Face Recognition Approaches and Implementation of Eigenfaces Based Recognition

Bhuwan Mehta (Y5827144)
Rahul Gupta (Y5340)

Feature-Based Face Recognition

Preprocessing Steps

- Segmentation: To eliminate the background
- Scaling: Performance decreases quickly if the scale is misjudged. (Can be controlled)
- Rotation: Symmetry operator to estimate head orientation.

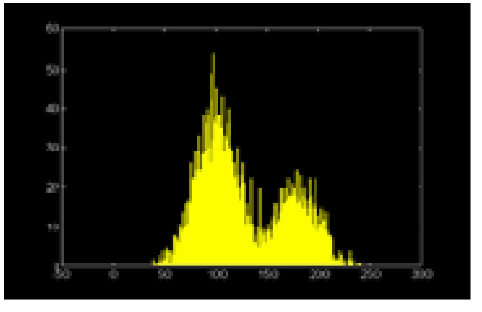
Segmentation

- Dividing images into meaningful regions that appear to be images of different surfaces.

- Approaches :
  - Histogram based segmentation
  - Spatial Coherence based Segmentation
    - Divide (split) or merge existing regions

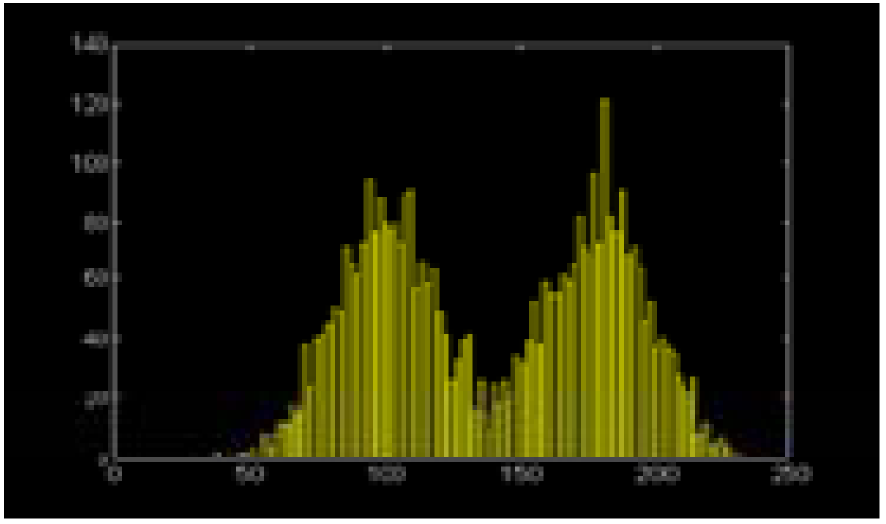
Histogram-based segmentation

- Gray level Image -> Binary mask using a Threshold
- Threshold can be found using Histogramming
- Gray-level histogram gives the number of cells having a particular gray-level



Histogram-based segmentation

- Ideally, object & background have constant different brightness inside their regions. Put a threshold between peak values in histogram.

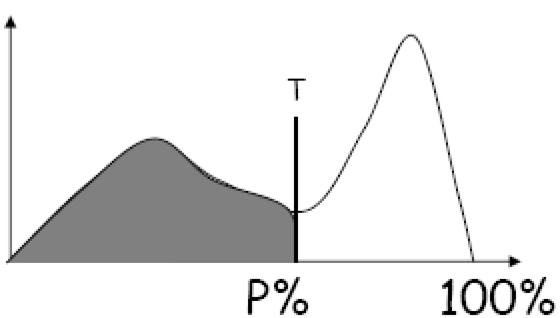


Histogram-based segmentation

- Problems:
  - Size differences between object and background
  - measurement noise
  - non-uniform illumination
  - non-uniform reflection from the surfaces
- Effect:
  - brightness is not constant; there is some spread

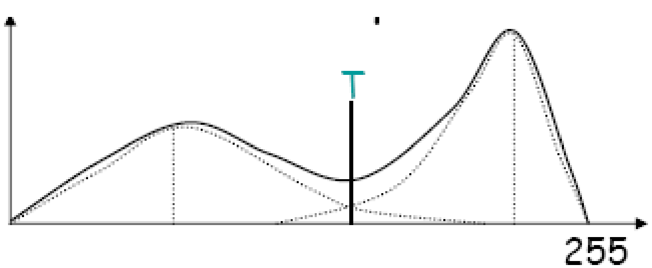
Histogram-based segmentation

- Solutions:
- P-tile method
  - Use the a priori knowledge about the size of the object :
    - Assume an object with size p
    - Choose the threshold such that %p of the overall histogram is determined



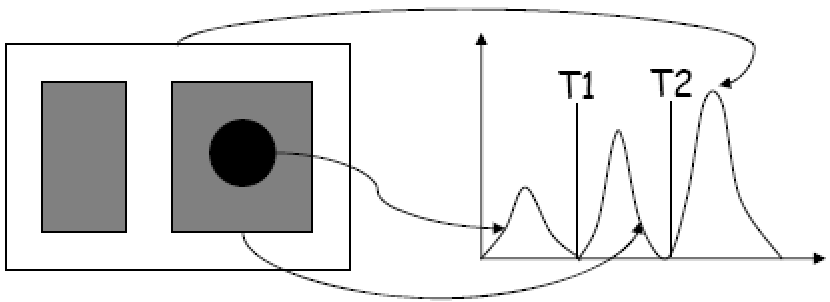
Histogram-based segmentation

- Mode method
  - Find the “peaks” and “valleys” of the histogram
  - Set threshold to the pixel value of the “valley”
- Non-trivial to find peaks/valleys :
  - Ignore local peaks, choose peaks at a distance
  - Find the valley between those peaks
  - Maximize “peakiness” (difference btw peaks & valleys) to find the threshold as valley



Histogram-based segmentation

- Double Thresholding:
  - Starting from a conservative initial threshold T_1 determine the “core” parts of the object
  - Continuing from this core part, grow this object by including neighboring pixels which are between T_1 and T_2

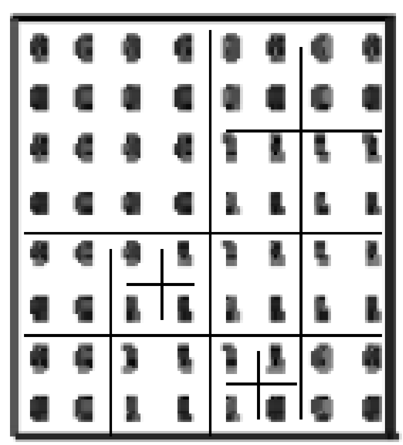


Spatial Coherence based Segmentation

- The dependency between neighboring pixels is taken into account.
- Neglecting this dependency may cause "salt-n-pepper" noise in the resulting image. (Randomly occurring black and white pixels)

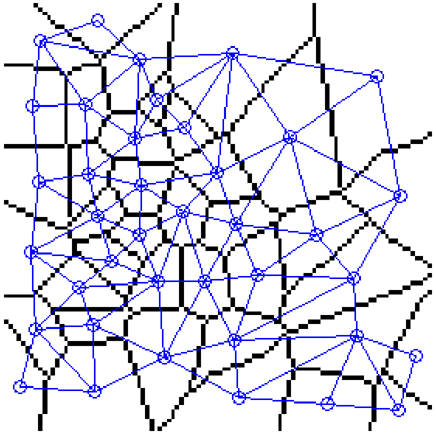
Spatial Coherence based Segmentation

- Region Representation:
  - Array representations : masks
  - Merging & Splitting based on:
    - Threshold
    - geometrical attribute : common boundary length



Spatial Coherence : Merging

- Region Adjacency Graphs (RAG) is created.
- Two regions are considered as neighbor if they are separated by a black (i. e. with color 0) pixel in the horizontal or vertical direction
- Matlab : ADJ = imRAG(IMG);



Spatial Coherence : Merging

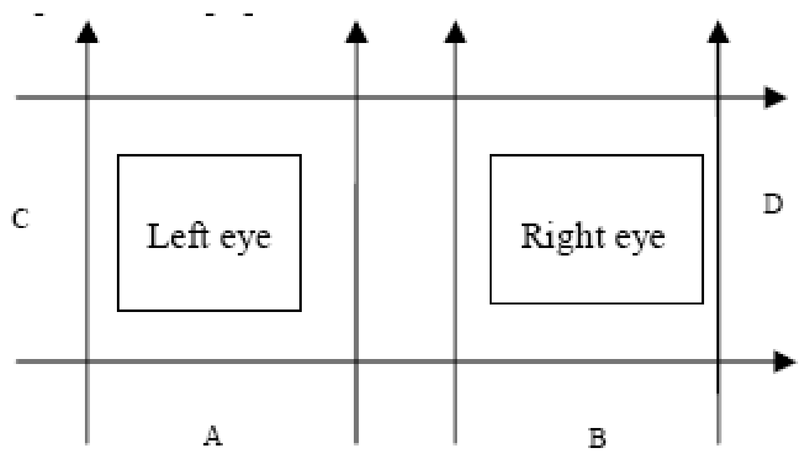
- A general region merge algorithm: Beginning from an initial segmentation, prepare an initial RAG
- For each region check whether its neighboring regions are similar, if so, merge these regions & modify RAG.
- For region similarity :
  - Compare their mean intensities and check with a predetermined threshold
  - Check “weakness” of the common boundary
    - weak boundary: intensities on two sides differ less than a threshold

Face Detection - Eyes Extraction

- Darker region
- Little gray value
- Gray changes greatly around eyes region so grads value of each point is also higher
- Eyes and skin have many differences so boundary detection is implemented in the candidate facial region and projected horizontally.
- Then the eye’s horizontal position in A and B is identified. Then, it is projected vertically above A and B and the first peak position is identified as C and D.

Face Detection - Eyes Extraction

- Eye's outline and left and right canthus are located in two areas made up of A,C and B,D.
- The mean of these two regions are considered as the position of pupils.



Face Detection - Mouth Extraction

- take lip color into account.
- In the bottom of face, we can consider the regions which satisfy the condition below as mouth:

$$\theta = \arccos\left(\frac{0.5 \times (2R - G - B)}{\sqrt{(R-G)(R-G) + (R-B)(G-B)}}\right), \theta < 0.2$$

Face Detection - Nose Extraction

- If the distance between two pupils is seen as 1 then the distance from nose to the middle of eyes is from 0.7 to 1.
- The darker region around this is the position of nostrils
- Greatest luminance point above the areas of two nostrils is identified as the top of the nose.

Holistic Matching

- An alternative to Feature based recognition is the Holistic Matching approach in which the machine automatically determines which features to use.
- These types of approaches can deal directly with complex, real-world images because the system is general and adaptive.
- The efficient selection of good features, however, is an important issue to consider.
- In this type of matching, we project the higher dimension image vector onto a lower dimension feature space constructed on training data.
- Various dimensionality reduction techniques are used .

Various Techniques of Holistic Matching

- In this lecture, we will be discussing the following two major holistic matching/recognition techniques based on different dimensionality reduction techniques:
  - Eigenfaces technique based on Principle Component Analysis(PCA).
  - Fisherfaces Technique based on Linear Discriminant Analysis(LDA).

Eigenfaces

- Suppose, there are K images in a data set with each image represented as a column vector X_i for $1 \le i \le K$.
- Let $\bar{X}$ be the mean of all X_i.
- Now consider a matrix $U = U_1 U_2 \ldots\ldots U_K$, where $U_i = X_i - \bar{X}$.
- Our goal is to calculate the eigenvectors of the covariance matrix UU^T. This cannot be done directly as the size of UU^T is N*N, which is very large even w_i for small sample images.
- The eigenvectors of UU^T can be however found by using a simple trick of linear algebra and using a linear combination of eigenvectors of $U^T U$ whose size is K*K and in practical situations, K<<N.
- The eigenvector w_j can be easily computed by using the

following formula:

$$w_j = \frac{\sum_{l=1}^{K} U_l E_{lj}}{\sqrt{\lambda_j}}$$

where E_{lj} is the l'th value of the j'th eigenvector of $U^T U$ and λ_j is the corresponding eigenvalue.

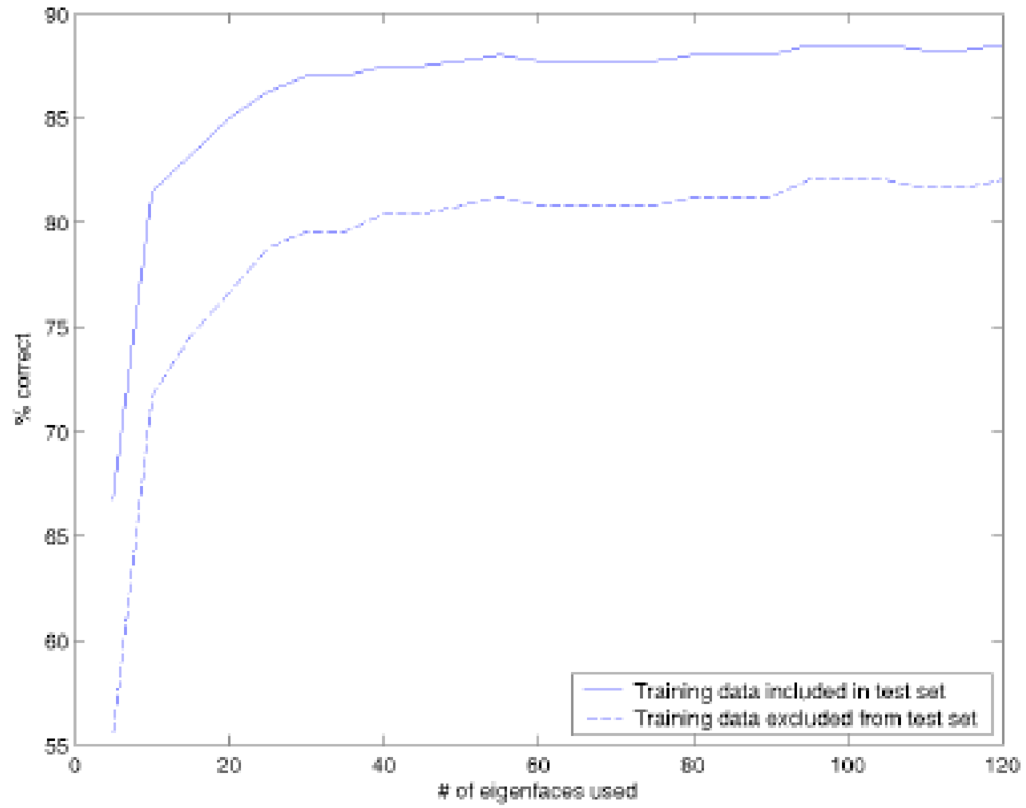
In this figure, averaging of two faces is shown

Figure shows nine eigenfaces generated from a face database. Each of these images represents the image interpretation of one of the w_j calculated previously. As one can see**,** the images appear almost as ghosts, each with a different portion of the face accented.

Significance of Eigenvectors/Eigenfaces

- Now that we have generated the eigenvectors for the covariance matrix of the differences faces, let us evaluate what we have actually created. An eigenvector whose corresponding eigenvalue is of greatest magnitude represents the direction of greatest variance in a covariance matrix.
- The eigenvector corresponding to the second largest eigenvalue represents the direction of greatest variance in the covariance that is perpendicular to the first eigenvector. This continues for all of the eigenvectors, as we have ordered them by the magnitude of their corresponding eigenvalues, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$.

- Since, the eigenvectors are perpendicular, so what we are essentially doing is creating a coordinate system (which we will refer to as face-space) that has the most possible discriminating power for the vectors we used in creating it.
- This means that when an image is represented in face space, it is really stored as a vector of coefficients that indicate how much each eigenface is to contribute to the final image.
- Since the eigenfaces have been ranked by their discriminating ability, it is not necessary to use all of the eigenfaces generated in classification. It is possible to only consider a small subset of the best eigenvectors and still maintain discriminating power.
- The actual number of eigenfaces needed is not known, and needs to be determined experimentally or taking those eigenfaces whose sum energy is the major portion of the total energy.



The number of eigenfaces used vs. success in classification for a database. For this database, the optimum number of eigenfaces comes out to be around 40.

Projecting Faces into Face Space

- Any image Z can be projected into face space by using the following formula:

$$W' = W^T (Z - \bar{X})$$

- Put simply, the vector of weights is found by multiplying the transpose of the matrix *W*(*W is* formed by letting each eigenface form a column of the matrix) by a vector that is found by subtracting the average face image ($\bar{X}$, a column vector) from a sample or test image (*Z*, a column vector).
- Note that, Z could be any training or test image converted into column vector.
- Now that a method of projecting images into face space has been defined, the problem of face recognition becomes one of everyday pattern recognition.

Reconstruction of image from Eigenfaces.

- The eigenface recognition method was derived from work on analyzing the loss of information by representing faces through weights of basis-faces.
- The basis faces were found through principle component analysis techniques. Because of this, it is possible to reconstruct an original face image from the known eigenface weights.

$$Z' = WU^T + \bar{X}$$

Images and their reconstruction from face-space representations; original image on far left; 5, 10, 50, 100, 150, 200 eigenface reconstructions from left to right. The three rows are the cases of known subject, unknown subject and not a face respectively.

Face Recognition Procedure

- All the training sample images have been projected onto face space. Form a cluster of all the images belonging to the same class (images of same user).
- Now, if a test image is fed as an input, we should first project the test image onto face space by the procedure discussed earlier.
- There can be 3 outputs possible for a test image as given below,

  1. **Not a Face**: It can be classified as ‘not a face’. The interpretation of “not a face” in the eigenface system is that the projection of an image into face-space not only does not yield a vector close to any know clusters formed by a single individuals, but it is also significantly far away from all clusters.

2. **Unknown Face**: The “unknown face” classification indicates that a test image contains a face, but the face is not recognized by the classification system. This classification is used to indicate that the face presented to the classification system does not closely match any face images on which it has been trained.

If one wishes the classification system to learn to recognize new individuals, the face space vectors of unknown faces could be recorded and then unsupervised clustering methods could be employed to attempt to recognize unknown recurring individuals.

3. **Recognized Face**: The “recognized face” classification indicates that the face recognition system was able to find a known individual that was sufficiently similar to the one presented in the test image. Along with the “recognized” classification, the system then provides the identity of the individual in the test image.

Salient Features of Eigenfaces Method:

- Run-time performance is very good.
- Construction: computationally intense, but need to be done infrequently.
- Need to rebuild the eigenspace if adding a new person.
- Starts to break down when there are too many classes.
- Retains unwanted variations due to lighting and facial expression.

Need for another approach?

- Although the Eigenfaces projection is well-suited to object representation, the features produced are not necessarily good for discriminating among classes defined by the set of samples.
- The Eigenfaces method describes some major variations in the class, such as those due to lighting direction; these variations may well be irrelevant to how the classes are divided.

The variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity. Here is an example of such a case, when the same person seen under different lighting conditions can appear dramatically different.

Fisherfaces/Fisher's Linear Discriminant

- Eigenfaces achieves larger total variance, FLD achieves greater between-class variance, and, consequently, classification is simplified.
- Let W be a projection matrix that projects a vector into the MDF subspace. Vector $Z = WY$ *is a new feature vector from samples of c* classes with class means M_i *, i = 1, 2, ..., c.*
- We can then define a within class scatter matrix as,

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{n_i} (Y_j - M_i)(Y_j - M_i)^T$$

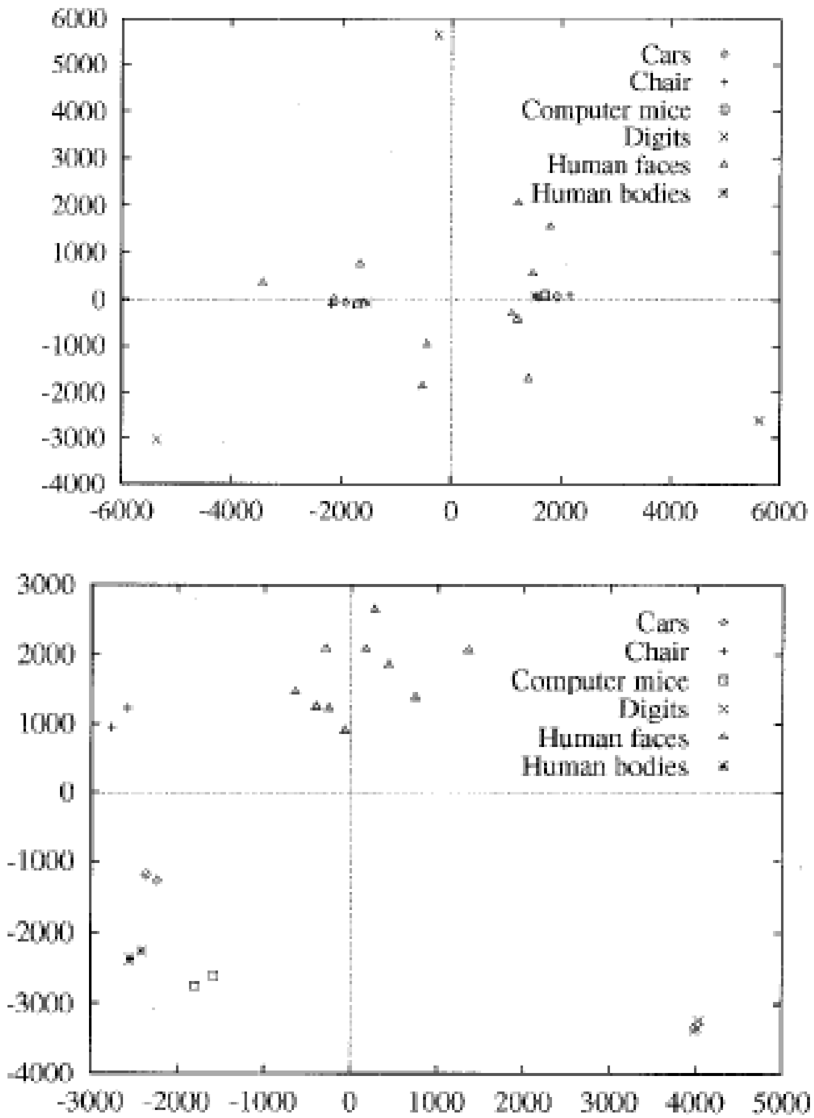
for n_i samples of class i.

- For a grand mean vector *M for all samples from all* classes, the between-class scatter matrix is defined as

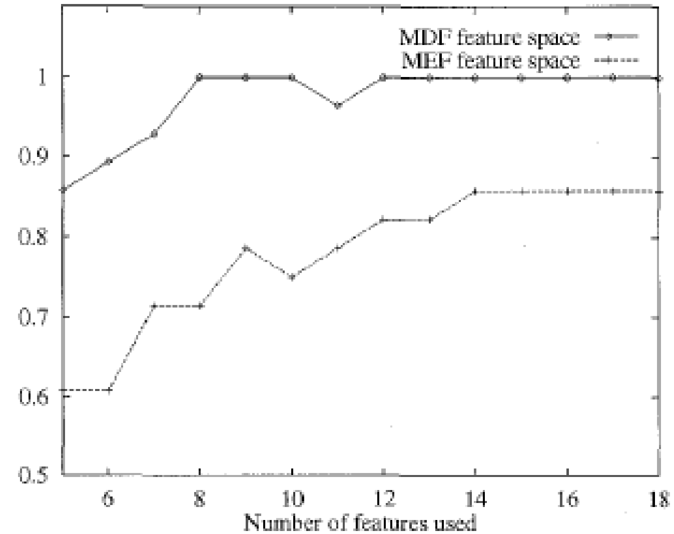
$$S_b = \sum_{i=1}^{c} (M_i - M)(M_i - M)^T$$

- In discriminant analysis, we want to determine the projection matrix *W that maximizes the ratio* $\frac{\det(S_b)}{\det(S_w)}$. In other words, we want to maximize the between-class scatter while minimizing the within-class scatter.
- It has been proven that this ratio is maximized when the column vectors of projection matrix W are the eigenvectors of $S_w^{-1} S_b$ *, associated with the largest eigenvalues.*

- FLD tries to project away variations in lighting and facial expression while maintaining discriminability.
- After getting the Fisherfaces or the Eigenvectors, remaining procedure is the same as described in the case of Eigenfaces.



Distribution of some samples using the best two features in the PCA and the FDA based spaces respectively. In the FDA based subspace, objects of the same class are clustered much more tightly than in the PCA based space



The performance of the system for different numbers of MEF(Eigenfaces) and MDF(Fisherfaces) features, respectively. The number of features from the subspace used was varied to show how the MDF subspace outperforms the MEF subspace. 95% of the variance for the MDF subspace was attained when 15 features were used; *95% of variance for the MEF* subspace did not occur until 37 features were used. Using 95% of the MEF variance resulted in an 89% recognition rate, and that rate was not improved using more features

Recent Developments

- One of the most recent methods that have shown up in the face recognition literature is a method based on blood vessels.
- Researchers have discovered that the skin is slightly warmer around blood vessels, so using pictures of people taken with infrared cameras or heat cameras, the blood vessels graph of the face appears.
- Thereafter, pattern matching can be done as discussed in the lecture.
- Accuracy is supposed to increase manifolds as this type of face graph would be indifferent to light illumination conditions, facial expressions, etc.

Bibliography

- J.R. Bach, S. Paul, and R. Jain, ”A Visual Information Management System for the Interactive Retrieval of Faces,” *IEEE Trans. Knowledge and Data Eng., vol. 5, no. 4, p. 619-628, Aug. 1993.*
- Face Recognition Technology program. (http://frvt.org/FERET/).
- B. Moghaddam, C. Nastar, A. Pentland. A Bayesian Similarity Measure for Direct Image Matching. In *Proceedings of Internatinal Confernce on Pattern Recognition, pages 350-358, Vienna, Austria, 1996.*
- J. R. Solar, P. Navarreto, " Eigen space-based face recognition: a comparative study of different approaches, IEEE Tran. , Systems man And Cybernetics- part c: Applications, Vol. 35, No. 3, 2005.
- A. M. Martinez, A. C. Kak, “PCA versus LDA”, IEEE Trans. Pattern Anal. Machine Intell, Vol. 23, pp. 228-233. 2004