Median Rank in Face Recognition

Hafþór Guðnason
Master of Science
June 2006

Supervisor:
Björn Þór Jónsson
Associate Professor

Reykjavík University - Department of Computer Science

M.Sc. Project Report

MEDIAN RANK IN FACE RECOGNITION

by

Hafþór Guðnason

Project report submitted to the Department of Computer Science at
Reykjavík University in partial fulfillment of the requirements
for the degree of
Master of Science

June 2006

Project Report Committee:

Dr. Björn Þór Jónsson, Supervisor
Associate Professor, Reykjavik University, Iceland

Dr. Laurent Amsaleg
IRISA-CNRS, France

Dr. Hákon Guðbjartsson
deCode, Iceland

Abstract

In recent years, the demand for face recognition systems has increased very much. Many researchers have tried to make reliable and accurate face recognition systems and developed many methods to do face recognition. One of the methods that have proved to be among the best is Elastic Bunch Graph Matching (EBGM). The drawback of EBGM is the slow distance calculations used to return good results. To increase the speed of the algorithm we tried to use a variant of the Median Rank algorithm instead of current distance calculation methods. The Median Rank with normalized data returned better results, and in less time, than the best EBGM distance methods.

Útdráttur

Á síðustu árum hefur eftirspurnin eftir kerfum sem geta borið kennsl á einstaklinga aukist mjög mikið. Samfara aukinni eftirspurn hafa fjölmargar rannsóknir verið gerðar, þar sem reynt hefur verið að smíða fullkomið andlitsmyndaleitarkerfi. Niðurstöðurnar hafa verið misjafnar en þó eru nokkur kerfi komin í notkun, til dæmis á stórum flugvöllum. Ein af þeim aðferðum sem hefur reynst hvað best er EBGM (teygjanleg bunka grafs greining) en gallinn við hana er hversu hægvirka fjarlægðarútreikninga hún notar til að skila sem bestum niðurstöðum. Til að auka afköstin prófuðum við að nota afbrigði af Median Rank algorithma í stað núverandi fjarlægðarútreikninga. Að nota Median Rank og staðla gögnin skilaði betri niðurstöðum, og á mun skemmri tíma, en upphaflegu aðferðirnar sem skiluðu bestu niðurstöðunum.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Face Recognition Applications

In the past, much research has been performed in the field of face recognition. This problem has received attention from people in many different disciplines such as medicine, engineering, computer vision, artificial intelligence and databases.

One reason why researchers have spent time in this field is that there has been a high demand from security services, such as police, army and customs, for a system that is able to recognize people. Applications for such recognition include databases of criminals and wanted or missing persons. Since these databases are typically very large, it is impossible for any user to memorize every face. Therefore, it is necessary to have machines aid users in the recognition process.

Another motivation for research in the face recognition field is the authentication problem. These days, most people in the western world have several access codes and credit cards with pin numbers, as well as login names for computer systems. In a security survey done by Price Waterhouse Coopers in the United Kingdom, they discovered that the average employee in British companies must remember three different usernames and passwords to be able to do his or her job [DTI06]. Personal access codes are not included in these figures.

Although some of these codes are necessary for security reasons, people tend to forget them, which leads to extra work or cost in getting a new one. In order not to forget these codes people may compromise security, e.g. by writing their pin numbers on their credit cards or using the same code for everything. As a result, the security that was supposed to stem from all these numbers is lost. To counter the lack of security associated with pin numbers, face recognition has been proposed as a means to authenticate persons. The advantages are that no numbers of authentication units are needed, only the face.

1.2 Face Recognition Methods

The FERET (Face Recognition Technology) [Feret] program was established to urge researchers to develop automatic face recognition systems. It was sponsored by the Department of Defence (DoD) Counterdrug Technology Development Program Office which also sponsored many different research groups that tried to develop usable face recognition applications.

These groups have invented many different methods for face recognition. To compare these methods the FERET program made a comparison protocol, including a database of facial images, so all the groups could test their applications with the same database and use the same test procedures. With an interval of a few years, these groups get together and test their products. These tests are called Face Recognition Vendor Tests (FRVT). To make it easier for researchers to start working on face recognition projects, Colorado State University distributes the source code for some of the most common algorithms in face recognition.

Some of the methods which have been tested in the FRVT are based on similar ideas. E.g. both principal component analysis (PCA) [Moghaddam96b] and linear discriminant analysis (LDA) [Zhao98] are based on eigenfaces (or Fischer faces as they are also known as). PCA and LDA have also been combined into a new method, in order to improve the recognition rate. One of the best methods in the FRVT tests is Elastic Bunch Graph Matching (EBGM) [Bolme03, Zhao03]. The EBGM method is based on the idea of finding several special points in faces, called landmarks. These points are for example the nose, right eye, left eye etc. Many various filters are convolved with these landmark locations to extract data for the landmarks. All these landmarks are then collected into a face graph. The face graphs for all facial images form a database which can be searched.

Most of these face recognition methods have in common that they first perform some operation on each image and store the result in a database. Then they use a distance functions to compare a query image to all the other images and find the best matches. The distance functions can be as simple as Euclidean distance to more advanced methods. The quality of results differs very much based on the distance function that is used. The rule of thumb is that more complicated (and slower) method, the better the result is.

In 2003, Fagin proposed using Median Rank distance for similarity retrieval [Fagin03]. Median Rank is not a real distance function. Instead it is a ranking method that ranks its data into rank positions. The nearest value is ranked in 1 and so forth. It is an approximation to other distance functions which has been observed to give good results in many cases. The algorithm is relatively simple and efficient. One of the biggest advantages of using Median Rank is that a single extreme value in one dimension does not affect the whole result, unlike e.g. Euclidean distance. Furthermore, recent research at Reykjavik University has demonstrated the scalability of Median Rank, using the PvS-index [Lejsek05].

1.3 Contributions of this project report

The goal of this project was to see whether Median Rank distance could improve the quality of the results for the EBGM algorithm and speed up the search. Because EBGM defines a face graph as a series of landmarks, and each landmark has many dimensions, we had to implement the Median Rank method in two steps: One step for the dimensions and another step for the landmarks. This two step Median Rank had not been implemented before and therefore it was interesting to see how it would perform in terms of speed and recognition rate.

The Median Rank method, on top of the EBGM algorithm, did not give good results; in fact they were much worse than expected. By investigating the results, we proposed that normalization would be a feasible technique to improve the results. We experimented with four different normalization methods.

- Normalizing by dimensions.
- Normalizing by landmarks.
- Normalizing by dimensions and then by landmarks.
- Normalizing by landmarks and then by dimensions.

The results showed that with the two latter approaches, the recognition rate is better than in any other method distributed by Colorado State University. While Median Rank was not the fastest method, the methods provided by CSU that ran faster returned much worse results.

1.4 Overview of the project report

The remainder of this project report is organized as follows. In Chapter 2 we give a brief background to face recognition methods and their performance. The Median Rank algorithm is then described in Chapter 3. In Chapter 4 we present the adaptation of the Median Rank algorithm and the EBGM database. In Chapter 5 we present the results of our experiments. Finally, in Chapter 6 we conclude.

2 Face Recognition Background

Researchers have developed many methods for face recognition. In order to facilitate their comparison, the Face Recognition Technology (FERET) program was established. The FERET protocol consists of a database of facial images of more than 1,200 persons, as well as search sets designed to test different aspects of face recognition methods.

Three Face Recognition Vendor Tests (FRVT) have been held (in 2000, 2002, and most recently in 2006), where researchers were invited to evaluate their algorithms and/or products against the competition. While this process has been useful for researchers and possible buyers to see which methods perform well and what their strengths and weaknesses are, it has also helped to define the requirements for future work.

Researchers at Colorado State University have collected most of the face recognition methods that have performed well and distributed them free of charge. In the remainder of this chapter, we first review the performance of the three main methods in this distribution, and then we describe each of these three methods briefly. Finally, we discuss some very recent developments in this field.

2.1 The FERET Protocol

The FERET program was established to stimulate researchers in the field of face recognition [Feret]. The ultimate goal of the program was to equip police and secret services with a fully automated application for recognizing people.

One of the first things that were done in this program was to build a database of human faces [Feret04]. The FERET database consists of facial images of around 1,200 persons. All the persons have more than one image in the database, which differ in illumination, appearance, rotation of the head and more. These images were taken over a period of a few years, so one person can either have only images taken on the same day or images taken with longer time in between. The database contains a total of 3,360 images. The original FERET database was in greyscale, but recently a colour database was also made available. The FERET program distributes this database for free to facilitate research in face recognition.

The FERET program was also intended to collect information about the quality and performance of face recognition methods. The FERET program made a protocol that defines test cases to evaluate face recognition systems, and held Face Recognition Vendor Tests (FRVT), where researchers were invited to come together and evaluate their designs.

In order to be usable, face recognition systems must return accurate results in real time. In one of the FRVTs, three methods were judged to be the most competitive, showing different results for different search sets [Zhao03]:

- Probabilistic Eigenface system [Moghaddam96b], uses principal component analysis for dimensionality reduction and making of eigenfaces. The recognition process is done with distance calculations between the eigenfaces.

- The Subspace Linear Discriminant Analysis system [Zhao98], uses linear discriminant analyses and principal component analysis to create eigenfaces. Distance functions are used to compare the eigenfaces.
- The Elastic Bunch Graph Matching [Wiskott99]. Creates face graph for every facial image, with filter convolution on certain points. The recognition process is done with distance calculations between the face graphs.

Colorado State University has collected most of the face recognition methods that have performed well in this FRVT test together. They are also distributing them free of charge to other researchers that want to experiment in the face recognition area. Some of these methods are improved by them but others are just for comparison. We have used the code distributed by CSU to generate the results displayed in Figure 1 to 4. Each figure shows the result quality of these three methods for one of the four search sets defined in the FERET protocol. For the EBGM method, three different variants are shown, which differ in their similarity estimation. In each figure, the Y-axis represents the cumulative fraction of correct matches at a particular rank (or better), while the X-axis represents the rank. The goal in all the methods is to have as many correct matches as possible, as early as possible; a perfect system would have 100% recognition at rank 1.

Each test case is based on two sets, the search set and the database set. For every image in the search set, each algorithm tries to find an image of the same person in the database set. The images in the database set are taken with good illumination, people looking into the camera with a straight face. The search sets are assembled according to the test cases that they illustrate. In the following, each search set is described along with its results.

- **FAFB:** The 1,195 images in the FAFB search set are taken immediately after the database images. The only difference is the facial gestures as people are typically smiling or frowning in the search set images. This is the easiest search set and does not represent reality well. The results for the FAFB search set are shown in Figure 1. As the figure shows, all algorithms perform well on this search set. The LDA method, which performs the worst, correctly identifies the correct image in over 70% of the cases and displays the correct image within the top ten images in over 80% of the cases. The EBGM methods return the best results, with EBGM Narrowing performing slightly better than the alternative variants. All EBGM variants identify the correct image in over 90% of the cases.
- **FAFC:** The 194 images in the FAFC search set are taken in bad light condition, so this test case is focusing on how illumination affects the recognition rate. The results for the FAFC search set are shown in Figure 2. As the figure shows, the quality of the results drops very much in this case, especially for the PCA method, which has a recognition rate of less than 18% at rank 10. The result quality of the other methods does not drop as much, and EBGM Narrowing is the best one with recognition rate around 75% at rank 10.
- **DUP1:** The 722 images in the DUP1 search set are taken between one minute and three years after the database images. The result for this search set is displayed in Figure 3. The worst method in this test case is PCA with correct matches around

53% at rank ten, but it is only about 15% worse than the best method, the EBGM Narrowing.

- **DUP2:** The 234 images in the DUP2 search set are taken between eighteen months and three years after the database images. This set is subset of DUP1 and the results are displayed in Figure 4. This is a very difficult test case and that is reflected in the results, which are worse than in all the other cases. As in the other test cases EBGM Narrowing search is the best search method with recognition rate around 59% at rank 10 while LDA is the worst method with recognition rate of only 30% at rank 10.

Overall, the most interesting effects in the results are 1) how badly PCA performs with different illumination, and 2) how much time affects the results. In almost all cases, the EBGM variants and EBGM Narrowing in particular, perform the best.

In all the face recognition methods, distance calculations are used to compare the facial images. These calculations determine how fast the search is and how good the results are. The rule of thumb in distance calculations is that the better the result is, the more time the calculations take. But of course there are exceptions to that and we want to find a good, reliable method that is also fast. The simplest distance function is Euclidean distance, which is based on the Pythagorean theorem. The formula for Euclidean distance between points $P=\{p_1, \ldots, p_n\}$ and $Q=\{q_1, \ldots, q_n\}$ is

$$D(P,Q) = \sqrt{\sum_{i=0}^{n} (p_i - q_i)} \qquad (1)$$

The biggest drawback of Euclidean distance is that one single extreme dimension has a significant impact on the results.

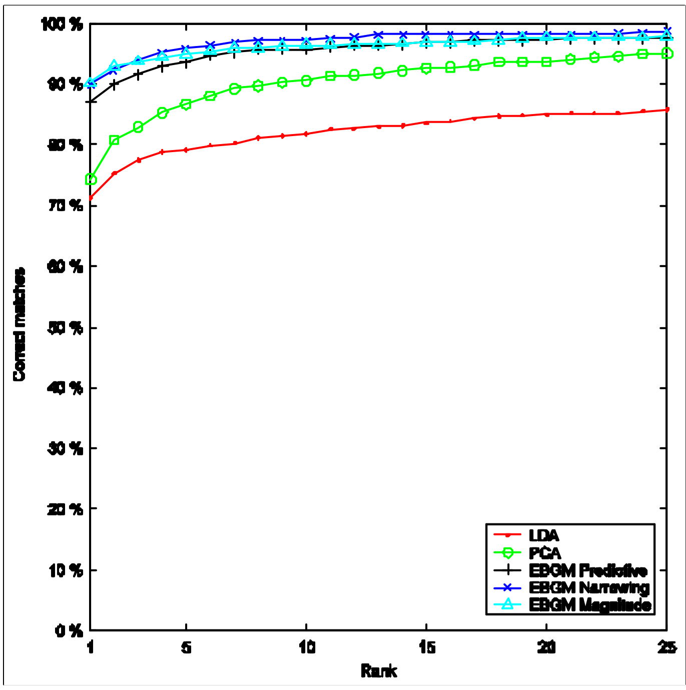


Figure 1. Search set FAFB.

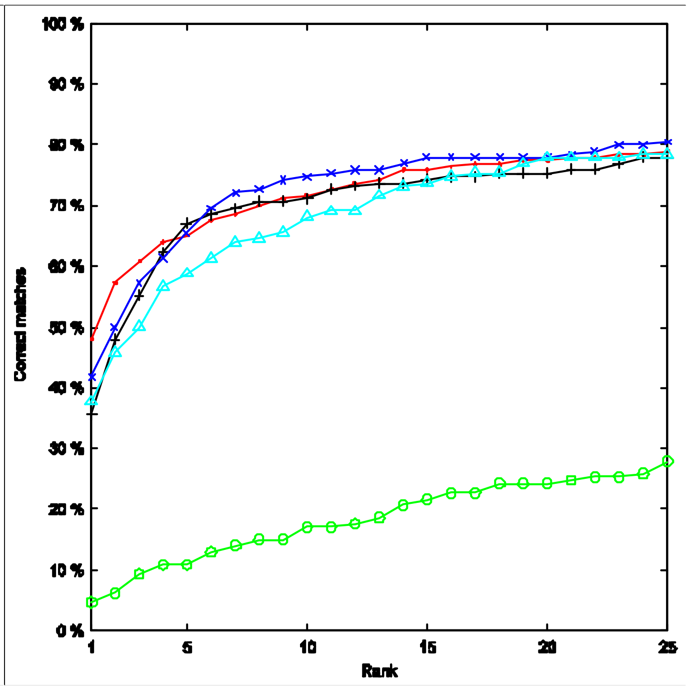


Figure 2. Search set FAFC.

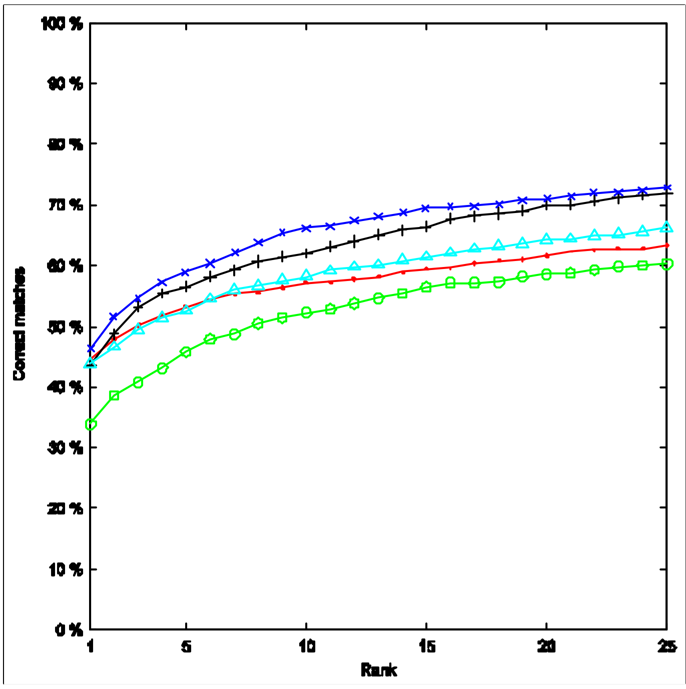


Figure 3. Search set DUP1.

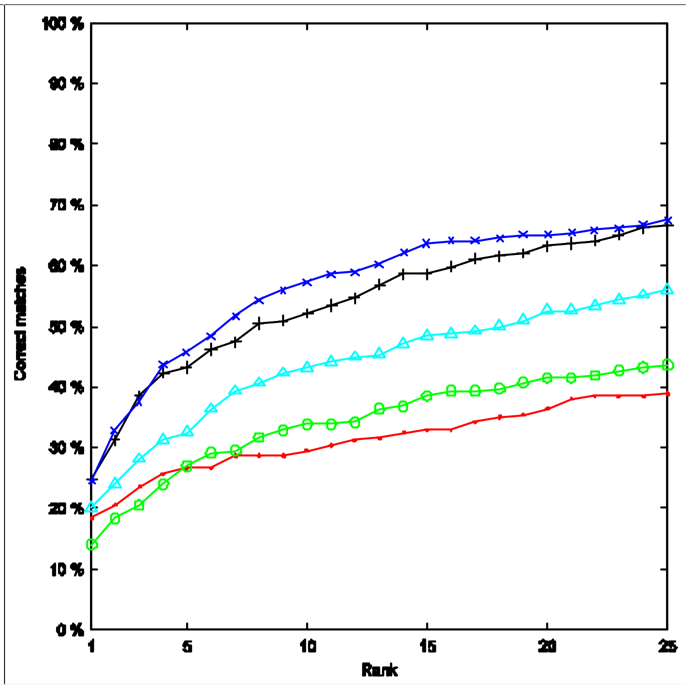


Figure 4. Search set DUP2.

The three EBGM variants use different distance functions, which are described in Section 2.4. In Section 3, we describe yet another distance function based on Median Rank, which we have applied to the EBGM method in the remainder of this project report. Before turning to Median Rank, however, we describe the PCA, LDA and EBGM methods briefly.

2.2 Probabilistic Eigenfaces (PCA)

Probabilistic eigenfaces are based on the idea that you can approximately rebuild a particular facial image using an average facial image along with information about what differs from the average [Moghaddam96b]. This idea comes from the information theory, where meaningful information is collected together but other information is discarded. Therefore we want to extract the meaningful information (the differences from the average) from all the images and only compare these differences, resulting in a faster and more effective search. To achieve this goal we use Principal Component Analysis.

Principal component analysis (PCA) is a feature extraction method, based on linear transformation that has been used in data analysis projects, e.g. face recognition, audio recognition [Gold00] and more. The main goals of PCA are two. The first is to identify patterns in data by maximizing variance between components (patterns) and the second is to reduce the dimensionality of the data set while not losing much of the information.

The PCA algorithm consists of a few steps. The first step is to create a data matrix of the pixel values in each facial image. Then we find the mean of each dimension and subtract it from each value in that particular dimension, producing a data set whose mean is zero. The second step is to calculate the covariance matrix C with the following formula:

$$C^{n\times n} = (c_{i,j}, c_{i,j} = \operatorname{cov}(Dim_i, Dim_j)) \qquad (2)$$

Dim_x is the x^{th} dimension. The third step in the process is to calculate eigenvalues and eigenvectors of the covariance matrix. The fourth step is to form feature vectors by choosing components. The eigenvector with the highest eigenvalue is the principal component. The eigenvectors should be ordered by the eigenvalues in descending order to form the feature vector matrix. Eigenvectors with low eigenvalues can be dropped. The fifth and final step is to create the principal component dataset, which is done by multiplying the transposed feature vector matrix with the original data set transposed [Smith02, Hollmen96, PCA].

In the face recognition problem the feature vector matrix is calculated for only a small fragment of the total database, called the training set. Then, after the feature vector matrix has been created for the training set, it is multiplied by the whole database and the principal component dataset made [Turk91].

When the eigenvectors are displayed they look like faces, which is why they are called eigenfaces. The last step in the probabilistic eigenface algorithm is the face recognition. It is done by performing distance calculations on the eigenface vectors. Euclidean distance is the simplest distance calculation method and therefore it is often used. The eigenface vector with the least distance in between is expected to be of the same person.

One of the greatest advantages of using the probabilistic eigenface method is that it discards less relevant data, which leads to faster retrieval. If the eigenfaces are stored to disk, they will also require less space.

In the FERET test, PCA was one of the best methods tested. But when we display just the best methods we observe that it does not perform as well as both LDA and EBGM. PCA works especially poorly for search set FAFC. The advantage of the method is that it is relatively simple and efficient.

2.3 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) has been used in face recognition systems, as well as other systems with complex data, for data classification and dimensionality reduction. The goal of LDA is to reduce the difference inside classes while increasing the difference between them. In face recognition systems class represents person, but the class consists of many images per face. Obviously it is a drawback of the method that it relies on many images per person. Therefore Zhao et al. [Zhao99] combined the Linear Discriminant Analysis with the PCA algorithm to try to increase the recognition rate of the faces using few images per person.

The PCA+LDA algorithm works in the following way. First it uses PCA (described in Section 2.2) to reduce the dimensionality of the feature vectors and then LDA is run to produce a linear discriminant function that maps the input into a classification space where the faces can be classified [Zhao98, Zhao99, Beveridge03] by using some distance function.

LDA is implemented with scatter matrix analysis. First within and between scatter matrixes are used to formulate criteria for class separability. The within scatter matrix S_w is calculated as:

$$S_w = \frac{1}{M}\sum_{i=1}^{M} \Pr(C_i)\Sigma_i \qquad (3)$$

where S_i is the average scatter of the sample vectors, C_i is the PCA output of particular class i, Pr is the prior class probability and M is number of classes. The between class scatter matrix S_b is calculated as:

$$S_b = \frac{1}{M}\sum_{i=1}^{M} \Pr(C_i)(m_i - m)(m_i - m)^T \qquad (4)$$

where m is the overall mean vector and m_i is class i mean vector. To find the discriminant projection we solve the following formula, where W is the discriminant projection and $?$ is generalized eigenvalue.

$$S_b W = ?S_w W \qquad (5)$$

To be able to compare the faces, a database of PCA projected faces is created, by performing PCA on the existing image database. Then all the faces in the PCA projected database are multiplied with the discriminant projection matrix that LDA creates, and the PCA+LDA database is created. The comparison between faces is performed by using a distance function, such as Euclidean distance, on the data in the new PCA+LDA database.

In Figures 1 through 4 we can see that LDA performs worse than the EBGM methods in most cases, but better than PCA for search sets FAFC and DUP1. Its main advantage is that it works well for searching for poorly illuminated images (Figure 2).

2.4 Elastic Bunch Graph Matching (EBGM)

The Elastic Bunch Graph Matching algorithm (EBGM) is based on the Bochum / USC [Wiscott97] face recognition algorithm that was used in the FERET evaluation. That algorithm appeared to work well in the FRVT test. The EBGM algorithm performs training on a set of facial images, where features are extracted from manually selected points in the images. These points represent facial features such as left eye, right eye, nose, etc. The extracted features are called model jets and contain frequency information for the selected points. Together the points and the model jets are called landmarks. All the landmarks for all the images in the training set are collected together into a graph called bunch graph. Every node in the bunch graph represents facial features and contains landmarks for that particular facial feature from all the images. The bunch graph can be used to locate landmarks in other images [Bolme03, Beveridge03].

The EBGM method then creates a face graph for every facial image in our database. The face graphs are similar to the bunch graph, except that each face graph represents only one facial image. These face graphs consist of many landmarks and for every landmark location in the facial image a Gabor jet is extracted, which consists of frequency information about the pixels surrounding the landmark locations. The Gabor jets are made by convolving the landmark location with a collection of Gabor wavelets and therefore they contain information on the landmark and the nearest area surrounding it. The Gabor jets are based on 40 complex wavelets where each wavelet has a real and imaginary component. These Gabor wavelets are similar to Fourier analysis, but they cover only the point and its nearest neighbourhood, so frequency information far away from the point does not affect the wavelet.

Each landmark stores information about magnitude and angles. In the version of EBGM that we used, 80 landmarks are extracted for every face and each landmark has magnitude in 40 dimensions, one dimension for every wavelet projection coefficient. The face graphs are created by collecting all the landmarks into one structure. The face graphs represent the faces in the images and we use the graphs to do the comparison. Therefore we do not need to use the images after the graphs have been created, which saves much memory. For the face recognition different distance formulas are used as described below.

In Figures 1 through 4 we can see that EBGM returns the best results of the three methods. Three versions of the EBGM algorithm are displayed in the figures. EBGM Magnitude uses a simple method which is related to Euclidean distance while EBGM Narrowing and EBGM Predictive step are more complicated methods. The latter two are based on the same method, but they use different methods to estimate the displacement of the phase angles in the calculations. EBGM Narrowing returns the best result but EBGM Predictive step is not far away. Because of this displacement estimation the latter two methods are quite slow.

In all the variants, the similarity between face graphs is defined as the average of the similarity between landmarks [Bolme03]. The formula for similarity of graph *G* and graph *G'* is given as:

$$L_{jet}(G,G') = \frac{1}{80}\sum_{i=0}^{79} S_x(J_i, J_i') \tag{6}$$

where S_x is the distance function between landmarks, J_i is landmark number *i* in graph *G* and J_i' is landmark number *i* in graph *G'*.

2.4.1 EBGM Magnitude

One of these similarity functions is EBGM Magnitude. It computes the similarity of the energy in the frequencies of the landmarks. The result of this formula is between 0 and 1, where the most similar data is 1 or nearly one. The distance function is given as:

$$S_a(J, J') = \frac{\sum_{n=1}^{N} a_n a'_n}{\sqrt{\sum_{n=1}^{N} a_n^2 \sum_{n=1}^{N} a'^2_n}} \tag{7}$$

where a_n is the n^{th} dimension of magnitude data in landmark *J*, *n* is dimension number and *N* is number of dimensions (40 in our case).

The drawback of this method is that information about phases of the frequencies is not used. So this method will give false positives if the frequencies are not in the same or similar phase. This method shares the same drawback as Euclidean distance has, where a single extreme dimension can ruin the search. When we use this function to compare every image to every other image this algorithm is $O(D^2)$ where *D* is the number of images.

2.4.2 EBGM Narrowing and EBGM Predictive Step

Another distance measure called EBGM Phase is bit more advanced and is similar to correlation. It uses magnitude and phase angles to find the difference between the faces. EBGM Phase is like EBGM Magnitude except that it weights the similarity of the magnitude with the similarity of the phase angles. The function is given as:

$$S_f(J, J') = \frac{\sum_{n=1}^{N} a_n a'_n \cos(f_n - f'_n)}{\sqrt{\sum_{n=1}^{N} a_n^2 \sum_{n=1}^{N} a'^2_n}} \tag{8}$$

where *a* is magnitude data in landmark *J* and *a'* is magnitude data for landmark *J'*, *n* is the number of dimension and *N* is the total number of dimensions, ø stands for phase angles. The result of this function is between -1 and 1 where the best match is 1.

The drawback of this method is that if the phase angles are slightly displaced, the calculated difference between the phases will be large and therefore the faces will not be recognized. On the other hand, methods that do not use phase angles like EBGM Magnitude distance could return false positives if the phase angles are displaced. In order to get rid of the false positives another method that uses phase angles and estimates the spatial displacement was created. The spatial displacement causes phase shift in the complex wavelets that were used to create the landmarks and makes it harder to recognize matching landmarks. The displacement vector estimates the displacement between the wavelets in the two landmarks. The distance function for EBGM with phase angles and displacement estimations is given as:

$$S_D(J, J', \vec{d}) = \frac{\sum_{n=1}^{N} a_n a'_n \cos(f_n - (f'_n + \vec{d} * \vec{k}_n))}{\sqrt{\sum_{n=1}^{N} a_n^2 \sum_{n=1}^{N} a'^2_n}} \tag{9}$$

where J and J' are landmarks in different face graphs, a_n is the magnitude variable in dimension n and landmark J, a'_n is the magnitude variable in dimension n and landmark J', N is the total number of dimensions in the landmarks, $ø_n$ is the phase angle variable in dimension n for landmark J, similarly the $ø'_n$ is the phase angle variable in dimension n for landmark J', d is displacement vector and k is spatial frequency vector. The displacement vector must be calculated separately for each comparison [Wiscott97].

There are a few methods to estimate the displacement and EBGM Predictive Step and EBGM Narrowing use two of them. EBGM Predictive Step uses Taylor expansion to approximate the cosine terms in the similarity function, and then it solves Equation 9 where the formula is equal to zero. The displacement estimation in the EBGM Predictive is single step and therefore it does not produce much work. The EBGM Narrowing search is a more accurate method but also much slower. It estimates the displacement by maximizing Formula 9 for four points in a grid around the landmarks, iteratively. The grid is typically 16 by 16 pixels and in every iteration we narrow the grid. First the grid is narrowed by 1 pixel per iteration but then we decrease the step size until it is only 0.125 pixels. The displacement vector that gave us the maximal result of Formula 9 will become the displacement vector for our distance calculation.

Both EBGM Predictive Step and EBGM Narrowing are $O(D^2)$ algorithms in order of magnitude, where D is the number of images, and we compare every image to every other. Though both these algorithms are $O(n^2)$, the work they have to perform is not the same. The estimation of the displacement is much more expensive for the EBGM Narrowing, where we typically have to iterate 50 times for every landmark in every image.

All these distance calculation methods are nearest neighbour methods and share the same drawbacks. The last function gives the best recognition rate but it is slow. It is used in both EBGM Predictive Step and EBGM Narrowing search.

2.5 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. Methods for fingerprint detection can then be used to find who the face belongs to.

The Japanese company Fujitsu has made ATM machines based on blood vessel graphs of hands and a few of these machines have already been installed in Japanese banks

[Fujitsu05, Watanabe05]. Fujitsu has demonstrated that this technique is very accurate and reliable, which indicates that it can be suitable for face recognition, since blood vessel graphs of faces should be at least as unique as graphs of hands. Because this technique is new and we did not have access to the required resources and equipment, we could not investigate it, but this method is very interesting.

3 Median Rank Distance

The Median Rank algorithm is a nearest neighbour ranking algorithm [Fagin03] for dimensional data, which can be used instead of distance formulas in many cases. It ranks the data into ranked lists where the best match is in the first position. It is simple but smarter than methods like Euclidean distance and does not have the same drawbacks.

The initial step of the Median Rank algorithm is to create a database to search in. A high-dimensional data set is projected to lines corresponding to the axes of the high-dimensional space.[1] Every projected instance in each line contains a value and an identifier to link the instance to the original data. All the lines are ordered in ascending order by value and together they form the database. To find the starting point of the query, binary search is used to locate the best match of the query instance, in all the lines separately. Two cursors are initialized at the best matches for all the lines. The cursors are used to search for the next best matches and its identifiers, either one position lower or higher, in a round-robin way through the lines. One vote is added to selected identifier, for every line and every iteration. When a certain identifier has received more votes than the number of half of the lines, it is returned to a result list. (Median Rank requires the number of votes to be half the number of lines because it is a *median* ranking algorithm. A different number could be specified, giving a different ranking algorithm.) The first instance to get enough votes is returned at rank 1, the second at rank 2, and so forth. The process continues until the specified number of nearest neighbours is found.

The order of magnitude of finding one neighbour with the Median Rank algorithm is $O(D^{1-2/(N+2)})$, where D is the number of images and N is the number of dimensions. While no available results exist on the complexity of finding many near neighbours, the complexity of finding similar images to all query images is much smaller than $O(D^2)$.

For large databases it is recommended to implement B+ trees on top of the lines and use the B+ tree instead of the binary search. But for small databases it is not needed and binary search is enough to locate the best match. The algorithm is the same in both cases, except that the B+ trees are used to locate the best matches and to retrieve the successive best matches.

The benefits of using Median Rank are many, but the most important ones are four. First it avoids returning an incorrect result when the data is out of range in one dimension (or a few). It is because of the voting system that Median Rank uses to calculate results. The second benefit is that its calculations are efficient, as it uses counting rather than complex

[1] The Median Rank is frequently combined with projection of the data points to random lines through the data space. Such random projection allows for a clever dimensionality reduction, by reducing the number of projected lines. In our application, however, we found that random projection reduced the quality of the results. Our data seem to be too “scrambled” after the random projection to increase recognition rate. Random projection is also based on the Euclidean metric and can improve recognition rate when using Euclidean distance, but we are working with more advanced and complicated methods. Therefore we do not include random projection in our discussion of Median Rank.

calculations. The third one is that it is scalable, as demonstrated in [Lejsek05]. The fourth one is that it has shown good quality of results [Fagin03, Lejsek05].

4 Method

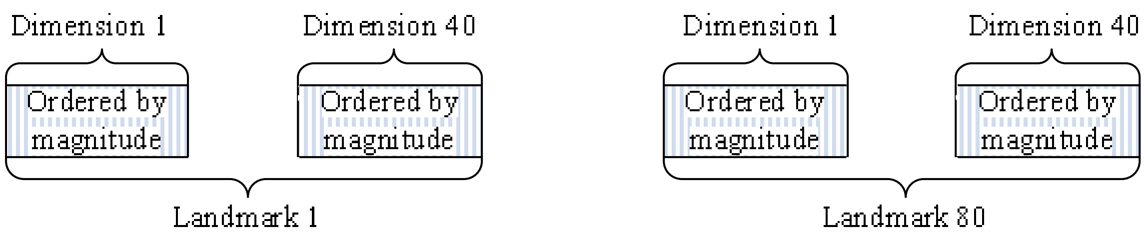
4.1 Database Index

We used the greyscale FERET facial image database in our experiments, because the EBGM algorithm requires manually selected landmark locations and a file with coordinates of the landmarks was provided for all the database images.

The EBGM algorithm creates face graphs that consist of 80 landmarks in 40 dimensions for each of the 3,360 database images. To run the Median Rank algorithm we had to index all the face graphs and since the database was small and static, the entire index could fit in memory. As the emphasis of this project is on the retrieval quality, we used a memory-based data structure, the three dimensional array shown in Figure 5 (rather than the B^+-trees used in [Fagin03]). Figure 6 shows a graphical representation of the array. All data for each landmark is stored together, and within each landmark each dimension is stored together. For each dimension, the list of dbRecords is ordered by Magnitude in ascending order.

```
struct dbRecord{
     int imageId;
     float magnitude;
}
dbRecord databaseArray[Landmark][Dimension][Image];
```

Figure 5: Data structure for the Median Rank algorithm



The first level of Median Rank was performed on the dimension lists inside each landmark separately. Since the values in the dimension lists are ordered in ascending order by magnitude, binary search is used to locate the starting point of the query in each dimension list. The Median Rank algorithm then traverses each array in a round robin fashion, as described in Chapter 3. Once an image has been seen in half of the dimensions (20 dimensions) it is returned as a match and put into a result list, as shown in Figure 7. We traverse the dimension lists and give votes until all the face graphs have been sent to the result list. This process is done for every landmark, resulting in 80 result lists. The result lists contain only image ids, but they are ordered, so the best matches are at the front of the lists, then the second best matches, and so forth.

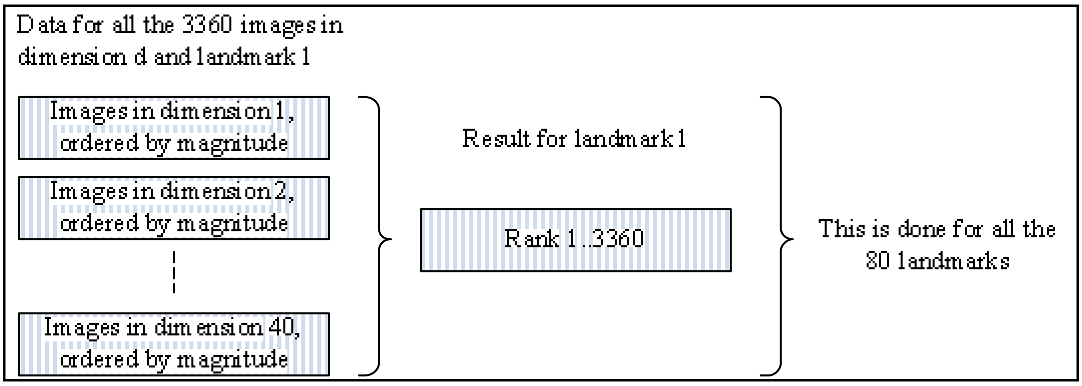
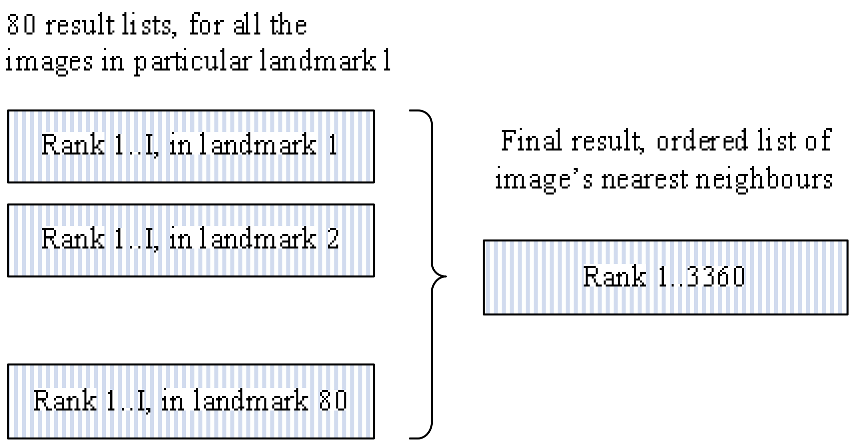


Figure 7: Median Rank algorithm for each landmark

To aggregate the result lists, we ran Median Rank again against those 80 result lists. The process is similar to the one described above, except that now the result lists are only traversed from the beginning to the end, as the lists are ordered by best matches. Also, every face graph requires 40 votes to be put in the final result list, as we have 80 landmarks and therefore 80 landmark result lists.



Our initial experiments with Median Rank showed the method performed quite poorly in many cases, in fact worse than PCA and LDA. Random projection [Bingham01] of the data, which sometimes may increase retrieval qualities, made the results even worse. We found, however, that normalizing the landmark data improved the results quality significantly; this process is described next.

4.3 Normalization

When we looked into the data for each face graph we observed that each value for a dark face graph was slightly higher than the corresponding value for a light graph. Furthermore, by examining the other distance calculation methods (Narrowing and Predictive Step) we saw that those methods contained some form of normalization, which in itself Median Rank does not. If the images are in a different scale it can be hard to match them, even though they could be of the same person and have the same pattern. The illumination affects the values of the face graphs, dark images have higher values than light ones and therefore it affects the scales of the images. Normalization could help getting the images into the same scale and make them easier to compare.

Therefore, we decided to normalize the face graph data. Normalizing the whole matrix does not change the results, as that simply amounts to rescaling the data from [0, 20] to [0, 1], and the difference between the patterns has the same ratio as before.

We tested four methods for normalization inside the data matrixes.

- First, we normalized the dimensions, by taking each dimension across all landmarks and normalizing its values such that the sum of the dimension became 1.
- Second, we normalized the landmarks, by taking each landmark across all dimensions and normalizing its values such that the sum of the landmark became 1.
- Third, we normalized the data first by dimensions and then by landmarks.
- Fourth, we normalized the data first by landmarks and then by dimensions.

These normalizations were performed before creating the database index. Therefore the normalizations did not affect the execution time of the algorithm. The first two normalization methods worked well for different search sets, and therefore we combined them in the second two methods in order to get the advantages of both and minimize drawbacks. By combining the methods we obtained better results than using them separately.

In the next chapter we present our experimental results, which show that the latter two methods perform better than the previous approaches.

5 Experimental results

In this chapter, we present the results of our performance study of using Median Rank for EBGM face recognition. We first describe the experimental setup in Section 5.1. Then we address the result quality in Section 5.2 and execution time in Section 5.3. Finally we summarize our findings in Section 5.4.

5.1 Experimental Setup

The computer that was used to run the experiment was Dell Intel Pentium 2.0 GHz PC machine. Its internal memory was 1.0 GB, the hard drive was 200 GB and the operating system was Linux. We used the FERET database (described in Section 4.1) as the facial image database. The images were greyscale and stored in pgm file format.

The face recognition algorithms we used are distributed free of charge by Colorado State University. The computer science department at CSU is distributing the code on their web page so others can experiment with it. It is also part of the FERET program. Their source code is written in C and runs on Linux. We used their source code to create the data for the database described in Chapter 4.

Our code to normalize the data, create the database index and run the Median Rank algorithm was also written in C and compiled with the GNU C compiler (gcc). We decided to use C for its explicit control of memory and for efficient execution.

5.2 Median Rank Results

In Figures 9 through 12, the results from our experiments with Median Rank and EBGM are displayed. Each figure shows the results with unnormalized Median Rank and the four normalization methods. EBGM Narrowing is also displayed for comparison purposes, since it is the best method distributed by CSU. Each figure depicts a different search set (described in Section 2). The FAFB set has different gestures, the FAFC different illumination, the DUP1 set is taken with at least one minute in between, and the DUP2 set with at least 18 months in between.

In each figure, the Y-axis represents the cumulative fraction of correct matches at a particular rank (or better), while the X-axis represents the rank. The goal in all the methods is to have as many correct matches as possible as early as possible; a perfect system would have 100% recognition at rank 1.

Overall, we observe that unnormalized Median Rank performs poorly in many cases, the same does normalization by landmark only and by dimension only. The figures also show that normalizing by both dimensions and landmarks, outperform the EBGM Narrowing. In the following sections, we look at each normalization method by itself.

5.2.1 Median Rank without Normalization

The results obtained using the Median Rank algorithm on unnormalized data, were not as good as we expected. In Figure 9 we can see that the result for search set FAFB is very good, but all the methods are returning very good results in that search set and it is hard to distinguish between them. In search set FAFC (Figure 10) the results are slightly worse than the average, but still acceptable. But the results for search sets DUP1 and DUP2 (Figures 11 and 12) are very poor; for DUP1 it is about 20% worse than the other results at average and in DUP2 the recognition rate is less than 10% at rank 20, which is even worse than with the simple EBGM Magnitude method.

5.2.2 Normalization by Landmarks

When we ran Median Rank on the database of face graphs normalized by landmarks, the results for the duplicate datasets improved very much. The percentage of correct matches rose from around 7% to 54% at rank 10 for the DUP2 dataset, which is very good compared to other methods. The improvement was not as good in DUP1, but still the percent of correct matches rose from being 40% to 60% at rank 10. The results for search set FAFB are also very good. However, an unexpected effect was seen in search set FAFC, as the result quality was very poor compared to the other methods (only 42% at rank 10) as can be seen in Figure 10.

Normalization by landmarks works well in the duplicate datasets because most of the images are well illuminated and this normalization makes facial features and boundaries that are not caused by light more clear.

Normalization by landmarks returns poor results for badly illuminated images, like we have in search set FAFC, because in dark images we need to rely on the strength of each filter (dimension) in the image, rather than the landmarks. When the normalization is performed, different dimensions are mixed up together. Dark landmarks have higher values than light landmarks and when we normalize by landmarks, the landmarks go into the same scale. Dimensions that were stronger than others in dark landmark will remain strong after the normalization, but the ones that were only little bit stronger in light landmarks will be significantly stronger after the normalization. When we plot the face graphs, we can see that the shape of dark images changes more than the shape of light images, which makes it harder to recognize them.

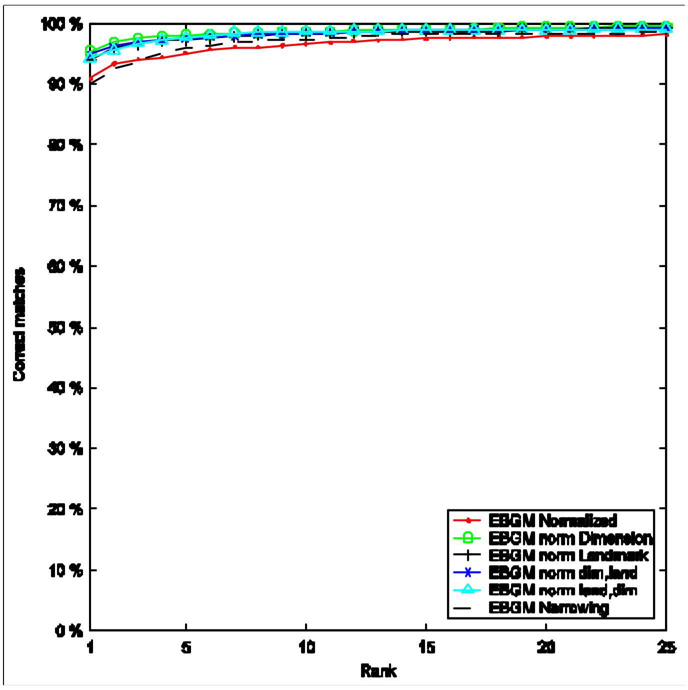


Figure 9. Search set FAFB.

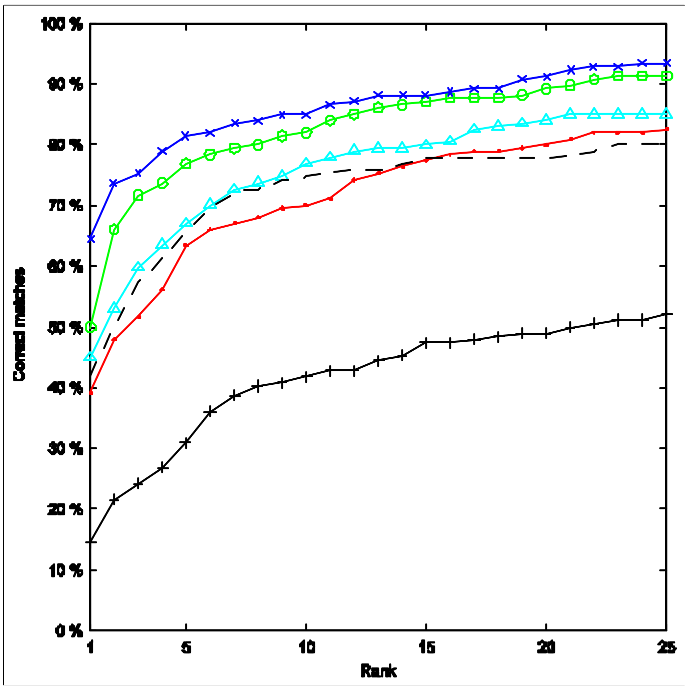


Figure 10. Search set FAFC.

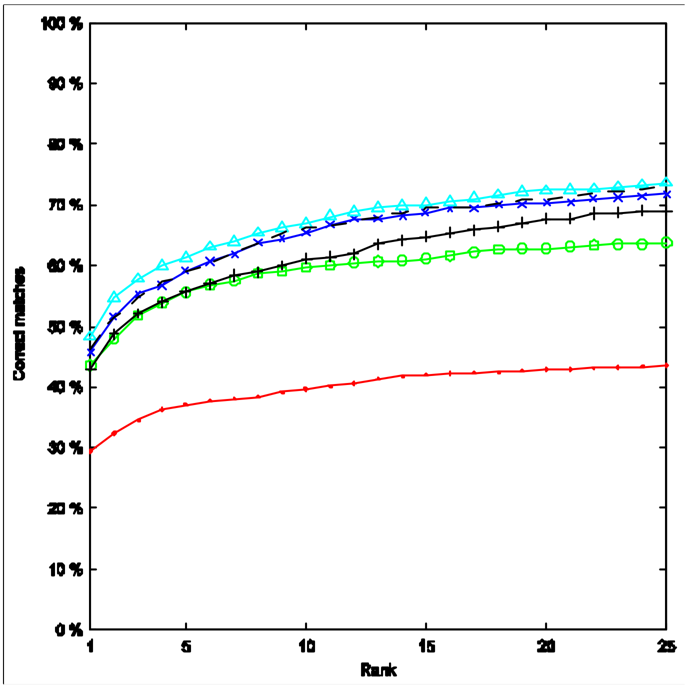


Figure 11. Search set DUP1.

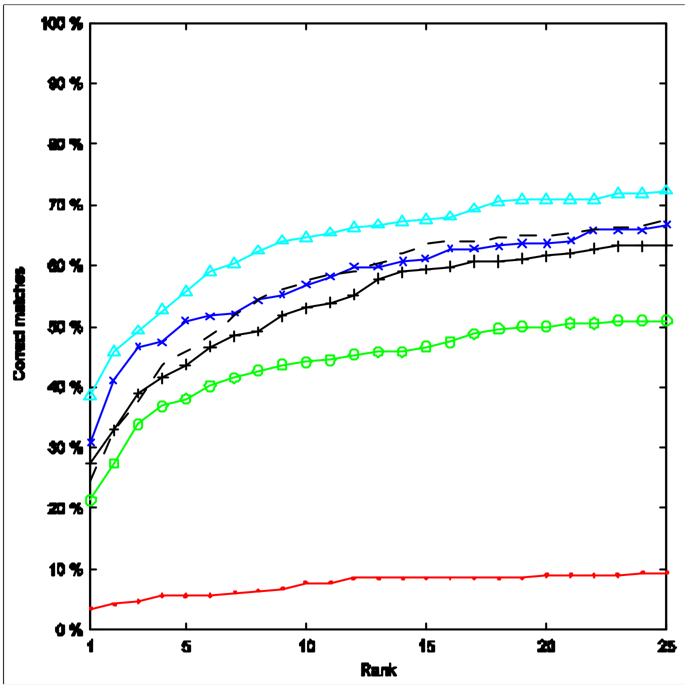


Figure 12. Search set DUP2.

5.2.3 Normalization by Dimensions

When we normalized by dimensions the result for search set FAFB and FAFC improved and in fact normalization by dimensions is the second best method in search set FAFC. The results for the duplicate datasets improved also, but not as much as it did when we normalized by landmarks.

Normalization by dimensions and Median Rank performs well in dataset FAFC, because the normalization makes one dimension equally as strong as each other and in search set FAFC we have to rely on the filters that were used to create the landmarks as it is very hard to distinguish between landmarks.

The reason why normalization by dimensions is worse in the duplicate search sets, is that distinction between landmarks are more clear and we need to rely more on each landmark than filter when comparing images taken with some time in between. The images are also well illuminated so we need to recognize facial gestures rather than changes in colour or other things.

5.2.4 Normalization by Landmarks and Dimensions

As we can see from the preceding sections, there were pros and cons with both previous normalizations. In order to get the best of both methods into one, we mixed these normalizations together. We normalized first by landmarks and then dimensions and vice versa, which in both cases yielded better results than before. The order of the normalizations, however, affected the results. When we normalized by dimensions first, we obtained slightly better results in the FAFB and FAFC search sets and when we normalized by landmarks we got slightly better results with the duplicated search sets.

Normalization by landmarks and then dimensions is the best for the duplicate sets. The normalization by landmarks was already good for the duplicate sets and by also normalizing by dimensions we improved the results for the FAFC dataset. The combination of normalizations gives us the advantages of both normalizations. That is because when we normalize by landmarks, the length of each landmark becomes 1 and that changes the whole face graph, but as this is done for all the images, it affects them all in a similar way. When we then normalize by dimensions it results in much less changes to the graphs than when normalizing the original data and therefore we can keep the advantages of normalization by landmarks and add some of the advantages of normalization by dimensions. In other words, we can balance the landmarks and dimensions by sharpening the facial features in the duplicate images and then sharpening the contrast and dimension differences.

5.2.5 Normalization by Dimensions and Landmarks

Normalization by dimensions and then landmarks also returns very good results and in fact it returns the best result for search set FAFC. In the duplicate datasets the results are also very good, but normalization by landmarks and then dimensions returns better results in those cases. This behaviour is similar to the results when normalizing only by landmarks or dimensions. Normalization by dimensions is better in test case FAFB and FAFC but normalization by landmarks is better in the duplicated sets.

By normalizing by dimensions and then landmarks we obtained much better results than by just normalizing by dimensions. The improvements for the duplicate dataset were very substantial while the improvements were smaller for search sets FAFB and FAFC.

What happens in this normalization is that we move the dark images closer to the lighter ones by normalizing by dimensions. Then we sharpen the facial gestures in the images by normalizing by landmarks so it is easier to compare the faces. This matches our previous conclusions that the dimensions have more weight in dark images while the landmarks themselves have more weight in images taken with a time difference.

5.3 Execution Time

The execution time of the search in all the Median Rank methods was the same. The only difference between these methods is how the database is built, but the search is unaffected. As stated before the execution time was a bit disappointing because we were hoping for Median Rank to be the fastest search method. But this voting mechanic costs and doing double Median Rank costs more so this execution time could be expected. Table 1 shows the running time for each search method, including full matching of all search sets, where we compare all the 3,360 images to each other.

EBGM search method	**Running time**
Median Rank	1 hour
Magnitude	10 minutes
Predictive step	1 hour 29 minutes
Narrowing search	39 hours

Table 1. Running time of the EBGM search methods.

The search using Median Rank took exactly one hour. Running EBGM Magnitude took only ten minutes, as the magnitude method is much simpler and has less overhead in controlling the calculation. On the other hand, EBGM Magnitude does not return good results.

The EBGM Predictive step method took 1 hour 29 minutes, but worst of all was the EBGM Narrowing method, which took 39 hours, which is completely unacceptable. Nobody will be able to use such a slow face recognition system. The reason why Predictive Step and Narrowing search are so slow is because of their displacement estimation methods. The Narrowing search uses iterative calculations to estimate displacement while the Predictive Step uses a single step calculation to estimate the displacement. The displacement estimation has to be performed in every landmark comparison between every two images, which adds a great deal of work to the algorithm.

In the Median Rank algorithm we always searched for 3360 images, but we are only interested in the top few images. Since the algorithm is incremental in nature, it is possible to avoid some of the work done for every landmark. We could not test the system for larger databases than this FERET database, but it is obvious that the option to search for only a fraction of the whole database is very valuable for very large databases.

5.4 Discussion

By considering the quality of results seen in Figures 9 to 12 we can see that Median Rank with normalization by landmarks and dimensions (or vice versa) is the best choice. When furthermore considering the search time compared to the EBGM Narrowing algorithm, we see that Median Rank is more efficient. We also note that Median Rank has been shown to be scalable in other applications, while it is very unclear how to index efficiently for the EBGM Narrowing algorithm. Finally, given that we feel that the duplicate datasets are more realistic than the others, we recommend normalizing first by landmarks and then by dimensions.

6 Discussion

In our experiments in face recognition we have seen that the EBGM algorithm used with normalization by landmarks and dimensions (or normalization by dimensions and then landmarks) and Median Rank performs better than all the methods distributed by Colorado State University. It is also much faster than the best methods distributed by CSU.

A further advantage of the Median Rank algorithm over the other methods is that it is scalable. As mentioned before, for large datasets we will not have to calculate the distance to every facial image in the database, which should save enormous time.

As described in Section 2, the latest thing in authentication is using images of the blood vessel network in hands. The Japanese company Fujitsu has built ATM machines that use images of the blood vessel network in a customers hand for identification and access to the machine, instead of pin numbers. The machine gets the image of the blood vessels with infrared cameras [Fujitsu05, Watanabe05]. According to their studies, this technology is safe and gives a very good recognition rate. Another study has shown that by taking facial images with heat cameras, a similar blood vessel network emerges, because where vessel is, the skin on top is slightly warmer. By merging those two technologies, it should be possible to make a very accurate face recognition system; this will be a very interesting topic in the next few years.

7 References

[BBC05] *Woman has first face transplant* (2005, November 30), Retrieved December 3, 2005, from http://news.bbc.co.uk/1/hi/health/4484728.stm.

[Beveridge01] J. R. Beveridge, K. She, B. A. Draper, G. H. Givens. A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR.* 2001

[Beveridge03] R. Beveridge, D. Bolme, M. Teixeira, B. Draper. The CSU Face Identification Evaluation System User's Guide. *Colorado State University*. May 2003

[Bingham01] E. Bingham, H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the seventh ACM SIGKDD conf.*, pages. 245-250. San Francisco, California, USA. 2001.

[Blackburn01] D. M. Blackburn, M. Bone, P. J. Phillips. Facial Recognition Vendor Test 2000, Evaluation Report. *FRVT*. 2001

[Bolme03] D. S. Bolme. Elastic Bunch Graph Matching. Master's thesis. *Colorado State University.* 2003.

[DTI06] Information security breaches survey 2006, technical report. *DTI's Publications Unit*. 2006

[Fagin03] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the ACM SIGMOD Conf, pages 301-312*, San Diego, CA, USA, 2003.

[Feret] Face Recognition Technology program. (http://frvt.org/FERET/).

[Feret04] The Facial Recognition Technology (FERET) Database. Retrieved October 7, 2004 from http://www.itl.nist.gov/iad/humanid/feret/feret_master.html.

[Fujitsu05] Fujitsu Palm Vein Technology. Retrieved September 20, 2005 from http://www.fujitsu.com/global/about/rd/200506palm-vein.html

[Garcia04] C. Garcia, M. Delakis. Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection. IEEE Transaction on Pattern Analysis and Machine Intelligence. 26(11), pages 1408-1423, 2004.

[Gold2000] B. Gold, N. Morgan. Speech and Audio Signal Processing. Processing and Perception of Speech and Music. John Whiley & Sons. 2000.

[Hollmen96] J. Hollmen. Principal Component Analysis (March 8, 1996). Retrieved September 16, 2005 from http://www.cis.hut.fi/~jhollmen/dippa/node30.html.

[Lejsek05] H. Lejsek, F. H. Ásmundsson, B. Þ. Jónsson and L. Amsaleg. Efficient and effective image copyright enforcement. In *Proceedings of BDA,* Saint Malo, France, 2005.

[Martinez01] A. M. Martinez, A. C. Kak. PCA versus LDA (2001). *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 23 (2), pages 228-233, 2001

[Moghaddam96a] 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

[Moghaddam96b] B. Moghaddam, A. Pentland. Probabilistic Visual Learning for Object Representation. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), pages 696-710, 1997.

[Phillips03] P. J. Phillips, P. Grother, R. J. Michaels, D. M. Blackburn, E. Tabassi, M. Bone. Face Recognition Vendor Test 2002, Evaluation Report. March 2003

[Photo02] Photobook/ Eigenfaces Demo. Retrieved October 9, 2005 from http://vismod.media.mit.edu/vismod/demos/facerec/basic.html.

[PCA] Principal Component Analysis. Retrieved December 10, 2005 from http://www.fon.hum.uva.nl/praat/manual/ Principal_component_analysis.html.

[Smith02] L. I. Smith. A tutorial on Principal Components Analysis. Retrieved November 15, 2005 from http://csnet.otago.ac.nz/cosc453/student_tutorials/ principal_components.pdf.

[Turk91] M. A. Turk, A. P. Pentland. Face Recognition Using Eigenfaces. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 586-590, Maui, Hawaii, 1991

[Watanabe05] M. Watanabe, T. Endoh, M. Shiohara, S. Sasaki. Palm vein authentication technology and its applications. *Fujitsu Laboratories Ltd., Kawasaki, Japan*. Sept, 2005

[Wiscott97] Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger, and Christoph von der Malsburg, Face Recognition by Elastic Bunch Graph Matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 19(7), pages 775-779, 1997.

[Zhao03] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Comput. Surv.* 35(4), pages 399-458, 2003

[Zhao98] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, J. Weng. Discriminant Analysis of Principal Components for Face Recognition. In *3rd International Conference on Automatic Face and Gesture Recognition*, pages 336--341, 1998.

[Zhao99] W. Zhao, R. Chellappa, P. J. Phillips. Subspace Linear Discriminant Analysis for Face Recognition. *Center for Automation Research, University of Maryland, Technical Report CAR-TR-914.* 1999

REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

Department of Computer Science
Reykjavík University
Ofanleiti 2, IS-103 Reykjavík, Iceland
Tel: +354 599 6200
Fax: +354 599 6201
http://www.ru.is