A dissertation submitted in partial fulfilment of the requirements for M.Tech(Computer Science) degree of the Indian Statistical Institute

Face Recognition Under Partial Occlusion

M.Tech(Computer Science) Dissertation Report

By

Mrinmoy Ghorai

Roll No : CS0717

Under the supervision of

Prof. Bhabatosh Chanda

Electronics and Communication Sciences Unit

INDIAN STATISTICAL INSTITUTE

203, B.T. ROAD

KOLKATA-700108

Indian Statistical Institute

Kolkata-700 108

CERTIFICATE

This is to certify that the thesis entitled "**Face Recognition Under Partial Occlusion**" is submitted in the partial fulfilment of the degree of M. Tech. in Computer Science at Indian Statistical Institute, Kolkata. It is fully adequate in scope and quality as a dissertation for the required degree.

The thesis is a faithfully record of bona fide research work carried out by Mrinmoy Ghorai under my supervision and guidance. It is further certified that no part of this thesis has been submitted to any other university or institute for the award of any degree or diploma.

Prof. Bhabatosh Chanda
(Supervisor)
ECSU Unit

Countersigned
(External Examiner)
Date:

Acknowledgement

I also take this opportunity to express my sincere gratitude to the supervisor of this study, Prof. Bhabatosh Chanda. His command over this matter has been a great help for my analysis. He helped me in all aspects including the preparation of this manuscript. This work has been possible only because of his continuous suggestions, inspiration, motivation and full freedom given to me to incorporate my ideas. I also take this opportunity to thank all my teachers who have taught me in my M. Tech. course and last but not the least I thank all my family members and friends for their endless support.

Place : Kolkata

Date : Mrinmoy Ghorai

Abstract

In today's world identification of a person is very important and necessary mostly in security purpose. Now a days many organisation use different type of identification techniques, one of them is Face Recognition. But in different conditions they may not have the capability to recognise efficiently w.r.t both success rate and time complexity of their algorithm. There are different as well as important issues in face recognition like expression, ageing, occlusion that may make trouble to any recognition system. In this report we have presented a novel approach of recognizing human face from various type of partially occluded frontal views.

The idea behind this approach is sparse representation that means the test face image should only be represented in terms of training face images of the same object. Here we consider that the occlusion is also sparse i.e. a fraction of the image pixels are occluded. We propose a simple and novel algorithm which uses pseudo inverse to express the test image as a sparse linear combination of training samples plus a sparse error due to occlusion. Lastly we use l^2 norm and k-NN classifier to recognize the face. One advantage of this approach is no need of feature selection, dimension reduction, domain-specific information about the image. We analysis the experimental results of this algorithm which shows how much occlusion the algorithm can handle and how to choose the training data to maximize robustness for different type of occlusion. It also shows where the maximum information present in the face image. To verify the algorithm we use the publicly available ORL database and MATLAB tool in windows system.

Contents

Chapter 1

Introduction

A **face recognition** is a computer application for automatically identifying or verifying a person from a digital image or a snapshot from a video source.
This is branch of a most popular and advanced topic **Biometrics** [1]. Facial scan is an effective biometric attribute or indicator. Different biometric indicators are suited for different kinds of identification applications due to their variations in intrusiveness, accuracy, cost and ease of sensing. There are six main biometric indicators namely face, finger, hand, voice, eye and signature. Among those facial features scored the highest compatibility, in a machine readable travel documents (MRTD) system based on a number of evaluation factors [2].

History of face recognition : The most famous of early example of a face recognition system is due to Kohonen[3], who demonstrated that a simple neural net could perform face recognition for aligned and normalized face images. The type of network he employed computed a face description by approximating the eigenvectors of the face image's autocorrelation matrix; these eigenvectors are now known as **eigenfaces**. Kohonen's system was not a practical success, however, because of the need for precise alignment and normalization.
Kirby and Sirovich (1989)[4] later introduced an algebraic manipulation (PCA) which made it easy to directly calculate the eigenfaces, and showed that fewer than 100 were required to accurately code carefully aligned and normalized face images.
Turk and Pentland (1991)[5] then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery, and to determine the precise location and scale of faces in an image. They then demonstrated

that by coupling this method for detecting and localizing faces with the eigenface recognition method, one could achieve reliable, real-time recognition of faces in a minimally constrained environment.

Face recognition scenarios can be classified into two types :
face verification or authentication is a one-to-one match that compares a query face image against a template face image whose identity is being claimed.
face identification or recognition is a one-to-many matching process that compares a query face image against all the template images in a face database to determine the identity of the query face.

Issues in Face Recognition

- 3D face projected into 2D image
- Illumination variation because of different lighting condition
- Facial expression
- Occlusion due to other objects or accessories(e.g. sunglasses, scarf etc.)
- Ageing is another great issue as the face changes over time in a non linear fashion over long periods.

1.1 Different approaches of Face Recognition

Image-based face recognition techniques can be mainly categorized into two groups based on the face representation which they use: (i)appearance-based which uses holistic texture features; (ii) model-based which employ shape and texture of the face, along with 3D depth information. A number of face recognition algorithms, along with their modifications, have been developed during the past several decades (see Figure 1.1).

Appearance-based approaches represent an object in terms of several raw intensity images. Most of these techniques depend on the vector space structure induced by the images. An image is considered as a point in high-dimensional vector space. These approaches use different statistical measurement to find out the distribution of the image vectors in the high-dimensional vector space and gives an explanation about the efficient feature space according to different applications. From the similarity between training

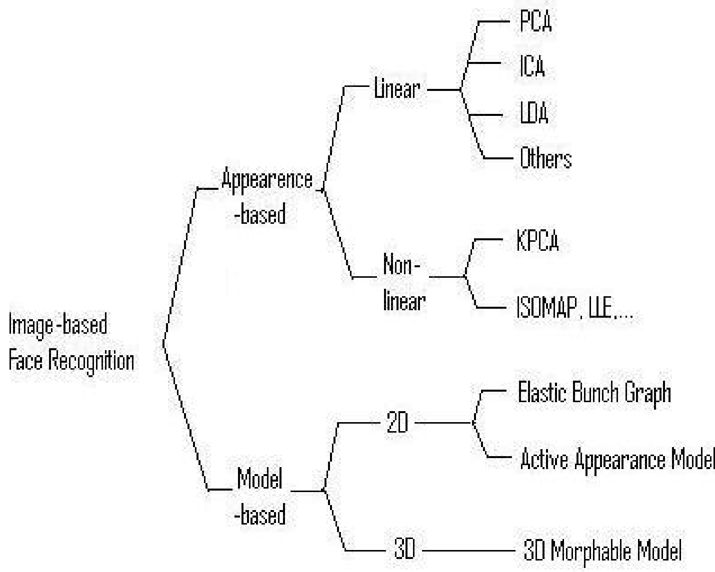

Figure 1.1: Face Recognition Methods

vectors and test vector in the feature space the system recognise the test object. Three classical linear appearance-based classifiers PCA , ICA[6] and LDA[7] are given in literature. Non-linear appearance-based approaches are Kernel PCA[8], ISOMAP[9] and LLE[10].

On the other hand model-based approaches construct a model of the human face for analysing facial variations.The prior knowledge of human face is highly utilized to design the model. For example, feature-based matching derives distance and relative position of features from the placement of internal facial elements (e.g. eyes,nose etc.). 2D morphable model-based approaches like Elastic Bunch Graph Matching [11][12] and Active Appearance Model[13][14] are introduced through which the face variations are learned. Blanz [15][16] proposed a method based on a 3D morphable face model that encodes shape and texture to handle pose, illumination.

In figure 1.3 we show a schematic diagram of a usual face recognition system.

Figure 1.2: Images from ORL database

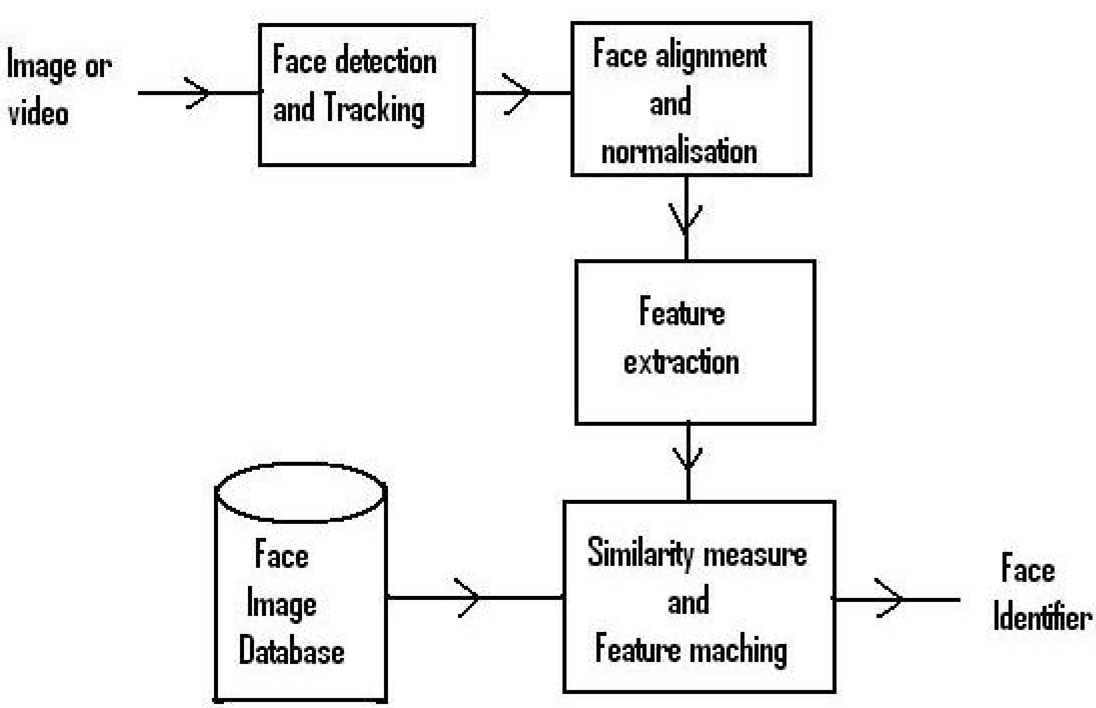

Figure 1.3: Schematic diagram of a face recognition system

1.2 Objective of the work

Face Recognition is an important component of any security system used all around the globe. But in reality several issues, some of them mentioned already, arise a great problem for a face recognition system. We mainly focused our attention on partial occlusion in the grey level test image. We restrict ourselves to images devoid of variation in ageing, pose etc. Training sample images are taken from frontal view without occlusion. Our proposed algorithm is efficient and computationally fast. We use the concept of pseudo-inverse and k-nearest neighbour classifier.

1.3 Organization of the report

Organization of rest of the report is as follows. We first formulate the problem for face recognition and then we give two algorithm, one of them is our proposed algorithm, with experimental results, in the second chapter. In the third chapter we reformulate the problem for face recognition under partial occlusion and modify the algorithm. In the fourth chapter we have presented experimental results.

Chapter 2

Face Recognition Technique

In the previous chapter we have mentioned some issues which are important in face recognition when we want to design an efficient algorithm. Here we consider the case where the face image, we want to recognize could be partially occluded. We will also deal with illumination but facial expression, ageing are the issues we do not explicitly account. The key of our approach is the *sparse* representation of objects to achieving robust and accurate recognition. Ideally, the test image can be represented in terms of just the training images of the same object, a small portion of the entire training set.

In this chapter we will formulate face recognition problem without any occlusion and describe two face recognition algorithm, one of them is our proposed algorithm. Both the algorithm have their own speciality though their is a trade off between efficiency in recognition rate and time complexity.

2.1 Problem Formulation

In face recognition(human face) the basic problem is to determine the identity of a test face image from k distinct object classes i.e. to recognize the class of the test image. Here we take $w \times h$ grey scale images both for training and testing with images as a $m = wh$ dimensional vector $v \in \mathbb{R}^m$ taking lexicographic ordering of the pixels. We arrange the given n training images as columns of a matrix $A = [v_1, v_2, ..., v_n]$ and let $A_i \in \mathbb{R}^{m \times n_i}$ denote the n_i training images of i-th subject.

A test image y from the i-th class is approximately lie in the linear span of the training images from the same class, mathematically $y = A_i x_i$, where $x_i \in \mathbb{R}^{n_i}$ is a vector of coefficients. y can also be represented as linear combination of the entire training images i.e.

$$y = Ax_0 \tag{2.1}$$

where $A = [A_1, A_2, ..., A_k]$, $A_i = [v_1, v_2, ..., v_{n_i}]$ and $x_0 = \left[0...0x_i^T 0...0\right]^T \in \mathbb{R}^n$ is a vector with zero entries everywhere except those associated with the corresponding subject i. So we can now formulate the problem as

Given sets of labelled training images $A_1, A_2, ..., A_k$ from k classes and a test image y is taken by sampling an image from one of the k classes, say i-th class, identify the correct class i.

2.2 Solution by Optimization Problem

Now we want to solve the system of linear equations $y = Ax$. If the number of image pixels, m, is grater than the number of training images, n, the system is overdetermined and may not have an exact solution. Then most popular technique is to take a solution in the least square sense, by minimizing the l^2 norm of the residual :

$$\hat{x}_2 = arg \min_x \parallel y - Ax \parallel_2 \tag{2.2}$$

But we use l^1 norm because in the next chapter we will show that l^1 norm is more effective than l^2 norm in case of partially occluded face images. The solution of the equation $y = Ax$ is

$$\hat{x}_1 = arg \min_x \parallel x \parallel_1 s.t. Ax = y \tag{2.3}$$

This is a convex optimization problem, whose solution is unique and can be efficiently computed by linear programming.

2.3 Face Recognition by l^1 norm minimization

For this we first define k functions $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ the i-th of which preserves the coefficients corresponding to the i-th group and sets rest to zero i.e. $\delta_i(x) = \left[0...0x_i^T 0...0\right]^T \in \mathbb{R}^n$. The approximation in terms of the coefficients associated with the i-th group is then

Figure 2.1: Some face images from ORL database

$\hat{y} = A\delta_i(\hat{x})$ and classification can be achieved by assigning y to the group that minimizes $\| y - A\delta_i(\hat{x_1}) \|_2$.The implementation minimizes the l^1 norm via a primal-dual algorithm for linear programming[17][18].

Algorithm 1 (Face Recognition by l^1 norm minimization)

---

1 : **Input** : n training samples partitioned into k classes $A_1, A_2, ...A_k$ and a test sample y.

2 : Normalize the training samples to have unit l^2 norm and set $B = [A_1 A_2 ... A_k]$, where $A_i = [v_{i1} v_{i2} ... v_{in_i}]$, n_i is no. of objects belonging to class i for $i = 1, 2, ...k$.

3 : Solve the l^1 minimization problem :

$$\hat{x}_1 = arg \min_x \| x \|_1 \qquad \text{s.t. } Ax = y$$

by linear programming.

4 : Compute the residuals $r_i\,(y) = \| y - A\delta_i(\hat{x_1}) \|_2$, for $i = 1, 2, ..., k$.

5 : **Output** : identity$(y) = arg \min_{i=1,2,...,k} r_i\,(y)$.

Experimental Result : We apply the above algorithm on ORL database. The database contain 380 face images, 10 face images from each of 38 person. The images are taken in different lighting condition for training and testing(see figure 2.1). We test the algorithm on 76 face images from the database. The recognition rate of this algorithm is 100%. Though algorithm is very efficient in face recognition the main disadvantage of this algorithm is it's time complexity. It takes on an average 10 minutes to identify an object. In some situation it may not much helpful. From the concept of it's sparse representation we have design an algorithm which is very faster than the above algorithm. We have used

Moore-Penrose pseudoinverse for our proposed algorithm.

2.4 MoorePenrose pseudoinverse

In mathematics, precisely in linear algebra, inverse of a matrix exists if it is a non singular square matrix.Obviously any matrix can not be invertible. E.H. Moore[19] and Roger Penrose[20] individually introduced the concept of *pseudoinverse* for any matrix which is generalisation of the inverse matrix.

Definition : The pseudoinverse A^+ of an m-by-n matrix A (whose entries can be real or complex numbers) is defined as the unique n-by-m matrix satisfying all of the following four criteria:

- $AA^+A = A$ (AA^+ need not to be the general identity matrix)
- $A^+AA^+ = A^+$ (A^+ is a weak inverse for the multiplicative semi-group)
- $(AA^+)^T = AA^+$ (AA^+ is Hermitian) and
- $(A^+A)^T = A^+A$ (A^+A is also Hermitian)

A common use of pseudoinverse is to compute a least squares solution to a system of linear equations.It can be computed using the singular value decomposition.

Computation Method : There are several methods available in the literature for computing the pseudoinverse of a matrix. A computationally simpler and more accurate way to get the pseudoinverse is by using the singular value decomposition(SVD). If $A = UDV^T$ is the singular value decomposition of A then $A^+ = VD^+U^T$. For a diagonal matrix such as D, we compute the pseudoinverse by taking the reciprocal of each non zero element on the diagonal and leaving the zeros in place. In real computation, elements smaller than some threshold are taken to be zero. MATLAB function *pinv* take this threshold as $\varepsilon * max(m, n) * max(D)$, where ε is the machine epsilon.

Application : Give a system of linear equations $Ax = b$, it is not easy to find a unique solution x which will satisfy the system. Our aim is to find such a x that minimize the

Euclidean norm $\| Ax - b \|_2$. Pseudoinverse fulfil that necessity at a high percentage. Thus the problem has a unique solution $x = A^+ b$.

2.5 Proposed Algorithm by Pseudoinverse

In our proposed algorithm we solve the system of linear equations $Ax = y$ using pseudoinverse. we find the unique solution x by computing $x = A^+ y$, where A^+ is the pseudoinverse of A. So we get an estimation $\hat{x}$ of x for the system of equations $Ax = y$ for which $\| Ax - y \|_2$ is minimized.

Algorithm 2(Face Recognition By Pseudoinverse)(proposed)

---

1 : **Input** : n training samples partitioned into k classes $A_1, A_2, ...A_k$ and a test sample y.

2 : Normalize the training samples to have unit l^2 norm and set $A = [A_1 A_2 ... A_k]$, where $A_i = [v_{i1} v_{i2} ... v_{in_i}]$, n_i is no. of objects belonging to class i for $i = 1, 2, ...k$.

3 : Compute : $\hat{x} = A^+ y$, where A^+ is the pseudoinverse of A

4 : Compute the residuals $r_i(y) = \| y - A\delta_i(\hat{x}) \|_2$, for $i = 1, 2, ..., k$.

5 : Output : identity$(y) = arg \min_{i=1,2,...,k} r_i(y)$.

Experimental Result : The same experiment we have done for our proposed algorithm also. The recognition rate for this algorithm is 100%. But in case of time complexity our proposed algorithm is better than algorithm 1. Our proposed algorithm is 600 time faster than algorithm 1.

Still now we have shown algorithm 2 is very efficient compare to algorithm 1 in face recognition. But our main purpose is which one perform better in face recognition when the test samples are partially occluded. In the next chapter we have discussed about various type of occlusion and how the above algorithms work in that case.

Chapter 3

Recognition Under Partial Occlusion

3.1 Robustness

Recognition of a human face image with out occlusion is a simple problem. In literature, many solutions are available for this problem. But real problem arises when test face image is occluded or corrupted with unknown magnitude and location. There are three type of errors due to occlusion in measured image,

- large in magnitude in the value of the pixels affected (gross errors)
- unpredictable in location of the pixels affected (randomly supported errors)
- concentrated only on relatively small portion of pixels of the image (sparse errors)

Finding robust recognition performance despite such errors incurred by occlusion is undoubtedly a challenging task.

Robustness from redundancy : A fundamental principle of coding theory is that *redundancy* in the measurement is essential to detecting and correcting gross errors. Redundancy arises in object recognition because the number of image pixels is typically far greater than the number of subjects that have generated the images. So, even if a fraction of pixels are completely corrupted by occlusion, recognition may still possible based on remaining pixels. On the other hand, schemes based on dimension reduction or feature extraction (e.g.,PCA,ICA,LDA) discard redundant information that could compensate for the occlusion.

Robustness from sparsity : The sparsity arises from two sources, one from the identity of the test image and another from the nature of the occlusion. Practically, the test image can be represented in terms of just the training images of the same object, a small portion of the entire training set. The corruption incurred by occlusion is also typically sparse, affecting only a relatively small fraction of the image pixels.

Recent works on signal processing and information theory [21][22][23] shows that l^1 norm is very effective for recovering sparse representations. Finding sparse solutions to systems of linear equations can be efficiently and exactly solved via convex optimization, by minimizing l^1 norm.

3.2 Reformulate

In section 2.1 we have formulate the problem for face recognition without any occlusion.

Now suppose the test face image is partially occluded and let ρ be the fraction of occluded pixels in the test image. Then a fraction $(1-\rho)$ of the pixels belong to an image y_0 from one of the k classes. Mathematically it can be expressed as

$$y = y_0 + e = Ax_0 + e \tag{3.1}$$

where $e \in \mathbb{R}^m$ is an error vector. Since a fraction ρ of its entries are non zero,e is *sparse*. The no. of occluded pixels i.e. non zero entries in the error vector is ρm. Later when we will reconstruct the error by the algorithm, will see that the entries corresponding to the uncorrupted pixels is not really zero but very small compare to that of the occluded pixels.

Here we do not make any assumption about the location of the corruption and the magnitude of the error. Also the type of corruption i.e. whether the corrupted pixels are in contiguous region or random in location, is totally unknown to us(see figure 3.1). The change in magnitude of the pixel values does not obey any prior probabilistic distribution. So we can now reformulate the problem as

Given sets of labelled training images $A_1, A_2, ..., A_k$ ***from k classes and a***

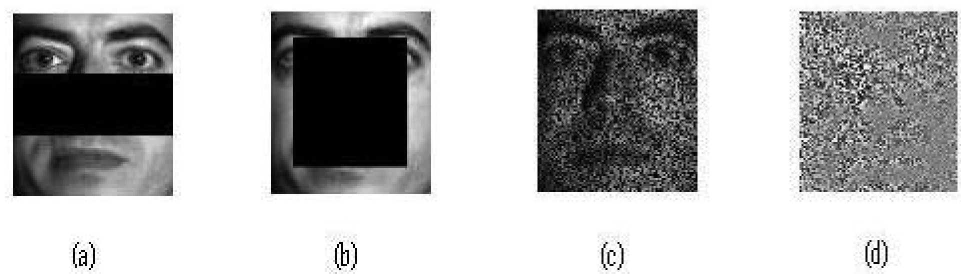

Figure 3.1: a and b are contiguous occluded, c and d are random pixels occluded images

test image y created by sampling an image from one of the k classes, say i-th class and then introducing the occlusion in the ρ fraction of pixels arbitrarily, identify the correct class i.

3.3 Solution By Convex Optimization Problem

Now we want to solve the system of linear equations $y = Ax$.If the number of image pixels, m, is grater than the number of training images, n, the system is overdetermined and may not have an exact solution. Then most popular technique is to take a solution in the least square sense, by minimizing the l^2 norm of the residual :

$$\hat{x}_2 = arg \min_x \parallel y - Ax \parallel_2 \tag{3.2}$$

But for highly non-Gaussian error, $\hat{x}_2$ can be arbitrarily bad. We will show how minimizing the l^1 norm, rather than the l^2 norm can achieve a efficient algorithm for recognition in the presence of occlusion.

Since the error e is known to be sparse, but of arbitrary magnitude, an alternative to minimizing l^2 norm of the residual is :

$$\hat{x}_0 = arg \min_x \parallel y - Ax \parallel_0 \tag{3.3}$$

Here, l^0 "norm" $\parallel x \parallel_0$ counts the number of nonzero entries of the vector x. Computing gives the the vector $A\hat{x}_0$ in the range of A such that the error, $e = y - A\hat{x}_0$, has the fewest nonzero entries.

But problem (3.3) is NP-hard and also difficult to approximate[24]. Unless P=NP, there is no procedure significantly more efficient than exhaustive search over all supports of e. It may therefore seem that computing the true x_0 is more costly.

Recently, some papers[21][22][23] have shown that if the error e is sufficiently sparse, then the l^0 minimizer $\hat{x}_0$ is equal to the l^1 minimizer : $\hat{x}_0 = \hat{x}_1$ where

$$\hat{x}_1 = arg\min_{x} \parallel y - Ax \parallel_1 \tag{3.4}$$

This is a convex optimization problem, whose solution is unique and can be efficiently computed by linear programming.

We need a simultaneously sparse solution for x and e. Rewriting (2.2)

$$y = \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} = Bw \tag{3.5}$$

Here $B = \begin{bmatrix} A & I \end{bmatrix} \in \mathbb{R}^{m \times (n+m)}$, so the system $Bw = y$ is underdetermined and does no have a unique solution.But from the above discussion we can say that if the sparsest solution w_0 is sufficiently sparse, then it is same as l^1 norm minimization,

$$\hat{w}_1 = arg\min_{[x|e]} \parallel w \parallel_1 \; s.t.Bw = y \tag{3.6}$$

When does the $l^0 - l^1$ equivalence hold ?

We require to know precisely when the occlusion e and the coefficients x_0 are "sufficiently sparse". Donoho[25] defines the *equivalence breakdown point* (EBP) of a matrix B as the minimum number k such that if $y = Bw_0$ for some w_0 with less than k nonzero entries, then the minimal l^1 norm solution $\hat{w}_1$ to the system $Bw = y$ is equal to the sparse generator w_0. In literature there is a proof of existence of a constant ρ_0, such that EBP(B)$> \rho_0 m$ with overwhelming probability $m \to \infty$. An upper bound on EBP(B) comes from the theory of centrally neighbourly polytopes[26] :

$$EBP(B) \leq \lfloor (m+1)/3 \rfloor \tag{3.7}$$

This result indicates that we should not expect to perfectly recover $[x_0|e]$ if $n_i + |support(e)| > m/3$. In our context, we would like to know for a given set of training images, how much occlusion it can handle.

3.4 Modified Algorithm (l^1 norm minimization)

For this we first define k functions $\delta_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ the i-th of which preserves the coefficients corresponding to the i-th group and sets rest to zero i.e. $\delta_i(x) = \left[0...0x_i^T 0...0\right]^T \in \mathbb{R}^n$. The approximation in terms of the coefficients associated with the i-th group is then $\hat{y} = A\delta_i(\hat{x}) + \hat{e}$ and classification can be achieved by assigning y to the group that minimizes $\parallel y - A\delta_i(\hat{x_1}) - \hat{e_1} \parallel_2$.The implementation minimizes the l^1 norm via a primal-dual algorithm for linear programming[17][18].

Algorithm 1.1 (Robust Recognition By l^1 norm minimization)[27]

---

1 :**Input** : n training samples partitioned into k classes, $A_1, A_2, ...A_k$ and a test sample y.

2 : Normalize the training samples to have unit l^2 norm and set $B = [A_1 A_2 ... A_k I]$, where $A_i = [v_{i1} v_{i2} ... v_{in_i}]$, n_i is no. of objects belonging to class i for $i = 1, 2, ...k$.

3 : Solve the l^1 minimization problem :

$$\hat{w}_1 = arg \min_{[x|e]} \parallel w \parallel_1 \qquad \text{s.t. } Bw = y$$

by linear programming.

4 : Compute the residuals $r_i\,(y) = \parallel y - A\delta_i(\hat{x_1}) - \hat{e_1} \parallel_2$, for $i = 1, 2, ..., k$.

5 : **Output** : identity(y) $= arg \min_{i=1,2,...,k} r_i\,(y)$.

In the above algorithm, the equation $y = Bw$ is considered as hard constraint in the optimization. In reality the equality would never be satisfied exactly, most numerical implementations of linear program is stable and can tolerate small amount of error in the constraints. So one can redefine the above optimization problem as

$$\min_{w=[x|e]} \parallel w \parallel_1 s.t. \parallel Bw - y \parallel_2 \leq \epsilon \tag{3.8}$$

so as to tolerate error upto ϵ .

Though the above algorithm is robust in face recognition under partial occlusion the main disadvantage of this algorithm is it's time complexity. It takes on an average 20 minutes to identify an object. In real situation it may not much helpful. From the concept of it's sparse representation we have design an algorithm which is very faster than the above algorithm. We have used Moore-Penrose pseudoinverse and k-nearest neighbour(k-NN) algorithm for our proposed algorithm. In later sections we will discuss about k-nearest neighbour algorithm and how they help to design our algorithm.

In our algorithm we solve the system of linear equations (3.5)using pseudoinverse. Since $B = \begin{bmatrix} A & I \end{bmatrix} \in \mathbb{R}^{m\times(n+m)}$, the system does not have a unique solution.For 'best fit' we find the unique solution w by computing $w = B^{+}y$, where B^{+} is the pseudoinverse of B. So we get an estimation $\hat{w} = [\hat{x}|\hat{e}]$ of w for the system of equations (3.5) for which $\| Bw - y \|_2$ is minimized.

Now intuitively we want to reconstruct the test image from each class assuming that only one of them is closest to the original test image y.After that we use these images as training samples of a k-NN classifier. The vector corresponding to j-th object in the i-th class is computed by

$$\bar{v}_{ij} = v_{ij}\mu_{ij}(\hat{x}) + \hat{e} \tag{3.9}$$

where $\mu_{ij}(\hat{x})$ = coefficient in $\hat{x}$ corresponding to the vector v_{ij}, for $i = 1, 2, ...k$ and $j = 1, 2, ...n_i$. So the training set of the classifier is $\bar{B} = [\bar{A}_1\bar{A}_2....\bar{A}_k]$, where $\bar{A}_i = [\bar{v}_{i1}\bar{v}_{i2}...\bar{v}_{in_i}]$ and $group(\bar{v}_{ij}) = i$. Using k-NN algorithm we shall classify the test sample y with training set $\bar{B}$ and class vector $group$. We will see that k-nearest neighbour classifier works better than Euclidean norm minimization of $\| y - A\delta_i(\hat{x}) + \hat{e} \|_2$. Now we will discuss a little bit about k-nearest neighbour classifier.

3.5 k-Nearest Neighbours Algorithm

A classification problem for objects in a particular domain is the problem of separating these objects into smaller classes and giving a criteria whether a particular object in the domain is in a particular class or not. In pattern recognition, the k-nearest neighbours algorithm is a method for classifying point based on closest training samples in the feature space. An object is classified in class i if among its k nearest neighbours maximum no. of object belongs to class i. Here the nearest neighbour corresponds to the minimum Euclidean distance.

Let $(x_i, \theta_i); i = 1, 2, ...m$ be given where $x_i \in \mathbb{R}^n$ $\forall i = 1, 2, ...m$ and θ_i denote the class of x_i $\forall i = 1, 2, ...m$. Let there be C classes and $x \in \mathbb{R}^n$ be the point to be classified. Let k be a positive integer.

Steps of k-nearest neighbours algorithm :

Step 1 Find k nearest neighbours of x among $\{x_1, x_2, ...x_m\}$.

Step 2 Let k_i of the nearest neighbours belongs to class i, $i = 1, 2, ...C$.

Step 3 Classify x to class i if $k_i > k_j$, $\forall j \neq i$ and $\displaystyle\sum_{i=1}^{C} k_i = k$.

The main advantage of k-NN algorithm is it's robustness to noisy training data and also effective if the training data is large. It provides good generalization accuracy for a variety of real-world classification tasks and application.

3.6 Modified Algorithm (Pseudoinverse)

This is our modified algorithm where we consider the test face image is partially occluded. Our algorithm recover the signal by pseudoinverse and find the class of the test sample by k- NN classifier. Also sparse representation takes a major role in our algorithm.

Algorithm 2.1 (Robust Recognition by Pseudoinverse and k-NN algorithm)

1: **Input** : n training samples partitioned into k classes $A_1, A_2, ...A_k$ and a test sample y.

2: Normalize the training samples to have unit l^2 norm and set $B = [A_1 A_2 ... A_k I]$, where $A_i = [v_{i1} v_{i2} ... v_{in_i}]$, n_i is no. of objects belonging to class i for $i = 1, 2, ...k$.

3: Find the solution : $\hat{w} = B^+ y$, where B^+ is the pseudoinverse of B and $\hat{w} = [\hat{x}|\hat{e}]$.

4: Using k-NN algorithm classify the test sample y with the training sample $\bar{B} = [\bar{A}_1 \bar{A}_2 .... \bar{A}_k]$, where $\bar{A}_i = [\bar{v}_{i1} \bar{v}_{i2} ... \bar{v}_{in_i}]$ with $\bar{v}_{ij} = v_{ij} \mu_{ij}(\hat{x}) + \hat{e}$ and $\mu_{ij}(\hat{x}) =$ coefficient in $\hat{x}$ corresponding to the vector v_{ij}, for $i = 1, 2, ...k$ and $j = 1, 2, ...n_i$.

5: **Output:** Output of k-NN algorithm is one of the class from $\{1, 2, ...k\}$.

In this chapter we have shown l^1-minimization and pseudoinverse both approaches have some speciality. We have used k-NN algorithm basically to see how much performance it provides than minimum distance classifier. In the next chapter we have presented some experimental results to compare the performance between the l^1-minimization approach and pseudoinverse approach.

Chapter 4

Experimental Result and Comparison

4.1 Preliminary

In this chapter we will show some experimental results and comparison between algorithm 1.1 and algorithm 2.1, depending upon their performances in face recognition and time complexity. Also we give our view about the formation of training data and some informations of the images. For the experiment, we use ORL database. This dataset contains frontal face images of 38 subjects . Each subject have 10 different images under various illumination conditions. The images are resized to 60×50 pixels for limited memory space in MATLAB. So in this case B is 3000×3380 matrix. Each of training images v_i is scaled to have unit l^2 norm.

Here we consider two type of corruption in the test image. In one case we corrupt certain percentage of randomly chosen pixels from each of the test images, replacing their values by (256 - original pixel value), see fig 4.1. In another case we corrupt the test image by placing a contiguous block on the image at some position, see fig. 4.3. To the human eye, beyond 50% corruption, the corrupted images are hardly recognizable as face images.

4.2 Recognition despite Random Pixel Corruption

Here we take 380 images as training sample and 38 as test sample. In figure 4.1 we give some test samples. Table 4.1 shows upto 25% occlusion our algorithm gives better

Figure 4.1: From left to right 10, 20, 30, 40 and 50 percentage of random pixels occlusion are incurred in the test images

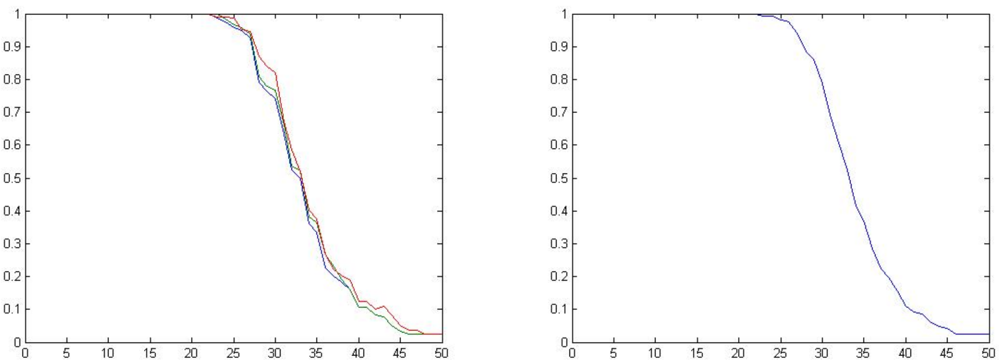

Figure 4.2: Algorithm 2.1 : 0-50 % random pixels occlusion along horizontal axis and success rate along vertical axis, first : blue(k=1), green(k=3), red(k=5) line represent the success rate, second : success rate for k=1

performance in face recognition under random pixels occlusion compare to the algorithm 1. For fig. 4.2(first) we take k=1,3,5 and random pixel corruption in the test sample. Here we take one sample from each class and delete those from training set and test those samples. We repeat this process for another set of images. Since there are 38 class we get 10 set of success rate, we take the average success rate. In figure 4.2(second) we take 380 as training sample and those are taken also as test sample. For random pixels occlusion, upto 25% our algorithm gives efficient recognition rate and after that it decreases rapidly. But in respect to time complexity our algorithm perform much faster than algorithm 1.1. Algorithm 1.1 takes on an average 12 minutes to recognize an image while our algorithm 2.1 takes 6 seconds in average.

Corruption(%)	0	5	10	15	20	25	30	35	40	45	50
Rec. rate(Algo.1.1)	100	100	100	100	97.3	94.7	84.2	60.5	18.4	13.2	5.3
Rec. rate(Algo.2.1,k=1)	100	100	100	100	100	98.2	77.4	35.6	10.2	3.4	2.5
Rec. rate(Algo.2.1,k=3)	100	100	97.4	97.4	97.4	94.7	81.6	28.95	7.9	2.63	2.63
Rec. rate(Algo.2.1,k=5)	100	100	100	100	97.4	92.11	71	31.6	5.26	2.63	2.63

Table 4.1: Recognition rate on ORL database with varying level of random pixels corruption

Figure 4.3: From left to right 10, 20, 30, 40 and 50 percentage of contiguous occlusion at eye level are incurred in the test images

4.3 Contiguous Block Occlusion starting from eye

For the result of table 4.2 we take 380 training images and 38 test images. Some test samples are shown in figure 4.3. Here we create the test samples by incurring contiguous block starting from almost the level of eye. Algorithm 2.1 gives better success rate for random pixel occlusion compare to contiguous block occlusion at the eye level, it shows that the pixels belonging to this region are very informative. As the previous case our algorithm is very faster compare to the algorithm 1.1. Figure 4.4 is graphical representation of success rate of algorithm 2.1 for 380 test samples.

Corruption(%)	0	5	10	15	20	25	30	35	40	45	50
Rec. rate(Algo.1.1)	100	100	100	100	94.7	94.7	71	60.4	39.5	32.8	25.6
Rec. rate(Algo.2.1,k=1)	100	100	100	86.8	65.8	57.9	52.6	52.6	31.6	15.8	7.9
Rec. rate(Algo.2.1,k=3)	100	100	100	89.47	68.42	57.89	55.26	47.37	26.32	18.42	7.89
Rec. rate(Algo.2.1,k=5)	100	100	100	90.2	73.68	60.13	57.24	50	42.11	36.84	18.42

Table 4.2: Recognition rate on ORL database with varying level of continuous block occlusion starting form the horizontal level of eye

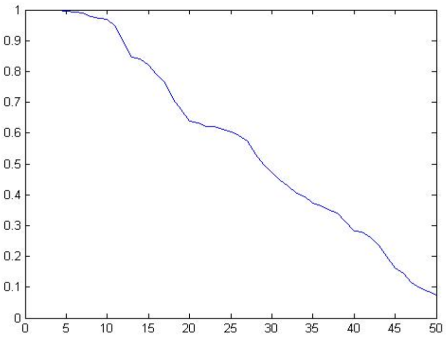

Figure 4.4: Algorithm 2.1 : k=1, 0-50 % contiguous block occlusion(from level of eye) along horizontal axis and success rate along vertical axis

Figure 4.5: From left to right 10, 20, 30, 40 and 50 percentage of contiguous occlusion are incurred at the center in the test images

4.4 Contiguous Block Occlusion at center

Figure 4.6 shows the graphical representation of success rate of algorithm 2.1 for 380 test samples. Some test samples are given in figure 4.5. Though in this case, the success rate of algorithm 1.1 is better than that of algorithm 2.1 (table 4.3), in real life situation when within few second we have to identify a person, algorithm 2.1 perform almost 120 times faster than algorithm 1.1.

Corruption(%)	0	5	10	15	20	25	30	35	40	45	50
Rec. rate(Algo.1.1)	100	100	100	100	100	100	81.5	71	65.2	55.26	39.5
Rec. rate(Algo.2.1,k=1)	100	100	100	98.3	93.4	86.4	81.7	64	36.7	27	20.4
Rec. rate(Algo.2.1,k=3)	100	100	100	100	100	92.11	68.42	55.26	28.95	21.05	18.42
Rec. rate(Algo.2.1,k=5)	100	100	100	100	100	92.11	73.68	52.63	36.84	28.95	26.32

Table 4.3: Recognition rate on ORL database with varying level of contiguous block occlusion at the center of the image

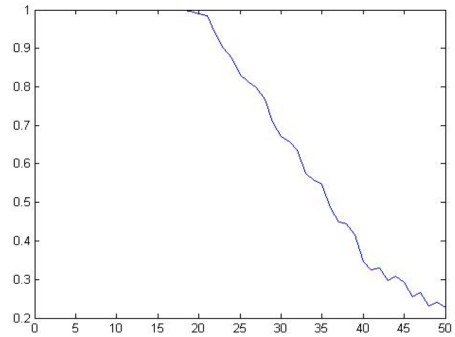

Figure 4.6: Algorithm 2.1 : k=1, 0-50 % contiguous block occlusion (at center) along horizontal axis and success rate along vertical axis

Figure 4.7: 20% of contiguous block occlusion of are incurred at random position in the test images

4.5 Contiguous Block Occlusion at random position

Here we create the test sample by placing a contiguous block at random position in the test images, see fig 4.7. Success rate in this case is given in the fig. 4.8. From figure it is clear that informations in the pixels are distributed through out the image in a irregular manner.

4.6 Other Experimental Results

Here we will discuss about the face images taken from ORL database.We experiment two aspect of the face images, the effect of the number of training images on the recognition rate and varying illumination in the face images.

Smaller Number Of Training Samples : Training face images are taken from different lighting condition, so illumination of the images vary from one face to another. Question is : How many training samples of each object is required for correct classification

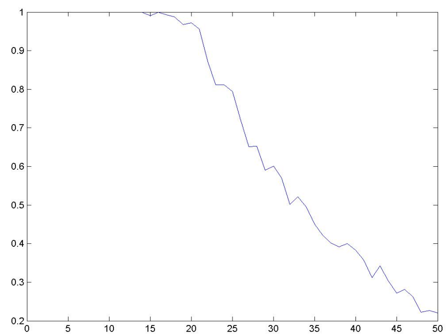

Figure 4.8: Algorithm 2.1 : k=1, 0-50 % contiguous block occlusion(at random position) along horizontal axis and success rate along vertical axis

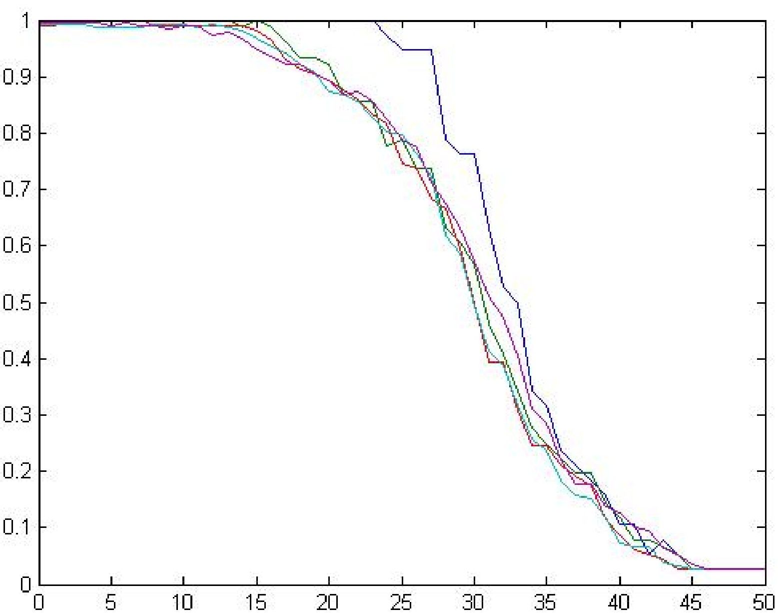

Figure 4.9: Algorithm 2.1 : 0-50 % random pixels occlusion along horizontal axis and success rate along vertical axis, blue, green, red, cyan, violet lines represent the success rates for training sample deleting 1, 2, 3, 4, 5 images respectively.

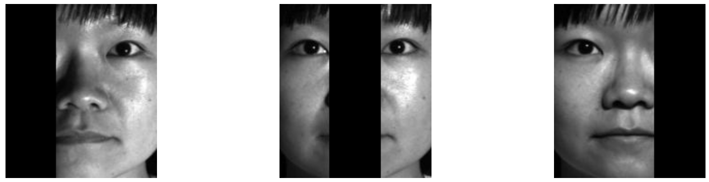

Figure 4.10: 33.3% of contiguous occlusion are incurred at vertical position in the test images

Position of Occlusion	Left	Middle	Right
Recognition rate(Algorithm 2.1)	0.77	0.55	0.52

Table 4.4: Recognition rate on ORL database with 33.3% of continuous block occlusion at vertical position of the image

of test sample? In this experiment we delete one training sample from one class and test it by remaining training samples. Similarly we delete two, three, four, five training sample respectively from each class and test those deleted face images by remaining training samples. In figure 4.9 we show the recognition rate for 0-50% occlusion incurred in the test sample for different training samples. We see that the recognition rate differed significantly for one to five deletion of training samples. But how many training samples are exactly required for efficient result is needed our further attention.

Varying Illumination : It is clear from the face images that images are taken from different lighting condition. We incurred contiguous occlusion(33.3%) at three vertical positions in the test face images, see figure 4.10. Recognition rate is given in the table 4.4. It shows that in the face images the left portion of the face is less informative than that of the right portion because in many images left side is not properly illuminated for varying lighting condition.

Figure 4.11: 33.3% of continuous occlusion are incurred at horizontal position in the test images

Position of Occlusion	Upper	Middle	Lower
Recognition rate(Algorithm 2.1)	0.13	0.8	0.51

Table 4.5: Recognition rate on ORL database with 33.3% of contiguous block occlusion at horizontal position of the image

Information in Eye Region : This experiment shows that the pixels in eye region and it's surrounding pixels contain maximum information. For this result we incurred 33.3% of contiguous occlusion at three horizontal positions in the test images, see figure 4.11. Table 4.5 gives the recognition rate for three different level of contiguous occlusion.

From the above results we see different type of behaviour of our algorithm since the test samples are partially occluded. Though our algorithm does not produce satisfactory result after certain percentage of occlusion, it takes a very little time to recognize face images. Here we can see k-NN classifier does not give much facility than Euclidean distance.

4.7 Discussion and Future Work

Here we have shown how the theory of sparse representation takes a major role in robust face recognition. Though l^1 norm minimization is efficient than pseudoinverse in solving system of linear equations, our proposed algorithm using pseudoinverse is very faster compare to algorithm using l^1 norm minimization. k-NN classifier improve the success rate in our proposed algorithm 2.1. These two algorithm have introduced a novel and comprehensive approach to face recognition which has provided new understanding on many issues such as feature selection, occlusion. We have shown that the pixels surrounding eye contain maximum information. In this case our approach does not produce satisfactory result, one can try to improve the result. The formulation of the problem via the perspective of sparse representation also leads to simple, efficient and robust algorithms based on the mathematical theory of compressed sensing. We strongly believe that this new approach and framework may provide new solutions to many other problems in the general area of pattern analysis and object recognition.

An intriguing problem for future work is whether this frame can be useful for object detection, in addition to recognition. From practical standpoint, it would also be useful to extend the algorithm to handle less constrained condition, such as variations in object pose. However it remains a problem how effective sparse representation will still be in this case and how many more training images it may require.

Bibliography

[1] Patrick Ross Anil K. Jain, Flynn. *Handbook of Biometrics.* Springer-verlag New York Inc.

[2] R. Hietmeyer. Biometric identification promises fast and secure processing of airline passengers. *The Intl Civil Aviation Organization Journal*, 5555, no. 9, pp. 10-11, 2000.

[3] T. Kohonen. *Self-organization and Associative Memory.* Springer-Verlag,Berlin, 1989.

[4] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Pattern Analysis and Machine Intelligence*, 12, no. 1, pp. 103-108, 1990.

[5] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cog. Neuroscience*, 3, no. 1, pp. 71-86, 1991.

[6] H.M. Lades M.S. Bartlett and T.J. Sejnowski. Independent component representations for face recognition. *Proceedings of SPIE*, 3299, pp. 528-539, 1998.

[7] J. P. Hespanha P. N. Belhumeur and D. J. Kriegman. eigenfaces vs. fisherfaces: Recognition using class specific linear projection.

IEEE Trans. Pattern Analysis and Machine Intelligence, 19, no. 7, pp. 711-720, July.

[8] A. Smola B. Scholkopf and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10, no. 5, pp. 1299-1319, 1998.

[9] V. de Silva J.B. Tenenbaum and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, pp. 2319-2323, 2000.

[10] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, pp. 2323-2326, 2000.

[11] T. Vetter and T. Poggio. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19, no. 7, pp. 733-742, 1997.

[12] J.M. Fellous N. Kruger, L. Wiskott and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19, no. 7, pp. 775-779, 1997.

[13] G.J. Edwards T.F. Cootes and C.J. Taylor. Active appearance models. *in Proc. European Conference on Computer Vision*, 2, pp. 484-498, 1998.

[14] T.F. Cootes G.J. Edwards and C.J. Taylor. Face recognition using active appearance models. *in Proc. European Conference on Computer Vision*, 2, pp. 581-695, 1998.

[15] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. *in Proc. ACM SIGGRAPH*, pp. 187-194,Mar. 1999.

[16] Sami Romdhani Volker Blanz and Thomas Vetter. Face identification across different poses and illuminations with a 3d morphable model. *in Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 202-207,2002.

[17] S. Boyd and L. Vandenberghe. Convex optimization. *Cambridge University Press*, 2004.

[18] E. Candes and J. Romberg. l^1-magic: Recovery of sparse signals via convex programming. *http://www.acm.caltech.edu/l1magic/*, 2005.

[19] E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26 394-395, 1920.

[20] Roger Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51 406-413., 1955.

[21] T. Tao E. Candes, M. Rudelson and R. Vershynin. Error correction via linear programming. *IEEE Symposium on FOCS*, 2005, pp. 295-308.

[22] E.Candes and P.A.Randall. Highly robust error correction by convex programming. *preprint,http://arxiv.org/abs/cs.IT/0612124*, 2006.

[23] D. Donoho. For most large underdetermined systems of linear equations the minimal l^1-norm near solution approximates the sparsest solution. *preprint, http://www-stat.stanford.edu/donoho/Reports/*, 2004.

[24] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, vol.209, pp. 237-260.

[25] D. Donoho. For most large underdetermined systems of linear equations the minimal l^1-norm near solution approximates the sparsest solution. *preprint, http://www-stat.stanford.edu/donoho/Reports/*, 2004.

[26] D. Donoho. Neighbourly polytopes and sparse solution of under determined linear equations. *IEEE Transactions on Information Theory*, 2006.

[27] Allen Yang Yi Ma John Wright, Arvind Ganesh. Robust face recognition via sparse representation. *IEEE Transactions,PAMI*, August,2007.