A STUDY OF THE MISSING DATA PROBLEMS IN 3D STRUCTURE RECONSTRUCTION AND 2D FACE RECOGNITION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Hongjun Jia, B.E., M.E.

* * * * *

The Ohio State University

2009

Dissertation Committee:

Prof. Aleix M. Martinez, Ph.D., Adviser
Prof. Yuan F. Zheng, Ph.D.
Prof. Alper Yilmaz, Ph.D.

Approved by

__________________________
Adviser
Graduate Program in
Electrical and Computer
Engineering
ABSTRACT

Missing data problems exist in many science and engineering fields. In this thesis, we discuss two important missing data problems in computer vision and pattern recognition: 3D reconstruction with structure from motion and 2D face recognition with occlusions.

The task of structure from motion for 3D reconstruction can be reduced to the problem of finding a low-rank \( r \) matrix that best fits an original data matrix of higher rank. The problem becomes especially difficult when the original data matrix has some missing entries and contains an unknown additive noise term in the remaining elements. The former problem can be solved by concatenating a set of \( r \)-column matrices which share a common, single \( r \)-dimensional solution space. Unfortunately, the number of possible submatrices is generally very large and, hence, the results obtained with one set of \( r \)-column matrices will generally be different from that captured by a different set. Ideally, we would like to find that solution which is least affected by noise. This requires that we determine which of the \( r \)-column matrices (i.e. which of the original feature points) are less influenced by the unknown noise term. A criterion which can successfully carry out such a selection is presented in this thesis. Our key result is to formally prove that the more distinct the \( r \) vectors of the \( r \)-column matrices are, the less they are swayed by noise. This key result is then combined with the use of a noise model to derive an upper-bound for the effect that noise and occlusions have on each of the \( r \)-column matrices. It is shown how this criterion can be
effectively used to recover the noise-free matrix of rank $r$. We derive affine and projective structure from motion (SFM) algorithms using the proposed criterion.

2D frontal face recognition with occlusions is another important problem in computer vision and pattern recognition. A more general problem of face recognition with occlusions is defined in this work as to classify a complete or partial face from a training set which may have partial face samples. We propose two new algorithms to solve it. One takes a reconstructive view and only uses the available information to reconstruct the test image from each class. The test face is labelled with the class within which the closest reconstruction is obtained. The second solution is in the framework of Support Vector Machines (SVM). The classical SVM cannot be applied when the feature vectors defining samples have missing entries. Here, the affine subspace which constituted with all possible fillings of the partial sample data is considered. A second term to maximize the probability of the accurate classification on the affine subspace is added to SVM criterion. The resulting optimization problem can be solved efficiently and we show how the global minimum of the error term is guaranteed under mild conditions.

Extensive validation on synthetic and real data sets for these two missing data problems shows the superiority of the proposed approaches over the state of the art.
To my parents ...
I thank everyone who has ever helped me.

First of all, I wish to greatly thank my advisor, Dr. Aleix M. Martinez, for his patience, support and instruction during the course of this work. I have learned a great deal of scientific knowledge and benefitted from his thoughts and intuition. He guides me towards the completion of this dissertation and my graduate study.

I would also like to thank all my labmates in the Computational Biology and Cognitive Science Lab for their support and stimulating discussions. They are Manli Zhu, Liya Ding, Onur Hamsici, Paulo Gotardo, Di You, Samuel Rivera, Shichuan Du and Fabian Benitez-Quiroz. In this fun environment, we learn and grow together. I would like to extend a special thanks to Jeff Fortuna for his helpful discussions and comments on my early draft. I am grateful to all my good friends who made it possible for me to enjoy a colorful life in Columbus, Ohio.

Finally, and most importantly, I wish to thank my parents and my girl friend for their love, understanding and encouragement accompanying me throughout this process.

This research was partially supported by the National Institutes of Health, grant R01 DC 005241, and the National Science Foundation, grant IIS 0713055.
VITA

October 24, 1978 . . . . . . . . . . . . . . . . . . . . . . . . . . . . Born - Shenyang, Liaoning, China

July, 2001 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . B.E. Electronics Engineering,
Tsinghua University, China

July, 2004 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . M.E. Electronics Engineering,
Tsinghua University, China

August, 2004 - present . . . . . . . . . . . . . . . . . . . . . . . Distinguished University Fellow,
Graduate Teaching Associate,
Graduate Research Associate,
Department of Electrical and Computer
Engineering, The Ohio State University,
Columbus, Ohio, USA

PUBLICATIONS

Research Publications


FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in:

Computer Vision and Pattern Recognition: Prof. Aleix M. Martinez
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstract</strong></td>
<td>ii</td>
</tr>
<tr>
<td><strong>Dedication</strong></td>
<td>iv</td>
</tr>
<tr>
<td><strong>Acknowledgments</strong></td>
<td>v</td>
</tr>
<tr>
<td><strong>Vita</strong></td>
<td>vi</td>
</tr>
<tr>
<td><strong>List of Tables</strong></td>
<td>xi</td>
</tr>
<tr>
<td><strong>List of Figures</strong></td>
<td>xii</td>
</tr>
<tr>
<td><strong>Chapters:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>1. Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Missing data problem in structure from motion</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Motivation and problem statement</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Literature review</td>
<td>10</td>
</tr>
<tr>
<td>1.2 Missing data problem in 2D face recognition</td>
<td>17</td>
</tr>
<tr>
<td>1.2.1 Motivation and problem statement</td>
<td>17</td>
</tr>
<tr>
<td>1.2.2 Literature review of appearance-based 2D face recognition</td>
<td>21</td>
</tr>
<tr>
<td>1.3 Contributions and thesis outline</td>
<td>23</td>
</tr>
<tr>
<td><strong>2. Low-rank Matrix Fitting Based on Subspace Perturbation Analysis with Applications to Structure from Motion</strong></td>
<td>26</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>26</td>
</tr>
<tr>
<td>2.2 Fitting a low-rank matrix with missing data</td>
<td>29</td>
</tr>
<tr>
<td>2.3 Deviation parameter criterion</td>
<td>32</td>
</tr>
<tr>
<td>2.3.1 Selecting the appropriate submatrices</td>
<td>32</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Percentage of times the deviation parameter criterion correctly predicts the ordering of the effects of noise. Percentages are given for a variety of noise terms and occlusions.</td>
</tr>
<tr>
<td>2.2</td>
<td>Denoising ability of the proposed factorization method.</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison between RAND, MME and DP.</td>
</tr>
<tr>
<td>2.4</td>
<td>RMSE results as given by ( \text{diff}_3 ) and average computation time.</td>
</tr>
<tr>
<td>2.5</td>
<td>Comparison between the MME and DP criteria in projective SFM</td>
</tr>
<tr>
<td>2.6</td>
<td>Notation</td>
</tr>
<tr>
<td>3.1</td>
<td>Training set ( {a \sim m} ). Testing set ( {a' \sim m'} ). Here, ( a = 54 ) and ( b = 39 ).</td>
</tr>
<tr>
<td>3.2</td>
<td>Experimental results (recognition rate in percentages) with a variety of training and testing sets. Here, ( a = 29 ) and ( b = 21 ).</td>
</tr>
<tr>
<td>3.3</td>
<td>Successful recognition rate (in percentages), with ( a = 54 ) and ( b = 39 ).</td>
</tr>
<tr>
<td>3.4</td>
<td>Successful recognition rate (in percentage) obtained using the specified image size and training and testing sets with the PWCM algorithm.</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The effects of noise on SVD. (a) The original data matrix is obtained by adding uniformly distributed Gaussian noise (with a standard deviation of 5). An additional error in one of the rows (with an extra 50% noise term) is then included. The dashed shape specifies the ground-truth; the solid shape the recovered result. (b) The recovered result obtained when the row containing most of the noise is eliminated before applying SVD.</td>
</tr>
<tr>
<td>1.2</td>
<td>An example of a sequence of images with occlusions. About 25% of 2D feature points are missing because of self-occlusions.</td>
</tr>
<tr>
<td>1.3</td>
<td>The illustration of the kinetic depth effect which demonstrates that the visual system can construct the 3D shape representations from motion information. Random 3D points are located on a rotating sphere, and the viewer can gain an impression of a rotating sphere from continuous 2D projections [72].</td>
</tr>
<tr>
<td>1.4</td>
<td>Some examples from the AR face database. About half of the samples have natural occlusions.</td>
</tr>
<tr>
<td>1.5</td>
<td>Different cases of face recognition with occlusions.</td>
</tr>
<tr>
<td>2.1</td>
<td>The distance between two spaces $\mathcal{X}$ and $\mathcal{Y}$ can be defined as the sine of the largest principal angles between them ($\theta(\mathcal{X}, \mathcal{Y})$).</td>
</tr>
<tr>
<td>2.2</td>
<td>Shown here is the relation between each pair of matrices in the process followed by the subspace constraint approach. The top part illustrates the relation between the noisy matrices, while the bottom part illustrates the noise-free case. In this figure, the empty squares in the matrix correspond to missing elements. Note that $\tilde{M}_i^T \tilde{A}_i = 0$ and $\tilde{N}_i^T \tilde{B}_i = 0$.</td>
</tr>
</tbody>
</table>
2.3 The effect of noise over each possible submatrix. (a) Actual distance between subspaces when the submatrices are sorted according to the number of deleted rows or DP value. In (b), the submatrices are ordered from having less to more missing elements. The abscissa corresponds to the index of the submatrices sorted by the numbers of missing entries, and the ordinate is the corresponding pb value. In (c) the submatrices are sorted by the actual distance between subspaces (dashed curve). The solid curve specifies the value of the deviation parameter derived in this chapter.  

44

2.4 (a) The average results over 100 trials. (b) A typical case for the relationship between DP and the number of missing rows. x-axis denotes the number of rows with missing entries.  

46

2.5 Two example curves of the RMSE with (a) \([m,n,r,\sigma,d\%] = [10, 16, 4, 1.00, 20\%]\) and (b) \([m,n,r,\sigma,d\%] = [15, 20, 3, 4.00, 30\%]\). The dashed curve is the true error (i.e. the difference between \(W\) and \(W_r\)), while the solid curve corresponds to that we can calculate (i.e. \(\hat{W}\) to \(W_r\)). The dashed lines indicate the number of submatrices used in each case and their corresponding RMSE. We can see that the RMSEs obtained with the true measure and with our estimate are practically identical.  

49

2.6 The average recovered error with 30\% missing data and different noise level, \(\sigma = 0.5, 1, 2, 4\).  

56

2.7 Shown here is (a) a frame of the box sequence, and (b) a frame of the dinosaur sequence.  

58

2.8 Plotted here are the reprojection errors obtained with Jacobs algorithm and the DP-based affine SFM. In (a) we show the RMSE over a total of 30 runs for each of the occluding percentages. In (b) we plot the MAE of each algorithm for each of the occlusions.  

58

2.9 Shown here are the reprojection errors for the proposed approach and Jacobs algorithm as functions of the noise term and the amount of occlusion. RMSE on (a) Jacobs algorithm [50] and (b) DP affine SFM. In (c) we show a slice of the plots in (a) and (b) at 40\% occlusion but with varying noise. This last plot allows for a one to one comparison.  

59

2.10 One example of over-fitting for global methods.  

62
2.11 The recovered tracks of the Dinosaur sequence: (a) on 4,983 points, (b) on 2,683 points and (c) on 336 points. (d) The 3D reconstruction of the dinosaur.

2.12 The average 2D reconstruction error after 25 iterations with the noise term as specified in the plot.

2.13 (a) One image in the wooden toy sequence. (b) The 3D structure of the object recovered by the DP-based projective SFM algorithm.

2.14 (a) One of the images in the model house sequence. (b) The recovered 3D structure of the scene.

2.15 The seven images of the wooden object sequence.

2.16 Originally marked (dots) feature points for the wooden object sequence.

2.17 Marked (dots) and recovered (squares) feature points for the wooden object sequence.

2.18 Projection of a set of 3D cubes onto each of the images.

3.1 An example of partial data filling with multiple weights. The visible parts are shown as shaded. The top and bottom occluded regions of the test data can be reconstructed with different sets of weights from the training set.

3.2 Classical SVM solutions for different (potential) filling-ins. \{p_1, p_2\} and \{q_1, q_2\} are in classes 1 and 2, respectively. The incomplete feature vector \( p_3 = (3, \bullet)^T \in \text{class 1} \).

3.3 The Correct Classification Probability (CCP) of a hyperplane. (a) Assuming a Gaussian distribution of \( S \), (b) the angle between \( S \) and \( l_i \) is proportional to the distance \( d(\bar{x}, q_0) \).

3.4 Iterative search to the optimization problem (3.24). In the \( k^{th} \) iteration, find the optimization solution \( w_k \) to (3.24) on the intersection region \( I_k \), and update \( \gamma : \gamma_k \rightarrow \gamma_{(k+1)} \) according to (3.26).

3.5 The Hyperplanes given by the solution to (3.16) with different regularizing parameters, (a) \( K = 0 \), (b) \( K = 1 \), (c) \( K = 10 \), (d) \( K = 100 \). The hyperplane becomes more parallel to the affine space \( S \) as \( K \) increases and the geometric margin becomes smaller at the same time.
3.6 (a-m) Shown here are the 13 images of the first session for one of the subjects in the AR face database. (n) An oval-shaped cropped example.

3.7 Two examples of face color detection and occlusion mask generation. In each row, from left to right: the full face, the face mask of the full face, the partially occluded face, the face mask of the occluded face and the final face occlusion mask.

3.8 Classification accuracy with synthetic occlusions. (a) Training image $a$, testing images $b, c, d$. (b) Training images $b, c, d$, testing image $a$. The image size is $54 \times 39$.

3.9 Classification accuracy with synthetic occlusions. (a) Training image $\{a, b, c\}$, testing images $d$. (b) Training images $\{a, b, c, d\}$, testing image $\{a', b', c', d'\}$. The image size is $29 \times 21$.

3.10 Successful classification rate using the proposed approach, PWCM, (with $r = 2$, 2-norm, $r = 1$, 1-norm, and $r = .5$, .5-quasi-norm). The results are compared to those in [68] and [100]. Here, $a = 170$ and $b = 120$.

3.11 Training set $\{a, b, c, a', b', c'\}$. Testing set $\{d, h, k, d', h', k'\}$. Here, $a = 100$ and $b = 52$.

3.12 Training set $\{a, b, c, d, a', b', c', d'\}$. Testing set $\{h, k, h', k'\}$. Here, $a = 83$ and $b = 60$.

3.13 Training set $\{a, b, c, d, e, f, g\}$. Testing set $\{h, i, j, k, l, m, h', i', j', k', l', m'\}$. Here, $a = 66$ and $b = 48$.

3.14 Training set $\{a, b, c, a', b', c'\}$. Testing set $\{e, f, e', f'\}$. Here, $a = 29$ and $b = 21$. 

xv
CHAPTER 1

INTRODUCTION

In general, the types of the data available will determine the algorithm we design. For example, weather forecasting is tackled in quite different ways depending on the data available. If we observe and record local and discrete data in weather stations, such as temperature, atmospheric pressure, wind direction and speed, humidity and precipitation, some simple statistical methods can be adopted to build a statistical model of the correlation between those data and the actual weather conditions. But if weather satellites are used to obtain the global and continuous information, numerical simulation becomes a better choice. In that case, a set of equations of how the state of a fluid changes with time are used.

The data type is not the only factor determining the design of an algorithm. There are many other variables that can determine the algorithms we need, e.g. data completeness, scale, consistency, accuracy and redundancy.

In this thesis, we focus on one of these factors: data completeness, which specifies which part of the data is available (visible) at a given time. In an ideal situation, one may obtain all the elements of the data we want. However, in practice, it is not always true that we are able to collect all features of the data. Let us again use the weather forecasting as an example. Let three data items including daily temperature, humidity and precipitation,
be described in a column vector. The data for a total of four days is collected to forecast tomorrow’s weather. In this case, there are a total of twelve elements in each data-set. If every entry in this data-set is well measured and recorded, this data-set is considered to be complete. But if one or several data entries are missing for some reason (e.g. thermometer failure), the data-set is said to be incomplete. An incomplete data-set needs to be dealt with an appropriate algorithm.

We call the incomplete data problem, the missing data problem. Missing data problems are inevitable in real world applications, and they appear in many disciplines of science, engineering, medicine and sociology. Examples are in computer vision [105], bioinformatics [28], statistics [41], epidemiology [46], data mining [122], environmental science [75] and marketing [3].

This thesis discusses two important yet challenging missing data problems in computer vision and pattern recognition: 3D structure from motion with missing entries and 2D face recognition with occlusions.

1.1 Missing data problem in structure from motion

1.1.1 Motivation and problem statement

The reconstruction of a 3D scene is one of the most important problems in the field of computer vision [25, 37]. It is the technique of how to extract the 3D structure as well as the relative motion of the camera from 2D images of a scene. This 3D reconstruction is a recurring problem in computer vision research, with applications in a variety of engineering and scientific areas, as for examples, in automation and robots [31], virtual reality [24, 65] and medical imaging [8].
Structure from Motion (SFM) is a technique used to recover the 3D structure and motion of world features as seen in 2D images. Popular ways to address the SFM problem are to extract and track points, lines or planes [1, 77]. Although extracting and tracking feature points has not been completely solved yet, in this thesis, we assume that a set of tracked feature points is available.

Theoretically, the 3D object structure can be recovered from two or three 2D views up to a projective ambiguity in the 3D space [37]. Without additional information of the camera or the scene, the affine or metric reconstruction cannot be computed. However, with multiple views of the 3D structure, it is possible to obtain a Euclidean reconstruction by means of the SFM technique. In this thesis, we focus on the general case of the SFM problem which requires a set of 2D point correspondences drawn from a set of images of an arbitrary scene obtained by uncalibrated cameras.

The factorization-based method originally proposed by Tomasi and Kanade [105] is a theoretically sounded approach, providing a closed-form solution to the problem. This solution is important because the algorithm is optimal in the least squares sense and non-iterative. Under the assumption of i.i.d. zero-mean Gaussian noise for each measurement point and the affine camera model, the factorization method achieves a maximum likelihood affine reconstruction [37].

Tomasi and Kanade provide a solution to the simplest case of SFM, where a single rigid object is photographed through an orthographic camera [105]. The main idea of the factorization method is that if there is no noise, the measurement matrix \( W \) containing all the 2D trajectories of 3D feature points can be expressed as the product of two other matrices, the shape matrix \( Q \), which contains the 3D positions of the tracked points, and the
motion matrix $P$, in which the information of the camera’s relative motion in the sequence can be found. We formulate this problem as follows.

A sequence of $q$ images with $n$ annotated points can be represented as a data matrix $W \in \mathbb{R}^{2q \times n}$

$$W = \begin{bmatrix} X_1 \\ \vdots \\ X_q \end{bmatrix}, \quad \text{where} \quad X_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad (1.1)$$

with each $X_i \in \mathbb{R}^{2 \times n}, i = 1, 2, \cdots, q$. $X_i$ is composed of the 2D coordinates of all the features in the $i^{th}$ image. All the $x$-coordinates are in $x_i$ and $y$-coordinates in $y_i$ ($x_i, y_i \in \mathbb{R}^{1 \times n}$), i.e.

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{in} \end{bmatrix},$$

$$y_i = \begin{bmatrix} y_{i1} & y_{i2} & \cdots & y_{in} \end{bmatrix}. \quad (1.2)$$

The mean of $x_i$ and $y_i$ has been subtracted separately.

In the affine SFM problem, this set of points is assumed to have resulted from the projection of a set of 3D point coordinates described by $Q \in \mathbb{R}^{3 \times n}$:

$$Q = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{bmatrix}, \quad (1.3)$$

where each column of $Q$ represents a point in 3D space.

The 3D point set undergoes a series of affine transformations between images to form the data matrix

$$W = P \cdot Q, \quad (1.4)$$

where $P \in \mathbb{R}^{2q \times 3}$ is constructed with $2 \times 3$ orthogonal affine transformation matrices $R_i$,

$$P = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_q \end{bmatrix}. \quad (1.5)$$
The rows in $R_i = (r_{i1}, r_{i2})^T, i = 1, 2, \cdots, q$, are the first two rows of a 3D rotation matrix $(r_{i1}, r_{i2}, r_{i3})^T$.

The rigidity and the orthography assumption guarantee that the rank of the 2D measurement matrix is fixed. Since the shape matrix $P$ has only 3 columns and the motion matrix $Q$ only 3 rows, the rank of the measurement matrix $W$ is at most 3. In a more realistic case, where a rigid object is observed under a projective camera, this rank would be at most 4 [98]. Now the structure from motion problem has been reduced to find the closest low-rank matrix approximation to the measurement matrix.

A Singular Value Decomposition (SVD) can be applied on $W$ to obtain the approximated rank-3 matrix, which gives the optimal solution in the least squares sense [61, 30]. In the noise-free case, if the data matrix $W$ has no missing elements, the motion and shape matrices can be recovered exactly (up to an arbitrary affine transformation) from the SVD of $W$ as

$$W = UDV^T.$$  \hspace{1cm} (1.6)

$P$ and $Q$ can be found from a rank-3 decomposition of $W$ by keeping the largest 3 singular values in $D$ and the corresponding singular vectors in $U$ and $V$. This defines the factorization method of Tomasi and Kanade [105].

The factorization technique based on SVD has been used in many research fields other than computer vision, as for example in mechanics [115], computer graphics [92] and activity recognition [12].

Factorization-based methods for recovering the structure and motion from an image sequence have many advantages, but they require a set of well-tracked features, which is not always provided in the real situation. The problems that the factorization-based algorithms may encounter include:
1. the feature points used to solve the correspondence problem cannot be detected with pixel accuracy;

2. some of the tracked feature points may not appear in all the views during the tracking process.

The former of these two problems is known as data noise, while the latter is usually referred to as missing data. These two problems limit the application of the factorization-based methods. Hence, it is necessary to consider other methods that can deal with an imperfectly tracked feature set.

The data noise problem can be stated as follows. Let the measurement matrix $\hat{W} \in \mathbb{R}^{m \times n}$ be the noisy version of an unknown matrix $W$ of rank $r \ll \min(m, n)$. Here, $m$ is a multiple of the number of frames in which each of the image feature points is being tracked (e.g. in affine SFM $m = 2q$, $q$ the number of frames, whereas in projective SFM $m = 3q$), and $n$ is the number of feature points. When the points in the 2D images are inaccurately measured, the SFM model becomes

$$\hat{W} = W + E = P \cdot Q + E,$$

where $\hat{W}$ is the actual measurement matrix and $E \in \mathbb{R}^{m \times n}$ is a matrix of the noise component at each feature point in each image. If the SVD is used to factorize $\hat{W}$, a least squares solution is obtained.

However, SVD does not generally work under the conditions of large noise. This is because the least squares solution for factorization favors those feature vectors associated to the largest variances (i.e. the outliers). In many practical cases, the points carrying most of the variance are actually those associated with noise, since it is the noise which makes the variance increase. If one fails to eliminate the subset of column vectors of the measurement
Figure 1.1: The effects of noise on SVD. (a) The original data matrix is obtained by adding uniformly distributed Gaussian noise (with a standard deviation of 5). An additional error in one of the rows (with an extra 50% noise term) is then included. The dashed shape specifies the ground-truth; the solid shape the recovered result. (b) The recovered result obtained when the row containing most of the noise is eliminated before applying SVD.

matrix carrying a large noise term, the SVD result (which is otherwise optimal in the least-squares sense) cannot guarantee a precise recovery of the 3D structure. Fig. 1.1 shows an example with one of the rows of $\hat{W}$ containing a large amount of noise. In this example the ground-truth (a 3D shape in the form of a letter L) is shown in (a) using dashed lines. The 3D shape recovered with SVD is delineated with solid lines, making it clear that the noisy row (i.e. the outlier) biases the whole result. In Fig. 1.1(b) we show the 3D shape recovered by SVD when the noisy row is deleted from $\hat{W}$. Here, we see that the recovered result is almost perfect. Note that this would in fact apply to any other method based on least squares not just SVD.

Unfortunately, the amount of noise in each feature point is a priori unknown. Hence, our main goal is to provide a good and useful estimate of which set of column vectors is (on average) less affected by the same noise term. This is the key to enhancing the accuracy of
the final result, because if we had such information, we would know which columns of the
data matrix are more appropriate to use to reconstruct the object structure.

The other problem that needs to be considered is the missing data one. For example,
a measurement matrix may have missing data due to the incomplete trajectories caused
by occlusion. The incomplete measurement matrix can also be contaminated with noise
or localization errors, which makes the problem yet more challenging. The missing data
problem in SFM cannot be solved directly with SVD. Although some algorithms have been
proposed to resolve the missing data problem recently [49, 67, 10], none or few of them can
deal with a high percentage of missing elements and a large noise term. This is especially
ture for projective cameras.

If the measurement matrix $\hat{W}$ has missing data, we will define a set $\Gamma$ representing all
those cells in the data matrix $\hat{W}$ that correspond to the feature points which are visible and
properly detected (tracked) in each of the frames of the image sequence. This set is thus
defined as

$$\Gamma = \{(i, j) \mid (i, j) \text{ successfully detected 2D visible points}\}. \quad (1.8)$$

Here, the $j$ component specifies the column of $\hat{W}$, corresponding to one of the coordinates
of the feature point visible in frame $i$. The non-missing elements in the measurement matrix
$\hat{W} = [\hat{w}_{ij}]$ can now be formally defined as

$$\hat{w}_{ij} = w_{ij} + e_{ij}, \forall (i, j) \in \Gamma,$$

where $E = [e_{ij}]$ is the noise matrix, and $[(\cdot)_{ij}]$ denotes a $m \times n$ matrix with the $(i, j)^{th}$
entry as $(\cdot)_{ij}$. Recall that in projective SFM, the third coordinate of the feature points is the
homogenous coordinate. We note that these do not carry a noise term. In our formulation
given above, this means that the $e_{ij}$ representing a homogenous coordinate will always need to be zero.

Now our goal is to recover the shape and motion matrix ($P$ and $Q$) from the incomplete and noisy matrix $\hat{W}$.

To resolve the problem caused by missing data, Jacobs [49, 50] proposes to construct a set of submatrices by randomly selecting three columns from the data matrix $\hat{W}$. If there was no missing data, any three columns would define the solution space. However, if some of the cells of the three randomly selected columns are missing, several solutions will be possible. By combining the solutions observed with a sufficient number of different triple-column submatrices, we can obtain a full reconstruction. This process can also be used to initialize global approaches, generally resulting in more accurate global fits. Jacobs’ fitting solution provides a mechanism to calculate the closest low-dimensional subspace without filling the missing data.

The main problem with Jacobs’ approach is the randomness of the triple-column selection process. The random submatrix selection impairs the overall performance and the algorithm does not always produce a consistent and accurate recovery. Also, since the selection of the three columns is random, there is no mechanism to know whether the selected columns carry most of the noise term.

It is our contention, in this thesis, that the selection of the columns constituting the submatrix should be carried out on the basis of how noise affects them and on the number of missing entries.

We have pointed out the fact that different subsets of $\hat{W}$ affect the final recovery distinctly. Therefore, an in-depth analysis of this effect will help us build a proper selection criterion to guarantee a precise yet stable reconstruction. In this thesis, we propose to first
sort the $r$-column submatrices based on an estimate of the effect that noise and occlusion have on them. Then, we select a sufficient number of submatrices to reconstruct the entire object.

An example of a missing data problem in structure from motion is demonstrated in Fig. 1.2. A wooden toy is placed on the table and some pictures are taken from close range. Because of self-occlusions, not all the marked points on the object can be located in all frames and about 25% of the 2D feature points are missing. The manually marking process will also introduce noise on the 2D coordinates. Since the distance between the object and the camera is comparable to the size of the object, the projective effect in those pictures is significant and cannot be ignored. This is an extremely difficult case of SFM. One of the main objective of this thesis is to calculate the 3D structure in cases such as this.

1.1.2 Literature review

There are several ways to obtain the 3D structure of a scene from 2D images. For example, we can use optical flow [4], motion parallax [59, 93], visual hull [34, 73, 60], a model-based approach [94] or a structure from motion (SFM) algorithm [105]. Compared with the other methods, SFM methods have less restrictions on the 3D structure. Moreover, no prior information of the scene is necessary. SFM represents an important class of methods for 3D reconstruction.

One of the first studies of SFM was that of Wallach and O’Connell [116]. They put forward the possibility of estimating the rigid 3D structure using the pattern of motion produced from a rotating object with a light source onto a translucent screen, and studied the Kinetic Depth Effect (KDE) as illustrated in Fig. 1.3. Based on this idea, Ullman
studied the three-dimensional description given by the SFM problem [112]. Once the correspondences between 2D points in different images are established, we can determine the resulting 2D transformations in terms of the 3D structure and motion of the viewed objects. Some of the first SFM algorithms were implemented in [88, 64]. Many different methods have been developed to address this SFM problem since.

In generally, according to different formulations of the SFM problem and the number of features or images used, SFM solutions fall into two classes: the factorization and the non-factorization approaches.
Figure 1.3: The illustration of the kinetic depth effect which demonstrates that the visual system can construct the 3D shape representations from motion information. Random 3D points are located on a rotating sphere, and the viewer can gain an impression of a rotating sphere from continuous 2D projections [72].

The non-factorization methods try to use the minimum information to achieve the 3D structure and motion based on the analysis of the view geometry and the camera imaging process [64, 110], or formulate the problem into a non-linear optimization framework and adopt some iterative methods in numerical optimization to obtain the final solution [95, 118].

The factorization-based methods generally track a large amount of feature points through a sequence of 2D images. They then put all the available 2D coordinates into one measurement matrix, which can then be decomposed into the product of two factor matrices as summarized above. For example, Kontsevich et al. [58] presented a similar formulation to that of Tomasi and Kanade [105]. After the factorization step, the 3D structure is only recovered up to a projective or affine ambiguity. To get the Euclidean structure, a Euclidean upgrade step is needed, providing a reconstruction up to an arbitrary similarity
transformation. In the factorization-based methods, any number of images can be handled simultaneously without preferential treatment for any subgroup of views.

The original factorization method has been extended in several ways. Tomasi and Kanade’s solution only applies to a single rigid object undergoing an orthographic projection. This factorization method was extended to deal with other camera models, e.g. the weak-perspective and the para-perspective models [85, 86]. Although some research shows that the affine camera is a good approximation of real camera, in some cases, a perspective projection is required [25]. Eagle and Hogervorst’s [18] indicates that the perspective information plays a very important role to recover the 3D structure in the human visual system. A full perspective factorization-based method was proposed by Triggs and Sturm [98, 107]. In the projective case, the projective depth of each 2D projection has to be recovered before one can apply the factorization approach.

Because of the introduction of a perspective camera model, a rank-4 constraint has to be considered. In [98, 107], the authors estimate the projective depth under the constraints of multi-view epipolar geometry. Another way to estimate the projective depth is to iteratively update it with respect to the re-projection errors. Here, some initial values of the projective depths have to be given, e.g. all ones after some proper transformation of the image coordinates [36]. A robust iterative method to estimate the projective depth was proposed by Hung and Tang [101, 45]. Heyden’s method [42, 43] and Chen and Suter’s method [11] rely on the subspace analysis to resolve the projective SFM problem. It is important to note that the low-rank matrix fitting result is especially critical to the iterative solution of the projective SFM [101, 45]. If the recoveries in each iteration are not accurate, the error will propagate to the estimation of the projective depths resulting in a system with positive feedback.
Since noise in the measurement matrix is inevitable, some researchers give an in-depth study of its effect on SFM. Irani and Anandan [48] showed that the actual uncertainty on feature positions has strong directionality and that the least squares error minimized by SVD is not any longer the correct error to minimize. The proper error is covariance-weighted squared-error, which is the Mahalanobis distance. Thomas and Oliensis’ method [103] is described as a new incremental algorithm to estimate and correct the error in computing the camera motion between images. This fusion algorithm computes the final reconstruction from intermediate reconstructions by analyzing the uncertainties in them. Chen and Suter [11] analyzed the denoising capacity of SVD and gave a constrained result in terms of the noise level, the sizes of the measurement matrix and the rank. Robust statistical techniques are used by Aanaes et al. [1].

The factorization approaches defined thus far can only be used on a complete measurement matrix. A matrix is complete only if each entry of that matrix has a specified value. In many real world SFM problems, we cannot get a complete data matrix.

Tomasi and Kanade [105] applied their original factorization method on the largest full submatrix, and used the imputation method to recover the missing components. However, to locate the largest full submatrix is a NP-hard problem, and thus, a sub-optimal approach is presented in [10].

The iterative approach proposed by Shum et al. [95] is based on the bi-linear formulation of the SFM problem. This method uses the classical decomposition of the measurement matrix \( \hat{W} \) into two matrices, one describing the camera motion \( \mathbf{P} \) and the other the object’s shape \( \mathbf{Q} \), using standard SVD but with zeros or average or random values in place of the missing elements. This decomposition allows for the definition of two optimization approaches. One is to optimize \( \mathbf{P} \) by keeping \( \mathbf{Q} \) fixed. The second optimization requires to
fix $P$ and solve for $Q$. This trick reduces the original bilinear problem to two linear ones where the goal is to minimize the norm of the difference between the measurement matrix and its SVD reconstruction when using the non-missing elements of the data matrix. This solution reduces to a weighted least-squares problem, which can be iterated until convergence. This method can converge to a locally optimal solution, but it cannot guarantee a globally optimal one. In Gruber and Weiss’ [33], the well known EM algorithm [74] is adopted. This method allows handling the missing data and the directional uncertainty on each single feature. [89, 106, 35] propose the use of other optimization approaches.

The factorization of the low-rank matrix, $W = PQ$, suggests an alternative solution using a subspace constraint: the spanning space of the column vectors of $W$ and $P$ should be identical. However, this constraint only makes sense when we have noise-free entries. To deal with noise, Jacobs [49, 50] proposes an approach where the subspace constraint can be derived from several submatrices. Jacobs’ approach is to then combine these local solutions to find the global one. To achieve this, each column of the measurement matrix is regarded as the coordinates of a point in a $m$-dimensional space. SVD can then be used to find the best 3D subspace, $\mathcal{W}$, fitting the $n$ available points. When there is neither noise nor missing data, $\mathcal{W}$ is the space spanned by any three linearly independent columns. If there are some missing elements in the measurement matrix, each column spans an affine subspace that accounts for all the possible missing elements. In this case, $\mathcal{W}$ lies in the space spanned by three such affine subspaces. By following this argument and letting $\mathcal{B}_k$ be the space spanned by the $k^{th}$ column triplet, we have $\mathcal{W} \subseteq \mathcal{B}_k$. Then, $\mathcal{W}$ should be a subset of the intersection of all possible $\mathcal{B}_k$,

$$\mathcal{W} \subseteq \mathcal{G} = \cap \mathcal{B}_k, \quad k = 1, 2, \ldots, l,$$

where $l$ is the number of all possible $\mathcal{B}_k$. 

15
If noise is introduced into the equation and we follow our previous notation, which uses the symbol $\hat{}$ to specify the corresponding noisy versions \(e.g., \hat{G} \text{ and } \hat{W}\), then, $\hat{G}$ will become empty because our target $\hat{W}$ cannot accurately lie in any $\hat{B}_k$. A null-space based method is used to solve this problem. All the matrix representations of the orthogonal complementary space of $\hat{B}_k$ are packed together to form a matrix representation of $\hat{G}^\perp$. This can now be decomposed using standard SVD, providing the least-squares solution. The three singular vectors corresponding to the three smallest singular values are selected to form a 3D linear space $\hat{W}$ to be orthogonal to the matrix considered to be closest to $W$. The affine shape of the original structure is thus recovered from $\hat{W}$ [49, 50]. This approach falls within the area of “subset selection,” where a set of columns is selected to generate a solution. A review and variants of this approach can be found in [76].

Jacobs’ solution is an elegant way to deal with missing data. However, in practice, we see that the recovered results vary extensively when measured by the Mean Square Error (MSE). The reason for this is simple. When the triple-columns carrying the least amount of noise are selected, the recovered shape will generally be very close to the ground-truth and the MSE will be small. Unfortunately, if one or more of the triple-columns carrying large amounts of noise are used, the result will be far from optimal, leading to a large MSE. Therefore, the remaining problem to be addressed within this framework is to find a criterion that determines which triple-columns are associated to less noise and thus are the best candidates for the algorithm.

Chen and Suter propose one such criterion in [10], where the fitness of each column is inversely proportional to the number of missing elements. This means that those columns with less missing elements are better candidates for reconstruction. This works well when the noise is evenly distributed. However, in many instances the columns with more missing
entries are precisely those carrying the least amount of noise or, equivalently, those less affected by it. In these cases, Chen and Suter’s approach would result in large MSEs. Our experimental results will show that indeed, many times, the recovery obtained from those columns with more missing elements carries a lower MSE. Our goal in this thesis is to propose an alternate approach and show that it generally provides better fits. The proposed criterion is based on the subset selection approach [76] as it was done in [50]. The derivations provided below are for the singular value decomposition of the submatrices. As detailed in [120], several of these results should extend to the eigen-decomposition as well.

1.2 Missing data problem in 2D face recognition

1.2.1 Motivation and problem statement

In the last several decades, face recognition has attracted increasing attention in both scientific research and industrial applications, e.g. access control, attendance management, information security, and surveillance. Compared with other biometric features, as for example, fingerprint, palm-print, iris and signature, faces can be recorded more easily and do not require special devices. Face recognition also has a large number of applications in human-computer interaction. Two examples are facial expressions of emotion and sign languages. However, face recognition is still not fully solved because of the significant appearance changes due to many factors.

There are two main categories of face recognition technique: feature-based and appearance-based methods. Feature-based approaches use local points of interest and their structure to differentiate faces [13]. The appearance-based approaches is proposed to process faces from raw image [111, 6]. In the appearance-based approach, the whole face region is used
as the raw input to the system. One of the most widely used appearance-based face recognition algorithm is eigenfaces which is based on Principal Component Analysis (PCA) [96, 57, 111].

In most of the appearance-based algorithms, face classification is usually preceded by face localization and registration to reduce the complexity and improve the performance of face recognition [16]. However, there are still many other factors (e.g., expression, pose, lighting and aging), which affect the performance of face recognition. To date, many appearance-based algorithms have been defined and some of them achieve success under some constrained conditions, such as changing expression [68], different pose [87] and varying illumination [29]. In this dissertation, we focus on yet another problem, the occlusion on faces, which occurs often in practice.
In Fig. 1.4, we show some examples selected from a popular face data-set, the AR face database [70]. This database includes face images with natural occlusions. Occlusions are usually caused by body parts (such as hair and hands) and personal accessories (such as glasses and scarf). This can easily result in a failure of the direct use of existing face recognition algorithms.

In Fig. 1.5 we show three possible scenarios for the problem of face recognition with occlusions. In the first row, we have the most studied case – non-occluded faces in training and occluded faces in testing. Many algorithms have been proposed to address this first case, ranging from dividing the test face image into a set of local regions to sophisticated statistical methods and sparse representation [68, 21, 121]. However, these approaches rely on a non-occluded training set, and most methods do not address the problem of constructing a model (or classifier) from occluded images. The second and third rows illustrate two other possible cases, which are rarely discussed in the literature: a) training with occluded and non-occluded faces, and b) training with occluded faces only. A realistic face recognition system ought to allow all these cases.
In general, there are two different ways to handle occlusions in face recognition. Some methods do not detect the face occlusions before the face recognition step, but try to reduce the effect of those parts [68, 100, 121]. In the proposed solutions for 2D face recognition with occlusions, we take the other approach. We first detect the occlusion on each training/testing image.

Let the training set have \( n \) labelled face images of size \( a \times b \) and corresponding to one of the \( C \) different classes. After the vectorization of all face images, we have \( n \) data samples, \( x_1, x_2, \ldots, x_n \), and each of them is defined on a feature set \( F = \{f_1, f_2, \ldots, f_d\} \). The labels on all training samples, \( y_1, y_2, \ldots, y_n \), are from the set \( \{1, 2, \ldots, C\} \). For every testing face image \( t \), the goal of the proposed algorithms is to classify \( t \) into one of these \( C \) classes with high accuracy.

We propose to resolve the problem of face recognition with occlusions with two different strategies. We first take a reconstructive view. In this approach, only the non-occluded facial information is used. After a total of \( C \) such reconstructions of the test image are created, one per each possible class, the reconstruction which is closest to the test image is chosen and the corresponding class label is assigned to the test image. The hypothesis behind this idea is that the most accurate reconstruction will be given when one uses the sample images belonging to the correct class. We argue that this is a grounded hypothesis, since the image reconstruction of a frontal face image will generally be most accurately obtained when combining face images of the person it represents rather than with images of other individuals.

The second solution is a Support Vector Machines (SVM) approach. The classical SVM requires a complete training set, which means it cannot be applied when the feature vectors defining the data samples have missing entries. Let us use a simple example to explain the
new SVM solution. When \( k \) features are missing in a sample vector of class 1, these define an affine subspace of \( k \) dimensions. The goal of the new SVM algorithm is to maximize the margin between the vectors of class 1 and class 2 on those dimensions with no missing elements and, at the same time, maximize the margin between the vectors in class 2 and the affine subspace of class 1. This second term of the SVM criterion will minimize the overlap between the classification hyperplane and the subspace of solutions in class 1, because we do not know which values in this subspace a test vector can take. The hyperplane minimizing this overlap is obviously the one parallel to the missing dimensions. However, this condition is too restrictive, because its solution will generally contradict that obtained when maximizing the margin of the visible data. To resolve this problem, we define a criterion which minimizes the probability of overlap. And the resulting optimization problem will be solved to obtain the classification hyperplane.

### 1.2.2 Literature review of appearance-based 2D face recognition

In the appearance-based 2D face recognition paradigm, only the texture of the face image is considered. We usually represent a face image of \( a \times b \) pixels with a vector in a space of \( ab \) dimensions. The two most popular solutions are the eigenface method [111] which is based on Principle Component Analysis (PCA) and Fisherface [6] which is based on Linear Discriminant Analysis (LDA).

PCA-based algorithms extract a low-dimensional subspace in which the variance is maximized. This is equivalent to minimizing the reconstruction error from the subspace. In the extracted PCA space, the distance between the test sample and the training samples can be measured in different ways followed by the nearest mean classifier or the nearest
neighbor classifier. Euclidean and Mahalanobis distances are two of the most popular distances being used, but other measures exist [123, 83]. In contrast, the supervised learning method LDA searches for the subspace where the between-class variance and the within-class variance are maximized and minimized, respectively. Some research [71] shows that when the training dataset is small, PCA is superior to LDA, but, in general, when the number of training images per class is large, LDA-based algorithms are superior.

Other subspace methods for face recognition include Laplacianfaces [38] and Independent Component Analysis (ICA) based approaches [56]. The Laplacianfaces approach uses the locality preserving projection to obtain a face subspace that best detects the essential face manifold. ICA-based face representation maximizes the statistical independence between the basis features by extracting the higher order statistical information from the training images.

Support Vector Machines [113] is another supervised learning algorithm and it focuses on maximizing the geometric margin between two classes in the feature space. It can also be used to classify faces as demonstrated in [39, 40]. Deniz et al. [15] combined SVM and ICA to gain invariances to some experimental settings.

However, the face recognition methods defined above depend on fully visible faces, and thus will fail when occlusions are present. Several methods specifically designed to resolve this problem have been proposed [79, 68, 2, 100, 19, 124, 21, 99, 121].

Some algorithms tackle the occlusion problem using a local approach without an explicit occlusion detection step. For example, the eigenwindow method based on local PCA is proposed in [79] with three criteria developed to help the windows selection.

Martinez [68] proposed a probabilistic approach to do recognition from a single sample per class. In this approach, a face is divided into local regions and each region is compared
in each class, yielding a probability for each. The global best match is determined by adding all the probabilities over all the regions. The occluded region is determined by computing how close the original image is to the face-space. This probabilistic approach can also be applied to face recognition from multiple images and video sequences [124]. Tan et al. extended the above probabilistic approach using the self-organizing maps (SOM) [100]. Non-metric similarity [91, 51] between sub-blocks can also be applied [99].

All the methods mentioned above do not need an explicit occlusion detection step, since those regions far away from a facial part will have a low effect to the final result. Note that, local methods cannot give a proper evaluation on those partially occluded sub-blocks and thus are sensitive to the initial division of faces.

Some algorithms use the occluded facial image as a whole without dividing it into sub-regions. For example, the sparse representation of the test image [121] is pursued in consideration of the following two facts. First, only a relatively small fraction of image pixels are usually occluded. Second, the training images of the same subject correspond to a small portion of the training set. Based on a sparse representation computed by $l_1$-minimization, the test sample can be assigned to the object class that minimizes the residual error.

In [21], the authors proposed to detect the occlusion first using a random sampling method followed by a robust LDA over the non-occluded pixels.

1.3 Contributions and thesis outline

As mentioned above, two important and popular missing data problems in computer vision and pattern recognition are discussed in this thesis. The first is the missing data
problem in 3D reconstruction of objects. The second is the missing data problem in 2D face recognition. The chapters of this thesis are organized as follows.

Chapter 2 introduces the solution to the problem of low-rank \((r)\) matrix fitting based on a novel Deviation Parameter (DP) criterion which can help select the best submatrices with missing data to be employed in the reconstruction process. The applications of this new algorithm on both the affine and projective structure from motion are presented. Our key result is to formally prove that the more distinct the \(r\) vectors of the \(r\)-column matrices are, the less they are swayed by noise. This key result is then combined with the use of a noise model to derive an upper-bound for the effect that noise and occlusions can have on each of the \(r\)-column matrices. Extensive validation on synthetic and real datasets will show the superiority of the proposed approach over the state of the art.

In Chapter 3, two new algorithms for 2D face recognition with occlusions are proposed. These two algorithms cannot only solve the most studied case of face recognition with occlusions (recognizing an occluded face from the non-occluded training set), but can also solve the overlooked situation where some training faces are partially occluded. The first algorithm takes a reconstructive view and tries to find the closest within-class reconstruction to label the test image. The key point in this formulation is to base this reconstruction solely on the visible data in the training and testing sets. A multi-weight linear estimation is provided to deal with different occlusion conditions of each feature in the training set. The second algorithm for face recognition is a Support Vector Machines method. We modify the objective function of the classical SVM solution by adding a new term which is directly associated with the probability of the accurate classification on the affine subspace defined by the incomplete data. This new approach extends the traditional SVM to the case
where some of the data points are not visible. We refer to this new algorithm the Partial-data Support Vector Machines (PSVM). The resulting optimization problem can be solved efficiently and we show how the global optimum is guaranteed under mild conditions. We show experimental results on a publicly available dataset.

A discussion of the results and potential future research are given in Chapter 4.
CHAPTER 2

LOW-RANK MATRIX FITTING BASED ON SUBSPACE PERTURBATION ANALYSIS WITH APPLICATIONS TO STRUCTURE FROM MOTION

2.1 Introduction

Many computer vision problems, as well as several others in computer graphics, pattern recognition and bioinformatics reduce to finding an appropriate low-rank matrix that successfully approximates the original data matrix [115, 92]. This problem becomes especially challenging when the original matrix contains noise and has several missing elements. One classical example is in the estimation of optical flow from video sequences, where several of the tracked fiducials can become occluded or be imprecisely detected (i.e. noisy measurements) [4, 47]. Other classical applications are in the recognition of faces using the so-called appearance-based approach [78], and in the classification of patients using microarray technology in medicine and bioinformatics [109, 26].

In this section, we will focus on yet another classical problem – that of structure from motion (SFM). SFM is one of the fundamental problems in computer vision, with a large variety of applications [64, 37]. The SFM problem requires that we compute the 3D structure of an arbitrary scene from a set of 2D image point correspondences. These point
correspondences are drawn from a set of images obtained by (usually) uncalibrated cameras. As above, the SFM problem becomes especially difficult when the feature points used to solve the correspondence problem cannot be precisely detected or become occluded for the duration of some frames. The former of these two problems is known as data noise, while the latter is usually referred to as missing data.

The noisy data matrix can be formally defined as

$$\hat{W} = W + E,$$

(2.1)

where $E$ is the matrix containing the unknown noise terms. In general, the addition of the noise matrix $E$ will enforce $\hat{W}$ to be of full-rank, while the original data matrix is known to be of a lower rank $r$. Hence, our objective is to find that rank-$r$ matrix $W_r$ that best approximates the noise-free data matrix $W$, i.e. we wish to minimize the difference between $W$ and $W_r$. Unfortunately, $W$ is not known and, hence, one is generally left to estimate $W_r$ from $\hat{W}$ by means of an appropriate metric.

A typical solution to this problem is to employ the classical Singular Value Decomposition (SVD) [30]. The popularity of SVD is due to it providing the least squares error solution. Among other problems, this has been applied to affine SFM [105, 86], appearance-based recognition of faces [96] and objects [78], optical flow [4, 47], and microarray analysis [109, 26].

Unfortunately, SVD does not work under large noise as we have pointed out in Chapter 1, and the amount of noise in each feature point is a priori unknown. Hence, we want to find a good estimate of the effect that the same noise term can have on a set of column vectors on average. This can help us know which columns of the data matrix to use to reconstruct our object. Note that this approach does not require that we know the actual noise term in each column, but rather how the noise affects the recovery.
In this chapter, we demonstrate that the same amount of noise does not affect every data matrix (or submatrix) equally. When a data matrix includes columns or rows defining very distinct vectors, the effect of noise is minimal. When the matrix rows or columns define very similar vectors though, the same amount of noise has a very large effect. We will prove this result formally and derive an upper-bound estimate of the effect of the noise term. This does not provide an optimal mechanism to reduce noise. Without any knowledge of the noise term, one can only hope to optimize a criterion. Our result shows that when no a priori information on the noise term is given, it is convenient to select those submatrices that are less swayed by the noise term.

Thus far, we have dealt with the problem caused by noise. The other typical problem addressed by researchers working in SFM is that of missing data caused by self-occlusions or by the incapacity of the tracking algorithm to successfully locate one or several of the feature points in some of the frames of our image sequence. In these cases, the measurement matrix becomes incomplete. To make this worse, and as already stated above, this incomplete data matrix will generally contain noise. Hence, we extend the definition given in (2.1) to include the missing data case as in Chapter 1.

To resolve the missing data problem in the low-rank matrix fitting, Jacobs [49, 50] proposes a linear fitting solution based on the subspace analysis. In this method, a column with missing entries will define an affine subspace. The orthogonal space of the subspace spanned by \( r \) different affine spaces is calculated. The final result is obtained by finding the \( r \) dimensional space which is most orthogonal to the spanning space of a number of those orthogonal spaces. The major problem of this approach is that the \( r \)-column selection process is on a random base. In this chapter, we propose to first sort the \( r \)-column submatrices
based on an estimate of the effect that noise and occlusion have on them. Then, we select a sufficient number of submatrices to reconstruct the entire object.

After a more formal, in-depth presentation of the problem in Section 2.2, we provide detailed derivations of our approach in Section 2.3. In Section 2.4, we show how the derived algorithm can be efficiently used to resolve the problems of noise and missing data in affine and projective SFM. We show that our SFM algorithm can recover the position of the 2D image points and corresponding 3D feature points with high accuracy. This is so even when the point has a large noise term or when it has been temporarily occluded. Experimental results are in Section 2.5. We conclude in Section 2.6.

2.2 Fitting a low-rank matrix with missing data

As summarized in the preceding section, missing matrix entries require that we redefine approaches such as SVD. Several solutions to this problem have been proposed over the years.

In SFM we need to find an appropriate low-rank matrix which contains the information of the structure and motion of the object being tracked. This suggests a direct approach where we search for those missing entries that maintain a low rank measurement matrix. This low-rank constraint means that only the first $r$ singular values of $W$ can be non-zero. Since $r$ is usually known (e.g. four in the general SFM problem), one can derive approaches for filling in the missing elements [17]. Two such solutions are proposed by Friedland et al. [26] and Troyanskaya et al. [109]. In these algorithms, the authors first fill in the missing elements with zeros, and then utilize SVD to find that $r$-dimensional subspace that best fits the data. This allows the authors to project the data matrix onto the subspace, resulting in a new low-rank representation. The measurement matrix can now be reconstructed using
the basis vectors selected by this process. This algorithm can be iterated, yielding better least-squares estimates of the missing elements.

In a related paper, Shum et al. [95] propose to iteratively update the shape matrix and the motion matrix by fixing only one of them and fill in the missing data in each iteration. The methods described in this section are related to earlier extensions of SVD with missing elements. One particular case is defined by Wiberg [119], whose derivations also provide the minimum number of observations needed to get a unique solution.

More recently, EM-based extensions of SVD and PCA have been proposed to address the problem of missing elements [90, 104, 114, 33]. In [14], this is further improved with robust statistics. And, it has been shown that projection pursuit could also be employed, since this compares favorably with other robust estimators [62]. Unfortunately, in general, the approaches defined in this section can only guarantee convergence to a local minimum.

The missing data in the low-rank data matrix are estimated before the factorization step in the methods described so far. Another class of methods to fit a low-rank matrix with missing data try to find the essential low dimensional structure directly without filling the matrix. To resolve the problem, one can use additional constraints, for example, a subspace constraint described in Chapter 1 that the spanning space of the column vectors of W and P should be identical. This constraint has been successfully applied in [50] and the detailed procedure will be presented in the following section.

Random Sample Consensus (RANSAC) [22] is a well-known, robust method to deal with outliers. In RANSAC-like procedures, a fixed number of data units are randomly selected to fit a model which will be measured over all the data. If there are no missing components, any r-column submatrix will span a r-dimensional subspace. However, in the case with missing data, the fixed number of columns which can fit a complete r-dimensional
subspace is no longer available. In such a case, we would need to first eliminate the rows with missing elements.

For the perspective model, the SFM problem becomes more difficult since the projective depths for all points have to recovered before the factorization can be implemented. Sturm and Triggs [98] applied the constraints in Epipolar geometry to formulate a set of linear equations. However, this approach will result in a noise-sensitive estimation of the projective depth. The iterative approaches to recover the projective depths are known to be less sensitive to the noise term than non-iterative ones [107]. In the projective SFM, the iterative solution for factorization with noise and missing data is thus widely used.

There are several iterative algorithms which adopt the factorization method in each step. For example, Tang and Hung [45, 101] use the iterative method to minimize the 2D reprojection errors to achieve the recovery of the projective depths for projective reconstruction. Mahamud and Hebert [66] introduce an iterative method based on the subspace constraints. In their algorithm, the projective depths in each column are computed as a generalized eigenvector problem with a complete matrix. The eigenvector corresponding to the largest eigenvalue is the solution to the projective depths. The missing data in the measurement matrix will be estimated based on the low-rank matrix factorization with the current projective depths. This process can be implemented iteratively. The iteration between the factorization and the projective depths calculation stops when the results are converged to some local or global optima based on different criteria [45, 101, 66]. To improve the performance of projective SFM, a non-linear bundle adjustment [27, 108] is usually adopted after the iterative factorization.
2.3 Deviation parameter criterion

A new method of determining which subset of the data is most suitable for fitting the low-rank matrix is introduced. This method only depends on the subset itself and does not require of a precalculation of the result as it would be the case in robust statistics [124].

2.3.1 Selecting the appropriate submatrices

The problem we need to address is that of determining which of the (triple-column) submatrices are less affected by the noise term. We first note that each of these submatrices defines a subspace. If the 3D subspace given by the complete (in the sense of not having any missing entries), noise-free data matrix $W$ were known, then our task would simplify to finding those submatrices spanning subspaces similar to that of $W$. In such an idealistic case, we would still need to define a mechanism that can compute the distance between two spaces – that of the ground-truth $W$ and that of the $i^{th}$ submatrix $\hat{B}_i$ (with $i = 1, \ldots, l$).

Since the dimensionality of these subspaces is identical, we can compute the actual distance between them, $dist(W, \hat{B}_i)$, by looking at the largest principal angle. More formally, if the dimensionality of our subspaces is $r$, then the principal angles $0 \leq \theta_{i,1} \leq \cdots \leq \theta_{i,r} \leq \pi/2$ between $W$ and $\hat{B}_i$ can be obtained recursively from

$$\cos(\theta_{i,k}) = \max_{u \in W} \max_{v \in \hat{B}_i} u^T v = u_{i,k}^T v_{i,k},$$

with the added constraints $\|u\| = \|v\| = 1$, $u^T u_{i,h} = 0$, $v^T v_{i,h} = 0$, $h = 1, \ldots, k - 1$, $1 \leq k \leq r$. Then, the distance between subspaces is $dist(W, \hat{B}_i) = \sin(\theta_{i,r})$ [30]. For simplicity of notation, we will use $\theta(X, Y)$ to specify the largest principal angle between the spaces defined by the matrices $X$ and $Y$. Also, we will refer to the space spanned by $X$ as $\mathcal{X}$, and the largest principal angle between two spaces as $\theta(\mathcal{X}, \mathcal{Y})$ (Fig. 2.1).
Figure 2.1: The distance between two spaces $\mathcal{X}$ and $\mathcal{Y}$ can be defined as the sine of the largest principal angles between them ($\theta(\mathcal{X}, \mathcal{Y})$).

In actuality, the space $\mathcal{W}$ defining the ground-truth is not known. Therefore, we need to find another mechanism to determine how noise influences each subspace $\hat{\mathcal{B}}_i$. To this end, we first note that the same amount of noise does not affect every submatrix defining a given subspace $\hat{\mathcal{B}}_i$ equally. In fact, when the vectors given by the columns of our submatrix are separated by a large angle (e.g. close to $90^\circ$), additive noise will have a limited effect. However, when the same noise is added to a submatrix with similar column vectors, the new resulting (noisy) subspace will be more different from the original noise-free version than the effect observed in submatrices with very distinct column vectors.

To clarify this point, one can think of the effect that noise has in a stereo vision system. When using two images describing a similar view of the same scene, noise will have a greater sway than that observed when the views are far apart. In other words, the 3D reconstruction will be generally less affected by noise when the vectors describing the scene correspond to sufficiently different views. This is so because a small amount of noise will correspond to a large percentage of the difference between two similar vectors but a small percentage in those that are far apart. This result will be formally proven next.
2.3.2 Subspace perturbation analysis

Since our approach is that of determining which submatrices are best to be employed based on their robustness to noise, our criterion needs to be related to the matrix sensitivity to noise. Subspace perturbation analysis [97] provides a way to formally derive a solution.

To this end, let the matrix \( \hat{W} \) be a perturbed version of the noise-free matrix \( W \in \mathbb{R}^{m \times n} \) with some missing elements, and let the rank of \( W \) be \( r \ll \min(m, n) \). The perturbing matrix \( E \) is considered to be the additive noise in the observation of \( W \) in (2.1).

As stated above, a classical approach to recover \( W \) is to find that matrix \( W_r \) of rank \( r \) which minimizes the difference between itself and \( W \). A convenient norm to calculate this difference is the Frobenius norm calculated over all non-missing elements of the matrix, \( \| W - W_r \|_{F, \text{nonmissing}} \).

If we select a \( r \)-column submatrix \( \hat{B}_i \) \((i = 1, 2, \cdots, l)\) from \( \hat{W} \) and follow the steps of Jacobs algorithm [50] presented in Chapter 1, we get the complementary space of \( \hat{B}_i \) which, if correct, should be orthogonal to the desired low-dimensional space \( \hat{W} \). After selecting a sufficient number of such submatrices, we build a large matrix which is composed of all these null-spaces. SVD can then be used to select the \( r \) singular vectors corresponding to the \( r \) smallest singular values. This generates a \( r \)-dimensional linear subspace \( \hat{W} \). This linear subspace can be utilized to recover the missing data and reconstruct a rank-\( r \) matrix, as we will see shortly.

For each \( \hat{B}_i \), let \( \hat{A}_i \in \mathbb{R}^{p \times r} \) \((p \leq m)\) be the corresponding reduced form of \( \hat{B}_i \) with no missing entries. This, we accomplish by removing all the rows having at least one missing entry. The spanning space of \( \hat{A}_i \) is \( \hat{A}_i \), with \( \hat{M}_i \) its null space and \( \hat{M}_i \) the \( p \times (p - r) \) matrix defining it. We can then expand \( \hat{M}_i \) by simply adding zeros in all the rows which were removed from \( \hat{B}_i \). This provides us with an original size matrix, \( \hat{N}_i \in \mathbb{R}^{m \times n} \).
Figure 2.2: Shown here is the relation between each pair of matrices in the process followed by the subspace constraint approach. The top part illustrates the relation between the noisy matrices, while the bottom part illustrates the noise-free case. In this figure, the empty squares in the matrix correspond to missing elements. Note that $\hat{M}_i^T \hat{A}_i = 0$ and $\hat{N}_i^T \hat{B}_i = 0$.

$\mathbb{R}^{m \times (p-r)}$, representing the null space. This process is illustrated in Fig. 2.2. All such $\hat{N}_i$ will then be packed together to form the matrix $\hat{N} = [\hat{N}_1 \hat{N}_2 \cdots \hat{N}_i]$. Our low-dimensional linear space $\hat{\mathcal{W}}$ will be orthogonal to the rank-$(n-r)$ matrix closest to $\hat{N}$ as given by the Frobenius norm. Note that the subspace will be correctly recovered only when all the rows are considered at least in one of the $\hat{A}_i$ matrices.

Had we applied the process just defined to the noise-free condition, we would have obtained the matrices $B_i$, $A_i$, $M_i$, $N_i$, and their corresponding space (e.g. $\mathcal{A}_i$ and $\mathcal{M}_i$)
Since \( \hat{N} \) is the final matrix to be decomposed by SVD, the difference between \( \hat{N} \) and \( N \) should be small. Similarly, we also require the difference between each \( \hat{N}_i \) and \( N_i \) to be as small as possible. Since \( N_i \) and \( M_i \) and \( \hat{N}_i \) and \( \hat{M}_i \) are the same except for added zeros, the principal angles between \( N_i \) and \( \hat{N}_i \) are the same as that between \( M_i \) and \( \hat{M}_i \). Also, since \( A_i \) and \( M_i \) and \( \hat{A}_i \) and \( \hat{M}_i \) are both orthogonal to each other, the principal angle between \( M_i \) and \( \hat{M}_i \) will be the same as that between \( A_i \) and \( \hat{A}_i \). Therefore, we have \( \theta(N_i, \hat{N}_i) = \theta(M_i, \hat{M}_i) = \theta(A_i, \hat{A}_i) \) and we can calculate the distance between subspaces directly from \( A_i \) and \( \hat{A}_i \), i.e., \( \text{dist}(N_i, \hat{N}_i) = \text{dist}(A_i, \hat{A}_i) = \sin \theta(A_i, \hat{A}_i) \).

The process defined in the preceding paragraph allows us to calculate the distance between the noise-free and noisy versions of the original data matrix with missing elements, by concentrating on spaces spanned by the matrices \( A_i \) and \( \hat{A}_i \) as

\[
\text{pb}(B_i, \hat{B}_i) = \sin \theta(R(\hat{A}_i), R(A_i)) = \sin \theta(A_i, \hat{A}_i),
\]

where \( R(X) \) denotes the range space \( \mathcal{X} \) spanned by the column vectors in \( X \), and \( \text{pb} \) stands for perturbation.

We note that, in the process described above, the farther the space of \( \hat{N} \) is from that of \( N \), the farther apart the recovered low-dimensional subspace \( \hat{W} \) will be from the ground-truth \( W \). Hence, we want to choose those \( \hat{N}_i \) that defines the smallest of all possible largest principal angles \( \theta(N_i, \hat{N}_i) \). This is equivalent to selecting those \( r \)-column submatrices \( \hat{B}_i \in \mathbb{R}^{m \times r} \) with the smallest \( \text{pb}(B_i, \hat{B}_i) \) values.

Since the ground-truth is not known, we need to resort to some other type of comparison. As argued in the previous section, those submatrices \( \hat{B}_i \) with most dissimilar column vectors will be generally less affected by additive noise. The framework derived in this section enables us to prove this result formally. Moreover, note the deleted rows do not
directly relate to the calculated \( pb \) value. In fact, this is not necessary because these rows will not enter in the computation of \( \hat{N}_i \) and, hence, have no effect in the final result.

**2.3.3 Upper-bound for the subspace distance**

Let \( \hat{B} \in \mathbb{R}^{m \times r} \) represent one of the \( r \)-column submatrices from \( \hat{W} \), and let \( \hat{A} \in \mathbb{R}^{p \times r} \) \((p \leq m)\) denote its complete part, which, as above, we construct by deleting all rows with at least one missing element. Further, let \( E_A \in \mathbb{R}^{p \times r} \) be the noise matrix defining the noise term on \( \hat{A} \). The matrix \( E_A \) can be obtained directly from \( E \) by deleting the same rows and columns that were eliminated to convert \( \hat{W} \) into \( \hat{A} \).

We begin by determining a bound on the distance between \( \mathcal{R}(\hat{A}) \) and \( \mathcal{R}(A) \), where \( A \) denotes the corresponding noise-free version of \( \hat{A} \), in terms of some \( f(\hat{A}) \) and \( \|E_A\|_2 \), where \( \|\cdot\|_2 \) denotes the 2-norm.

To do this, we extend on a perturbation theorem provided by Wedin [117] to determine a bound on the \( pb \) value between \( \hat{B} \) and \( B \), which is equal to the sine of the largest principal angle between the spanning spaces of \( A \) and \( \hat{A} \). Before the theorem is presented, we need to introduce some definitions. Let

\[
\hat{A} = A + E_A,
\]

with SVDs \( A = U \Sigma V^T \) and \( \hat{A} = \hat{U} \hat{\Sigma} \hat{V}^T \). \( A \) and \( \hat{A} \) can be decomposed as

\[
A = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}^T,
\]

\[
\hat{A} = \begin{bmatrix} \hat{U}_1 & \hat{U}_2 \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \hat{V}_1 \\ \hat{V}_2 \end{bmatrix}^T,
\]

where \( U_1, \hat{U}_1 \in \mathbb{R}^{p \times s}, U_2, \hat{U}_2 \in \mathbb{R}^{p \times (p-s)}, V_1, \hat{V}_1 \in \mathbb{R}^{r \times s}, V_2, \hat{V}_2 \in \mathbb{R}^{r \times (r-s)}, s \leq r, \Sigma_1 = \text{diag}(\sigma_1, \ldots, \sigma_s), \Sigma_2 = \text{diag}(\sigma_{s+1}, \ldots, \sigma_r), \hat{\Sigma}_1 = \text{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_s) \) and \( \hat{\Sigma}_2 = \text{diag}(\hat{\sigma}_{s+1}, \ldots, \hat{\sigma}_r) \).
In the theorem that follows, the representation of $E_A$ in the orthonormal subspace $\hat{V}_1$ and $\hat{U}_1$ is used, rather than $E_A$ directly, since we are defining bounds for subspaces. As such, define
\[
R = A \hat{V}_1 - \hat{U}_1 \hat{\Sigma}_1 = -E_A \hat{V}_1 \\
D = A^T \hat{U}_1 - \hat{V}_1 \hat{\Sigma}_1 = -E_A^T \hat{U}_1.
\]
From these definitions we note that
\[
\|R\| = \|E_A \hat{V}_1\| \leq \|E_A\| \\
\|D\| = \|E_A^T \hat{U}_1\| \leq \|E_A\|,
\]
where $\|\cdot\|$ represents an appropriate norm, such as the 2-norm or the Frobenius norm. We can now state Wedin’s theorem [117] as follows.

**Theorem 1** If $\exists \alpha, \delta > 0$ such that
\[
\min \sigma(\hat{\Sigma}_1) \geq \alpha + \delta \quad \text{and} \quad \max \sigma(\Sigma_2) \leq \alpha,
\]
then
\[
\max\{\|\sin \Phi\|, \|\sin \Theta\|\} \leq \frac{\max\{\|R\|, \|D\|\}}{\delta},
\]
where $\Phi$ is a matrix of angles between $\mathcal{R}(U_1)$ and $\mathcal{R}(\hat{U}_1)$, $\Theta$ is a matrix of angles between $\mathcal{R}(V_1)$ and $\mathcal{R}(\hat{V}_1)$, and the operator $\sigma(\cdot)$ denotes the singular value spectrum. Here, $\sin \Phi = U_2^T \hat{U}_1$ and $\sin \Theta = V_2^T \hat{V}_1$.

Returning to the original problem, to determine a bound on the distance between the spanning spaces $\mathcal{R}(\hat{A})$ and $\mathcal{R}(A)$, we use the sine of the largest principal angle between $\mathcal{R}(\hat{A})$ and $\mathcal{R}(A)$ to measure the distance between two submatrices, defined as $\sin \theta(\mathcal{R}(\hat{A}), \mathcal{R}(A))$. Using the inequality in (2.4), we have
\[
\sin \theta(\mathcal{R}(\hat{A}), \mathcal{R}(A)) \leq \max\{\|\sin \Phi\|, \|\sin \Theta\|\} \\
\leq \frac{\max\{\|R\|, \|D\|\}}{\text{gap}} \leq \frac{\|E_A\|}{\text{gap}},
\]
where $\Phi$ now represents the matrix of angles between $\mathcal{R}(A)$ and $\mathcal{R}(\hat{A})$, and $\Theta$ is a matrix of angles between $\mathcal{R}(A^T)$ and $\mathcal{R}(\hat{A}^T)$. From the conditions in Theorem 1, it follows

$$\text{gap} = \min \sigma(\hat{\Sigma}_1) - \max \sigma(\Sigma_2),$$

with $\Sigma_2$ and $\hat{\Sigma}_1$ the diagonal matrices of the singular values of $A$ and $\hat{A}$ as shown in (2.3).

Finally, for the case where $p > r$, the first $r$ left singular vectors of $\hat{U}$ will span $\mathcal{R}(\hat{A})$ and the remaining singular vectors will have corresponding singular values equal to 0. To illustrate, consider the following SVD for $p > r$, rewritten with a square matrix of singular values

$$A = [U_1' \ U_2'] [\Sigma_1' \ 0 \ 0 \ \Sigma_2'] [V_1' \ V_2']^T,$$  \hspace{1cm} (2.6)

where $U_1' \in \mathbb{R}^{p \times r}$, $U_2' \in \mathbb{R}^{p \times (p-r)}$, $V_1' \in \mathbb{R}^{r \times r}$, $V_2' \in \mathbb{R}^{r \times (p-r)}$, $\Sigma_1' = \text{diag}(\sigma_1, \ldots, \sigma_r)$, and $\Sigma_2' = \text{diag}(\sigma_{r+1}, \ldots, \sigma_p)$. Similarly the decomposition of $\hat{A}$ is

$$\hat{A} = [\hat{U}_1' \ \hat{U}_2'] [\hat{\Sigma}_1' \ 0 \ 0 \ \hat{\Sigma}_2'] [\hat{V}_1' \ \hat{V}_2']^T,$$  \hspace{1cm} (2.7)

where $\hat{U}_1' \in \mathbb{R}^{p \times r}$, $\hat{U}_2' \in \mathbb{R}^{p \times (p-r)}$, $\hat{V}_1' \in \mathbb{R}^{r \times r}$, $\hat{V}_2' \in \mathbb{R}^{r \times (p-r)}$, $\hat{\Sigma}_1' = \text{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_r)$ and $\hat{\Sigma}_2' = \text{diag}(\hat{\sigma}_{r+1}, \ldots, \hat{\sigma}_p)$.

Comparing (2.6) and (2.7) to (2.3) with $s = r$, yields $U_1' = U_1$, $U_2' = U_2$, $\Sigma_1' = \Sigma_1$, $\Sigma_2' = 0$, $V_1' = V_1$ and $V_2' = I$. From this result, we see that the condition in Wedin’s theorem is $\max \sigma(\Sigma_2') = 0$ and $\min \sigma(\hat{\Sigma}_1') = \min \sigma(\hat{A})$. And, hence, $\text{gap} = \min \sigma(\hat{\Sigma}_1') - \max \sigma(\Sigma_2') = \min \sigma(\hat{A})$. Finally, from (2.5) we get our expression for the bound on the distance between $\mathcal{R}(A)$ and $\mathcal{R}(\hat{A})$ as

$$\sin \theta(\mathcal{R}(\hat{A}), \mathcal{R}(A)) \leq \frac{\|E_A\|}{\min \sigma(\hat{A})}. \hspace{1cm} (2.8)$$

### 2.3.4 The deviation parameter criterion

In order to use the perturbation bound derived in (2.8), we must have some knowledge of the nature of the noise matrix $E$. Since this cannot be measured directly, we will resort
to a statistical model. Specifically, we assume that the elements of the noise matrix are Gaussian distributed according to $N(0, \sigma^2)$ and are independent of each other. This is a reasonable characterization of noise that arises from inaccurate measurements, i.e. it is most probable that the measurement will be close to the actual value.

Let $X$ be a $p \times r$ Gaussian random matrix with entries distributed according to $N(0, 1)$. As shown by Johnstone [55], the mean and variance of the largest eigenvalue $\lambda_1$ of the covariance matrix $X^T X$ can be approximated by

$$
\mu_{\lambda_1} = (\sqrt{p-1} + \sqrt{r})^2
$$

$$
\sigma_{\lambda_1} = (\sqrt{p-1} + \sqrt{r}) \left( \frac{1}{\sqrt{p-1}} + \frac{1}{\sqrt{r}} \right)^{\frac{1}{2}}.
$$

(2.9)

Given the fact that $\lambda_1$ is the square of the largest singular value of $X$ and the largest singular value of a matrix is its 2-norm, the expected value of the 2-norm of our error matrix $E_A \in \mathbb{R}^{p \times r}$ with elements distributed as $N(0, \sigma^2)$ is

$$
\mu_{\|E_A\|} \approx (\sqrt{p-1} + \sqrt{r}) \sigma.
$$

(2.10)

It is important to note from (2.9) that while the estimate of the mean increases almost linearly with the dimensionality of the matrix, the standard deviation increases at a lower rate. This means that the ratio of standard deviation over mean, which indicates the percentage of error in the estimate of the mean, decreases as the size of the matrix increases. This is relevant because, in most applications, the size of our matrices is quite large. And, in such cases, it becomes appropriate to substitute the 2-norm of our matrix by the mean derived in (2.9) as was done in (2.10).

Now considering the results in (2.2), (2.8) and (2.10) together, we obtain the expectation of the upper-bound of $pb(B, \hat{B})$. We refer to this value as the Deviation Parameter (DP)
[52, 53] of submatrix $\hat{\mathbf{B}} \in \mathbb{R}^{m \times r}$ with $b$ rows of missing data ($b = m - p$), and define it as

$$DP(\hat{\mathbf{B}}) = \frac{(\sqrt{m - b - 1} + \sqrt{r})\sigma}{\min(\sigma(\hat{\mathbf{A}}))}.$$  \hspace{1cm} (2.11)

In this result, $\sigma$ represents the variance of the noise given by the 2-norm of the noise matrix $E_A$. As mentioned earlier, it is common to assume that the norm of the noise in each submatrix is identical – although the distribution of the noise in each column/row can vary considerably from matrix to matrix. Under this condition, all $\sigma$’s are the same and can thus be eliminated from the computation of the DP criterion presented above.

Given a measurement of a low-rank matrix with noise and missing data, the difference between it and the ground-truth can be estimated with (2.11). This $DP$ value gives a measure of the sensitivity of the low-rank matrix to perturbation due to i.i.d. additive Gaussian noise. If a matrix has a larger $DP$ value, the corrupted matrix will generally be farther from the original one under the same noise situation than that with a smaller $DP$ value. Note that this result is based on an upper-bound and may thus not lead to optimal solutions. Nonetheless, the Deviation Parameter provides an appropriate mechanism to select the candidate submatrices in the problem of finding a low-dimensional linear space representation of the data matrix.

### 2.3.5 Analysis of the DP criterion

In the preceding sections, we have given an estimate of the possible noise carried on an incomplete, noisy matrix based on results borrowed from matrix perturbation [97] and random matrix theory. In our derivations provided thus far, we use the expectation of the upper-bound to measure the distance between $\mathcal{R}(\mathbf{A})$ and $\mathcal{R}(\hat{\mathbf{A}})$ instead of the actual distance because this cannot be calculated. Although in our previous section we provided
grounded arguments for such a definition, we now turn to a study of the behavior of this newly defined criterion to demonstrate its effectiveness.

We start with a $m \times n$ matrix $\mathbf{W}$ of rank $r$ and with every entry in the matrix in $[0, 100]$. This matrix is then contaminated with additive Gaussian noise, $N(0, \sigma^2)$, followed by a random occlusion mask with $d\%$ of missing entries. Let the resulting matrix be $\hat{\mathbf{W}}$. The deviation parameter $DP_i$ for each of the possible submatrices $\hat{\mathbf{B}}_i$ (each constructed with $r$ columns) is computed. Now, the resulting $DP_i$ values need to be compared to the actual distance $\theta_i$ between the submatrix $\hat{\mathbf{A}}_i$ and its corresponding noise-free submatrix $\mathbf{A}_i$.

Our first study will test how many times the deviation parameter criterion correctly selects that submatrix leading to a closer estimate of the ground-truth. For this to happen, the $DP_i$ and $DP_j$ computed from two submatrices $\hat{\mathbf{B}}_i$ and $\hat{\mathbf{B}}_j$ should be in the same order as $\theta_i$ and $\theta_j$. That is, if $DP_i > DP_j$, then $\theta_i > \theta_j$, and vice-versa. This can be readily computed as the percentage ($\rho\%$) of times that $\hat{\mathbf{B}}_i$ and $\hat{\mathbf{B}}_j$ yield $(DP_i - DP_j)(\theta_i - \theta_j) > 0$. This is shown in Table 2.1. These results are calculated from a total of 30 trials with the matrix size, noise and missing data as specified in the table. From these results, we see that even for reasonably large amounts of missing elements and noise, the ordering provided by our deviation parameter criterion is almost always consistent with that of the ground-truth.

The DP criterion thus provides a convenient estimate of how useful each submatrix is. This is further illustrated in Fig. 2.3(a). In this case, we first generate a $50 \times 10$ rank 4 matrix $\mathbf{A}$ and its corresponding noisy version $\hat{\mathbf{A}}$ (with zero-mean Gaussian noise). Then, we obtained the 210 possible 4-column submatrices. Now, for each of these 210 submatrices, we generate another subset of 46 submatrices by deleting $0, 1, 2, \ldots, 45$ rows. Each resulting set of 46 submatrices is sorted from smallest to largest $DP_i$. A plot of the actual distance as given by (2.2) is shown in Fig. 2.3(a) as a solid curve. This plot is the
Table 2.1: Percentage of times the deviation parameter criterion correctly predicts the ordering of the effects of noise. Percentages are given for a variety of noise terms and occlusions.

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>r</th>
<th>σ</th>
<th>d%</th>
<th>ρ%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>10%</td>
<td>91.80%</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>20%</td>
<td>87.14%</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>6</td>
<td>2</td>
<td>20%</td>
<td>85.98%</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>4</td>
<td>2</td>
<td>10%</td>
<td>93.31%</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>4</td>
<td>2</td>
<td>40%</td>
<td>84.16%</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>4</td>
<td>5</td>
<td>40%</td>
<td>82.78%</td>
</tr>
</tbody>
</table>

average over all possible 210 sets of 46 submatrices. As seen in the figure, the DP criterion results in the monotonically increasing function we need. This result is further compared to the one obtained when the submatrices are sorted from less to more missing rows (dashed curve in the figure). Adding rows is equivalent to adding samples to the least-squares fitting, generally resulting in better estimates. This result is however not as accurate as that of the DP criterion. Therefore, it is expected to provide better recoveries in practice, a point we will demonstrate in Section 2.5.

The problem with the above result is that it uses the same submatrices to generate the results with different percentages of missing entries. In actuality, the $r$-column submatrix with more missing entries will be generally constructed with different columns than those used to build a submatrix with less missing entries. Therefore, our criterion should also work under this condition. To test this, a $50 \times 10$ random matrix of rank 4 with 10\% missing elements (also randomly selected) and 1\% random Gaussian noise is divided into all possible 210 submatrices $\hat{B}_i$. First, the submatrices $\hat{B}_i$ are sorted from those with the smallest to those with the largest number of missing elements. This means that $\hat{B}_1$ corresponds to the matrix with the least number of missing elements, while $\hat{B}_{210}$ is that with the most.
Figure 2.3: The effect of noise over each possible submatrix. (a) Actual distance between subspaces when the submatrices are sorted according to the number of deleted rows or $DP$ value. In (b), the submatrices are ordered from having less to more missing elements. The abscissa corresponds to the index of the submatrices sorted by the numbers of missing entries, and the ordinate is the corresponding $pb$ value. In (c) the submatrices are sorted by the actual distance between subspaces (dashed curve). The solid curve specifies the value of the deviation parameter derived in this chapter.
This index corresponds to the $x$-axis in Fig. 2.3(b). Then, we use Eq. (2.2) to calculate the actual distance between the ground-truth and each of the perturbed submatrices, which is shown in the $y$-axis in Fig. 2.3(b). We see that the selection of those matrices with less missing entries does not always help choose an appropriate set.

We now order the submatrices according to the actual distance between them and the ground-truth, dashed curve in Fig. 2.3(c). Here, we also plot the $DP_i$ values for each of the submatrices. As it can be seen in this figure, the $DP_i$ values follow the plot of the true distance very closely and thus result in a very convenient and efficient way to order the submatrices according to their sensitivity to noise. This result is contrasted against the previous one shown in Fig. 2.3(b) where the submatrices were sorted according to the missing elements. Clearly, the estimate provided by the DP criterion is much preferred. This is because the deviation parameter provides information about both, the sensitivity to noise and the amount of missing elements. If the number of the non-complete rows ($b$) increases, both \((\sqrt{m - b - 1} + \sqrt{r})\) and \(\min(\sigma(\hat{A}))\) will decrease.

We provide the proof for this result and show that although both the numerator and denominator decrease as the number of missing elements increases, the value of $DP$ increases with it. This means that we favor matrices with less missing entries in general.

We construct a reduced matrix $\hat{A}_1 \in \mathbb{R}^{p_1 \times r}$ from $p_1$ rows of $\hat{B} \in \mathbb{R}^{m \times r}$ \((b_1 = m - p_1)\), and another matrix $\hat{A}_2 \in \mathbb{R}^{p_2 \times r}$ from $\hat{A}_1$ \((b_2 = m - p_2)\). So we have $p_2 < p_1$ and $b_2 > b_1$. We compare the smallest singular values of two matrices,

\[
\min(\sigma(\hat{A}_1)) = \min_{\|x\|_2=1} \|\hat{A}_1 x\|_2 = \min_{\|x\|_2=1} \sqrt{\sum_i (a_i^T x)^2} \\
\geq \min_{\|x\|_2=1} \sqrt{\sum_j (a_j^T x)^2} = \min_{\|x\|_2=1} \|\hat{A}_2 x\|_2 = \min(\sigma(\hat{A}_2)),
\]

(2.12)
The number of rows with missing entries

\[ \sqrt{m - b - 1} + \sqrt{r} \min(\sigma(\hat{A})) \]

DP

Figure 2.4: (a) The average results over 100 trials. (b) A typical case for the relationship between DP and the number of missing rows. x-axis denotes the number of rows with missing entries.

where \( a_k^T \in \mathbb{R}^{1 \times r} \) is the \( k^{th} \) row of \( \hat{A}_1 \), \( i = 1, 2, \cdots, p_1 \), \( j = l_1, l_2, \cdots, l_{p_2} \), and \( \{l_1, l_2, \cdots, l_{p_2}\} \) is a subset with \( p_2 \) elements drawn from \( \{1, 2, \cdots, p_1\} \).

Fig. 2.4 shows two examples where the minimum singular value (i.e. the denominator of DP) and the numerator of (2.11) decrease. In the figure we see that although both the denominator and numerator decrease, the value of DP increases. On average, the more one deletes rows with missing entries, the larger (2.11) becomes. Thus, in general, our criterion favors matrices with less missing rows, Fig. 2.4(a). Nevertheless, in each instance, some matrices with more missing rows may have lower DP values and they will be chosen for use in the proposed algorithm, Fig. 2.4(b).

As a final note, it is important to understand the complexity of the algorithm defined in this section. Recall that for a \( m \times n \) matrix, there is a total of \( C_n^r \) r-column submatrices. If the percentage of missing elements in this matrix is \( p \), each r-column submatrix has \( m(1-p)^r \) full rows on average (that is, assuming the occlusions are uniformly distributed).
This means that SVD needs to be performed on the resulting \( m(1 - p)^r \times r \) submatrix. The computational complexity of SVD is \( O(m(1 - p)^r r^2 + r^3) \), which is polynomial (of degree 3) in each iteration. The computational complexity of our criterion is thus given by \( C''_n O(m(1 - p)^r r^2 + r^3) \). For large matrices this is a cost to be considered. In these cases, we can divide the large data matrix into more tractable submatrices. We will address this issue in the section to follow.

### 2.3.6 Low-rank matrix fitting with DP

Before the data selection criterion described above can be employed in an actual low-rank matrix fitting approach, we have to mention two points which are essential to its successful implementation. The first practical issue to attend to is given by the process we have selected to eliminate the missing entries from the original data matrix. In our approach, we eliminate all rows that have at least one missing element. The problem is that we need to guarantee that there is at least one submatrix in \( \hat{N} \) which contains the information of each row. In our algorithm, we first sort the submatrices according to their \( DP_i \) values – from smallest to largest. Then, as to how many submatrices to select, we will choose the minimum number of submatrices needed to include the information of every row.

The second problem is that of determining the appropriate number of columns for the final matrix \( \hat{N} \). Since the number of possible \( r \)-column submatrices is \( C^r_n \) and this is much larger than \( n \), it is not necessary to include all these submatrices to construct \( \hat{N} \). In [50], the width of the matrix \( \hat{N} \) is set to a fixed size, e.g. 10\( m \) or 100\( m \). Here, we go one step further and propose a method which is based on the performance of the algorithm in recovering the matrix defining the null-space. To illustrate this, let us look at a couple of examples.
Let a $m \times n$ matrix of rank $r$ have $d\%$ of its elements missing, and contain additive Gaussian noise at level $\sigma$. Also, as above, let the value of each entry be bounded by zero and 100. In our first case study, we generate a matrix with the parameters $m = 10$, $n = 16$, $r = 4$, $\sigma = 1$, and $d = 20$. This provides us with the ground-truth matrix $W$ and its noisy version $\hat{W}$. We can use our algorithm (as described above) to find the best, minimum number of submatrices needed to recover $W_r$. This allows us to analyze how good the recovery is when the minimum number of submatrices $\hat{B}_i$ is used and how much improvement one gets when we keep adding additional submatrices. This is illustrated in Fig. 2.5, where the $x$ axis specifies the number of submatrices used to compute $W_r$ and the $y$ axis indicates the Root Mean Square Error (RMSE). The dashed curve in the figure corresponds to the RMSE between $W$ and $W_r$ obtained by using all the matrix elements, even those that were occluded to the algorithm and had to be recovered by it. Hence, this first measure provides the holy grail of measures, because it will show how our estimate relates to it. The solid curve in the figure corresponds to the actual estimate, given by the RMSE between $\hat{W}$ and $W_r$ over the non-missing elements of the matrices. The global minima of these two curves are shown in the figure as the lines travelling from those minima to the $x$ and $y$ axes. These specify the optimal number of submatrices needed to achieve the minimum RMSE.

We see in Fig. 2.5(a) that the true RMSE (dashed curve) and the one we can calculate (solid curve) have a similar behavior. They both decrease at first and then start increasing (indicating the noisy submatrices have started to overcome the recovery). This is also the case in the other example provided in Fig. 2.5(b), where $m = 15$, $n = 20$, $r = 3$, $\sigma = 4$, and $d = 30$. Most importantly, we see that the global minima in these two curves are located at a very proximal $x$ value, i.e. a very similar number of submatrices is needed to
Figure 2.5: Two example curves of the RMSE with (a) \([m, n, r, \sigma, d\%] = [10, 16, 4, 1.00, 20\%]\) and (b) \([m, n, r, \sigma, d\%] = [15, 20, 3, 4.00, 30\%]\). The dashed curve is the true error \(i.e.\) the difference between \(\hat{W}\) and \(W_r\), while the solid curve corresponds to that we can calculate \(i.e.\) \(\hat{W}\) to \(W_r\). The dashed lines indicate the number of submatrices used in each case and their corresponding RMSE. We can see that the RMSEs obtained with the true measure and with our estimate are practically identical.

minimize our computed measure \(i.e.\) the difference between \(\hat{W}\) and \(W_r\)\) and that provided by the ground-truth. We have further observed this pattern in a large number of simulations we have carried out. Therefore, the number of submatrices needed to carry out the recovery of \(W_r\) is conveniently given by the global minima of our estimate (solid line).

A final point needs to be made about large data matrices too. When \(\hat{W}\) is very large, the number of possible \(r\)-column submatrices grows very fast. While this may be computationally demanding, in actuality, there is no need for creating all the possible submatrices. Here, we propose a three-step iterative DP method, which can improve the performance of the DP method in the case of a large measurement matrix. First, the large measurement matrix is divided into several overlapping submatrices \(e.g.\) in the “dinosaur” sequence to be used in Section 2.5, we employed three overlapping submatrices). Second, the original DP method is applied to each of these (overlapping) submatrices, and the rows with the
largest reconstruction error are iteratively removed. This process continues until the average reprojection error over all the visible data begins to increase. This process is similar to that in [67]. The difference is that in [67] a threshold representing the highest tolerable reprojection error has to be pre-determined by the user. In our approach, this is automatically determined by the algorithm. In the third and final step, the whole recovery will be obtained by combining all these sub-recoveries. To do this, we work as follow. The data matrix $\hat{W}$ is partitioned into two overlapping submatrices, $\hat{W}_1$ and $\hat{W}_2$. This process is done to ensure that the number of overlapping columns in these two submatrices is at least $r$. Hence, $\hat{W}_1 = [\hat{W}_{11}, \hat{W}_{12}]$ and $\hat{W}_2 = [\hat{W}_{21}, \hat{W}_{22}]$, where $\hat{W}_{12}$ and $\hat{W}_{21}$ come from the same part of the measurement matrix of rank $r$. The matrix representations of the recovered $r$-dimensional (row) spaces, $\hat{F}_1$ and $\hat{F}_2$, can be written as $\hat{F}_1 = [\hat{F}_{11}, \hat{F}_{12}]$ and $\hat{F}_2 = [\hat{F}_{21}, \hat{F}_{22}]$. Then, to find the $r \times r$ matrix $K$ which minimizes $\|\hat{F}_{12} - K\hat{F}_{21}\|_F$, we can follow a simple linear least-squares method where $K = \hat{F}_{12}\hat{F}_{21}^T(\hat{F}_{21}\hat{F}_{21}^T)^{-1}$. For the case with overlapping rows, a similar process can be followed.

Note that the division process described in this section works best when the occlusion follows a continuous pattern – meaning that the points occluded in a first set of images are visible in another set and vice-versa. This is the case, for example, when one moves the face left to right or up and down or when an object is placed on a turntable.

### 2.4 Application to SFM

The approach described thus far can be directly applied to the problem of affine SFM. In projective SFM, the measurement matrix needs to be scaled by a set of proper projective depths. The projective depth can be recovered either using the fundamental matrix in epipolar geometry [98] or using the iterative estimation approach [45, 66]. The iterative
estimation method has many advantages but requires a good low-rank matrix fitting solution to ensure the convergence of the projective depths. The DP approach described in this chapter provides such a solution.

2.4.1 Projective SFM with missing data

Assume that we have $q$ views of a scene, each with $n$ 3D points generated from different projective projections. We want to recover the projective structure of the scene as well as the camera motion (or projection) for all $q$ views. Denote the $3 \times 4$ projection matrix of view $i$ by $P_i$, $i = 1, 2, \ldots, q$, and the 3D point $j$ by $Q_j$; in homogeneous coordinates $Q_j = [x_j, y_j, z_j, 1]^T$, $j = 1, 2, \ldots, n$. Then, the projection equation for point $j$ in view $i$ can be written as $\lambda_{ij} q_{ij} = P_i Q_j$, where $q_{ij} = [x_{ij}, y_{ij}, 1]^T$ is the homogeneous coordinate of point $j$ in image $i$, and $\lambda_{ij}$ is the corresponding projective depth. Further, let $p_{ij} = [x_{ij}, y_{ij}]^T$ be the inhomogeneous 2D coordinate of point $j$ in image $i$.

If we write all the projective matrices and all the 3D point coordinates in a single matrix, we have $P = [P_1^T, P_2^T, \ldots, P_q^T]^T$ and $Q = [Q_1, Q_2, \ldots, Q_n]$. Then the sequence of all $q \times n$ tracked 2D points with the scaling factors $\lambda_{ij}$ can be represented as a scaled measurement matrix,

$$S = [\lambda_{ij} q_{ij}] = P \cdot Q.$$  

The size of the scaled measurement matrix $S$ is $m \times n$, where $m = 3q$. If there is no noise and all the 3D points are visible in all views, it is clear that the scaled matrix $S$ is of rank 4 [98].

In general, however, we only have the non-scaled measurement matrix $W$ or its homogeneous version $H$ obtained from the actual 2D tracked points,

$$W = [p_{ij}] \quad \text{and} \quad H = [q_{ij}]^T.$$
Had we known the correct projective depths $\lambda_{ij}$, the low-rank matrix fitting method could have recovered $P$ and $Q$ up to a $4 \times 4$ homography.

The other major problem is that the (non-scaled) measurement matrix usually comes with some associated noise and missing elements, resulting in

$$\hat{W} = [\hat{p}_{ij}] \quad \text{and} \quad \hat{H} = [\hat{q}_{ij}],$$

where for all $(i, j) \in \Gamma$, $\hat{p}_{ij} = [\hat{x}_{ij}, \hat{y}_{ij}]^T$ and $\hat{q}_{ij} = [\hat{x}_{ij}, \hat{y}_{ij}, 1]^T$ are the inhomogeneous and homogeneous coordinates with unknown additive noise. The goal is to recover the full, noise-free matrices $P$ and $Q$ from the incomplete and inaccurate measurement matrix.

### 2.4.2 DP-based projective SFM

The proposed DP-based approach will be used iteratively to recover the projective depths, which will allow us to fit a rank 4 matrix to our current result. To facilitate convergence, the minimization criteria used in these two steps should have a similar form. Mahamud and Hebert [66] introduce a projective depth update where they propose to minimize the angle between each column vector and the low dimensional linear space. This method is based on a complete measurement matrix. The convergence of this algorithm has been proven under the case without missing data. We now extend this method to work in the missing data case.

For all those points that are occluded, we set $\hat{q}_{ij} = [0, 0, 0]^T$, i.e. if $(i, j) \notin \Gamma$. In the $k^{th}$ iterative step, we have the current projective depth $\Lambda^{(k)}$ as

$$\Lambda^{(k)} = [\lambda_{ij}^{(k)}] \quad \text{and} \quad \lambda_j^{(k)} = [\lambda_{1j}^{(k)}, \lambda_{2j}^{(k)}, \ldots, \lambda_{mj}^{(k)}]^T,$$

where $j = 1, 2, \ldots, n$. The scaled measurement matrix $\hat{S}^{(k)}$ satisfies

$$\hat{S}^{(k)} = \hat{H} \odot \Lambda^{(k)},$$
where $\odot$ indicates the Hadamard product ($[a_{ij}] \odot [b_{ij}] = [a_{ij} \cdot b_{ij}]$). The proposed method is used on the current scaled measurement matrix $\tilde{S}^{(k)}$ to find its best rank-4 fitting and factorize it into a product,

$$\tilde{S}^{(k)} \rightarrow \hat{P}^{(k)} \cdot \hat{Q}^{(k)},$$

where $\hat{P}^{(k)}$ is a matrix composed by 4 orthonormal vectors.

The second part of this step is to update the current projective depths. First, we need to fill in the missing data based on the current projective depths, and then we have the fill-in version of $\tilde{H}$, denoted as $\tilde{H}^{(k)}$. We want to find $\Lambda^{(k+1)}$, such that the range space of $\tilde{S}^{(k+1)} = \tilde{H}^{(k)} \odot \Lambda^{(k+1)}$ is closest to that of $\hat{P}^{(k)}$, which is given by the sine of the largest principal angle. Here we use $\hat{P}^{(k)}$ to update each column of $\Lambda^{(k)}$ and $\hat{Q}^{(k)}$ to update the rows of $\Lambda^{(k)}$. Each column vector in $\tilde{S}^{(k+1)}$ should be as close as possible to the space $\mathcal{R}(\hat{P}^{(k)})$. The $j^{th}$ column of $\tilde{S}^{(k+1)}$ is given by

$$s_j \equiv [\lambda_{1j} \cdot \hat{q}_{1j}^T, \lambda_{2j} \cdot \hat{q}_{2j}^T, \cdots, \lambda_{mj} \cdot \hat{q}_{mj}^T]^T,$$

where $[\lambda_{1j}, \lambda_{2j}, \cdots, \lambda_{mj}]^T = \lambda_j$ is a column vector with some projective depths. Now, let $\theta_j$ be the angle between $s_j$ and $\mathcal{R}(\hat{P}^{(k)})$. We then have

$$\lambda_j^{(k+1)} = \arg \min_{\lambda_j} \theta_j = \arg \max_{\lambda_j} \cos^2 \theta_j = \arg \max_{\lambda_j} \frac{\| \hat{P}^{(k)} \hat{P}^{(k)^T} s_j \|^2}{\| s_j \|^2}$$

$$= \arg \max_{\lambda_j} \frac{s_j^T \hat{P}^{(k)} \hat{P}^{(k)^T} s_j}{s_j^T s_j}.$$

This result can be rewritten as

$$\lambda_j^{(k+1)} = \arg \max_{\lambda_j} \frac{\lambda_j^T C_j C_j^T \lambda_j}{\lambda_j^T T_j \lambda_j}, \quad (2.13)$$

where the $i^{th}$ row of the $m \times 4$ matrix $C_j$ is given by $\hat{q}_{ij}^T \hat{P}_i$ and $\hat{P}_i$ is a $3 \times 4$ matrix constructed with the $i^{th}$ triplet of rows of $\hat{P}^{(k)}$, and $T_j$ is a diagonal matrix with the $i^{th}$ diagonal entry equal to $\hat{q}_{ij}^T \hat{q}_{ij}$. Eq. (2.13) is in fact a generalized eigenvalue-decomposition.
problem, where the correct projective depth $\lambda_j^{(k+1)}$ corresponds to the eigenvector associated with the largest eigenvalue. This result directly provides a solution for the scale matrix $\Lambda^{(k+1)}$ at each iteration. While iterative methods such as [66] have recently been found to lead to trivial solutions in some cases [80], the DP-based algorithm just presented is shown to converge to the correct solution in a large number of experiments detailed in the section to follow.

2.5 Experimental results

We now provide extensive experimental validation for the proposed approach. A statistical analysis is first presented using synthetic data. We then conclude with the application of the proposed approach on four real datasets.

2.5.1 Fitting low-rank matrices

We begin by testing the denoising ability of the proposed method. For this, we generate a $30 \times 20$ matrix $W$ of rank 4 with the absolute values of its entries set to no more than 100, and then add Gaussian noise with variance $\sigma$ and randomly occlude $d\%$ of the matrix entries, $\hat{W}$. We recover the low-rank matrix $W_r$ and then provide several measures of performance: $i)$ the difference over the non-occluded entries between $W_r$ and $\hat{W}$ (which we denote as $\text{diff}_1$), $ii)$ the difference over the visible data between $W_r$ and $W$ ($\text{diff}_2$), and $iii)$ the difference between $W_r$ and $W$ over all entries ($\text{diff}_3$). All these measures are given in RMSE and MAE (Maximum Absolute Error). The averages over a total of 30 trials are listed in Table 2.2 for each of the specified values of noise and missing elements. In this table we have added a result with a very large noise term ($\sigma = 5$) and extreme occlusion (40%), which results in a large $\text{diff}_3$. In this case, we see that the deletion of the rows with missing entries is problematic because it may eliminate other useful (visible) information.
This was not a problem when the noise and occlusion were smaller, because these columns were included in other submatrices. However, in this extreme conditions, it is common to have missing elements in many rows and too much noise in the remaining ones. These are thus the limits of the algorithm.

As we can see from this table, the recovered low-rank matrix $W_r$ is closer to the noise-free version, $W$, than to its noisy version $\hat{W}$. This is indeed a most desirable property, since it shows the algorithm is capable of denoising the data matrix. Furthermore, we see that the missing elements recovered by our algorithm do not include much additional error; demonstrating that the proposed approach does a good job in recovering the missing information too. (This point will also be shown to hold true for real data.) By plotting these results as a function of the noise parameter and the RMSE, Fig. 2.6, we see that the error increases linearly with the amount of noise that is added to the data matrix. This is also a very desirable property.

The comparison to other data selection criteria is provided in Table 2.3. In this table, we compare the RMSE, as given by $\text{diff}_1$, $\text{diff}_2$, $\text{diff}_3$, of the proposed algorithm and those obtained with a random selection of the submatrices [50] (indicated in the table as RAND).
Figure 2.6: The average recovered error with 30% missing data and different noise level, \( \sigma = 0.5, 1, 2, 4 \).

<table>
<thead>
<tr>
<th>( \sigma, d% )</th>
<th>RAND</th>
<th>MME</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \text{diff}_1 )</td>
<td>( \text{diff}_2 )</td>
<td>( \text{diff}_3 )</td>
</tr>
<tr>
<td>0.5, 10%</td>
<td>0.5032</td>
<td>0.3623</td>
<td>0.3770</td>
</tr>
<tr>
<td>1.0, 20%</td>
<td>1.1491</td>
<td>0.9312</td>
<td>1.1117</td>
</tr>
<tr>
<td>2.0, 20%</td>
<td>2.2586</td>
<td>1.8609</td>
<td>2.9495</td>
</tr>
<tr>
<td>2.0, 40%</td>
<td>3.1655</td>
<td>2.9333</td>
<td>34.068</td>
</tr>
<tr>
<td>4.0, 40%</td>
<td>5.0738</td>
<td>4.4706</td>
<td>33.697</td>
</tr>
<tr>
<td>5.0, 50%</td>
<td>5.9846</td>
<td>5.3507</td>
<td>76.851</td>
</tr>
</tbody>
</table>

Table 2.3: Comparison between RAND, MME and DP.

and the selection of the columns with less missing elements (MME, Minimal Missing Elements). In Table 2.3, we tabulate the RMSE results of random selection for RAND, MME and DP, averaging over 30 trials. We generate \( 30 \times 20 \) rank-3 matrices with additive Gaussian noise (\( \sigma \)) and missing data (\( d\% \)). As expected, the larger the noise and the amount of missing elements, the more sense it makes to use the criterion presented in this chapter.
2.5.2 Affine SFM

In our first experimental result with real data, we employ a publicly available sequence\(^1\) of 8 frames where a box with a calibration grid drawn onto it is shown, Fig. 2.7(a). This dataset comes with a total of 40 points tracked over each of the 8 frames with no occlusions.

In Fig. 2.8, we compare the results of the DP affine SFM algorithm and that of Jacobs’ [50] using the \( \text{diff}^2 \) measure given above. To generate these results we randomly occluded a percentage of the image points. This is specified by the index in the \( x \) axis, while the \( y \) axis represents the RMSE in (a) and the MAE in (b). In these results, Jacobs algorithm has been labelled \( \text{Jacobs}_{\text{trans}} \) because it includes a row of all ones. This is based on the observation that when the translation is included in the formulation of the affine model, a row with all entries equal to one should always be present in the solution space [50]. Hence, in general, using this approach results in better estimates. We have also extended our method to include this step. This extension is labelled \( \text{DP}_{\text{trans}} \) in the figure.

Fig. 2.9 illustrates how the RMSE increases as the amount of occlusion and noise increase. In (a) and (b) we show the RMSE as a function of both, noise and occlusion, for each of the two algorithms. From this result, we see that the sway noise performs over the DP-based approach is minimal. Most importantly, this effect is constant regardless of the occlusion term. The two algorithms are further compared in Fig. 2.9(c) for the particular case of 40% occlusion. Since the algorithm precision is consistently equated with the additive noise term, this could be further used as an initialization of a linear iterative optimization algorithm, such as, the bilinear method defined in [95] or the iteration-refining step given in [10].

\(^1\)http://www.cs.umd.edu/˜djacobs/missing-data.tar
Figure 2.7: Shown here is (a) a frame of the box sequence, and (b) a frame of the dinosaur sequence.

Figure 2.8: Plotted here are the reprojection errors obtained with Jacobs algorithm and the DP-based affine SFM. In (a) we show the RMSE over a total of 30 runs for each of the occluding percentages. In (b) we plot the MAE of each algorithm for each of the occlusions.
Figure 2.9: Shown here are the reprojection errors for the proposed approach and Jacobs algorithm as functions of the noise term and the amount of occlusion. RMSE on (a) Jacobs algorithm [50] and (b) DP affine SFM. In (c) we show a slice of the plots in (a) and (b) at 40% occlusion but with varying noise. This last plot allows for a one to one comparison.
The comparison is given between the following approaches: DP, RAND (i.e., Random Selection) [50], MME (Minimal Missing Elements) [10], the variant of RANSAC described in the text, CF (the Closed-Form solution of [7]), RPCA (Robust PCA) [14].

Table 2.4: RMSE results as given by $\text{diff}_3$ and average computation time.

<table>
<thead>
<tr>
<th>$\sigma$, d%</th>
<th>DP</th>
<th>RAND</th>
<th>MME</th>
<th>RANSAC</th>
<th>CF</th>
<th>RPCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5, 30%</td>
<td>0.9738</td>
<td>2.8692</td>
<td>2.1526</td>
<td>2.1666</td>
<td>3.3184</td>
<td>35.667</td>
</tr>
<tr>
<td>1.0, 30%</td>
<td>2.3501</td>
<td>5.6598</td>
<td>3.9688</td>
<td>6.6115</td>
<td>2.3672</td>
<td>34.098</td>
</tr>
<tr>
<td>2.0, 30%</td>
<td>3.6381</td>
<td>16.963</td>
<td>8.6395</td>
<td>9.8347</td>
<td>3.8571</td>
<td>35.816</td>
</tr>
<tr>
<td>4.0, 30%</td>
<td>6.4969</td>
<td>36.960</td>
<td>13.369</td>
<td>11.686</td>
<td>15.040</td>
<td>37.602</td>
</tr>
<tr>
<td>0.5, 40%</td>
<td>1.9068</td>
<td>10.256</td>
<td>3.2569</td>
<td>3.6903</td>
<td>14.561</td>
<td>39.539</td>
</tr>
<tr>
<td>1.0, 40%</td>
<td>3.1088</td>
<td>11.261</td>
<td>7.5623</td>
<td>8.4500</td>
<td>18.077</td>
<td>46.470</td>
</tr>
<tr>
<td>2.0, 40%</td>
<td>6.3400</td>
<td>23.653</td>
<td>15.320</td>
<td>13.009</td>
<td>22.906</td>
<td>45.234</td>
</tr>
<tr>
<td>4.0, 40%</td>
<td>8.5015</td>
<td>36.523</td>
<td>18.965</td>
<td>17.528</td>
<td>46.200</td>
<td>45.469</td>
</tr>
<tr>
<td>time (s)</td>
<td>100</td>
<td>0.1</td>
<td>100</td>
<td>100</td>
<td>30</td>
<td>0.3</td>
</tr>
</tbody>
</table>

These results can now be contrasted to other global methods such as RANSAC [22]. In RANSAC one could randomly choose $r$ columns, eliminate the rows with missing entries, and then obtain the subspace directly (without the need to compute the null space). We can then calculate how each of the unused columns fits to this subspace result and sort the $r$-column submatrices according to the fitting error. This approach results in higher RMSE than DP. Comparisons between the proposed approach and this variant of RANSAC are in Table 2.4. This table also includes comparative results with the robust approach presented in [14] and the closed-form solution of [7].

The review paper of Buchanan and Fitzgibbon [9] lists several methods providing a global criterion (e.g., minimization of the reprojection error over all visible data) and solve the problem iteratively as an EM-like approach [7, 33, 45]. For example, Brandt [7] proposes an EM-based method to solve the missing-data problem under affine projection. At each iteration of the algorithm (and assuming that the camera motion matrices are given), a closed-form maximum likelihood solution is derived to minimize the reprojection error. Global criteria such as this usually over-fit to the visible data. Buchanan and Fitzgibbon
[9] introduce an extra penalty term to alleviate this effect, such as orthogonality. Finally, Robust Principle Component Analysis (RPCA) (as well as robust SVD) is proposed by De la Torre and Black [14]. These approaches are based on robust statistics and can deal with noise and missing data – the approach will treat them as outliers. This should work reasonably well for low noise and occlusion terms, but will fail for large ones.

We present a comparison of our algorithm to the methods described in [7, 14] and to the variant of RANSAC defined above. All comparisons are obtained using the images of the box sequence. The results are in Table 2.4. Hence, the number of submatrices and the threshold used to discriminate between inliers and outliers are not the classical ones used in RANSAC, but are intrinsically given by the mechanism described above.

The major problem with global approaches is that they are based on the fitting to the non-occluded entries. This may lead to over-fitting. The main reason why our approach outperforms global methods is that it cares about the difference between the approximation matrix ($W_r$) and the ground-truth ($W$), i.e. $\text{diff}_3$. Over-fitting should not generally happen in our algorithm since our DP criterion is based on the upper-bound of the difference to the ground-truth. In Fig.2.10, an example of over-fitting is shown. In this example $\text{diff}_3$ becomes diverging while $\text{diff}_1$ is still converging, i.e. global methods will increase the value of $\text{diff}_3$ for smaller values of $\text{diff}_1$.

The other real dataset used in this section is the Dinosaur sequence [23]. This sequence has 36 frames and 4,983 tracked feature points which become occluded for the duration of several frames. One of these frames is shown in Fig. 2.7(b). What makes this sequence of interest is the large amount of missing (occluded) elements. Overall, the matrix has 90.84% of its entries missing (occluded). In fact, 2,300 of the feature points appear in only two frames. To facilitate convergence, at the end of each iteration of our algorithm, we will
Figure 2.10: One example of over-fitting for global methods.

employ the optimization of [95]. To provide a direct comparison with the results given in [10], we provide the results of the DP-based affine SFM with: all the data points, a subset of 2,683, and a yet smaller set of 336. These results are shown in Fig. 2.11(a-c). The 3D reconstruction of the Dinosaur’s shape when using all the data points is shown in Fig. 2.11(d). In [10], the authors provide a reprojection error by first computing the reconstruction on the smallest set and then extending this result to the two other (larger) sets. Their average reprojection errors and their maximum errors are (represented here as average/maximum): 1.8438/72.4467 for the set with 4,983 (i.e. all the data), and 2.4017/72.4467 when using the subset of 2,683 feature points. By repeating this procedure with the approach presented in this chapter, we obtain the following lower errors: 1.6419/39.9988, and 1.9340/39.9988, respectively.

2.5.3 Projective SFM

To test the DP-based projective SFM algorithm, we will use a synthetic dataset to provide quantitative results and two sets of real data to show actual applications of the method.
Figure 2.11: The recovered tracks of the Dinosaur sequence: (a) on 4,983 points, (b) on 2,683 points and (c) on 336 points. (d) The 3D reconstruction of the dinosaur.
The synthetic dataset consists of fifty randomly selected 3D points in the range of a sphere of radius 100 centered at the origin. Eight views of the resulting structure are generated. The cameras are located at random positions outside that sphere within the range of 200 to 500. The relatively large size of the scene produces large perspective distortions that could not be correctly recovered by affine SFM algorithms. Next, we add Gaussian noise with different variances \(\sigma\) and randomly occlude \(d\%\) of the data points. The average distance between the recovered 2D coordinates (obtained with the proposed algorithm) and the (2D) ground-truth of each of the images is given in Fig. 2.12. Similar to our results in affine SFM, here too the average recovered errors remain almost unchanged for different quantities of noise.

The RMSE between the recovered 3D structure obtained with our algorithm and the ground-truth is in Table 2.5. Since this is a 3D matching problem, the table illustrates the percentage of error added to our reconstruction. In this table, we also provide the RMSE obtained using the MME criterion in the projective SFM approach defined above. Since Jacobs’ random selection cannot guarantee a consistent recovery result at each iteration, it is not an appropriate candidate to be embedded in an iterative framework and was excluded from this comparison. In the table, the results are averaged over 30 trials.

We now apply the DP-based projective SFM algorithm to two real image sequences. The first example has 7 real images of a small wooden object shown in Fig. 2.13(a). Here,
we manually select 32 feature points and track them for the duration of the video sequence with 25% of the points missing. The other sequence we will use is the Model House sequence, which includes 10 images and 672 3D feature points. The percentage of missing points in this second sequence is 57.65%. One of the frames is shown in Fig. 2.14(a). The projective effect on this second sequence is larger than in the first one. This is because the distance between the camera and the turntable used to take the images is comparable to the size of the object.

The 3D reconstructions obtained with the proposed algorithm are in Fig. 2.13(b) and Fig. 2.14(b). In the first sequence (wooden object), the 2D locations of all feature points are recovered at pixel accuracy. The precision of the projective recovery is clear in this case from the figure too. In the second sequence, the RMSE and the MAE between the recovered and the given data are 0.6479/11.9946.
Figure 2.13: (a) One image in the wooden toy sequence. (b) The 3D structure of the object recovered by the DP-based projective SFM algorithm.

Figure 2.14: (a) One of the images in the model house sequence. (b) The recovered 3D structure of the scene.
2.6 Conclusions

Many problems in computer vision, pattern recognition and related areas reduce to finding that low-rank matrix that best fits an original data matrix with noisy and missing entries. A classical example is the SFM problem, where the 3D shape and motion of the object need to be recovered from a sequence of 2D images. In SFM, many of the points are generally imprecisely detected (noisy), while others are occluded in some of the frames.

In this chapter, we have shown that the missing and noise problems can be simultaneously addressed by first dividing the data matrix into an appropriate set of submatrices with no missing elements and, then, using a criterion that determines which of these submatrices are less affected by the noise term. Our key result was to provide a formal proof for the relation between the effect of noise in a submatrix and the similarity of its column vectors. That is, when the vectors given by the columns of one of our submatrices are separated by a large angle (say, close to $90^\circ$), additive noise has a limited influence. However, when the same noise is added to a submatrix with similar column vectors, the resulting subspace will be more different from its original noise-free version than the sway observed in submatrices with very distinct column vectors. The reason for this is grounded in the fact that dissimilar measurements do not get affected as much by a relatively small error term as do very similar ones. Note that in the latter case the dimensions of the subspace of the noisy matrix that correspond to the error term have a deviation from the original basis vectors similar to that seen in the original bases. In this case, it is unclear which small variations (between vectors) correspond to the noise term and which define the underlying subspace.

We have then shown how we can employ this formulation and a noise model to derive an upper-bound for the effects of noise in each of the submatrices. The derived criterion, referred to as DP (for Deviation Parameter), has been shown to be a very consistent and
reliable criterion for estimating low-rank matrices from synthetic and real data. In particular, we have shown how the criterion can be successfully applied to the problems of affine and projective SFM. In these cases, our criterion was able to work under large occlusions (about 40%) and noise terms (with variances around 5).
Notation

In this chapter, matrices are represented with bolded capital letters (e.g., $W$). Their $(i, j)^{th}$ entry is denoted as $w_{ij}$. Furthermore, $[\cdot]_{ij}$ denotes a matrix with entry $(\cdot)_{ij}$ in the $(i, j)^{th}$ position. This allows us to write $W$ as $[w_{ij}]$. Noisy matrices, with or without missing entries, are represented with a hat symbol $\hat{\cdot}$, e.g., $\hat{W}$. The spanning space of the column vectors in a matrix, also known as the range space, is described using the corresponding calligraphic capital letters, e.g., $\mathcal{W}$. The same letter is used to represent the space and the data matrix. $\mathcal{W} \text{ representing the range space of the matrix } W$. We can also write $\mathcal{W} = \mathcal{R}(W)$, where $\mathcal{R}$ means range space. The orthogonal space to $G$ is described using the capitalized letter and an orthogonal symbol, $G^\perp$. A detailed list of symbols and variables is in Table 2.6.
noise-free data matrix (in SFM, it contains the 2D coordinates of the feature points)
\( \mathbf{W} \)

range space of \( \mathbf{W} \)
\( \mathcal{W} \)

measurement matrix with noise and missing entries
\( \widetilde{\mathbf{W}} \)

rank-\( r \) recovered matrix from \( \widetilde{\mathbf{W}} \)
\( \mathbf{W}_r \)

noise matrix
\( \mathbf{E} \)

number of rows in the matrix \( \mathbf{W} \)
\( m \)

denotes \( \mathbf{W}_r \)

number of columns in the matrix \( \mathbf{W} \) (i.e. number of feature points)
\( n \)

number of images
\( q \)

t he rank of \( \mathbf{W} \)
\( r \)

percentage of missing entries in \( \widetilde{\mathbf{W}} \)
\( d\% \)

variance of the additive Gaussian noise
\( \sigma \)

visible mask of \( \widetilde{\mathbf{W}} \)
\( \Gamma \)

camera motion matrix in SFM
\( \mathbf{P} \)

shape matrix in SFM
\( \mathbf{Q} \)

number of \( r \)-column submatrices of \( \widetilde{\mathbf{W}} \) or \( \mathbf{W} \)
\( l \)

range space of the \( k \)th \( r \)-column submatrix of \( \mathbf{W} \)
\( \mathcal{B}_k \)

range space of the \( k \)th \( r \)-column submatrix of \( \widetilde{\mathbf{W}} \)
\( \mathcal{B}_k \)

intersection space of all \( \mathcal{B}_k \)
\( \mathcal{G} \)

orthogonal space to \( \mathcal{G} \)
\( \mathcal{G}^\perp \)

column vector
\( \mathbf{u} \)

largest principal angle between the spaces defined by the matrices \( \mathbf{X} \) and \( \mathbf{Y} \)
\( \theta(\mathbf{X}, \mathbf{Y}) \)

largest principal angle between the spaces \( \mathcal{X} \) and \( \mathcal{Y} \)
\( \theta(\mathcal{X}, \mathcal{Y}) \)

\( \mathbf{B}, \mathbf{B}_i \)

\( r \)-column submatrix from \( \widetilde{\mathbf{W}} \)
\( \hat{\mathbf{B}}, \hat{\mathbf{B}}_i \)

complete part of \( \mathbf{B} \)
\( \hat{\mathbf{A}}, \hat{\mathbf{A}}_i \)

submatrix of \( \mathbf{E} \) defining the noise term on \( \hat{\mathbf{A}} \)
\( \mathbf{E}_\mathbf{A} \)

matrix defining the null space of \( \hat{\mathbf{A}}_i \)
\( \mathbf{\tilde{M}}_i \)

matrix extending \( \mathbf{\tilde{M}}_i \) with zeros
\( \mathbf{\tilde{N}}_i \)

matrix containing all \( \mathbf{\tilde{N}}_i, \mathbf{\tilde{N}} = [\mathbf{\tilde{N}}_1 \mathbf{\tilde{N}}_2 \cdots \mathbf{\tilde{N}}_l] \)
\( \mathbf{\tilde{N}} \)

singular value spectrum of \( \hat{\mathbf{A}} \)
\( \sigma(\hat{\mathbf{A}}) \)

number of rows in \( \mathbf{\hat{B}} \) with at least one missing entry
\( b \)

deviation parameter of \( \mathbf{\hat{B}} \)
\( DP(\mathbf{\hat{B}}) \)

\( \mathbf{p}_{ij} \)

2D coordinates of the feature point \( j \) in image \( i \), \( \mathbf{p}_{ij} = [x_{ij}, y_{ij}]^T \)
\( \mathbf{q}_{ij} \)

2D homogeneous coordinates of the feature point \( j \) in image \( i \), \( \mathbf{q}_{ij} = [x_{ij}, y_{ij}, 1]^T \)
\( \lambda_{ij} \)

projective depth of \( \mathbf{q}_{ij} \)
\( \mathbf{S} \)

scaled measurement matrix

Table 2.6: Notation
The wooden object sequence

The wooden object sequence used in the paper consists of 7 images. These images are shown in Fig. 2.15(a-g). In these images, we have selected a total of 32 feature points, which are tracked over the video sequence. Several of the points get occluded when the object rotates, leading to 25% of missing entries in the data matrix, as in Fig. 2.16.

Fig. 2.17 shows the 2D recovered feature points (large squares) by the proposed DP-based algorithm. These have been overlaid on top of the originally marked positions (specified with small dots). The reconstructed 3D object structure is shown in Fig. 2.13. To further verify the validity of our reconstruction, we now add three virtual cubes to the 3D structure and project them onto the individual image. The results are in Fig. 2.18.
Figure 2.15: The seven images of the wooden object sequence.
Figure 2.16: Originally marked (dots) feature points for the wooden object sequence.
Figure 2.17: Marked (dots) and recovered (squares) feature points for the wooden object sequence.
Figure 2.18: Projection of a set of 3D cubes onto each of the images.
CHAPTER 3

FACE RECOGNITION WITH OCCLUSIONS IN BOTH TRAINING AND TESTING SETS

3.1 Introduction

The appearance-based approach to face recognition has resulted in the design of highly successfully computer algorithms in the last several years [125]. In this approach, the brightness values of the image pixels are reshaped as a vector and then classified using a classification algorithm.

One primary advantage of this appearance-based approach is that it is not necessary to create representations or models invariant to some of the (natural) object variations, since the object’s model is implicitly defined by the selection of the sample images. For example, when some key illumination variations are included, we can build representations invariant to some lighting choices [5]. Similarly, we can build representations modeling a large variety of facial expressions [69].

A major disadvantage of the appearance-based framework is that it cannot be directly used when some of the features (i.e. face pixels) are occluded. Partial occlusions in face images pose a great problem for most face recognition algorithms. In this case, the values for those dimensions are unknown. This problem has received considerable attention in
recent years. To date, the major approach used to resolve this problem is as follows. First, learn the appearance representation of the face as stated above using non-occluded faces. When attempting to recognize a partially occluded face, use only the visible dimensions (i.e. features) common to the model and the test images.

Since the work of Martinez [68], a variety of methods have been proposed for matching non-occluded training samples to partially occluded test images [20, 21, 82, 99, 100, 121]. The goal is to define a matching technique that omits the large matching errors due to occlusions while concentrating with those of the non-occluded parts. However, most methods do not address the problem of constructing a model (or classifier) from occluded images. As stated in Chapter 1, the problem of face recognition with occlusions is not strictly confined to recognizing partially occluded face from full faces only, but it also includes recognizing faces from a training dataset which has partial faces. The extreme case is that all the training samples are partially occluded. In these cases, the methods relying on the information gained from the full face training set are no longer feasible, such as [79, 68, 100, 99]. For example, the random sampling solution to detect the occlusion [21] may not work since the detection is based on the consistency of pixels between the occluded test image and the full training images. The sparsity of the face representation in [121] is no longer true when the training set has occlusions. New solutions are necessary to solve the general problem of face recognition with occlusions defined above.

In this chapter, two new approaches are proposed to solve the general problem of face recognition with occlusions. We first redefine the face recognition problem as a reconstruction one and only the information from the non-occluded facial regions is used. A Support Vector Machines solution is exploited in the second part in which a new criterion to measure the classification accuracy is defined. Extensive experimental results using a large
variety of comparative studies are demonstrated to show the superiority of the proposed algorithms.

3.2 Reconstructive face recognition with occlusions

We first pose the problem as a reconstruction one. In this approach, each test image is described as a linear combination of the training samples in each class. The class samples providing the best reconstruction determine the class label. Here, “best reconstruction” means that reconstruction providing the smallest matching error when using an appropriate metric to compare the reconstructed and test images. A key point in our formulation is to base this reconstruction solely on the visible data in the training and testing sets. This allows to have partial occlusions in both the training and testing samples, while previous methods only dealt with occlusions in the testing set.

In this approach, the training samples of a class are linearly combined to create a new image that is as close as possible to the test image. The hypothesis is that the most accurate reconstruction will be given when one uses the samples of the correct class. This is a grounded hypothesis, since the image reconstruction of a frontal face image will generally be most accurately obtained when combining face images of the person it represents rather than with images of other individuals.

Under this reconstructive view, the major problem is to define an appropriate mechanism to reconstruct the test image from the training samples. The linear reconstruction problem based on a complete dataset can been solved by the linear least squares method. Let the training set have \( n \) training data samples, \( x_1, x_2, \ldots, x_n \), and there is one testing data \( t \). Each of the training and testing data is defined on a feature set \( F = \{ f_1, f_2, \ldots, f_d \} \).
with the size \( |F| = d \), where \( |\cdot| \) is the set size. Both \( t \) and \( x_i \) are the \( d \)-dimensional vectorized images, \( d = ab \), and \( a \times b \) defines the image size. A sample is considered to be a complete one only if all features have certain values. Thus a complete data sample can be treated as a point in a \( d \)-dimensional space, \( x_j = (x_{j1}, x_{j2}, \cdots, x_{jd})^T \in \mathbb{R}^d \) and \( t = (t_1, t_2, \cdots, t_d)^T \in \mathbb{R}^d \). All the training data \( x_j \in \mathbb{R}^d \) can form a data matrix \( X = [x_1, x_2, \cdots, x_n] \) with the size \( d \times n \).

If the training and testing images had no occlusions, one of the simplest approaches would be to try to represent the test image \( t \) as a linear combination of the \( n \) training samples, \( \{x_1, \ldots, x_n\} \), \( t \approx \sum_{j=1}^{n} w_j x_j \), where \( w_j \in \mathbb{R} \) are the weights describing the contribution of each image.

The problem of estimating the weights \( w_j \) can be stated as,

\[
\mathbf{w} = \arg \min_{(w_1, \ldots, w_n)^T} \left\| t - \sum_{j=1}^{n} w_j x_j \right\|_r.
\] (3.1)

The selection of the metric above, which is given by the \( r \)-norm \( \| \cdot \|_r \), regulates how each feature (dimension) can be used to reconstruct the test image. The selection of the norm also defines the space \( \ell_r \).

The most commonly used norms in minimization problems such as that defined in (3.1) are the 2-norm, which provides the least-squares solution, and the 1-norm, which generates sparse representations. A recent result [121] shows that \( \ell_1 \) is preferred because it can handle sparse occlusions in the test image. In this approach, the sample images of all classes are used to estimate an occluded test image. Since \( \ell_1 \) emphasizes sparseness, only a very small number of sample images will be used to linearly reconstruct the test image. However, the samples which will generally best reconstruct the test instance, are those associated to the same class label (i.e. identity), facilitating the recognition process. Another concern of
this approach is that the $\ell_1$-minimization is computationally expensive \[121\]. So, here, we choose 2-norm to do the $\ell_2$-minimization in (3.1).

In the following sections, we rework the general reconstruction framework defined in (3.1) to efficiently address the problem that the training and testing samples have missing entries. By taking advantage of the natural partitions provided by the occluders, we derive a formulation that allows different weighting factors in distinct parts of the image based on the occlusion conditions. A multi-weight linear combination method is proposed to solve the problem of approximating the testing sample from an incomplete training set. This solution is included in the Partial Within Class Match (PWCM) method to do the face recognition without any filling step. We demonstrate that this framework outperforms the general definition given in (3.1), and that the derived solution can efficiently work with a $\ell_2$-minimization, resulting in very fast processing time.

3.2.1 Multi-weight reconstruction from partial data

For a partial testing data $t = (t_1, t_2, \cdots, t_d)^T \in \mathbb{R}^d$, we can use a set $\Gamma$ to include all indices corresponding to the visible features in $t$ and another set $\Pi$ to include all invisible feature indices, that is

$$\Gamma = \{i|\text{the } i^{th} \text{ feature is visible in } t, 1 \leq i \leq d\},$$

$$\Pi = \{i|\text{the } i^{th} \text{ feature is invisible in } t, 1 \leq i \leq d\}.$$

Thus, we can attach the superscript $(\cdot)^\Gamma$ (or $(\cdot)^\Pi$) to a matrix or a vector to denote the corresponding part by keeping only those rows with the indices in $\Gamma$ (or $\Pi$). Since we do not want to fill in the missing part of the test data, only the visible part of the test data $t$, $t^\Gamma$, can be used to do the classification. Consequently, only the corresponding entries in $x^\Gamma_j$ will appear in the following reconstruction. In the linear least squares method, $t^\Gamma$ can be
written as a linear combination of the data in the training set,

\[ t^\Gamma \approx \sum_{j=1}^{n} \omega_j x_j^\Gamma, \]  

(3.2)

and the weights \( \omega = (\omega_1, \ldots, \omega_n)^T \) can be estimated by the linear least squares solution to

\[ \omega = \arg \min_{(\omega_1, \ldots, \omega_n)^T} \| t^\Gamma - \sum_{j=1}^{n} \omega_j x_j^\Gamma \|_2. \]  

(3.3)

Thus, we can obtain the best reconstruction (in the linear least squares sense) from the training set of \( t^\Gamma \) as

\[ \hat{t}^\Gamma = \sum_{j=1}^{n} \omega_j x_j^\Gamma. \]  

(3.4)

However, this solution only fits the case where the test image is incomplete and the training set is complete. It cannot be extended to the case with training set having partial data. The major problem with the reconstruction approach defined above, is that one is only allowed to use a single weight per training image. That is, all the pixels in each sample image are weighted equally. In most instances, it would be useful to be able to weight distinct areas differently. For example, assume that we have two sample images per class in our database – one with close eyes and mouth and another with open eyes and mouth. We now wish to reconstruct a test image showing open eyes and a close mouth. This can be readily achieved if we allow different weights for the top and bottom parts of each image. The same requirement is also necessary in the partial data filling problem from a training set with partial data. Different features may have observed values in different samples, and different weights need to be assigned to different features based on their occlusion conditions. Hence, our goal is to define a reliable and fast mechanism to do this weight assignment.

Here, we derive a multi-weight linear least squares approach. The idea of the proposed step is demonstrated in Fig. 3.1. In this example, we have three training sample images,
Figure 3.1: An example of partial data filling with multiple weights. The visible parts are shown as shaded. The top and bottom occluded regions of the test data \( t \) can be reconstructed with different sets of weights from the training set.

\[ x_1, x_2, x_3 \] and one testing image \( t \). The shaded region in the figure represents the visible data in each sample. \( x_1 \) and \( x_3 \) have some occluded part, while \( x_2 \) does not. The test image is also incomplete, as shown in the figure. Each image is divided into several parts. For example, the pixels in \( t_1^1 \) are visible, and as shown in [54] they can be described as a linear combination of the corresponding visible elements of \( x_{11} \) and \( x_{21} \). This linear combination is defined by the weights \( \omega_{11}, \omega_{21} \) for \( x_{11} \) and \( x_{21} \), respectively. Similarly, another set of weights, \( \omega_{22}, \omega_{32} \), can be used to describe \( t_2^1 \) using the corresponding parts in \( x_{22} \) and \( x_{32} \).
To formally express the above idea, we need the occlusion masks for both the training set and the test data. We defined the occlusion mask $m_i$ for each training data $x_i$ which satisfies:

$$m_{ij} = \begin{cases} 
1 & \text{if } x_{ij} \text{ has an observed value,} \\
0 & \text{otherwise,}
\end{cases}$$

where $x_{ij}$ and $m_{ij}$ are the $j^{th}$ element, $x_{ij}$ and $m_{ij}$, of the $d$-dimensional vector $x_i$ and $m_i$, respectively, $i = 1, 2, \cdots, n, j = 1, 2, \cdots, d$. Now we use all $m_i$ to form the occlusion mask of the whole training set, $M = [m_1, \cdots, m_n]$. Let $M^j$ denote the $j^{th}$ row of this matrix, thus $M^j$ defines the sample images that can be used to reconstruct the $j^{th}$ image pixel of $t$, $t_j$. For test image $t$, we define $\tilde{m} \in \mathbb{R}^d$ as its occlusion mask.

Note that since each $M^j$ has $n$ values, there are $2^n$ possible patterns of useful values to reconstruct $t_j$ in total. That is, each feature in the feature set can be observed by either zero times from the sample images (when all are occluded), a subset of them, or all of them (when there are no occlusions). Let these patterns be denoted by $P_l$, $l = 1, \ldots, 2^n$, and $L_l$ is a set containing the indices of those training data with the features having observed values in the $l^{th}$ pattern.

Now consider those features in the feature set $F$ that can be reconstructed using the same pattern $P_l$, and denote the index set of the feature observed in $l^{th}$ pattern, $\Delta_l$. The set $\Delta_l$ can be further divided into two subsets, $\Gamma_l$ and $\Pi_l$, based on the occlusion condition of the feature in $\Delta_l$ in the test data $t$, i.e., $\Gamma_l$ contains the indices of the observable features in $t$ and $\Pi_l$ defines the indices of the occluded ones. Thus $\Gamma_l \cup \Pi_l = \Delta_l$, $\Gamma_l \cap \Pi_l = \emptyset$ and they can be used as the superscripts in the same way as in (3.2) ~ (3.4). Use these notations, a linear approximation for the pattern $P_l$ can be expressed as

$$t^{\Gamma_l} \approx \sum_{j \in L_l} \omega^{\Gamma_l}_j x^{\Gamma_l}_j,$$  \hspace{1cm} (3.5)
where the weights \( \{ \omega^l_j | j \in L_l \} \) are given by

\[
\arg \min_{\{ \omega^l_j | j \in L_l \}} \left\| \mathbf{t}^\Gamma_l - \sum_{j \in L_l} \omega^l_j \mathbf{x}^\Gamma_l \right\|_2. \tag{3.6}
\]

Thus the best reconstruction is given by

\[
\hat{\mathbf{t}}^\Gamma_l = \sum_{j \in L_l} \omega^l_j \mathbf{x}^\Gamma_l. \tag{3.7}
\]

The whole reconstruction process of \( \mathbf{t}^\Gamma \) is done after we go through all different patterns. The reconstructive version of \( \mathbf{t}^\Gamma \) denoted by \( \hat{\mathbf{t}}^\Gamma \) is given by the combination of all the rows in \( \hat{\mathbf{t}}^\Gamma_l, l = 1, \ldots, 2^n \).

### 3.2.2 Partial within class match

Now we are in the place to introduce the first new method to solve face recognition with occlusions based on the multi-weight reconstruction solution mentioned above. We refer to our method presented below as the Partial Within Class Match (PWCM) method. In this method, only the available data in the training and testing sets are used, and no filling steps are involved in either the training stage or the testing stage. In PWCM, for each testing sample \( \mathbf{t} \), one reconstruction of the visible part is generated from those samples in the training set belonging to the same class by the multi-weight linear reconstruction method described in the preceding section. If there are totally \( C \) different classes, \( C \) reconstruction of the test image \( \mathbf{t}, \hat{\mathbf{t}}^\Gamma(1), \hat{\mathbf{t}}^\Gamma(2), \cdots, \hat{\mathbf{t}}^\Gamma(C), \) are given, one per class. The best reconstruction of the testing data \( \mathbf{t} \) can chosen from \( \hat{\mathbf{t}}^\Gamma(1), \hat{\mathbf{t}}^\Gamma(2), \cdots, \hat{\mathbf{t}}^\Gamma(C) \). The next task is to determine which of these reconstructions is most similar to the original test image.

The simplest mechanism to test how accurate the reconstruction is, would be to look at the reconstruction error, given by

\[
\| \hat{\mathbf{t}}^\Gamma(i) - \mathbf{t}^\Gamma \|_r. \tag{3.8}
\]
For example, in $\ell_2$, this corresponds to the least squares fitting error. In $\ell_1$, it represents the error given by the sparse representation selected by the optimization mechanism.

However, the metric used to reconstruct the image is not always the most adequate for classification. For instance, if we employ $\ell_1$ to find a sparse set of training samples to describe the test image, it is generally adequate to compare the reconstructed and original images using the Euclidean distance (i.e. in $\ell_2$). In this case, while a $\ell_1$-minimization provides the advantages of a sparse representation, $\ell_2$ is adequate for comparing images (or, equivalently, vectors). This is in fact one of the most used approaches.

In contrast, in our approach derived above, Eqs. (3.5)-(3.6), it makes sense to do the minimization in $\ell_2$, because the goal is to use as much information from each sample of the same class as possible in an attempt to get more accurate reconstructions of the test image. However, once the reconstruction is obtained, it is generally prefer to compare the reconstruction and the original test image using a metric which emphasizes the overall similarity. Recall that the 2-norm is not a good choice for that, because it emphasizes those distances that are large while diminishing those that are small. This is a consequence of the quadratic term, which emphasizes large components and minimizes small ones. This is the same as saying that we would like to validate or invalidate a reconstruction based on the similarity of those areas that are most dissimilar, rather than those that are most similar. In fact, the 2-norm is well known to be sensitive to outliers (i.e. the large distances), which are typically found in the types of reconstructions obtained with a linear fit. The area of robust statistics usually employs the 1-norm to resolve these issues.

In our application too, we can use the 1-norm to be robust to outliers and to base our judgment on the overall similarity. The advantage of this norm is that it does not emphasize
the large or the small distances, since it is simply given by

$$\|a\|_1 = |a_1| + \cdots + |a_p|,$$

where $a \in \mathbb{R}^p$. Note that, in this norm, all components are treated equally—regardless of their size.

We can now go one step further and use a measure that deemphasizes large distances while emphasizing small ones. This would help put more emphasis on the similarity between the reconstructed and original test image, rather than on their dissimilarity as in $\ell_2$. To achieve this we can use the 0.5-quasi-norm, given by

$$\|a\|_{0.5} = (a_{0.5}^1 + \cdots + a_{0.5}^p)^2.$$

Recall that this is not a norm, because it does not satisfy the triangular inequality, which needs to be replaced by $\|a + b\|_r \leq h(\|a\|_r + \|b\|_r)$, for some $h > 1$. In our case, $r = 0.5$ and $h = 2$.

The important concept here, is that the .5-quasi-norm will deemphasize large distances (including the outliers) and emphasize the small ones (i.e. the areas where the reconstruction was possible). This effect is due to the fact that the exponential term in the .5-quasi-norm is smaller than 1. One could use an $r < .5$, but this would only emphasize points of agreement and would no longer consider the overall similarity of the two images.

These are thus the two measures that we use for classification, i.e. the 1-norm and the .5-quasi-norm. And, the class label, $c_t$, of our testing image $t$ is given by

$$c_t = \arg \min_{i=1,\ldots,C} \|\hat{t}^\Gamma(i) - t^\Gamma\|_r,$$

where $r$ is either 1 or 0.5.
3.3 Support vector machines in face recognition with occlusions

A classification algorithm that has successfully been used in the appearance-based face recognition is the well known Support Vector Machines (SVM) [113], which can be applied to the original appearance space or a subspace of it obtained after applying a feature extraction method [81, 40, 102].

However, the classical SVM algorithm cannot be applied directly on the problem of face recognition with occlusions because a complete training set is required in the training stage of SVM. In this chapter we derive a criterion for SVM that can be employed in the three cases defined in Fig. 1.5. Note that the classical criteria of SVM cannot be applied to any of the three cases, because SVM assumes all the features are visible. In the sections to follow, we derive a criterion that can work with missing components of the sample and testing feature vectors. We will refer to the resulting algorithm as Partial-data Support Vector Machines (PSVM) to distinguish it from the standard criteria used in SVM.

The goal of PSVM is, nonetheless, similar to that of the standard SVM – to look for a hyperplane that separate the samples of any two classes as much as possible. In contrast with traditional SVM, in PSVM the separating hyperplane will also be constrained by the incomplete data. In the proposed PSVM, we treat the set of all possible values for the missing entries of the incomplete training sample as an affine space in the feature space to design a criterion which minimizes the probability of overlap between this affine space and the separating hyperplane. To model this, we incorporate the angle between the affine space and the hyperplane in the formulation. The resulting objective function is shown to have a global optimal solution under mild conditions. Experimental results demonstrate that the proposed PSVM approach provides superior classification performances than those defined in the literature.
3.3.1 Classical SVM algorithm

In the classical SVM framework, a set of complete data samples is needed in the training stage to find the hyperplane which can maximize the geometric margin in the feature space. Since a complete data sample can be considered as a point in the feature space, the geometric margin is defined as the minimum distance between data points and the hyperplane.

We first briefly review the classical SVM algorithm. In the training stage of SVM, a hyperplane is obtained from a complete dataset with labels by maximizing the geometric margin. Let the training set have \( n \) samples \( \{x_1, \ldots , x_n\} \), with labels \( y_i = \pm 1, \ i = 1, \ldots , n \), each of them defined by a feature set \( F = \{f_1, f_2, \ldots , f_d\} \). In this setting, a complete data sample can be treated as a point in a \( d \)-dimensional space, \( x_i = (x_{i1}, \ldots , x_{id})^T \in \mathbb{R}^d \).

The best hyperplane, \( w^T x = b \), to separate two classes is achieved by maximizing the geometric margin,

\[
\max_{w, b} \frac{1}{\|w\|}, \quad s.t. \quad y_i(w^T x_i - b) \geq 1, \ i = 1, \ldots , n, \tag{3.10}
\]

where \( \| \cdot \| \) is the 2-norm of a vector. Eq. (3.10) is equivalent to minimizing the quadratic term \( \frac{1}{2}\|w\|^2 \) with the same constraints,

\[
\min_{w, b} \frac{1}{2}\|w\|^2, \quad s.t. \quad y_i(w^T x_i - b) \geq 1, \ i = 1, 2, \cdots , n, \tag{3.11}
\]

which has an efficient solution [113].

Typically, the original set will not be linearly separable. To resolve this problem, it is common to define a soft margin by including the slack variables \( \xi_i \geq 0 \) and a regularizing
parameter $C > 0$,

$$\min_{w, \xi, b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i,$$

$$s.t. \quad y_i(w^T x_i - b) \geq 1 - \xi_i, \quad i = 1, \ldots, n. \tag{3.12}$$

In (3.12), the slack variables $\xi$’s have positive values for those incorrectly classified data, and the regularizing constant $C$ controls the tradeoff between the classification accuracy on the training set and the generalization performance of the output model. This is also a quadratic programming problem which has an efficient solution as well.

However, when some of the features are missing, these distances can no longer be computed. One possible way to solve this problem is to attempt to fill-in the missing entries of each feature vector before using SVM. Unfortunately, the filling-in step leads us to a worst problem: how to know the correct (or appropriate) values of the missing entries? A classical approach is to use the average over the samples with no missing elements for those features. This solution has two problems of its own: 1st) there may be one or multiple features with missing entries for which no sample mean can be computed, and 2nd) the mean of two samples may not provide a realistic solution (e.g. the average eye region of one sample with open eyes and another with close eyes is a region of semi-transparent eyelids, which has no real bases).

If we consider the affine space $S_i$ defined by all possible fill-ins of the corresponding partial data $x_i$ as one single data unit, the ideal solution to the partial data classification is that it can classify the affine space correctly. That means the hyperplane should ideally be parallel to all the affine spaces defined by the incomplete data, which is generally impossible.

To illustrate this point, we show a simple example in Fig. 3.2. In this figure, two sets of points, $\{p_1, p_2\}$ and $\{q_1, q_2\}$, defined on the feature plane $\{f_1, f_2\}$ and corresponding to
classes 1 and 2, are generated. The additional sample vector $p_3$ has a known value for $f_1$ but a missing entry in $f_2$. Three possible filling-ins of $p_3$ are shown in the figure – denoted $p_3^1, p_3^2$ and $p_3^3$. For each of them, the classical SVM would give the hyperplanes denoted by $l_1, l_2$ and $l_3$, respectively. We can see that none of these three hyperplanes can give correct classifications for all $p_j^i$.

To resolve the problem illustrated above, we resort to a new solution which focuses on classifying partial data correctly with the help of probabilities. In particular, we show how to add a new term to (3.10).
3.3.2 The angle between the hyperplane and the affine space

The values of the missing elements of our \( d \)-dimensional feature vector define an affine space in \( \mathbb{R}^d \). We now show that the correct classification probability of a hyperplane on the affine space is determined by two factors: \( a) \) the relative position between them, and \( b) \) the classification result of the actual missing elements.

To get started, let us assume that there is only one missing element in \( x \) in class 1. Denote the affine space defined by this missing element as \( S \), and the hyperplane which separates the two classes by \( l \). This hyperplane can be readily obtained with the standard SVM criterion if we can fill in the missing entries in \( x \), \textit{e.g.} simply substituting the missing entry by that of the mean feature vector \( \bar{x} \). If the hyperplane \( l \) and the affine space \( S \) are not parallel to each other, the intersection between the two divides the affine space into two (non-overlapping) parts, \( S_1 \) and \( S_2 \). This partition is illustrated in Fig. 3.3(a). We see from this figure that the possible values of the missing entry that fall in \( S_1 \) will be correctly classified as class 1, whereas the values now in \( S_2 \) will be misclassified. Using this argument, we can compute the Correct Classification Probability (CCP) of \( l \) over the affine space \( S \) as

\[
CCP(l, S) = \int_{q \in S_1} p(q) dq,
\]

where \( p(q) \) is the probability density function (pdf) and \( q \in S \).

Under the above defined model, the goal is to minimize the probability of overlap between the most probable values of the samples in class 1, \textit{i.e.} we want to prevent \( l \) to cut over plausible values of the missing entries. To calculate this probability, we assume the sample data is Gaussian distributed, \( p(q) \in N(\bar{x}, \sigma) \) with \( \bar{x} \) the mean and \( \sigma \) the variance. This is shown in Fig. 3.3(a). The intersection between \( S \) and \( l \) is at \( q_0 \). Maximizing CCP is thus equivalent to maximizing the distance between the value given by \( \bar{x} \) and \( q_0 \), \( d(\bar{x}, q_0) \).
Figure 3.3: The Correct Classification Probability (CCP) of a hyperplane. (a) Assuming a Gaussian distribution of $S$, (b) the angle between $S$ and $l_i$ is proportional to the distance $d(\bar{x}, q_0)$.

Note that, if the classification of $\bar{x}$ is incorrect, a minimization problem of the distance $d(\bar{x}, q_0)$ is needed to maximize CCP.

For a fixed set of sample vectors, the angle between the subspaces $S$ and $l$, $\theta(S, l)$, decreases proportionally to the increase of $d(\bar{x}, q_0)$ (Fig. 3.3(b)). Hence, $\theta(S, l)$ is the term needed to account for the possible values of the missing elements of $x$. Note also that $\theta(S, l)$ is invariant to the position of $\bar{x}$ on $S$ and there is only one principal angle between two spaces $S$ and $l$ since $l$ has only one dimension [30].

Now we can add one angle-related term into the traditional SVM objective function ((3.10) or (3.11)) to penalize those hyperplanes having a large angle from the affine space under the condition of correct classification of $\bar{x}$. The new objective of PSVM is to maximize the geometrical margin between the visible dimensions and the classification accuracy on the occluded dimensions (which define the affine subspace) simultaneously.
3.3.3 The objective function of PSVM

We are now in a position to formulate the criterion which will properly model the aforementioned penalty term. This will take us to the definition of the PSVM algorithm. We start by presenting the solution for the linearly separable case.

To address the incomplete data problem efficiently, we use the occlusion mask \( m_i \in \mathbb{R}^d \) as defined in Section 3.2 for each sample vector \( x_i, i = 1, \ldots, n \). The elements of the occlusion mask \( m_i \) will be 0 wherever the corresponding feature in \( x_i \) is occluded and 1 otherwise. The affine space which is formed by all possible filling-ins of incomplete sample \( x_i \) is denoted \( S_i \), and the hyperplane separating the two classes by \( l : w^T x = b \), where \( w = (w_1, \ldots, w_d)^T \).

The angle between \( S_i \) and \( l \) is the same as the angle between the orthogonal space of \( S_i, S_i^\perp \), and the normal vector of \( l, w \). The projection of \( w \) on \( S_i^\perp \) is \( w_i^1 = w \odot m_i \), where \( \odot \) is the Hadamard product. The angle between \( S_i \) and \( l, \theta(S_i, l) \), is given by

\[
\cos \theta(S_i, l) = \cos \theta(S_i^\perp, w) = \frac{\|w_i^1\|}{\|w\|}. \tag{3.14}
\]

For each incomplete data, the corresponding principal angle can be computed, and a new term can now be formulated as a weighted summation, \( i.e., \sum_{i=1}^{n} K_i \|w_i^1\|/\|w\| \), where the weights \( K_i \geq 0 \) for each term defined in (3.14) are chosen to be positive when \( x_i \) is incomplete and zero otherwise. To obtain the highest possible CCP, this term is to be maximized. This can be readily achieved by adding it to SVM optimization problem as follows

\[
\max_{w, b} \quad \frac{1}{\|w\|} + K \sum_{i=1}^{n} K_i \frac{\|w_i^1\|}{\|w\|} \tag{3.15}
\]

\[ s.t. \quad y_i(w^T \tilde{x}_i - b) \geq 1, \quad i = 1, \ldots, n, \]
where $K > 0$ is the regularizing parameter to control the overall tradeoff between the
generalization performance of the hyperplane (defined by the maximal geometric mar-
gin, $1/\|w\|$) and the classification accuracy on the incomplete data. Note that we have
used $\{\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n\}$ to substitute the original and incomplete dataset $\{x_1, x_2, \cdots, x_n\}$
in (3.15).

The objective function in (3.15) is neither linear nor quadratic, which usually does
not yield efficient solutions. Nonetheless, we can transform (3.15) into a more tractable
criterion (with the quadratic form of $w$ in both denominator and numerator) as follows

$$
\max_{w, b} \quad 1 + K \sum_{i=1}^{n} K_i \|w_i\|^2 \\
\text{s.t.} \quad y_i (w^T \bar{x}_i - b) \geq 1, \quad i = 1, \ldots, n.
$$

(3.16)

### 3.3.4 Optimization for linearly separable case

We now show how to solve the above optimization problem in the linearly separable
case. Without loss of generality, let us rework the above derived SVM solution in $\mathbb{R}^d$, $d$ the
number of dimensions. In the above equation (3.16), the variable $w$ appears in different
ways, either the original complete form $w$, or the partial form $w_i^1 = w \odot m_i$. And thus the
manipulation of the variable becomes difficult. It will be more convenient if we consider all
the elements $w_1, w_2, \cdots, w_d$ in $w$ as the separate variables. This transformation is detailed
as follows.

We use $d$ new parameters $u_i, i = 1, 2, \cdots, d$, to replace the weights $K_i, i = 1, 2, \cdots, n,$
and the regularizing constant $K$. This is just for the convenience of the derivation later. The
new parameters satisfy the following equation,

$$
\sum_{i=1}^{d} u_i w_i^2 = K \sum_{i=1}^{n} K_i \|w_i^1\|^2,
$$

(3.17)
where each \( u_i \) is the coefficient of the quadratic term \( \| w_i \|^2 \) after combining the similar terms in the expansion of \( K \sum_{i=1}^{n} K_i \| w_i \|^2 \) with respect to \( d \) variables, \( w_1, w_2, \cdots, w_d \). Or, more specifically, \( u_i, i = 1, 2, \cdots, d, \) can be calculated in the following way.

Denote the occlusion mask matrix of the whole training set by \( M = [m_1, m_2, \cdots, m_n] \in \mathbb{R}^{d \times n} \). Now we use the weights \( K K_i \) to multiply each column of \( M \) and generate a new matrix \( \mathbf{\hat{M}} = [K K_1 m_1, K K_2 m_2, \cdots, K K_n m_n] \in \mathbb{R}^{d \times n} \). Finally, we can concatenate the summation of each row of \( \mathbf{\hat{M}} \) to form a column vector \( u \in \mathbb{R}^d \) with \( u_1, u_2, \cdots, u_d \) as its entries.

Now we have the objective function as

\[
\max_{w, b} \quad f(w) = \frac{1 + \sum_{i=1}^{d} u_i w_i^2}{\sum_{i=1}^{d} w_i^2} \tag{3.18}
\]

\[s.t. \quad y_i (w^T \bar{x}_i - b) \geq 1, \quad i = 1, \ldots, n, \]

where \( w = (w_1, w_2, \cdots, w_d)^T \).

Since \( b \) only appears in the linear constraint as an offset of the separation hyperplane, it will not affect the convexity of the defined region. Therefore, in the following analysis, we focus on \( w \), which still needs to be shown to yield convex regions to allow optimal solutions wrt the derived criterion.

To do this, note that the optimization problem in (3.18), with respect to \( w \), is defined on a polyhedral convex region in a \( d \)-dimensional space. We see that this region in the space of \( w \) does not cover the origin point \( w = 0 \). If the above statement were not true, then we would need to use \( 0 \) to replace \( w \) in the constraint to get \( y_i (-b) \geq 1, i = 1, \ldots, n \). Since \( y_i \) is either \( \pm 1 \), and noting that each of these two values must be assigned at least once to \( y_i \), we can choose \( y_j = +1 \) and \( y_k = -1 \) \((j, k \in \{1, \ldots, n\} \text{ and } j \neq k)\) to get \( b \geq 1 \) and \(-b \geq 1\). This results in a null set.
The target function is not convex on \( w \). Nonetheless, it has some good properties we can exploit to facilitate the optimization. Consider two points \( w_1 \) and \( w_2 \) (e.g. \( w_2 = r w_1, r > 1 \)), then the corresponding function values satisfy

\[
\begin{align*}
f(w_1) &= \frac{1}{\sum_{i=1}^{d} w_{1i}^2} + \frac{\sum_{i=1}^{d} u_i w_{1i}^2}{\sum_{i=1}^{d} w_{1i}^2} \\
&\leq \frac{1}{\sum_{i=1}^{d} (rw_{1i})^2} + \frac{\sum_{i=1}^{d} u_i (rw_{1i})^2}{\sum_{i=1}^{d} (rw_{1i})^2} \\
&= \frac{1}{\sum_{i=1}^{d} w_{2i}^2} + \frac{\sum_{i=1}^{d} u_i w_{2i}^2}{\sum_{i=1}^{d} w_{2i}^2} = f(w_2).
\end{align*}
\]

The above result implies that the objective function is monotonically increasing on a line passing through the origin.

Since (3.18) is defined on a convex region not covering the original point (\( w = 0 \)) and has the monotonically increasing property proved above, the optimal solution of (3.18) must be on the boundary of that region. Therefore, if we use the solution to the classical SVM as the initial point (on the completed training set), we can apply a gradient-descent method to solve for (3.18). The question is whether this procedure can provide the \textit{global} optimal solution wrt our criterion. We can now show that under mild conditions, this global optimal is guaranteed.

Here, instead of maximizing \( (1 + \sum_{i=1}^{d} u_i w_i^2) / (\sum_{i=1}^{d} w_i^2) \), we take the maximization term out of the objective function and add a new constraint, \( (1 + \sum_{i=1}^{d} u_i w_i^2) / (\sum_{i=1}^{d} w_i^2) \geq \gamma \), to get

\[
\begin{align*}
\max_{w, b} & \quad \gamma \\
\text{s.t.} & \quad 1 + \frac{\sum_{i=1}^{d} u_i w_i^2}{\sum_{i=1}^{d} w_i^2} \geq \gamma \quad \text{and} \quad y_i(w^T \bar{x}_i - b) \geq 1, i = 1, 2, \cdots, n.
\end{align*}
\]
In (3.20) we maximize the lower bound of the objective function instead of maximizing the function itself. The resulting maximization problem is equivalent to (3.18) since the lower bound $\gamma$ can always be maximized further until it is tightly reached.

The first constraint above can be rewritten in an equivalent way as

$$1 + \sum_{i=1}^{d} u_i w_i^2 \geq \gamma \sum_{i=1}^{d} w_i^2, \quad \text{or} \quad \sum_{i=1}^{d} (\gamma - u_i) w_i^2 \leq 1.$$  \hspace{1cm} (3.21)

Now for any fixed value of $\gamma \geq \max\{u_1, u_2, \ldots, u_d\}$, (3.21) defines a convex region in a $d$-dimensional space (with respect to $w$), so it defines a convex region in a $(d+1)$-dimensional space (with respect to $w$ and $b$) as well. Therefore, the target function and the constraints are convex, which ensures a global optimization solution. This means that a global solution exists under the condition

$$\gamma_{max} \geq \gamma_0 = \max\{u_1, u_2, \ldots, u_d\},$$  \hspace{1cm} (3.22)

where $\gamma_{max}$ is the solution to (3.20). This is indeed a very mild condition. In fact, it holds in all our experiential results to be presented later.

We now can write our optimization problem as a convex optimization problem by adding a new constraint,

$$\begin{align*}
\max_{w, b} \quad & \gamma \\
\text{s.t.} \quad & \sum_{i=1}^{d} (\gamma - u_i) w_i^2 \leq 1, \gamma \geq \gamma_0, \text{ and } y_i (w^T \bar{x}_i - b) \geq 1, i = 1, 2, \ldots, n.
\end{align*}$$  \hspace{1cm} (3.23)

Given $\gamma = \gamma_0$, (3.23) becomes a feasibility problem following the general structure of a problem called Second Order Cone Program (SOCP) [63], which is convex and has an efficient solution [32]. If the feasibility of the above problem is verified by the SOCP solution, we can look for the maximal $\gamma$ further starting from $\gamma = \gamma_0$. 

97
To get the maximal $\gamma$ for (3.23), we need to find $\gamma_{\text{max}}$ which is the maximal value to keep the feasibility of (3.23). With $\gamma_{\text{max}}$ and the corresponding solution $w_{\text{max}}, b_{\text{max}}$, it can be readily shown that any $\gamma \in (\gamma_0, \gamma_{\text{max}})$ will provide a solution to (3.23) since

$$\sum_{i=1}^{d} (\gamma - u_i)w_{\text{max}}^2 \leq \sum_{i=1}^{d} (\gamma_{\text{max}} - u_i)w_{\text{max}}^2 \leq 1.$$ 

Hence, the bisection search over $\gamma \in (\gamma_0, +\infty) \subset \mathbb{R}^+$ is an efficient and direct way to determine the value of $\gamma_{\text{max}}$.

If the optimal value $\gamma_{\text{max}}$ is far from the initial guess $\gamma_0$, the number of the feasibility problems we need to run could be very large and thus it slows down the search process. The reason is that there is not any optimization on $\gamma$ at all. In the following derivations, we will slightly modify (3.23) to achieve a fast search algorithm based on the new observation of the property of the optimization problem.

We investigate the connection between the two constraints in (3.23), i) $\sum_{i=1}^{d} (\gamma - u_i)w_i^2 \leq 1, \gamma > \gamma_0$ and ii) $y_i(w^T \bar{x}_i - b) \geq 1, i = 1, 2, \ldots, n$, in a geometric way.

First, constraint ii) defines a convex region in a $(d+1)$-d space with respect to $w$ and $b$. The projection of this region onto the $d$-d subspace defined by $w$ is a polyhedral convex one, and it will not change as $\gamma$ changes. We have proven that this projected polyhedral convex region does not cover the origin point of the $d$-d subspace.

On the other hand, constraint i) defines a high dimensional ellipsoid in the $d$-d space with respect to the variables in $w$ centered at the origin. When $\gamma$ increases starting from $\gamma_0$, the ellipsoid will shrink in all dimensions. Thus, the optimization problem of (3.23) comes with a geometric interpretation of finding the smallest ellipsoid which has intersections with the polyhedral convex region defined above. The optimal solution to this geometric
intersection problem is reached when the ellipsoid is tangential to the polyhedral convex region, i.e. $\gamma = \gamma_{\max}$.

As we have discussed before, the bisection search is applied to find $\gamma_{\max}$ by updating its upper and lower bounds ($\gamma_{ub}, \gamma_{lb}$) after each run. But those runs of SOCP optimization with $\gamma > \gamma_{\max}$ to lower the upper bound of $\gamma_{\max}$ are not necessary. We can approach $\gamma_{\max}$ from only one side of $\gamma_{\max}$ if we modify (3.23) to the following optimization problem,

$$\min_{w,b} g_0(w, b) = \sum_{i=1}^{d} (\gamma_{lb} - u_i)w_i^2$$

s.t. $\sum_{i=1}^{d} (\gamma_{lb} - u_i)w_i^2 \leq 1$, $\gamma_{lb} = \gamma_0$, and $y_i(w^T\bar{x}_i - b) \geq 1, i = 1, 2, \cdots, n,$

where the current lower bound of $\gamma$ is set to $\gamma_{lb} = \gamma_0$.

The modified optimization problem still satisfies the SOCP rules and can help us approach the maximum of $\gamma$ by iteratively updating the lower bound of $\gamma_{\max}$. The geometric interpretation of the above equation (3.24) and the corresponding iterations are illustrated in Fig. 3.4.

In the $k^{th}$ iteration, we set $\gamma_{lb} = \gamma_k$ in (3.24). Two constraints $i)$ and $ii)$ define two convex regions: the polyhedral convex region $G$ and the ellipsoid $E_k$. $E_k$ changes in each iteration while $G$ keeps the same. The intersection between $G$ and $E_k$ is $I_k$ (the shaded region). Any point, that is within $I_k$ but not on the surface of $E_k$, gives a value smaller than 1 on the objective function $g_k(w, b)$ as

$$\min_{w,b} g_k(w, b) = \sum_{i=1}^{d} (\gamma_{lb} - u_i)w_i^2$$

s.t. $\sum_{i=1}^{d} (\gamma_{lb} - u_i)w_i^2 \leq 1$, $\gamma_{lb} = \gamma_k$, and $y_i(w^T\bar{x}_i - b) \geq 1, i = 1, 2, \cdots, n,$

If $w_k = (w_{k1}, w_{k2}, \cdots, w_{kd})^T$ is the optimal solution to (3.25), we can use it to update the lower bound $\gamma_{lb}$ by calculating $\gamma_{k+1} = (1 + \sum_{i=1}^{d} u_iw_{ki}^2)/(\sum_{i=1}^{d} w_{ki}^2)$.
Figure 3.4: Iterative search to the optimization problem (3.24). In the $k^{th}$ iteration, find the optimization solution $w_k$ to (3.24) on the intersection region $I_k$, and update $\gamma : \gamma_k \rightarrow \gamma_{(k+1)}$ according to (3.26).

Formally, in the $k^{th}$ iteration with $\gamma_{lb} = \gamma_k$, the solution $w_k = (w_{k1}, w_{k2}, \cdots, w_{kd})^T$ given by (3.25) satisfies

$$t_k = \sum_{i=1}^{d} (\gamma_k - u_i)w_{ki}^2 \quad \text{and} \quad 0 < t_k \leq 1.$$

Thus

$$t_k = \sum_{i=1}^{d} (\gamma_k - u_i)w_{ki}^2,$$

$$\Leftrightarrow t_k + \sum_{i=1}^{d} u_i w_{ki}^2 = \gamma_k \sum_{i=1}^{d} w_{ki}^2,$$

$$\Leftrightarrow \frac{1 + \sum_{i=1}^{d} u_i w_{ki}^2}{\sum_{i=1}^{d} w_{ki}^2} = \gamma_k + \frac{1 - t_k}{\sum_{i=1}^{d} w_{ki}^2}.$$

So we can update $\gamma_{lb}$ with

$$\gamma_{lb} = \gamma_{(k+1)} = \gamma_k + \frac{1 - t_k}{\sum_{i=1}^{d} w_{ki}^2} > \gamma_k, \quad (3.26)$$
and step to solve the updated optimization problem

\[
\min_{w,b} \quad g_{k+1}(w, b) = \sum_{i=1}^{d} (\gamma_{lb} - u_i)w_i^2
\]

s.t. \[
\sum_{i=1}^{d} (\gamma_{lb} - u_i)w_i^2 \leq 1; \gamma_{lb} = \gamma_{k+1}, \quad \text{and} \quad y_i(w^T\bar{x}_i - b) \geq 1, \quad i = 1, 2, \ldots, n.
\]

A smaller ellipsoid \(E_{k+1}\) is defined in (3.27), and the point \(w_k\) is on the border of \(E_{k+1}\). Repeat the above process iteratively starting from \(\gamma_{lb} = \gamma_0 = \max_{i=1}^{d}\{u_1, u_2, \ldots, u_d\}\), and we get a monotonically increasing sequence \(\{\gamma_0 < \gamma_1 < \cdots < \gamma_N < \gamma_{(N+1)} < \cdots\}\) approaching \(\gamma_{\text{max}}\). This process stops when any one of the following two stop criteria is satisfied: \(|\gamma_{(N+1)} - \gamma_N| < \epsilon_1\) or \(t_N > 1 - \epsilon_2\) for pre-specified \(\epsilon_1 > 0, \epsilon_2 > 0\). The synthetic experiments show that the iterative search method is \(3 \sim 10\) times faster than the bisection search algorithm.

### 3.3.5 Non-linearly separable case

Many classification problems are not linearly separable. These cases can be tackled with the inclusion of a soft margin. In this case, the slack variables \(\xi = (\xi_1, \ldots, \xi_n)^T\) and the regularizing parameter \(C > 0\) need to be added to (3.16). Since some incomplete data may now be incorrectly classified, we need to adjust the weights of the angle term according to the value of the slack variables. This can be done as follows,

\[
\max_{w,b} \quad \frac{1+K}{\|w\|^2} \sum_{i=1}^{n} sgn(1-\xi_i)K_i\|w_i\|^2 - C \sum_{i=1}^{n} \xi_i
\]

s.t. \[y_i(w^T\bar{x}_i - b) \geq 1 - \xi_i; \xi_i \geq 0, \quad i = 1, \ldots, n,
\]

where \(sgn(\cdot)\) is the sign function to adjust the maximization of the corresponding cosine term based on the potential values taken by the missing entries of the incomplete feature vectors.
Although (3.28) is defined on a convex region, this equation is difficult to solve because the function \( sgn \) is not continuous. As it is common in such cases, we choose to optimize a closely related cost function

\[
\max_{w, b} \frac{1 + K \sum_{i=1}^{n} (1 - \xi_i)K_i\|w_i\|^2 - C \sum_{i=1}^{n} \xi_i\|w\|^2}{\|w\|^2} 
\]  
(3.29)

s.t. \( y_i(w^T\bar{x}_i - b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, n, \)

This defines the PSVM algorithm as

\[
\max_{w, b, \xi} h(w, b, \xi) = \frac{1 + \sum_{i=1}^{d} u_i(\xi)w_i^2}{\sum_{i=1}^{d} w_i^2} 
\]  
(3.30)

s.t. \( y_i(w^T\bar{x}_i - b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, n, \)

where \( u_i(\xi) \) is a function of \( \xi \). Using the solution of (3.12) as an initialization and using the iterative method defined above to solve for \( w, b \) and \( \xi \), we arrive at the desirable solution.

To see this, note that if \( \xi \) is fixed, \( h(w, b, \xi) \) can be maximized in the same way as the linearly separable case presented above; if \( w \) and \( b \) is fixed, \( h(w, b, \xi) \) becomes an easy linear optimization problem defined on a convex region.

After the hyperplane that separates the two classes has been learned, it can be readily used to classify a new test feature vector. If the test image is incomplete, however, we need to first determine the probability of its values. To do this, we will use the probabilistic view defined earlier. This we do in the section to follow.

### 3.3.6 Multi-weight data filling

A SVM algorithm was derived to find the optimal hyperplane separating two classes with incomplete data. However, in this formulation, a complete training set \( \{\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n\} \) is need in the training stage. If the test image \( t \) is occluded, its completed version, \( \bar{t} \), is also needed for classification.
To determine the values of the missing elements of a partial vector from a complete dataset, there are different ways. For example, we can use zero (or the mean value of that feature over all the available observations) to fill in the missing elements of the corresponding feature. As we have discussed in Section 3.2, a linear combination of visible data in several training data samples can be used to approximate the visible data in the test sample. A multi-weight solution is provided if the training data have missing entries. The same idea can also be applied to fill in the missing part of partial data. As in Fig. (3.1), the weights $\omega_{11}, \omega_{21}$ calculated to describe $t_1^\Pi$ from the corresponding parts in $x_{11}$ and $x_{21}$ can be used to estimate the missing part, $t_0^1$. Similarly, $t_0^2$ can be imposed as the linear combination of the parts in $x_{22}$ and $x_{32}$ weighted by $\omega_{22}$ and $\omega_{32}$.

Formally, with the same notations in Section 3.2, the weights calculated in (3.5) can be used to give the estimation of the missing part on pattern $P_l$,

$$\hat{t}^\Pi_l = \sum_{j \in L_l} \omega_l^j x_j^\Pi_l.$$  (3.31)

If for some pattern $P_l$, the feature set $\Pi_l$ is not empty but $\Gamma_l$ is, it means that the corresponding weights cannot be computed. In this case, we use the average value of the training set to determine the most probable value of the missing entries (i.e. the value with highest probability term assuming the data is normally distributed).

### 3.3.7 Analysis of PSVM

In the optimization function of PSVM, there are several parameters to control the balance between the classification accuracy of the partial data and the generalization performance of the hyperplane. In this section, we are going to see how these parameters will change the hyperplane.
In (3.15) and (3.18), two different but equivalent sets of parameters, \( \{K, K_1, K_2, \ldots, K_n\} \) and \( \{u_1, u_2, \ldots, u_d\} \) are used. \( \{K_1, K_2, \ldots, K_n\} \) control the relative weights among different partial data. The simplest way is to set them equally. However, we may want to further differentiate different data. The only information we know is the number of missing features of each partial data unless more prior knowledge about the importance of different partial data is provided. The data with more missing feature becomes less important (in general) since it provides less information to the classifier. Therefore, we can set \( K_i \) to an inverse proportion of the number of missing features in the corresponding data. But these two settings do not give much difference in the experiments we implement in Section 3.4, thus we set all \( K_i \) to 1.

The regularizing parameter \( K \) (or equivalently, the norm of \( u \)) plays an important role in the performance of different solutions to (3.15) as it controls the tradeoff between the accuracy and the generalization. In Fig. 3.5 we give different hyperplanes with its corresponding \( K \) in a 2D simple case. In this example, two classes have two complete data samples each. Class 1 has one more partial data \( x \) and its fill-in \( \bar{x} \) is located on the affine space \( S \) defined by \( x \) (indicated as the square point on the line \( S \) in Fig. 3.5). Different values are assigned to \( K \) to calculate the corresponding hyperplane according to (3.16), and the results are shown in four plots. We can see that as the regularizing parameter \( K \) increases (or equivalently, \( \|u\| \) increases), the hyperplane becomes more parallel to the affine space \( S \). That is to say, the classifier we are trying to find will correctly classify the partial data with a higher probability. At the same time, the geometric margin becomes smaller and smaller, which means a weaker generalization ability to the hyperplane. This property of PSVM is useful to help improve the over-all performance of the classifier by choosing the proper parameter \( K \) after cross validation in the training set.
Figure 3.5: The Hyperplanes given by the solution to (3.16) with different regularizing parameters, (a) $K = 0$, (b) $K = 1$, (c) $K = 10$, (d) $K = 100$. The hyperplane becomes more parallel to the affine space $S$ as $K$ increases and the geometric margin becomes smaller at the same time.
3.3.8 Classification of partial data

The above multi-weight linear least squares filling method can work in both training and testing stage of PSVM. In the training stage, the partial data can be filled based on all other samples in the same class, and there is only one fill-in for each partial data. After we have completed the partial training data, PSVM algorithm proposed in this chapter can be applied to give the optimal hyperplane which is readily used for testing the complete samples.

If the test sample is not complete, we cannot classify it directly based on the distance to the hyperplane. Before the classification, we have to fill in it to make it a complete one. But one problem arises, that if it is a good idea to generate one single fill-in of this partial test data based on the whole training set. The answer is no. If we do so, the estimated value of the missing feature is very likely to have the same distance to all classes, which makes the estimation useless for the purpose of classification.

Thus a more suitable way to do the classification of the partial data is to generate one fill-in for each class and make the decision based on the overall comparison. For example in the two-class PSVM, the partial test data $t$ will have two different fill-ins $\bar{t}_1$ and $\bar{t}_2$ (one for each class). These two fill-ins will have different distances to the separation plane in the feature space, and the one with a larger distance to the hyperplane from the side of its corresponding class will give us the correct label.

For the multi-class case of PSVM, there are two different ways to do the classification as that in the classical multi-class SVM. The first solution is one vs. one plus majority voting. That is to do $C\left(C - 1\right)/2$ two-class PSVM ($C$ is the number of classes) and each one will give a vote to the winning class, and the class with the maximal votes is chosen.
as the final label of the test data. The other way is to do $C$ one vs. all PSVM and find the fill-in with the largest distance to the hyperplane.

In our experiments, one vs. one with majority voting method provides better results. The running time is a little longer since we have to solve $(C-1)/2$ times more optimization problems. But we notice that the running time is not much longer since the scale of the optimization problem solved in one vs. one is much smaller than that in one vs. all.

### 3.4 Experimental results

We refer to the reconstructive face recognition approach presented in Section 3.2 as the Partial Within-Class Match (PWCM$_r$) method, where the subscript $r$ specifies the metric (or quasi-metric) used in (3.9). Our approach is to use the 2-norm in (3.6) and the 1-norm or .5-quasi-norm in (3.9). Nonetheless, for comparison purposes, we also provide results where we have employed the 2-norm in (3.9). Our results are consistent with our theoretical argument and, hence, the 1-norm in (3.9) results in superior results to those of the 2-norm. We have also experimented with the use of the 1-norm in (3.6) and the 2-norm in (3.9). This resulted in slightly worse results than the ones reported below and with the added disadvantage of a high computational cost – typically, a 10-fold increase.

In the Partial-data Support Vector Machines approach, the parameters $\{K_1, \ldots, K_n\}$ controlling the relative weights among different incomplete observations are set to 1. The norm of $u$ controlling the tradeoff between the accuracy and the generalization, needs to be fixed. A 5-fold cross validation is implemented in the training set to determine $\|u\|$ from the set $\{0.1, 0.5, 1, 5, 10, 40\}$. To relieve the difficulty in the cross validation caused by the ”small sample size” problem, we use the following methods to increase the number of training samples [68, 96]. Each one training image is shifted one pixel vertically and
horizontally to generate four more images. Additionally, the mirror image of each one is also used. Thus ten times the number of the face images are produced in the training set. Note that, the test image does not need to be shifted or mirrored.

3.4.1 Database and experimental settings

In the present chapter, we employ the AR face database [70], which is one of the most popular databases, and one of the very few to include natural occlusions. The AR face database consists of more than 100 people’s frontal-view color images. Other image variations include different illuminations and distinct facial expressions. This database is considered very challenging, since \( \sim 50\% \) of the images have large or very large occlusions. The first 13 images, for one of the subjects in the database, are shown in Fig. 3.6(\( a-m \)). These correspond to the images taken during a first session. Another set of 13 images taken under the same occlusions, illuminations, and expressions was taken two weeks after the first session. We will refer to the images in this second session as \( a’ \) to \( m’ \). The images in the first session are labelled \( a \) to \( m \) following the notation shown in Fig. 3.6.

In our experiments, we first detect and warp the face (without the inclusion of any hair or background) to generate the registered face image with a fixed size, and calculate the occlusion mask for each of them (as described in Section 3.4.2). This localization and warping process is known to improve recognition rates [68]. Then, we convert the face images to gray-scale and resize them to an \( a \times b \) size. These \( a \) and \( b \) are selected in each experiment to match those used by other authors. This facilitates a direct comparison to a large number of results reported in the literature. We randomly choose 100 persons (50 male and 50 female) from the database. Each face image is segmented using an oval-shaped mask as shown in Fig. 3.6(n).
3.4.2 Computing occlusion masks

As mentioned in previous sections, in our formulation for face recognition with occlusions, we require of the occlusion masks $m_i$ ($i = 1, 2, \cdots, n$) and $\hat{m}$, which specify the pixels that are occluded in each image. We now present an algorithm to calculate $m_i$ and $\hat{m}$.

In face detection and segmentation, color is a practical cue for robust detection, because human skin color can be reliably modelled using statistical methods [44, 84]. Detection of an occlusion is a bit trickier, because we need to distinguish between these and the background. Here, we adopt a variant of the approach presented in [68]. After a principal component analysis face detection step, the approach models the skin color using a mixture of Gaussians and then employs morphological operators to tune the result. Two examples on non-occluded faces and two on occluded faces are given in Fig. 3.7. In the last step, the
morphological operators of erosion and dilation are used to eliminate isolated segments and refill the eroded local areas, respectively. After these steps, we can delineate the face limits with the use of the color map previously obtained. The right and left most pixels with skin color for each image row and the top and bottom pixels in each column are used to achieve this. The result of the entire process is shown to the right of each of the occluded image examples in Fig. 3.7. The final occlusion detection results shown in the right-most image of each row in Fig. 3.7 are obtained after a second round of morphological erosion and dilation, where 0s (black) represent occluded facial pixels and 1s (white) non-occluded. The binary occlusion map is vectorized to get the occlusion masks $m_i$ and $\hat{m}$.

Note that the process described above also determines the limits of the face. This will be used to separate the face from its background – to prevent misclassifications due to background noise.
3.4.3 Synthetic occlusions

The first part of the experiments is to test the proposed algorithms under synthetic occlusions and different facial expressions.

**Experiment 1:** PWCM algorithm is tested on different training sets. We use the first image (neutral face) of the first session (a) of each subject as the training set, and the happy (b), sad (c) and scream face (d) from the first session for testing. The occlusion is simulated by placing a random square mask of $s \times s$ to both the training and testing images. In Fig. 3.8(a), we can see that the proposed algorithm can successfully handle occlusions of up to $20 \times 20$ pixels, corresponding to a $\sim 20\%$ occlusion in both the training and testing sets. Note that this is different to the previous results reported in the literature, since we also enforce occlusions in the training set, not only the testing one.

Fig. 3.8(b) shows the reverse example. That is, using the face images b, c and d for training and a for testing. Here too, the maximum allowable occlusion is of $\sim 20\%$ of the image.

A similar experiment is implemented on PSVM. We use the neutral, happy and sad faces in the first session (a, b, and c) for training, and the screaming face (d) for testing (images). Occlusions are added to the training images only by overlaying a black square of $s \times s$ pixels in a random location. Fig. 3.9(a) shows the results with $s = 0, 3, 6, 9, 12$. Next, we use the images of the first session (a, b, c, d) for training, and the duplicates (a’, b’, c’, d’) for testing. In this case, the $s \times s$ occlusion masks are randomly added to the images in both the training and testing sets. The results are in Fig. 3.9(b). We see that in both cases, occlusions of up to $6 \times 6$ pixels do not affect the recognition rates.
3.4.4 Real occlusions

We divide the experiments in this section into two parts. The first set of experiments only considers occlusions in the testing set – allowing for a comparison with the state of the art. The second set of experiments considers occlusions in both the training and testing sets.

**Experiment 2:** We use the neutral non-occluded face $a$ for each of the 100 individuals for training, and the occluded faces $h$ and $k$ (of the first session) and $h'$ and $k'$ (of the second session) for testing. The PWCM results are compared with two other methods: the local probabilistic method of [68] and the Self Organizing Map (SOM) of [100]. The comparative results are in Fig. 3.10.
Consistent with our theory of PWCM algorithm, the use of the 1-norm in (3.9) results in more accurate classifications. We also see that the results of the proposed approach are (on average) superior to those of [68] and [100]. The method of [100] provides comparative results with ours for those images within the same session, but not with those in the other session.

**Experiment 3:** Our previous experiments only considered a single training image per class. We now consider the case where the number of training images is larger, as it was done in [21, 99, 121]. In the first of these experiments, the training set is \{a,b,c,a',b',c'\} (i.e. the non-occluded faces), and the testing set is \{d,h,k,d',h',k'\} (i.e. the scream face and the occluded set). The experimental results are plotted in Fig. 3.11. In this comparison, we note that the results reported in [21] only used 50 people from the AR database. The second experiment uses \{a,b,c,d,a',b',c',d'\} as the training set and \{h,k,h',k'\} as the testing. Comparative results against the method of Wright *et al.* [121] are in Fig. 3.12, where
Figure 3.10: Successful classification rate using the proposed approach, PWCM$_r$ (with $r = 2$, 2-norm, $r = 1$, 1-norm, and $r = .5$, .5-quasi-norm). The results are compared to those in [68] and [100]. Here, $a = 170$ and $b = 120$.

Figure 3.11: Training set $\{a, b, c, a', b', c'\}$. Testing set $\{d, h, k, d', h', k'\}$. Here, $a = 100$ and $b = 52$.

“Wright_S” is a method using a single block and “Wright_M” a set of multiple blocks as defined in [121]. The method of [121] does provide comparative results with ours. However, as reported in [121], their algorithm requires about 75 seconds of processing time for a single $83 \times 60$ test image (on a PowerMac G5). The proposed PWCM algorithm achieves a slightly superior result with each image while requiring of less than 1 second of processing time. Finally, comparative results with Tan [99] are in Fig. 3.13. Since no occlusion exists in the training set, the classifier hyperplane obtained with PSVM is the same as that of the
Figure 3.12: Training set \( \{a, b, c, d, a', b', c', d'\} \). Testing set \( \{h, k, h', k'\} \). Here, \( a = 83 \) and \( b = 60 \).

Figure 3.13: Training set \( \{a, b, c, d, e, f, g\} \). Testing set \( \{h, i, j, k, l, m, h', i', j', k', l', m'\} \). Here, \( a = 66 \) and \( b = 48 \).

standard SVM. The occluded test images need to be completed and classified as introduced in Section 3.3.8. The comparison between PSVM and [21] is also given in Fig. 3.14.

**Experiment 4:** There are very little results reported in the literature where a method can deal with partially occluded faces in both the training and testing sets. In [126], Zhu and Martinez provide experimental results using several subspace algorithms when using the images of the first session in the AR database for training and the images of the second session for testing. The authors report superior results for the Subclass Discriminant Analysis (SDA) algorithm [126]. In Table 3.1, we show the results obtained with the proposed PWCM and that of SDA. We also included the results obtained with a Nearest Neighbor
Figure 3.14: Training set \{a, b, c, a', b', c'\}. Testing set \{e, f, e', f'\}. Here, \(a = 29\) and \(b = 21\).

<table>
<thead>
<tr>
<th>SDA</th>
<th>PWCM(_2)</th>
<th>PWCM(_1)</th>
<th>PWCM(_{5})</th>
<th>PSVM</th>
<th>NN(_2)</th>
<th>NN(_1)</th>
<th>NN(_{5})</th>
<th>WCM(_2)</th>
<th>WCM(_1)</th>
<th>WCM(_{5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>78.2</td>
<td>85.0</td>
<td>89.6</td>
<td>90.6</td>
<td>85.6</td>
<td>64.0</td>
<td>66.9</td>
<td>66.2</td>
<td>76.0</td>
<td>83.2</td>
<td>84.6</td>
</tr>
</tbody>
</table>

Table 3.1: Training set \{a \sim m\}. Testing set \{a' \sim m'\}. Here, \(a = 54\) and \(b = 39\).

approach with the \(r\)-norm (denoted as \(\text{NN}_r\)), and those obtained when using Eq. (3.4) in place of (3.7) (denoted Within-Class Match, \(\text{WCM}_r\)). Again, our approach consistently outperforms the others, with the \(\frac{1}{2}\)-quasi-norm providing the top results.

More experimental results are provided in Table 3.2. In this table, all the training datasets are mixed ones, \textit{i.e.} both occluded and fully uncovered faces exist in the training set. We can see that PWCM provides a higher recognition rate than PWCM.

**Experiment 5:** Our next experiment considers the extreme cases where all the training images have partial occlusions and the test image can be either a partial image or a full one. There are many such cases. In Table 3.3, we report on seven possible scenarios. Since the occlusion in the AR face database each occludes almost 50\% of the image, this is a very challenging case. We see that, as expected, the 1-norm provides superior results to the 2-norm and that our results are consistently high. The only slightly low classification rate is
Table 3.2: Experimental results (recognition rate in percentages) with a variety of training and testing sets. Here, $a = 29$ and $b = 21$.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Testing Set</th>
<th>PSVM</th>
<th>PWCM$_2$</th>
<th>PWCM$_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[a,h,k]$</td>
<td>$[b,c,d]$</td>
<td>91.3</td>
<td>75.7</td>
<td>85.7</td>
</tr>
<tr>
<td>$[a,h,k]$</td>
<td>$[a',b',c',d']$</td>
<td>74.5</td>
<td>63.0</td>
<td>70.3</td>
</tr>
<tr>
<td>$[a',h',k']$</td>
<td>$[a,b,c,d]$</td>
<td>75.2</td>
<td>61.0</td>
<td>70.7</td>
</tr>
<tr>
<td>$[a',h',k']$</td>
<td>$[b',c',d']$</td>
<td>92.0</td>
<td>77.0</td>
<td>84.7</td>
</tr>
<tr>
<td>$[a',e,a',h',k']$</td>
<td>$[h,c,d,b',c',d']$</td>
<td>92.8</td>
<td>77.0</td>
<td>87.3</td>
</tr>
<tr>
<td>$[a,b,c,h,k]$</td>
<td>$[d]$</td>
<td>92.0</td>
<td>68.0</td>
<td>82.0</td>
</tr>
<tr>
<td>$[a,b,c,h,k]$</td>
<td>$[d']$</td>
<td>63.0</td>
<td>41.0</td>
<td>52.0</td>
</tr>
<tr>
<td>$[a,b,c,h,k,a',b',c',h',k']$</td>
<td>$[d,d']$</td>
<td>88.5</td>
<td>58.8</td>
<td>75.5</td>
</tr>
</tbody>
</table>

Table 3.3: Successful recognition rate (in percentages), with $a = 54$ and $b = 39$.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Testing set</th>
<th>PSVM</th>
<th>PWCM$_2$</th>
<th>PWCM$_1$</th>
<th>NN$_2$</th>
<th>NN$_1$</th>
<th>WCM$_2$</th>
<th>WCM$_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[h,k]$</td>
<td>$[a]$</td>
<td>98.0</td>
<td>90.0</td>
<td>96.0</td>
<td>50.0</td>
<td>90.0</td>
<td>55.0</td>
<td>32.0</td>
</tr>
<tr>
<td>$[h,k]$</td>
<td>$[a']$</td>
<td>82.0</td>
<td>60.0</td>
<td>77.0</td>
<td>33.0</td>
<td>54.0</td>
<td>32.0</td>
<td>27.0</td>
</tr>
<tr>
<td>$[h,k]$</td>
<td>$[b,c,d]$</td>
<td>87.0</td>
<td>70.3</td>
<td>87.6</td>
<td>39.0</td>
<td>69.7</td>
<td>43.0</td>
<td>31.3</td>
</tr>
<tr>
<td>$[h,k]$</td>
<td>$[b',c',d']$</td>
<td>61.0</td>
<td>44.7</td>
<td>60.3</td>
<td>21.7</td>
<td>37.7</td>
<td>29.0</td>
<td>19.7</td>
</tr>
<tr>
<td>$[h,k,h',k']$</td>
<td>$[a,a']$</td>
<td>98.5</td>
<td>94.0</td>
<td>$99.0$</td>
<td>53.5</td>
<td>89.5</td>
<td>37.5</td>
<td>28.5</td>
</tr>
<tr>
<td>$[h,k,h',k']$</td>
<td>$[h,c,d,b',c',d']$</td>
<td>91.0</td>
<td>72.5</td>
<td>90.0</td>
<td>38.5</td>
<td>70.5</td>
<td>27.0</td>
<td>15.5</td>
</tr>
<tr>
<td>$[h,k]$</td>
<td>$[h',k']$</td>
<td>59.0</td>
<td>51.0</td>
<td>57.0</td>
<td>26.5</td>
<td>33.0</td>
<td>32.5</td>
<td>35.0</td>
</tr>
</tbody>
</table>

for the case where we train with the set $\{h,k\}$ and test with $\{b',c',d'\}$. This is challenging because the images in the training and testing sets are quite distinct and correspond to different sessions. Since there is no other method available that allows a direct comparison, we show the results obtained with the nearest neighbor and the WCM approaches defined above. The results of the proposed approach are much superior.
Training set | Testing set | Image size | PSVM | PWCM₂ | PWCM₁
--- | --- | --- | --- | --- | ---
\[a\sim m\] | \[a'\sim m'\] | 108 × 78 | 86.9 | 85.2 | 89.8
\[a\sim m\] | \[a'\sim m'\] | 54 × 39 | 85.6 | 85.0 | 89.6
\[a\sim m\] | \[a'\sim m'\] | 36 × 26 | 84.6 | 84.5 | 89.2
\[a\sim m\] | \[a'\sim m'\] | 29 × 21 | 83.1 | 83.5 | 88.4
\[h,k,h',k'\] | \[a,a'\] | 108 × 78 | 98.5 | 94.0 | 99.0
\[h,k,h',k'\] | \[a,a'\] | 54 × 39 | 98.5 | 94.0 | 99.0
\[h,k,h',k'\] | \[a,a'\] | 36 × 26 | 97.5 | 93.5 | 99.0
\[h,k,h',k'\] | \[a,a'\] | 29 × 21 | 97.0 | 92.0 | 99.0

Table 3.4: Successful recognition rate (in percentage) obtained using the specified image size and training and testing sets with the PWCM algorithm.

### 3.4.5 Effect of different image sizes

Although all the algorithms aforementioned [21, 68, 99, 121, 126] use the AR database, the image size tends to be different in each experiment. This is important, because the performance of most algorithms goes up as the image size increases. In our final experiment, we demonstrate that the proposed algorithms achieve similar results to those shown above for a variety of image sizes. The results are in Table 3.4.

### 3.5 Conclusions

This chapter presents two novel face classification algorithms to deal with occluded faces in the training and testing sets. The first algorithm introduced a new mechanism to do reconstruction of partially occluded faces from the visible data only. We argued for the use of the 1-norm and the $\frac{1}{2}$-quasi-norm for comparison with this reconstruction. Our Matlab implementation of the algorithm classifies a new test image in less than a second.

We also introduced a SVM approach for face (object) recognition with partial occlusions. One major advantage of the proposed algorithm is that it allows for partial occlusions.
to occur in both, the training and testing sets, while previous methods would generally per-
mit occlusions only in the testing set. To achieve this goal, the derived algorithm considers
the affine subspace in the feature space formed by all possible fill-ins of a partial data, and
incorporates an additional term to the SVM formulation indicating the probable range of
values for the missing entries. This range is learned (estimated) from the available data. We
have shown that the resulting criterion is convex under very mild conditions. The proposed
method has then been shown to obtain higher recognition rates than the algorithms defined
in the literature in a variety of experiments.

In a large number of experimental results, we demonstrated the superiority of the pro-
posed approaches to those reported in the literature.
CHAPTER 4

CONCLUSIONS AND FUTURE STUDIES

4.1 Conclusions

Missing data problems appear in many fields of science, engineering, medicine and sociology. In this dissertation, we addressed two missing data problems in detail. The first one was the missing data problem in the structure from motion (SFM). The second problem was that of face recognition with occlusions.

The problem of SFM with missing data problem was presented in Chapter 2. SFM is a technique to be used to reconstruct the 3D structure of an arbitrary scene. This problem can be reduced to fitting a low-rank matrix to a full rank matrix. If the full measurement matrix containing all the trajectories of feature points can be obtained, SFM can be solved by a simple singular value decomposition (SVD), providing the optimal solutions in the least squares sense. However, the measurement matrix is usually contaminated by large noise and some of its entries do not have values. In this situation, SVD cannot provide a reconstruction.

The noise in the measurement matrix is caused by imprecise localization, while the missing data is usually caused by self-occlusions or tracking failure. We have proposed to solve this problem with the newly defined Deviation Parameter (DP) criterion which
is based on the results from subspace perturbation analysis. The DP criterion can help us choose the submatrices that are least affected by the noise term and the missing data. This new algorithms can be applied to the affine SFM directly and to the projective SFM in an iterative framework. Experimental results on both synthetic and real datasets have illustrated the robustness and efficacy of the proposed approach.

In Chapter 3, we focused on another active topic in computer vision and pattern recognition – the face recognition with occlusions. We first reviewed the recent advance in the literature and pointed out that most of the existing algorithms try to solve the problem of recognizing an occluded face from a full face training set. However, this is not encompassed all the face recognition problems with occlusions. A realistic face recognition system should also be able to recognize a partially and a non-occluded face from a training set in which some samples are partially occluded. We proposed two new algorithms to handle the task of face recognition in the situation where the training set contains occluded faces.

In the first algorithm, we treat the face recognition problem as a reconstructive one. Here, the key assumption is that the closest reconstruction of the test face sample is given by that generated from the training faces of the correct class. Only the visible data in both training and testing faces is used. We refer to our first algorithm as the Partial Within Class Match (PWCM) algorithm. The second approach is defined within the SVM framework. As opposed to the traditional SVM, which requires a full training set, our new SVM algorithm can generate a classifier (a hyperplane) from non-complete datasets. We refer to this algorithm as the Partial-data SVM (PSVM). In PSVM, all possible fill-ins of a partial data vector define an affine subspace. The classifier we want to find does not only need to
maximize the geometrical margin, it also needs to maximize the probability of the accurate classification on the affine subspace. Under a mild condition, the resulted optimization problem can be optimized globally with efficient solutions. The recognition performance of PWCM and PSVM on the partial training set has been tested on the publicly available database and the results are shown to be superior to other methods.

4.2 Future studies

The deviation parameter criterion presented in this thesis has been applied to the problem of structure from motion. Nonetheless, this criterion is very general and can be employed in many other problem where a low-rank fitting step is required. This is the case, for example, in the problem of face and object recognition, in optical flow, and in the modeling and classification of microarray data in bioinformatics. Further research will determine how the proposed criterion compares to previously defined approaches in these other domains. Extensions of this approach should also include other estimates of the upper-bound and extensions to non-Gaussian noise.

In face recognition problems, there are many different factors that can affect the performance of our algorithms. These factors often take effect jointly. Occlusions usually co-occur with different illuminations and pose variations. We have shown that our algorithms can tolerate different illuminations to some extent, since the AR database has images taken under different illuminations. But the use of PWCM and PSVM under different poses has not been studied. Our future work on face recognition should include tests on datasets with different poses and different natural occlusions. To the best of our knowledge, such a database is not available as of yet.
Since SVM is applied to face recognition under different poses [39], we will require of fundamentally new ideas to resolve these two problems simultaneously. We will apply PSVM in such a situation where both the occlusion condition and the poses change in the training and testing sets. Improvement on the pose invariance can also be expected, for example, in PWCM, if we adopt the pose estimation as a preceding step to reconstruction and take the results from pose estimation to re-weight different samples in the linear combination. Another promising step towards achieving face pose invariance can be anticipated from using 3D face model. This 3D face models may be either predefined or extracted from the training data using, for example, SFM.
BIBLIOGRAPHY


