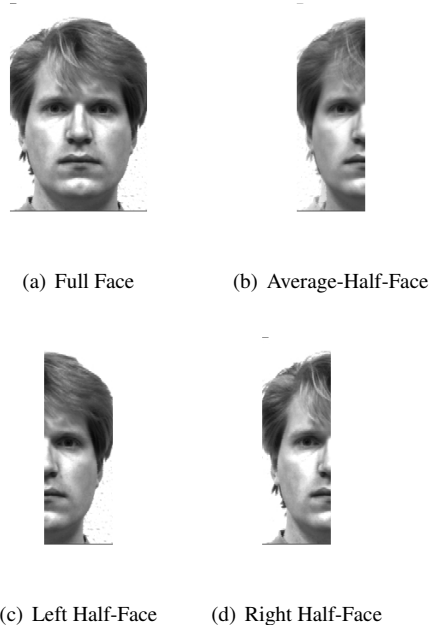
(c) Left Half-Face   (d) Right Half-Face

Figure 1. (a) 2D full face image; (b) its average-half-face; (c) its left half-face; and (d) its right half-face.

Next, the image is divided into two symmetric halves and they are averaged together (reversing the columns of one of the halves first). The final step of performing the singular value decomposition (SVD) on the image is skipped, since we desire a full rank SVD for comparison to the original full face image.

Figure 1 displays the full face image, the left and right faces after centering, and the average-half-face of an example image from the Yale Face database [1].

The average-half-face can be seen as a preprocessing step to a face recognition algorithm. Feature selection as well as subspace computation can be performed on the set of average-half-faces just as is done on a set of full faces. Therefore, the average-half-face can be applied to any face recognition algorithm that uses full frontal faces as an input.

3. Face Recognition Algorithms

We will briefly discuss each of the 6 face recognition methods used in our experiments.

3.1. Eigenfaces

Eigenfaces was introduced early [15] on as powerful use of principal components analysis (PCA) to solve problems in face recognition and detection. Eigenfaces is a subspace projection face recognition method that relies on computing the PCA of a training set which will in return a set of orthogonal basis vectors that maximize the variance that is

inherent in the training data. PCA is an unsupervised technique, so the method does not rely on class information. In our implementation of eigenfaces, we use the nearest neighbor (NN) approach to classify our test vectors using the Euclidean distance.

3.2. Multilinear Principal Components Analysis

One extension of PCA is that of applying PCA to tensors or multilinear arrays which results in a method known as multilinear principal components analysis (MPCA) [8]. Since a face image is most naturally a multilinear array, meaning that there are two dimensions describing the location of each pixel in a face image, the idea is to determine a mulitlinear projection for the image, instead of forming a one-dimensional (1D) vector from the face image and finding a linear projection for the vector. It is thought that the multilinear projection will better capture the correlation between neighborhood pixels that is otherwise lost in forming a 1D vector from the image.

3.3. MPCA + Linear Discriminant Analysis

A further extension of MPCA is to use linear discriminant analysis on the projected multilinear arrays to perform feature selection, which results in MPCA+LDA. Example code for the MPCA and MPCA+LDA methods was provided in [8].

3.4. Fisherfaces

Fisherfaces is the direct use of (Fisher) linear discriminant analysis (LDA) to face recognition [3]. In eigenfaces, the variance between vectors is used to find a linear subspace for projection, without taking into consider the class associations of each vector. In LDA, the class information is explicitly used to form a linear subspace. The purpose of LDA is to maximize the objective function:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \qquad (1)$$

where S_B and S_W are the *between class scatter* and the *within class scatter* matrices respectively and where w is the normal vector to the discriminant hyperplane. The solution can be posed as the eigenvalue problem

$$S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}} v = \lambda v \qquad (2)$$

by defining $v = S_B^{\frac{1}{2}} w$. It turns out that we desire the eigenvectors that correspond to the largest eigenvalues for our solution. Projecting our training and test vectors into this new subspace, we can classify the test images using NN.

3.5. Independent Components Analysis

When applying PCA to a set of face images, we are finding a set of basis vectors using lower order statistics of the relationships between the pixels. Specifically, we maximize the variance between pixels to separate linear dependencies between pixels. ICA is a generalization of PCA in that it tries to identify high-order statistical relationships between pixels to form a better set of basis vectors. We utilize Architecture II as described in [2], where the pixels are treated as random variables and the face images as outcomes. In a similar fashion to PCA and LDA, once the new basis vectors are found, the training and test data are projected into the subspace and a method such as NN is used for classification. The code for ICA was provided by the authors for use in face recognition research [2].

3.6. Support Vector Machine

Support Vector Machines (SVM) are a type of binary classifier that are designed to maximize the *margin* of the decision boundary between positive and negative examples, or support vectors [12]. Since we represent each face as a 1D vector, this amounts to finding the most informative positive and negative support vectors and maximizing the margin between them to form the optimal decision boundary for classifying new vectors.

For implementation of the SVM, we utilized the powerful LIBSVM library [5] that uses the "one vs. one" approach to multi-class problems. We used default parameters for our SVM, including a radial basis function (rbf) kernel.

4. Databases

The experiments were performed on three face image databases; A: The Yale Face database, B: The AR Face database, and C: 3D face database.

The Yale Face database (A) [1, 4] consists of a total of 165 gray scale, frontal, 2D face images. There are a total of 15 subjects with 11 images per subject representing changes in illumination and facial expressions. For each of the algorithms, we maintained a consistent use of the database by forming the training data from the first 8 images per subject and using the remaining 3 images per subject for testing.

We used images from 109 subjects (66 men and 43 women), each with 26 configurations from the AR Face database (B) [9]. The different configurations consist of expression changes (such as neutral, smile, anger, and scream), lighting changes, and occlusions. Two different sessions, each with 13 different configurations, were taken to form the database. We used the first 21 configurations per subject for training and the remaining images for testing.

We have additionally utilized a 3D face range image database (C) acquired using an MU-2 stereo imaging system manufactured by 3Q Technologies Ltd. (Atlanta, GA)

by the former company Advanced Digital Imaging Research, LLC, Friendswood, TX [6]. The database consists of a total of 1126 images of 104 subjects. There are anywhere from 1 to 55 images per subject. We trained the algorithms using a combination of 360 images from 12 subjects and a single neutral expression from 104 different subjects. The test database consisted of the remaining 662 images from all 104 subjects.

5. Experiments

The parameters for each algorithm were kept constant between experiments to maintain a fair comparison of each algorithm's performance on the average-half-face and the full face. Also, the images were centered for both the average-half-face and full face recognition results. Table 1 summarizes the results of our experiments. Each of the numbers in the table represents the rank-1 accuracy rate for recognition, meaning that we report only the accuracy of the closest match of the test data to a corresponding training sample. It is crucial to recall when studying that table that the purpose of these experiments is to compare the full face to the average-half-face for recognition, not to compare the accuracy of the algorithms themselves.

Figures 2, 3, and 4 display these results more clearly for each of the three databases involved. From the results of Figures 2 and 4, we can clearly see that the average-half-face outperforms the full face in every method for the Yale Face database and the 3D face database. However, there are mixed results shown in Figure 3 when using the AR Face database. The best performing method for the Yale Face database was ICA with the average-half-face at 100% accuracy. For the AR Face database, the MPCA-LDA method was the best with the full faces at 91.9%. The 3D database saw the best result of 93.8% accuracy with the MPCA-LDA method and the average-half-face.

Figures 5, 6, and 7 display cumulative match characteristic (CMC) curves for each of the databases and algorithms used (except for the SVM, since the SVM classifies test samples based on class boundaries, not on training samples). The CMC curve plots the identification rate of each algorithm versus the list of the top $1 : n$ (rank-n) matches from the training set. For instance, when $n = 1$, these plots simply return the rank-1 recognition values that are displayed in Table 1. In Figure 7, the average-half-face and the full face CMC curves are displayed for the 3D database The curves for MPCA, and MPCA+LDA follow one another closely up to a rank of $n = 50$ or greater. For PCA, LDA and ICA, the average-half-face is clearly more accurate. The curves in Figure 5 give similar results for the average-half-face and the full face up to a rank of around $n = 10$. Beyond rank of $n = 10$, the average-half-face does lag behind in accuracy for the MPCA+LDA and LDA methods. Finally, the CMC curves for the AR Face database

Database	A		B		C	
	Yale		AR		3D	
Algorithms	Full	AHF	Full	AHF	Full	AHF
PCA	77.8	86.7	49.4	52.3	72.8	80.4
MPCA	80.0	93.3	59.4	57.6	81.0	81.3
MPCA-LDA	66.7	68.9	91.9	88.1	91.8	93.8
LDA	91.1	97.8	54.1	78.0	79.8	82.6
ICA	93.3	100	65.3	60.0	76.9	84.3
SVM	91.1	91.1	44.8	36.1	50.8	51.4

Table 1. Rank-1 accuracy results using the full face (Full) and the average-half-face (AHF).
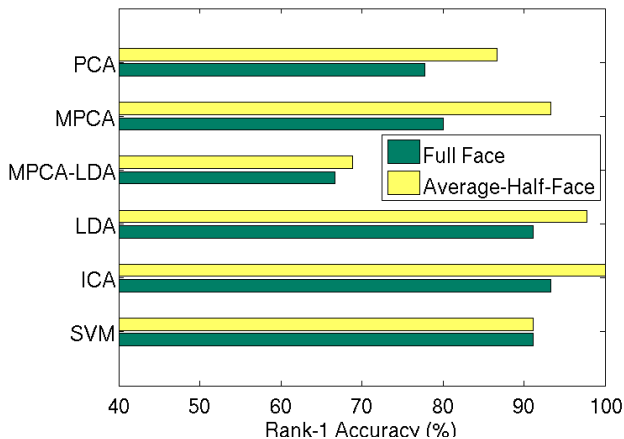

Figure 2. Accuracy of Full Face and Average-Half-Face on Yale Face database (A).

are shown in Figure 6. The curves comparing the average-half-face and the full face for MPCA and MPCA-LDA are almost indistinguishable. The full face performs better with ICA while the average-half-face has a clear advantage with LDA.

6. Discussion

It is abundantly apparent from Table 1 that regardless of the algorithm using average-half-face with the Yale Face database and the 3D database produces an equal or higher accuracy rate than when using the original full face. This is not the case for every method when using the AR Face database. For instance, when using the AR Face database, the rank-1 recognition rates of the ICA and SVM methods are notably better when using the full face versus using the average-half-face. The other methods are very close in accuracy for the AR Face database, except for the Fisherfaces (LDA) method which shows a drastic improvement for the average-half-face. For the Yale Face database, the LDA, MPCA and eigenfaces (PCA) methods perform 6 - 13% bet-
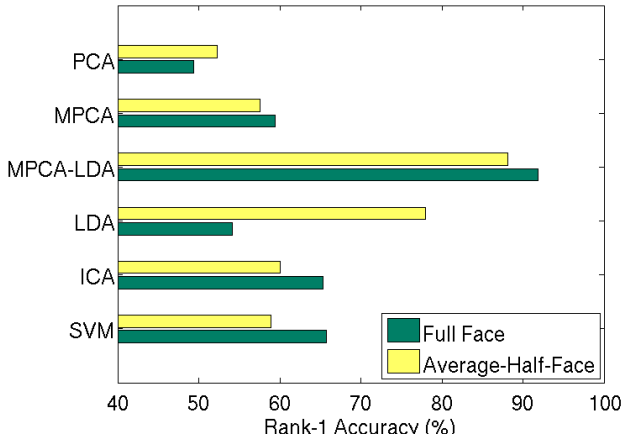
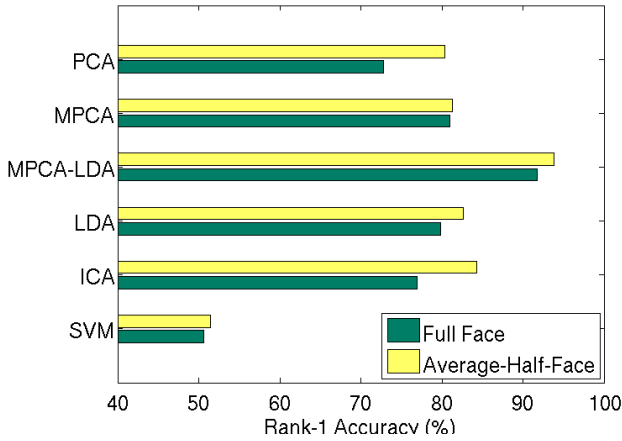Figure 3. Accuracy of Full Face and Average-Half-Face on AR Face database (B).

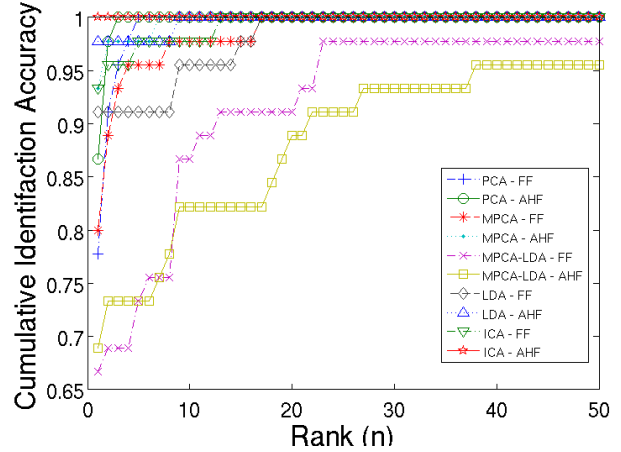Figure 4. Accuracy of Full Face and Average-Half-Face on 3D Face database (C).

Figure 5. CMC plot of Full Face and Average-Half-Face on Yale Face database (A).
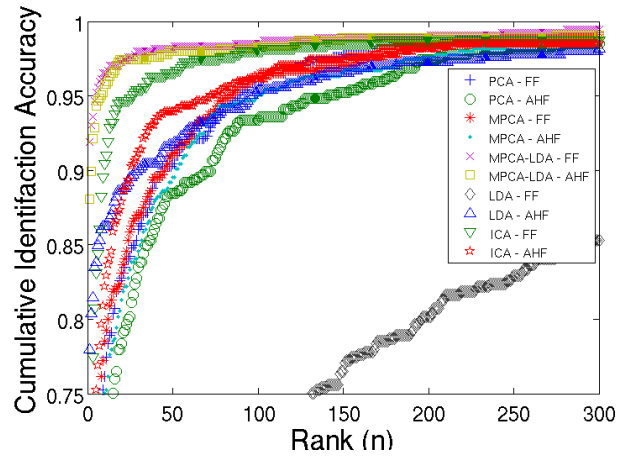
Figure 6. CMC plot of Full Face and Average-Half-Face on AR Face database (B).

ter with the average-half-face than with the full face. All other methods with the Yale Face database are comparable, but usually have better results with the average-half-face. The 3D database gives consistently better results when using the average-half-face with a maximum accuracy increase of around 8% with the eigenfaces (PCA) method.

Another comparison between the full face and the average-half-face is found in the CMC curves provided in Figures 5, 6, and 7. The information present in these curves is revealing to the difference between the performance of the average-half-face and the full face. For most methods and databases, the difference between the curves is negligible. When there is a difference in the curves, the difference only becomes apparent after at least 10% of the possible ranks have been considered. Clearly there are some cases where the average-half-face outperforms the full face and vice versa. However, there does not seem to be a clear advantage of one over the other. Therefore, the average-half-face has proven itself to be comparable, and in some cases superior, to the full face when considering rank n identification of subjects.

At first glance, the results of using the AR Face database might give evidence that average-half-face is inferior to the full face. However, there are only 2 clear instances out of a total of 18 experiments that give evidence to the full face producing a higher accuracy. Therefore, the average-half-face is clearly of interest, especially since the data stored in the average-half-face is exactly half that of the full face, yet the information stored may be more discriminatory for face identification, especially in the case of the 3D database.

The computation of the average-half-face, given the full face and the position of the middle of the face, is simple.
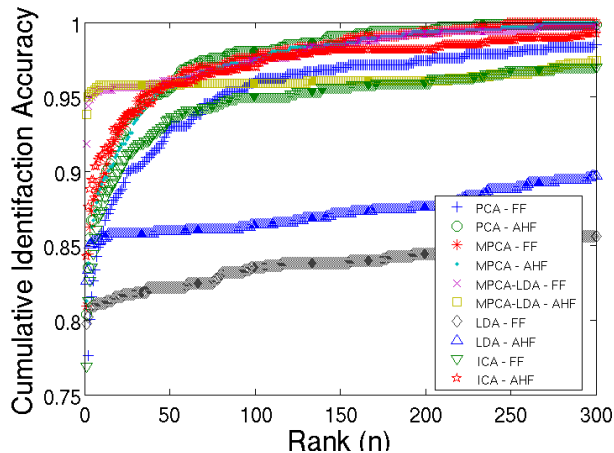
Figure 7. CMC plot of Full Face and Average-Half-Face on 3D Face database (C).

Therefore, with a simple computation step, the accuracy of the majority of the algorithms tested was improved. We believe that this gain in accuracy has it's origin in the averaging operation, which produces a new face that contains a set of features that are more discriminatory that those of the full face. More work must be done to verify this claim and to complete the picture of the origin of this accuracy gain.

It is important to note that in the results in Table 1 may not be the best accuracy possible for each algorithm because some of the algorithms' parameters can be fine tuned depending on the data set. We utilized each algorithm to compare the accuracy of using the average-half-face and the full face, since we were interested in their relative accuracies, not their absolute accuracies.

7. Conclusion

We have shown that with simple calculation of the average-half-face computed from the full face, the rank-1 accuracy of recognition is improved in most cases, regardless of database or algorithm utilized. Future work on this topic begins with a deeper analysis of the source of the accuracy gain of the average-half-face. We would also like to apply the average-half-face to facial feature extraction methods, such as those using wavelets. Naturally, applying the average-half-face to additional algorithms and databases (2D and 3D) and analyzing the effect of illumination, facial expressions, occlusions and other difficulties would be helpful in identifying the most useful applications of the method.

Acknowledgments

References

[1] Yale Univ. Face DB, 2002. `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.

[2] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13:1450–1464, 2002.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 45–58, London, UK, 1996. Springer-Verlag.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] S. Gupta, J. K. Aggarwal, M. K. Markey, and A. C. Bovik. 3D face recognition founded on the structural diversity of human faces. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–7, 2007.

[7] J. Harguess, S. Gupta, and J. K. Aggarwal. 3D face recognition with the average-half-face. *International Conference on Pattern Recognition ICPR*, 2008.

[8] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE Trans. on Neural Networks*, 19(1):18–39, 2008.

[9] A. Martinez and R. Benavente. The AR face database. *CVC Technical Report*, (24), June 1998.

[10] G. Pan and Z. Wu. 3D face recognition from range data. *Int. J. Image Graphics*, 5(3):573–594, 2005.

[11] N. Ramanathan. Facial similarity across age, disguise, illumination and pose. In *Proceedings of International Conference on Image Processing*, 1999.

[12] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Comput.*, 12(5):1207–1245, 2000.

[13] M. I. Shah and D. C. Sorensen. A symmetry preserving singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 28(3):749–769, 2006.

[14] G. Shakhnarovich, G. Shakhnarovich, B. Moghaddam, and B. Moghaddam. Face recognition in subspaces. In *S.Z. Li, A.K. Jain (Eds.), Handbook of Face Recognition*, pages 141–168. Springer, 2004.

[15] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[16] L. Zhang, A. Razdan, G. Farin, J. Femiani, M. Bae, and C. Lockwood. 3D face authentication and recognition based on bilateral symmetry analysis. *The Visual Computer*, 22(1):43–55, 2006.

[17] W. Zhao and R. Chellappa. Illumination-insensitive face recognition using symmetric shape-from-shading. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1286, 2000.