

H124/3808

No Restrictions.

MONASH UNIVERSITY
THESIS ACCEPTED IN SATISFACTION OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
ON..... 16 November 2004.....

Sec. Research Graduate School Committee

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing for the purposes of research, criticism or review. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

**Note: the Errata/Addenda
are affixed to the end of the thesis**

**An Investigation of Statistical Aspects of
Linear Subspace Analysis
for
Computer Vision Applications**

Pei Chen

A dissertation submitted to the
Department of Electrical & Computer System Engineering
Monash University
Clayton, Victoria 3168, Australia
in fulfillment of the requirements for the degree of
Doctor of Philosophy

June, 2004

Contents

List of Figures.....	VI
List of Tables.....	VIII
Abstract.....	IX
Declaration.....	X
Acknowledgements.....	XI
Publications.....	XII
Chapter 1: Introduction.....	1
1.1 Issues to be addressed and motivations.....	1
1.1.1 i.i.d. Gaussian noise in the SVD.....	1
1.1.2 Missing data problem under low-rank constraint.....	2
1.1.3 Low rank approximatoin and parameter estimation in a hetaroscedastic system.....	3
1.2 Contributions of this thesis.....	4
1.3 Organization of this thesis.....	5
Chapter 2: SVD and perturbation expansion theory.....	8
2.1 Notation.....	8
2.2 Singular value decomposition.....	9
2.2.1 SVD theorem.....	10
2.2.2 Power method and orthogonal iteration method for the computation of one or a few singular values and vectors.....	10
2.2.2.1 The power method (Golub <i>et al.</i> 1996).....	11
2.2.2.2 The orthogonal iteration method (Golub <i>et al.</i> 1996).....	11
2.3 Matrix perturbation theory.....	12
2.3.1 Perturbation theory.....	13

2.3.2	New perturbation theory, corresponding to a multiple eigenvalue/ singular value.....	15
2.4	Conclusion.....	20
Chapter 3: The factorization method in SFM and PCA-based face recognition		
3.1	Overview of the development of the factorization method for SFM.....	21
3.2	The factorization method under orthographic settings	23
3.2.1	Measurement matrix and its rank 4 property.....	23
3.2.2	Metric transformation	25
3.3	PCA-based face recognition	26
Chapter 4: Analysis of noise effect in the SVD-based applications.....		
4.1	Issues to be addressed.....	29
4.1.1	Noise Effects.....	29
4.1.2	Performance Questions.....	30
4.1.2.1	Denoising capacity of SVD	30
4.1.2.2	Learning capacity of linear subspace analysis.....	31
4.2	Major results.....	32
4.2.1	Major results	32
4.3	Denoising capacity of SVD	33
4.3.1	Case of distinct singular values	34
4.3.2	Case of multiple singular value	36
4.3.3	Extension to the rectangular matrix	38
4.4	Learning capacity of LSA-based recognition system.....	39
4.4.1	Perturbation of the basis images.....	39
4.4.2	Projection of a new test image on the basis images.....	41
4.4.3	Performance analysis over the learning samples	42
4.4.4	The optimal learning set	44
4.5	Simulation results	45
4.5.1	Simulation of the denoising capacity of SVD	46
4.5.2	Simulation of the <i>learning</i> capacity for LSA-based recognition.....	46
4.5.3	Relationship with some experimental observations	51
4.6	Conclusion	52

Chapter 5: Recovering the missing components in a large noisy low-rank matrix: application to SFM	53
5.1 Introduction.....	53
5.1.1 Missing-data problem in SFM.....	55
5.1.2 Other missing data problems under low rank constraint	58
5.1.3 Contributions of this chapter	59
5.1.4 Overview of the chapter.....	59
5.2 The definition of the problem and its nonlinear nature	60
5.2.1 The problem.....	60
5.2.2 Non-linearity of the problem	61
5.3 An iterative imputation method.....	62
5.3.1 Minimization of the distance of a vector with missing entries to a known subspace.....	62
5.3.2 An iterative algorithm for the problem (<i>Iter</i>)	63
5.3.3 The convergence of the iterative algorithm.....	65
5.4 SVD's denoising capacity vs. missing data.....	65
5.4.1 SVD's denoising capacity and its extension to an incomplete matrix	66
5.4.2 The minimal unreliability ratio in SFM.....	68
5.4.3 Algorithm (<i>IterPart</i>).....	69
5.4.4 Discussion.....	70
5.5 Experiments	71
5.5.1 Only one unknown entry	71
5.5.2 Synthetic data in a 8-frame-and-40-point sequence	73
5.5.3 Box sequence.....	78
5.5.4 Dinosaur sequence.....	80
5.6 Conclusion	87
Appendices	87
A.1 "Bad-behaviour" and a bootstrapping strategy.....	87
A.2 Revisiting the objective function in (5.5.3)	89
A.3 Difference from the imputation in (Troyanskaya <i>et al.</i> 2001).....	90
A.4 RMS and Re-projection error	90

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting	93
6.1 Introduction.....	93
6.1.1 Parameter estimation in a heteroscedastic system.....	94
6.1.2 Bilinear approach to the low-rank matrix approximation.....	96
6.1.3 The issues to be studied and the organization of this chapter.....	98
6.2 The parameter estimation problem	98
6.2.1 Objective function to be minimized	98
6.3 The bilinear approach to the heteroscedastic parameter estimation	100
6.3.1 Update of R.....	101
6.3.1.1 Case with zero singular values in the covariance matrix C.....	101
6.3.2 Update of S	103
6.3.2.1 Constant column in the measurement matrix	104
6.3.2.2 Discussion of the convergence of the bilinear approach.....	105
6.3.3 A more general update.....	105
6.3.4 Disussion of the optimality.....	106
6.4 Application in conic fitting.....	106
6.4.1 Covariance matrix in the conic fitting	107
6.4.2 Noise level estimation.....	109
6.5 Experimental results	110
6.5.1 Noise level=2 over a half ellipse	111
6.5.2 Noise level=2 over 3/8 ellipse	114
6.5.3 Noise level=1 over a quarter ellipse	116
6.5.4 Noise level=2 and a quarter ellipse.....	118
6.5.5 Comments on the experimental results.....	119
6.5.6 Question raised	121
6.6 Conclusion	122
 Chapter 7: Orthographic projection of distances: a low-dimensional approximation	 124
7.1 Introduction.....	124
7.2 Distance vector, spherical harmonics, and linear subspace.....	124
7.2.1 Definition of distance vector (distance matrix).....	124
7.2.2 Spherical harmonics.....	125

7.2.3	Spherical harmonics for the <i>sine</i> function	125
7.2.4	Low dimensionality of the distance vectors	126
7.2.5	<i>Sine</i> function on a sparse set of points.....	128
7.3	Discussion and conclusion.....	129
Chapter 8: Subspace-based face recognition: outlier detection and a new distance criterion.....		
		131
8.1	Introduction.....	131
8.2	Lambertian reflection and low dimension subspace.....	131
8.2.1	Lambertian reflectance and 3-dimensional subspace	132
8.2.2	Attached shadow and low-dimensional subspace.....	133
8.2.3	Generation of the image basis from synthetic images	133
8.3	Learning capacity of low-dimensional subspace and a new distance criterion.....	135
8.3.1	Noise level estimation.....	136
8.4	Non-Lambertian pixel detection.....	137
8.4.1	Performance evaluation of the outlier detection strategy	138
8.5	Experimental results	139
8.6	Conclusion	142
Chapter 9: Conclusion and future directions		
		144
9.1	Summary of the contributions of this thesis	144
9.2	Directions of future work.....	145
References.....		147

List of Figures

- Figure 3.1: The coordinate systems: world system and camera system.
- Figure 4.1: The average error that still resides in the approximation matrix.
- Figure 4.2: The dependency of SSD on the size of learning samples.
- Figure 4.3: The effects of three parameters in (4.23) on SSD.
- Figure 5.1: The optimal performance of the iterative algorithm.
- Figure 5.2: Two 10-by-10 examples with one unknown entry.
- Figure 5.3: The convergence rate of four iterative methods against the missing entry fraction.
- Figure 5.4: The reprojection RMS error of the eight methods, as described in the beginning of section 5.5.
- Figure 5.5: One frame of the box sequence.
- Figure 5.6: The performance of the RMS reprojection error by the eight methods on the box sequence are depicted.
- Figure 5.7: The convergence rate of the four iterative methods are depicted.
- Figure 5.8: The 20th frame of the dinosaur sequence.
- Figure 5.9: The missing data (grey) and measured data (black) for the dinosaur sequence.
- Figure 5.10: The 336 tracked feature points over 36 frames.
- Figure 5.11: The 336 recovered tracks by the Jacobs' and Shum's and the proposed methods.
- Figure 5.12: The recovered tracks over 36 frames.
- Figure 5.13: The RMS errors in the low-rank approximation matrix and the reprojection RMS error.

- Figure 6.1: The comparison of the bilinear approach, with the FNS, HEIV and KAN, TLS.
- Figure 6.2: Figure 6.2: See the caption of Figure 6.. Noise =2 over 3/8 ellipse.
- Figure 6.3: Two example of "good" estimates, falling in the 10% and 20% range.
- Figure 6.4: Figure 6.3: The "good" estimates in 1000 trials of the Bilinear, FNS, HEIV and KAN approaches for noise=2 over 3/8 ellipse.
- Figure 6.5: Figure 6.4: The "good" estimates in 1000 trials of the Bilinear, FNS, HEIV and KAN approaches for noise=1 over 1/4 ellipse.
- Figure 6.6: Figure 6.5: The "good" estimates in 1000 trials of the Bilinear, FNS, HEIV and KAN approaches for noise=2 over 1/4 ellipse.
- Figure 6.7: A bad estimate, with details.
-
- Figure 7.1: The amplitudes of the spherical harmonics up to the tenth order.
- Figure 7.2: The approximations of the *sine* function.
- Figure 7.3: The singular values of the distance matrices with 20 (denoted by circles "o") or 100 (denoted by crosses "+") feature points.
-
- Figure 8.1: 10 persons in Yale-B face database.
- Figure 8.2: Different images under different illumination conditions, for person 7 in Figure 8.1.
- Figure 8.3: Mask for the outliers, which do not obey the 3-dimensional constraint. The black pixels denote the outliers.
- Figure 8.4: The 7D image basis of person 1 and person 4 in the Yale-B face database.

List of Tables

- Table 4.1: A typical example of the approximation error, $B^3 - A$.
- Table 6.1: The statistics of the estimated major length, minor length, x and y coordinates of the center, and the deflected angel. The truth is listed in the first row. For every method, its mean, with its standard deviation in the bracket, is listed in each row. For every method, its mean, with its standard deviation in the bracket, is listed in each row. Noise =2 over $\frac{1}{2}$ ellipse.
- Table 6.2: See the caption of Table 6. noise=2 over $\frac{3}{8}$ ellipse.
- Table 6.3: The "good" estimates for noise=2 over $\frac{3}{8}$ ellipse.
- Table 6.4: The "good" estimates for noise=1 over $\frac{1}{4}$ ellipse.
- Table 6.5: The "good" estimates for noise=2 over $\frac{1}{4}$ ellipse.
- Table 7.1: Energy distribution for the sphere and a sparse set of direction.
- Table 8.1: The ratio between the third singular value and the fourth singular value.
- Table 8.2: Comparison of the error classification rate on Yale-B face database.

Abstract

This thesis mainly studies linear subspace analysis and its applications in computer vision. Using first-order matrix perturbation expansion theory, we present a statistical analysis of the noise effect on subspace approaches to computer vision tasks. More specifically, we derive an explicit formula for the denoising capacity of the low rank matrix, in terms of the noise level, the sizes of the measurement matrix and the dimensionality. Similarly, by using first-order matrix perturbation expansion theory, we also derive an explicit formula for the learning capacity of algorithms using a learnt low-rank subspace approximation.

In the missing data problem under a low rank constraint, we present a criterion to recover the most reliable submatrix, in terms of deciding when the inclusion of extra rows or columns, containing significant numbers of missing entries, is likely to lead to poor recovery of the missing parts. This is based on the aforementioned theory of the denoising capacity of a large low-rank matrix. The superiority of our algorithm is validated in the structure from motion problem.

We propose a new error distance for the subspace-based recognition problem. This is based on the theory about the learning capacity in low-rank subspace approaches. In face recognition, we employ the iterative reweighted least square (IRLS) technique to detect the pixels that do not following the Lambertian reflectance model.

In this thesis, we also study other rank-constraint problems. We prove that the distance vectors under different views approximately lie in a linear subspace with a dimension of 6. We study the parameter estimation problem in a general heteroscedastic linear system, by putting the problem in the framework of the bilinear approach to low-rank matrix approximation.

Declaration

I declare that:

1. this thesis contains no material which has been accepted for the award of any other degree or diploma in any university or institution, and
2. to the best of my knowledge, it contains no material previously published or written by another person.

A solid black rectangular box used to redact the author's signature.

Pei Chen

Acknowledgements

First of all, I wish to sincerely thank my supervisor, Professor David Suter, for his guidance, encouragement, support, inspiration, patience, persistence and enthusiasm during my studies at Monash University. In particular, I would like thank David for the academic freedom in the Digital Perception Laboratory and for his guidance into the computer vision field.

Thanks to all the members of the Digital Perception Laboratory, including Daniel, Hanzi, Mohamed, James, Tat-jun, and former members, Dr. Fang Chen, Dr. Paul Richardson and Dr. Prithiviraj Tissainayagam.

I would like to express my gratitude to Jane for her help on English.

I also grateful to my friends, Homer Chen, Sara Shi, John Davis, and Lei Ma, for their daily help and suggestions.

I am also grateful to the Monash Graduate Scholarships from Monash Graduate School and the tuition scholarships from the Engineering faculty.

Publications

P. Chen and D. Suter, Recovering the missing components in a large noisy low-rank matrix: Application to SFM. *IEEE Trans. Pattern Analysis and Machine Intelligence*, no 8, vol. 26, 2004.

P. Chen and D. Suter, Subspace-based face recognition: outlier detection and a new distance criterion. In *Proceedings ACCV2004, Asian Conference on Computer Vision*, pages 830-835, 2004.

P. Chen and D. Suter, Subspace-based face recognition: outlier detection and a new distance criterion. Accepted by *International Journal of Pattern Recognition and Artificial Intelligence*, subject to minor revision. (Selected in a special issue of ACCV 2004).

P. Chen and D. Suter, Shift invariant wavelet denoising using interscale dependency, in *Proceedings ICIP2004, International Conference on Image Processing*.

Chapter 1

Introduction

Linear subspace analysis (LSA) has become rather ubiquitous in the solution of a wide range of problems arising in pattern recognition and computer vision. The essence of these approaches is that certain structures are intrinsically (or approximately) low dimensional: for example, the factorization approach to the problem of structure from motion (SFM) (Tomasi *et al.* 1992) and the principal component analysis (PCA)-based approach to face recognition (Turk *et al.* 1991; Hallinan 1994; Eipstein *et al.* 1995). Such approaches have also been employed in a wide range of other fields: like DNA prediction (Troyanskaya *et al.* 2001), recommender system (Sarvar *et al.* 2000; Brand 2003), and even the general “knowledge or opinion networks” (Maslov *et al.* 2001).

Computational approaches to such problems have been well established. Their central idea is to employ the singular value decomposition (SVD) (Golub *et al.* 1996) to obtain the low-dimensional representation of the high-dimensional structures. This representation, obtained by the SVD, is the best approximation, measured by the Frobenius norm or the 2-norm (Gold *et al.* 1996). More importantly, it is the maximum likelihood (ML) estimate, assuming the signal is corrupted with i.i.d. Gaussian noise (Press *et al.* 1992; Reid *et al.* 1996; Hartley *et al.* 2000; Irani *et al.* 2000; Anandan *et al.* 2002).

1.1 Issues to be addressed and motivations

1.1.1 i.i.d. Gaussian noise in the SVD

Although it is well known that the estimate, by the SVD, is the ML estimate in the presence of *i.i.d. Gaussian* noise, little effort has been spent on the analysis of the noise effects in the SVD-based applications. The lack of such a performance analysis impedes the careful design of optimal systems. For example, in the factorization

approach to SFM, it is widely accepted that processing more frames produces a more accurate result than depending on just a few ("few" typically being little more than 3) frames. It was even claimed (Thomas *et al.* 1999) that the 3D scene could be reconstructed to arbitrary accuracy given enough frames. *However, there are many questions one could pose about this, for example: what is the gain of adding the data from one extra frame to a very large measurement matrix? What precision does the 3D reconstruction have as the number of the frames approaches infinity?*

The questions above relate to the performance analysis of the SVD in the presence of noise. Fundamentally, they are specific examples of the following questions. Supposing the noise level is small enough, how much signal is retained by keeping the largest r components? Or, how much has the noise been reduced, as a result of discarding the other components? In this sense, we are *blindly* using a SVD, *without knowing* its capacity for separating the signal from the noise: how close the low-rank approximation matrix to the noise-free matrix is, or how close the SVD-based subspace to the ground-truth subspace is.

One of the major tasks of this thesis is to analyze the performance of the SVD in the presence of noise. This will be done in the context of the two computer vision tasks: in the factorization method for the structure from motion and in PCA-based face recognition.

1.1.2 Missing data problem under low-rank constraint

One prominent drawback with the SVD-based approaches is their inability to deal with the missing data. A SVD can only be applied to complete matrices. However, in computer vision, it is common to require operations on matrices with "missing data", for example because of occlusion or tracking failures in the SFM problem.

Take the SFM as an example. SFM with missing data has been well studied in computer vision community, eg (Tomasi *et al.* 1992; Shum *et al.* 1995; Jacobs 1997; Heyden *et al.* 1998; Kahl *et al.* 1999; Jacobs 2001; Brand 2002; Guerreiro *et al.* 2003), etc. In terms of the computation, we divide these approaches into two categories: iterative and non-iterative. The non-iterative approaches include: (Tomasi

et al. 1992; Heyden *et al.* 1998; Jacobs *et al.* 1998; Kahl *et al.* 1999; Jacobs 2001; Brand 2003), etc; and the iterative approaches include: (Shum *et al.* 1995; Guerreiro *et al.* 2003), etc.

Both approaches have their advantages and disadvantages. Iterative methods are usually more computation expensive. Moreover, they can suffer from the divergence problem when there is a large amount of missing data and when the observed data has been badly corrupted by noise. In contrast, the non-iterative methods cannot make full use of the information in the observed data, even though they are generally fast and stable.

A compromise approach is possible and, indeed, very efficient, as will be presented in chapter 5.

1.1.3 Low rank approximation and parameter estimation in a heteroscedastic system

Another drawback with the SVD (Golub *et al.* 1996) is its “inefficiency” in dealing with the heteroscedastic noise. Here, we mean the “inefficiency” the fact that the estimate, by the SVD, is not the ML estimate in the heteroscedastic system, as pointed out in (Irani *et al.* 2000; Anandan *et al.* 2002).

The Mahalanobis distance is always employed as the minimization objective function, in such a heteroscedastic system (Leedan *et al.* 1999; Matei *et al.* 1999; Irani *et al.* 2000; Leedan *et al.* 2000; Matei *et al.* 2000; Anandan *et al.* 2002). If the noise can be modelled as Gaussian, either i.i.d. or correlated, the minimizer of the Mahalanobis distance is the ML estimate.

A closely related problem is the parameter estimation in a heteroscedastic system, where some good approaches, like FNS (Chojnacki *et al.* 2000; Chojnacki *et al.* 2004) and HEIV (Leedan *et al.* 1999; Matei *et al.* 1999; Leedan *et al.* 2000; Matei *et al.* 2000), are already available. However, no convergence in these iterative approaches is ensured, as pointed out in the literature. Another limitation of these

approaches is that they can't model the correlations among different rows of the "general measurement matrix".

In this thesis, we employ the bilinear approach (Shum *et al.* 1995; Guerreiro *et al.* 2003; Vidal *et al.* 2004) to calculate the optimal estimate of the low-rank matrix approximation, measured by the Mahalanobis distance. Notably, a large set of parameter estimation can be put in the framework of the low-rank matrix approximation. In this way, we solve the parameter estimation problem in a heteroscedastic system.

1.2 Contributions of this thesis

This thesis has two major contributions. First, it presents an analysis of the i.i.d. Gaussian noise effects in SVD-based computer vision tasks, especially in the context of the factorization method for the SFM problem and in PCA-based face recognition. This is achieved by using the matrix perturbation expansion theory (Wilkinson 1965; Stewart *et al.* 1990) as the major tool. Second, we employ the bilinear approach (Shum *et al.* 1995; Guerreiro *et al.* 2003; Vidal *et al.* 2004) to calculate the low-rank matrix approximation in a heteroscedastic system, and consequently solve the parameter estimation problem in a heteroscedastic system.

More specifically, we study the following problems:

- In the factorization method, we study the error between the noise free measurement matrix and the approximated (low rank) measurement matrix. Using the matrix perturbation expansion theory, we derive an explicit formula for the mean square error (MSE) of the approximated (low rank) matrix, in terms of the noise level, the sizes of the measurement matrix and the dimensionality. We call this the *denoising capacity* of a large low-rank matrix.
- With this characterization of the denoising capacity, we can explain the accepted fact that more frames produce more accurate result than a few ("few" typically being little more than 3) frames. Moreover, we can explain

other phenomenon: that one more frame helps little in improving the reconstruction precision after 20-40 frames (Morita *et al.* 1997). More importantly, by using the denoising capacity, we present, in chapter 5, a criterion to recover the most *reliable* submatrix, in terms of deciding when the inclusion of extra rows or columns, containing significant numbers of missing entries, is likely to lead to poor recovery of the missing parts.

- In PCA-based face recognition, we study the reprojection error of a new face image upon a low-dimensional subspace, which is learned from a few samples. By using the first-order matrix perturbation theory, we also derive an explicit formula for such a reprojection error measure. This error comes from two independent sources: one source is the noise in the training samples and another in the noise in the test image. Based on this learning capacity theory, we propose a new error distance for subspace-based recognition problems.
- We study the parameter estimation problem in a general heteroscedastic linear system, by putting the problem in the framework of the bilinear approach to the low-rank matrix approximation. We extend the bilinear approach to the heteroscedastic system and consequently solve the problem of parameter estimation in a heteroscedastic system.

1.3 Organization of this thesis

Although the core contribution of this thesis is a *general* theoretical analysis of the noise effect in the SVD-based applications, this thesis mainly concerns two computer vision tasks: the factorization method for the SFM problem and the PCA-based face recognition, in order to make the *general* problem concrete. Thus, a significant portion of this thesis is devoted to developing such a context.

A brief description of the subsequent chapters is as follows:

- **Chapter 2: SVD and perturbation expansion theory.** The requisite knowledge of the matrix theory is covered, including the SVD theorem and the first-order matrix perturbation theory. We also include our new result concerning the first-

order perturbation expansions of the singular vectors (eigenvectors), associated with a multiple singular value (eigenvalue).

- **Chapter 3: The factorization method in SFM and PCA-based face recognition.** We review the development of the factorization method for the SFM problem and describe the seminal factorization method in detail. Then, we present an overview of the PCA-based face recognition.
- **Chapter 4: Analysis of noise effect in the SVD-based applications.** The noise effect in the SVD is analyzed in the context of two computer vision tasks: the factorization method for the SFM problem and PCA-based face recognition. This analysis is important in since the results in this chapter will be used in chapter 5 and chapter 8.
- **Chapter 5: Recovering the missing components in a large noisy low-rank matrix: application to SFM.** This chapter concerns the missing data problem under a low rank constraint. First, an iterative approach, with weak convergence, to this problem is proposed. More importantly, by using the denoising capacity theory from chapter 4, we present a criterion to recover the most *reliable* submatrix, in terms of deciding when the inclusion of extra rows or columns, containing significant numbers of missing entries, is likely to lead to poor recovery of the missing parts.
- **Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting.** We study the parameter estimation problem in a general heteroscedastic linear system, by putting the problem in the framework of the bilinear approach to the low-rank matrix approximation. The conic fitting problem is studied as a specific example of the general theory.
- **Chapter 7: Orthographic projection of distances: a low-dimensional approximation.** We prove that the distance vectors under different views approximately lie in a linear subspace with a dimension of 6.
- **Chapter 8: Subspace-based face recognition: outlier detection and a new distance criterion.** We propose a new error distance for the subspace-based recognition problem, based on the theory about the *learning capacity* introduced in chapter 4. Moreover, in order to remove points not following the Lambertian reflectance model, we employ the iterative reweighted least square (IRLS)

Chapter 1: Introduction

technique to detect the pixels that do not obey the dimension-3 subspace constraint.

- **Chapter 9: Conclusion and future directions.** We summarize the main work in this thesis and give some comments of the future research.

Chapter 2

SVD and perturbation expansion theory

In this chapter, we intend to provide a background, in terms of the relevant matrix theory for the following chapters. However, this chapter does a little more than just survey known results: a result that appears to be new is also presented.

We first introduce (section 2.1) some notations used in this chapter and in the rest of the thesis. In section 2.2, we introduce the singular value decomposition (SVD) theorem (Golub *et al.* 1996), with two iterative approaches to compute the singular vectors, associated with the largest (few) singular values. Section 2.3 includes a treatment of Matrix Perturbation Expansion theory (Wilkinson 1965; Stewart *et al.* 1990); including our new result, which can be found in section 2.3.2.

2.1 Notation

The notation introduced here will be used through the rest of the thesis. A matrix will be denoted by a bold capital letter, such as \mathbf{M} , and a bold lowercase letter represents a vector, e.g. \mathbf{x} . The i^{th} column of \mathbf{M} is denoted by \mathbf{M}_i . A scalar entry in a vector or in a matrix will be denoted by, for example, x_1 or $M_{1,2}$ respectively. \mathbf{I}_n denotes the $n \times n$ identity matrix, and $\mathbf{0}_{m,n}$ for an $m \times n$ zero-matrix. \mathbf{e}_i is the i^{th} column of \mathbf{I}_n . $\mathbf{M}_{i,j,k,l}$, a notation from *Matlab*, denotes for the submatrix of \mathbf{M} : the intersection of the i -to- j rows and the k -to- l columns. The set of $m \times n$ matrices with orthonormal columns is denoted by $O^{m,n}$. That is, $\mathbf{U} \in O^{m,n}$ iff $\mathbf{U}^T \mathbf{U} = \mathbf{I}_n$. Such a matrix will always be denoted by \mathbf{U} or \mathbf{V} .

The Frobenius norm of a matrix \mathbf{M} (or a vector) will be denoted as $\|\mathbf{M}\|_F$, where

$$\|\mathbf{M}\|_F = \sqrt{\sum_{i,j} M_{i,j}^2}. \text{ Span}(\mathbf{M}) \text{ denotes the subspace spanned by the columns of } \mathbf{M}.$$

The distance of a vector \mathbf{m} , $\mathbf{m} \in R^m$, to the subspace $\text{Span}(\mathbf{M})$, $\mathbf{M} \in R^{m,n}$, is

denoted as $d(\mathbf{m}, \text{Span}(\mathbf{M}))$ and it is sometimes described as the distance of a vector \mathbf{m} to a matrix \mathbf{M} , denoted as $d(\mathbf{m}, \mathbf{M})$. If $\mathbf{U} \in O^{m \times n}$, $d(\mathbf{m}, \mathbf{U}) = \|\mathbf{m} - \mathbf{U}\mathbf{U}^T \mathbf{m}\|_F$.

The hat symbol, “^”, denotes an estimate of the quantity beneath the symbol. Usually, a signal is observed with noise. We denote the observed signal with an unadorned symbol that depends on its nature; for example: s , \mathbf{s} , or \mathbf{S} , for a scalar, a vector or a matrix respectively. Although the true signal, without noise, is generally unknown, we need to refer to it in some cases, especially when we evaluate the performance. In this thesis, we use the symbol “~” to denote the noise-free signal, eg \tilde{s} .

\mathbf{M}^r denotes the closest rank- r approximation of \mathbf{M} , which can be obtained by SVD (Golub *et al.* 1996). The SVD, itself, will be introduced in the next section. The symbol “ \approx ” means the first order perturbation, explained in section 2.3. Finally, “ \cong ” means equality, in the sense of statistical expectation.

2.2 Singular value decomposition

The singular value decomposition plays an important role in many applications, where a few of the largest singular values, and possibly their associated singular vectors, are needed. This includes the calculation of the low rank approximation of a matrix. Such applications can be found in the factorization method for structure from motion in computer vision (Tomasi *et al.* 1992; Poelman *et al.* 1997), object modeling from range images in computer vision (Shum *et al.* 1995), data analysis (Mardia 1979), image processing and coding (Pratt 1975), model reduction (Moore 1981), pattern recognition (Chien *et al.* 1967), the minimal realization of linear system (Kung *et al.* 1987) and antenna array processing (Schmidt 1979; Bienvenu *et al.* 1983).

2.2.1 SVD theorem

The principle behind the SVD (Golub *et al.* 1996) states that any matrix, $\mathbf{M} \in R^{m \times n}$, can be decomposed into

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (2.1)$$

where $\mathbf{U} \in O^{m \times m}$, $\mathbf{V} \in O^{n \times n}$ and $\mathbf{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_p\} \in R^{m \times n}$, with $p = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$.

Without loss of generality, suppose $m \geq n$. $\{\sigma_i^2 \mid i = 1, 2, \dots, n\}$ are the eigenvalues of $\mathbf{M}^T \mathbf{M}$, or the first n largest eigenvalues of $\mathbf{M} \mathbf{M}^T$. The first n left singular vectors of \mathbf{M} are $\{\mathbf{U}_i \mid i = 1, 2, \dots, n\}$, where \mathbf{U}_i is the eigenvector, corresponding to the eigenvalue of λ_i^2 , of $\mathbf{M} \mathbf{M}^T$. Similarly, the right singular vectors of \mathbf{M} are $\{\mathbf{V}_i \mid i = 1, 2, \dots, n\}$, where \mathbf{V}_i is the eigenvector, corresponding to the eigenvalue of λ_i^2 , of $\mathbf{M}^T \mathbf{M}$.

Another important fact (Golub *et al.* 1996), is that one can easily construct \mathbf{M}^k , the closest rank k approximation of \mathbf{M} , measured by 2-norm or Frobenius-norm, by:

$$\mathbf{M}^k = \sum_{i=1}^k \sigma_i \mathbf{U}_i \mathbf{V}_i^T \quad (2.2)$$

Specifically,

$$\|\mathbf{M} - \mathbf{M}^k\|_2 = \sigma_{k+1} \quad (2.3)$$

$$\|\mathbf{M} - \mathbf{M}^k\|_F = \sqrt{\sum_{j=k+1}^n \sigma_j^2} \quad (2.4)$$

2.2.2 Power method and orthogonal iteration method for the computation of one or a few singular values and vectors

In many applications, such as in the PCA (Mardia 1979) and the factorization method for SFM (Tomasi *et al.* 1992), only a few of the largest singular values and their associated singular vectors are needed. From the SVD theorem above, it can be observed that the right (left) singular vectors of \mathbf{M} are the eigenvectors of

$M^T M$ (MM^T). In this subsection, we concentrate on the computation of a few of the most dominant eigenvalues and their associated eigenvectors.

2.2.2.1 The power method (Golub *et al.* 1996)

Suppose a matrix $M \in R^{n \times n}$ is diagonalizable and that $X^{-1}AX = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ with $X = [X_1, X_2, \dots, X_n]$ and $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$. Starting from an arbitrary initial vector $v^{(0)}$, a sequence $\{v^{(k)}\}$ will be produced by the power method:

for $k=1, 2, \dots$
 $z^{(k)} = Mv^{(k-1)}$; $v^{(k)} = z^{(k)} / \|z^{(k)}\|_F$
 end

If $v^{(0)} = c_1 X_1 + c_2 X_2 + \dots + c_n X_n$ and $c_1 \neq 0$, then it follows that

$$M^k v^{(0)} = c_1 \lambda_1^k \left(X_1 + \sum_{j=2}^n \frac{c_j}{c_1} \left(\frac{\lambda_j}{\lambda_1} \right)^k X_j \right) \quad (2.5)$$

and

$$v^{(k)} = \xi_k \lambda_1^k \left(X_1 + \sum_{j=2}^n \frac{c_j}{c_1} \left(\frac{\lambda_j}{\lambda_1} \right)^k X_j \right) \quad (2.6)$$

where ξ_k is a constant.

From (2.6), $\{v^{(k)}\}$ approaches the direction of X_1 and the convergence ratio is $|\lambda_2 / \lambda_1|$.

2.2.2.2 The orthogonal iteration method (Golub *et al.* 1996)

Only the most dominant eigenvector, with its associated eigenvalue, is computed in the power method. To overcome this drawback, a straightforward generalization of the power method can be used to compute an invariant subspace, which is spanned by a few dominant eigenvectors. Following the assumption in section 2.2.2.1, we suppose that the starting matrix with r orthogonal columns, $Q^{(0)}$, has the property that $X_i^T Q^{(0)} \neq 0$ for $i = 1, 2, \dots, r$. Under this assumption, a sequence of matrices with r orthogonal columns $\{Q^{(k)}\}$ can be obtained:

for $k=1, 2, \dots$

$$\begin{aligned}
 Y^{(k)} &= MQ^{(k-1)} \\
 Q^{(k)}R^{(k)} &= Y^{(k)} \quad (QR \text{ factorization}) \\
 \text{end}
 \end{aligned}$$

It has been proved in (Golub *et al.* 1996) that the subspace $\text{span}\{Q^{(k)}\}$ approaches $\text{span}\{X_1, X_2, \dots, X_r\}$ and the convergence ratio is $|\lambda_{r+1} / \lambda_r|$.

2.3 Matrix perturbation theory

Noise is inevitably introduced in almost all practical signals, for example, due to the measurement uncertainty. The central question in matrix perturbation theory is to characterize how a small variation (or more precisely, perturbation) produces an error in functions of that matrix. Two issues are usually addressed in perturbation theory: the development of a perturbation expansion and the calculation of a perturbation bound (Stewart *et al.* 1990). In this section, we concentrate on the former issue, the perturbation expansion, which will be used in the analysis of the performance of the SVD-based applications in noisy environments.

Only the perturbation expansion concerning *singular values/vectors* is needed in this thesis. However, we also include the perturbation expansion theory concerning the *eigenvalues/eigenvectors* as a useful way to arrive at our results. With our objective, though, in the eigenvalues/eigenvectors form, we need only consider *symmetric* matrices.

To our best knowledge*, the perturbation expansion of the eigenvectors/singular-vectors is available only for those that correspond to a *simple* eigenvalue or singular value (Wilkinson 1965; Stewart *et al.* 1990). In section 2.3.1, we review such theory; and present, in section 2.3.2, our new results for those that correspond to a *multiple* eigenvalue or singular value. In order to have a complete description of the perturbation theory, we give all the proofs, including those available in the textbooks (Wilkinson 1965; Stewart *et al.* 1990), plus those leading to our new results.

* Here, we'd like to express our appreciation to Prof. G. W. Stewart (Stewart *et al.* 1990), who, by private correspondence, pointed this out to us.

2.3.1 Perturbation theory

Here, we do not follow the notation in (Wilkinson 1965), where an arbitrarily small positive number, ε , was introduced. Because we only consider the first-order perturbation, a simpler and straightforward form works. For example, suppose \mathbf{M} has a simple eigenvalue λ , and the associated eigenvector is \mathbf{x} . If \mathbf{M} is corrupted with $\Delta\mathbf{M}$ and $\Delta\mathbf{M}$ is small enough, the first-order perturbations of the eigenvalue and the eigenvector, denoted as $\Delta\lambda$ and $\Delta\mathbf{x}$ respectively, will also be small enough, from Ostrowski's continuity theorem (Wilkinson 1965). Suppose their higher-order terms are $\delta\lambda$ and $\delta\mathbf{x}$, respectively. From $(\mathbf{M} + \Delta\mathbf{M})(\mathbf{x} + \Delta\mathbf{x} + \delta\mathbf{x}) = (\lambda + \Delta\lambda + \delta\lambda)(\mathbf{x} + \Delta\mathbf{x} + \delta\mathbf{x})$, we have the first-order perturbation, by dropping the higher-order terms:

$$\mathbf{M} \cdot \mathbf{x} + \mathbf{M} \cdot \Delta\mathbf{x} + \Delta\mathbf{M} \cdot \mathbf{x} \approx \lambda \cdot \mathbf{x} + \lambda \cdot \Delta\mathbf{x} + \Delta\lambda \cdot \mathbf{x} \quad (2.7)$$

Of course, this first-order perturbation is same as that in (Wilkinson 1965), despite the difference in notation.

Theory 1 (Wilkinson 1965): Consider a symmetric matrix, $\mathbf{M} \in R^{m,m}$. Suppose \mathbf{M} has m distinct eigenvalues, $\{\lambda_i | i = 1, 2, \dots, m\}$ and the corresponding eigenvectors are $\{\mathbf{x}_i | i = 1, 2, \dots, m\}$. If \mathbf{M} is perturbed by a matrix \mathbf{N} , the eigenvalues and the eigenvectors of $\mathbf{M} + \mathbf{N}$ are $\{\lambda'_i | i = 1, 2, \dots, m\}$ and $\{\mathbf{x}'_i | i = 1, 2, \dots, m\}$ respectively. Supposing every entry in \mathbf{N} is small enough, the first-order perturbations of eigenvalues and eigenvectors are:

$$\lambda'_i = \lambda_i + \beta_{i,i} \quad (2.8)$$

$$\mathbf{x}'_i = \mathbf{x}_i + \sum_{j \neq i} \frac{\beta_{j,i}}{\lambda_i - \lambda_j} \mathbf{x}_j \quad (2.9)$$

where $\beta_{i,j} = \mathbf{x}'_i{}^T \mathbf{N} \mathbf{x}_j$.

Proof: Suppose $\mathbf{x}'_i = \mathbf{x}_i + \sum_{j \neq i} c_{j,i} \mathbf{x}_j$ and $\lambda'_i = \lambda_i + b_i$. From the first-order perturbation, we have $\mathbf{M} \mathbf{x}_i + \mathbf{M} \sum_{j \neq i} c_{j,i} \mathbf{x}_j + \mathbf{N} \mathbf{x}_i \approx \lambda_i \mathbf{x}_i + \lambda_i \sum_{j \neq i} c_{j,i} \mathbf{x}_j + b_i \mathbf{x}_i$, and

$$\sum_{j \neq i} c_{j,i} (\lambda_j - \lambda_i) \mathbf{x}_j + \mathbf{N} \mathbf{x}_i = b_i \mathbf{x}_i \quad (2.10)$$

Because \mathbf{M} is symmetric and has m distinct eigenvalues, $\{\mathbf{x}_i\}$ are orthogonal to each other. Pre-multiplying (2.10) by \mathbf{x}_i^T , we obtain $b_i = \mathbf{x}_i^T \mathbf{N} \mathbf{x}_i = \beta_{i,i}$. Pre-multiplying \mathbf{x}_j^T , we have $c_{j,i} = \frac{\beta_{j,i}}{\lambda_i - \lambda_j}$.

Theorem 2 (Stewart *et al.* 1990): Suppose \mathbf{A} (not necessarily symmetric) is corrupted with \mathbf{N} and we observe \mathbf{B} : $\mathbf{B} = \mathbf{A} + \mathbf{N}$. According to the SVD theorem, we have $\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, where $\mathbf{U} \in O^{m \times m}$, $\mathbf{\Sigma} = \text{diag}\{\kappa_1, \kappa_2, \dots, \kappa_m\}$, $\mathbf{V} \in O^{m \times m}$. Define $\mathbf{C} = \mathbf{U}^T \mathbf{N} \mathbf{V}$. Suppose κ_i is a simple non-zero singular value of \mathbf{A} . Then, the first order perturbations of the singular values λ_i , the right singular vector \mathbf{x}_i , and the left singular vector \mathbf{y}_i , of \mathbf{B} are respectively

$$\lambda_i = \kappa_i + C_{i,i} \quad (2.11)$$

$$\mathbf{x}_i = \mathbf{V}_i + \sum_{j \neq i} \frac{\kappa_j C_{j,i} + \kappa_i C_{i,j}}{\kappa_i^2 - \kappa_j^2} \mathbf{V}_j \quad (2.12)$$

$$\mathbf{y}_i = \mathbf{U}_i + \sum_{j \neq i} \frac{\kappa_i C_{i,i} + \kappa_j C_{i,j}}{\kappa_i^2 - \kappa_j^2} \mathbf{U}_j \quad (2.13)$$

Proof: Suppose $\mathbf{\Omega} = \mathbf{\Sigma} + \mathbf{C}$. Obviously, $\{\kappa_j\}$ and $\{\mathbf{e}_j\}$ are respectively the singular values and the right/left singular vectors of $\mathbf{\Sigma}$. First, we prove that the singular values λ'_i , the right singular vectors \mathbf{x}'_i , and the left singular vectors \mathbf{y}'_i of $\mathbf{\Omega}$ are respectively

$$\lambda'_i = \kappa_i + C_{i,i} \quad (2.14)$$

$$\mathbf{x}'_i = \mathbf{e}_i + \sum_{j \neq i} \frac{\kappa_j C_{j,i} + \kappa_i C_{i,j}}{\kappa_i^2 - \kappa_j^2} \mathbf{e}_j \quad (2.15)$$

$$\mathbf{y}'_i = \mathbf{e}_i + \sum_{j \neq i} \frac{\kappa_i C_{i,i} + \kappa_j C_{i,j}}{\kappa_i^2 - \kappa_j^2} \mathbf{e}_j \quad (2.16)$$

Suppose $\lambda'_i = \kappa_i + \Delta \kappa_i$, $\mathbf{x}'_i = \mathbf{e}_i + \sum_{j \neq i} f_{j,i} \mathbf{e}_j$, and $\mathbf{y}'_i = \mathbf{e}_i + \sum_{j \neq i} g_{j,i} \mathbf{e}_j$

From the properties of SVD, we have $\mathbf{\Omega} \mathbf{x}'_i = \lambda'_i \mathbf{y}'_i$ and $\mathbf{\Omega}^T \mathbf{y}'_i = \lambda'_i \mathbf{x}'_i$. Equating their first order parts, we have:

$$\Sigma \mathbf{e}_i + \mathbf{C} \mathbf{e}_i + \Sigma \sum_{j \neq i} f_{j,i} \mathbf{e}_j \approx \kappa_i \mathbf{e}_i + \Delta \kappa_i \mathbf{e}_i + \kappa_i \sum_{j \neq i} g_{j,i} \mathbf{e}_j \quad (2.17)$$

$$\Sigma^T \mathbf{e}_i + \mathbf{C}^T \mathbf{e}_i + \Sigma^T \sum_{j \neq i} g_{j,i} \mathbf{e}_j \approx \kappa_i \mathbf{e}_i + \Delta \kappa_i \mathbf{e}_i + \kappa_i \sum_{j \neq i} f_{j,i} \mathbf{e}_j \quad (2.18)$$

Then

$$\mathbf{C} \mathbf{e}_i + \sum_{j \neq i} \kappa_j f_{j,i} \mathbf{e}_j = \Delta \kappa_i \mathbf{e}_i + \kappa_i \sum_{j \neq i} g_{j,i} \mathbf{e}_j \quad (2.19)$$

$$\mathbf{C}^T \mathbf{e}_i + \sum_{j \neq i} \kappa_j g_{j,i} \mathbf{e}_j = \Delta \kappa_i \mathbf{e}_i + \kappa_i \sum_{j \neq i} f_{j,i} \mathbf{e}_j \quad (2.20)$$

First, by equating \mathbf{e}_i , we have $\Delta \kappa_i = C_{i,i}$. And from \mathbf{e}_j ($j \neq i$),

$$\begin{cases} \kappa_i g_{j,i} - \kappa_j f_{j,i} = C_{j,i} \\ -\kappa_j g_{i,j} + \kappa_i f_{j,i} = C_{i,j} \end{cases} \quad (2.21)$$

$$\begin{cases} g_{j,i} = (\kappa_i C_{j,i} + \kappa_j C_{i,j}) / (\kappa_i^2 - \kappa_j^2) \\ f_{j,i} = (\kappa_j C_{j,i} + \kappa_i C_{i,j}) / (\kappa_i^2 - \kappa_j^2) \end{cases} \quad (2.22)$$

So far, (2.14-2.16) have been proved.

From

$$\mathbf{B} = \mathbf{U} \mathbf{\Omega} \mathbf{V}^T \approx \mathbf{U} [\mathbf{y}'_1, \dots, \mathbf{y}'_m] \text{diag}\{\lambda'_1, \dots, \lambda'_m\} [\mathbf{x}'_1, \dots, \mathbf{x}'_m]^T \mathbf{V}^T \quad (2.23)$$

\mathbf{B} has λ'_i , $\mathbf{V} \mathbf{x}'_i$, and $\mathbf{U} \mathbf{y}'_i$ respectively as its singular values, right and left singular vectors.

The perturbation theory above, concerning the singular values/vectors, holds only for positive (and especially only for significantly large) singular values (Stewart *et al.* 1990) (Note: singular values have to be non-negative.) In this thesis, only linear subspace analysis based applications are of concern, where the noise-free signal lies in a low-dimensional subspace, for example, of dimension r . In these rank- r problems, only the first r largest singular values are needed, where $r \ll m$. Thus, we do not have to consider the behavior of the perturbation for the zero (or near zero) singular values.

2.3.2 New perturbation theory, corresponding to a multiple eigenvalue/ singular value

In this section, we present our new result concerning the perturbation expansions, corresponding to the case where the matrix has at least one multiple eigenvalue/singular value.

First, we want to shed some light on the perturbation expansions concerning singular vectors that correspond to a multiple singular value. We do this by considering the perturbation expansions of the eigenvectors of a symmetric square matrix:

Theorem 3: Suppose $\mathbf{M} \in R^{m,m}$, $\mathbf{M} = \mathbf{M}^T$, and it has m eigenvalues $\{\lambda_i\}$ and m eigenvectors $\{x_i\}$, which are orthogonal to each other*. Without loss of generality, suppose the first k eigenvalues of \mathbf{M} are same, $\lambda_i = \lambda$ for $i = 1, 2, \dots, k$. \mathbf{M} is corrupted with \mathbf{N} , which, compared with \mathbf{M} , is small enough. Define $\mathbf{Q} = [x_1, \dots, x_m]^T \mathbf{N} [x_1, \dots, x_m]$. Then, the first-order perturbation of the first k eigenvalues and eigenvectors of $\mathbf{M} + \mathbf{N}$ are:

$$\lambda'_i = \lambda + \delta_i \quad (2.24)$$

$$x'_i = \sum_{j=1}^k S_{j,i} x_j + \sum_{j=k+1}^m \frac{Q'_{j,i}}{\lambda - \lambda_j} x_j \quad (2.25)$$

where δ_i (supposing $\delta_i \neq \delta_j$ if $i \neq j$) and $\mathbf{S}_i = [S_{1,i}, S_{2,i}, \dots, S_{k,i}]^T$ are the eigenvalues and eigenvectors of $\mathbf{Q}_{1:k,1:k}$ respectively, i.e.

$\mathbf{Q}_{1:k,1:k} = \mathbf{S} \text{diag}\{\delta_1, \dots, \delta_k\} \mathbf{S}^{-1}$ and $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_k]$. Define

$\mathbf{Q}' = \begin{bmatrix} \mathbf{S}^{-1} & \\ & \mathbf{I}_{m-k} \end{bmatrix} \mathbf{Q} \begin{bmatrix} \mathbf{S} & \\ & \mathbf{I}_{m-k} \end{bmatrix}$. The other $m-k$ eigenvalues/eigenvectors can be

obtained as in theorem 1.

* For an r -ple multiple eigenvalue, we, first, have its r eigenvectors, $\{x_i | i = 1, \dots, r\}$, which may not be orthogonal. Then, the r orthogonal eigenvectors can be obtained by applying Schmidt orthogonalization on $\{x_i | i = 1, \dots, r\}$.

Proof: From the perturbation expansion about the eigenvectors corresponding to a multiple eigenvalue (Wilkinson 1965), we can suppose $x'_i = \sum_{j=1}^k c_{j,i} x_j + \sum_{j=k+1}^m f_{j,i} x_j$ and $\lambda'_i = \lambda + \Delta\lambda_i$. Note: $c_{j,i}$ are different from $f_{j,i}$. $c_{j,i}$ can possibly take any value within $[0,1]$, while $f_{j,i}$ approach zeroes if N is small enough.

$$(M + N)x'_i = \lambda'_i x'_i \quad (2.26)$$

Equating the first order parts:

$$M \sum_{j=1}^k c_{j,i} x_j + M \sum_{j=k+1}^m f_{j,i} x_j + N \sum_{j=1}^k c_{j,i} x_j \approx \lambda \sum_{j=1}^k c_{j,i} x_j + \lambda \sum_{j=k+1}^m f_{j,i} x_j + \Delta\lambda_i \sum_{j=1}^k c_{j,i} x_j \quad (2.27)$$

Then

$$\sum_{j=k+1}^m \lambda_j f_{j,i} x_j + [x_1, \dots, x_m] Q_{1:m,i:k} c_i \approx \lambda \sum_{j=k+1}^m f_{j,i} x_j + \Delta\lambda_i \sum_{j=1}^k c_{j,i} x_j \quad (2.28)$$

where $c_i = [c_{1,i}, c_{2,i}, \dots, c_{k,i}]^T$. Equating the coefficients of x_j for $(j = 1, \dots, k)$, we have

$$Q_{1:k,i:k} c_i = \Delta\lambda_i c_i \quad (2.29)$$

where $Q_{1:k,i:k}$ is the left-up $k \times k$ submatrix of Q . If $Q_{1:k,i:k}$ has k distinct eigenvalues, the solution of $\Delta\lambda_i$ and c_i is unique, as in (2.29). Obviously, c is the same as S , as defined in the theorem. After substituting $\Delta\lambda_i$ and c_i in (2.28), the equality of x_j for $(j = k+1, \dots, m)$ produces the first order perturbations of $f_{j,i}$ as in the theorem.

Following the same notation as used in theorem 2, we consider the perturbation expansion, where the matrix has at least one multiple *singular* value. This result appears to be a new one.

Theorem 4: A , B , C and Σ are defined as those in theorem 2. Define $\Omega = C + \Sigma$. Without loss of generality, suppose the first k singular values of A are the same: $\{\kappa_i = \kappa \mid i = 1, \dots, k\}$. By SVD, $\Omega_{1:k,1:k} = FSE^T = F \text{diag}\{S_1, \dots, S_k\} E^T$. Let

$$U' = \begin{bmatrix} F & \\ & I_{m-k} \end{bmatrix}, \quad V' = \begin{bmatrix} E & \\ & I_{m-k} \end{bmatrix}, \quad \text{and } \Omega' = U'^T \Omega V'.$$

$$\mathbf{B} = (\mathbf{U}\mathbf{U}')\mathbf{\Omega}'(\mathbf{V}\mathbf{V}')^T \quad (2.30)$$

The first order perturbation of the singular values, $\{\lambda'_i\}$, right singular vectors $\{\mathbf{x}'_i\}$, and left singular vectors $\{\mathbf{y}'_i\}$ for $0 \leq i \leq k$, of $\mathbf{\Omega}'$ are respectively

$$\lambda'_i = \Omega'_{i,i} = S_i \quad (2.31)$$

$$\mathbf{x}'_i = \mathbf{e}_i + \sum_{j=k+1}^m \frac{\kappa_j \Omega'_{j,i} + \kappa \Omega'_{i,j}}{\kappa^2 - \kappa_j^2} \mathbf{e}_j \quad (2.32)$$

$$\mathbf{y}'_i = \mathbf{e}_i + \sum_{j=k+1}^m \frac{\kappa \Omega'_{j,i} + \kappa_j \Omega'_{i,j}}{\kappa^2 - \kappa_j^2} \mathbf{e}_j \quad (2.33)$$

From (2.30), $\{\lambda'_i\}$ are also the first k singular values of \mathbf{B} , and, the right singular vectors $\{\mathbf{x}_i\}$ and left singular vectors $\{\mathbf{y}_i\}$ of \mathbf{B} are respectively: $\{\mathbf{V}\mathbf{V}'\mathbf{x}'_i\}$ and $\{\mathbf{U}\mathbf{U}'\mathbf{y}'_i\}$. The perturbations, corresponding to other non-zero simple singular values, can be obtained as in theorem 2.

Proof: Let $\mathbf{\Omega}$ has $\{\lambda''_i\}$, $\{\mathbf{x}''_i\}$ and $\{\mathbf{y}''_i\}$ as its first k singular values, right singular vectors and left singular vectors respectively. For $i > k$, $\{\lambda''_i\}$, $\{\mathbf{x}''_i\}$ and $\{\mathbf{y}''_i\}$ can be obtained as in theorem 2. Thus, we concentrate on the first-order perturbation of $\{\lambda''_i\}$, $\{\mathbf{x}''_i\}$ and $\{\mathbf{y}''_i\}$, for $i \leq k$.

First, we only consider one singular value and the corresponding singular vectors. Combining the techniques in the proof of theorem 2 and theorem 3, we assume that the first-order perturbations of the right and the left singular vectors, \mathbf{x}'' and \mathbf{y}'' respectively, have the following forms:

$$\mathbf{x}'' = \sum_{i=1}^k p_i \mathbf{e}_i + \sum_{i=k+1}^m q_i \mathbf{e}_i \quad (2.34)$$

$$\mathbf{y}'' = \sum_{i=1}^k f_i \mathbf{e}_i + \sum_{i=k+1}^m g_i \mathbf{e}_i \quad (2.35)$$

Note: p_i and f_i can possibly take any value within $[0,1]$, while q_i and g_i approach zeroes if N is small enough. From the continuity of the eigenvalues of $\mathbf{M}\mathbf{M}^T$, the singular values of \mathbf{M} also obey Ostrowski's continuity rule (Wilkinson 1965), because the singular values of the matrix, \mathbf{M} , are the square roots of the

eigenvalues of \mathbf{MM}^T . Supposing the corresponding singular value is $\lambda'' = \kappa + \Delta\kappa$, equality of the first order parts of $\mathbf{\Omega}\mathbf{x}'' = \lambda''\mathbf{y}''$ and $\mathbf{\Omega}^T\mathbf{y}'' = \lambda''\mathbf{x}''$ produces:

$$\kappa \sum_{i=1}^k p_i \mathbf{e}_i + \sum_{i=k+1}^m q_i \kappa_i \mathbf{e}_i + \sum_{i=1}^k p_i \mathbf{C}_i = (\kappa + \Delta\kappa) \sum_{i=1}^k f_i \mathbf{e}_i + \kappa \sum_{i=k+1}^m g_i \mathbf{e}_i \quad (2.36)$$

$$\kappa \sum_{i=1}^k f_i \mathbf{e}_i + \sum_{i=k+1}^m g_i \kappa_i \mathbf{e}_i + \sum_{i=1}^k f_i (\mathbf{C}^T)_i = (\kappa + \Delta\kappa) \sum_{i=1}^k p_i \mathbf{e}_i + \kappa \sum_{i=k+1}^m q_i \mathbf{e}_i \quad (2.37)$$

From (2.36) and (2.37), we have, by equating \mathbf{e}_s (for $s = 1, \dots, k$):

$$\kappa p_s + \sum_{i=1}^k p_i C_{s,i} = (\kappa + \Delta\kappa) f_s \quad (2.38)$$

$$\kappa f_s + \sum_{i=1}^k f_i C_{i,s} = (\kappa + \Delta\kappa) p_s \quad (2.39)$$

In matrix form, they are:

$$(\mathbf{C}_{1:k,1:k} + \kappa \mathbf{I}) \mathbf{p} = (\kappa + \Delta\kappa) \mathbf{f} \quad (2.40)$$

$$(\mathbf{C}_{1:k,1:k}^T + \kappa \mathbf{I}) \mathbf{f} = (\kappa + \Delta\kappa) \mathbf{p} \quad (2.41)$$

where $\mathbf{C}_{1:k,1:k}$ is the left-up k by k submatrix of \mathbf{C} , $\mathbf{p} = [p_1, p_2, \dots, p_k]^T$ and $\mathbf{f} = [f_1, f_2, \dots, f_k]^T$. Note that $\mathbf{Q}_{1:k,1:k} = \mathbf{C}_{1:k,1:k} + \kappa \mathbf{I}_k$. Obviously, $\kappa + \Delta\kappa$, \mathbf{p} and \mathbf{f} are respectively the singular value, the right and the left singular vectors of $\mathbf{C}_{1:k,1:k} + \kappa \mathbf{I}_k$; and \mathbf{p} and \mathbf{f} correspond to the columns of \mathbf{E} and \mathbf{F} in the theorem. $\mathbf{C}_{1:k,1:k} + \kappa \mathbf{I}_k$ just has k singular values, right and left singular vectors, which correspond to $\{\lambda_i''\}$, $\{\mathbf{x}_i''\}$ and $\{\mathbf{y}_i''\}$, for $i \leq k$, of $\mathbf{\Omega}$.

Equating the \mathbf{e}_i in (2.36) and (2.37), for $i > k$, we have

$$\sum_{l=1}^k C_{l,i} p_l = \kappa g_i - \kappa_i q_i \quad (2.42)$$

$$\sum_{l=1}^k C_{l,i} f_l = \kappa q_i - \kappa_i g_i \quad (2.43)$$

Note for $i > k$ and $l \leq k$, $C_{l,i} = \Omega_{l,i}$ and $C_{i,l} = \Omega_{i,l}$. Considering the i^{th} (for $1 \leq i \leq k$) singular value:

$$\kappa g_i - \kappa_i q_i = \sum_{j=1}^k \Omega_{i,j} p_{j,i} = \Omega'_{i,i} \quad (2.44)$$

$$\kappa q_i - \kappa_i g_i = \sum_{j=1}^k \Omega_{j,i} f_{j,i} = \Omega'_{i,i} \quad (2.45)$$

Combining (2.40, 2.41) and (2.44,2.45),

$$x'' = \sum_{i=1}^k p_i e_i + \sum_{j=k+1}^m \frac{\kappa_j \Omega'_{j,i} + \kappa_j \Omega'_{i,j}}{\kappa_i^2 - \kappa_j^2} e_j \quad (2.46)$$

$$y'' = \sum_{i=1}^k f_i e_i + \sum_{j=k+1}^m \frac{\kappa_j \Omega'_{j,i} + \kappa_j \Omega'_{i,j}}{\kappa_i^2 - \kappa_j^2} e_j \quad (2.47)$$

are respectively the right and left singular vectors of Ω . After the system transformation as (2.30), the first k left/right singular vectors of Ω' are as defined in (2.32, 2.33).

2.4 Conclusion

In this chapter, we reviewed some requisite knowledge about matrix theory: including the SVD, and perturbation expansion theory in relation to the singular values and singular vectors (the eigenvalues and eigenvectors). The SVD theorem will be used throughout the rest of this thesis; yet the perturbation theory will be used only in chapter 4, which is the core contribution of this thesis (however, the results in chapter 4 will be employed in other chapters, such as chapter 5 and chapter 8).

The essential contribution of this chapter is that we present the *new* perturbation expansion theory for a multiple singular value (eigenvalue).

Chapter 3

The factorization method in SFM and PCA-based face recognition

In this chapter, we review the factorization method for the problem of structure from motion (SFM) and PCA-based face recognition. In these two tasks, the singular value decomposition (SVD) is the major tool, to calculate the motion matrix and structure matrix up to a non-singular transformation in SFM; or to calculate the eigenfaces from a few training face images. In section 3.1, we provide an overview of the development of the factorization method for the problem of Structure from motion. In section 3.2, we review the seminal work on the factorization method (Tomasi *et al.* 1992). In section 3.3, a brief description of the development of the PCA-based face recognition is given.

3.1 Overview of the development of the factorization method for SFM

The original work by Tomasi and Kanade (Tomasi *et al.* 1992) restricted itself to the orthographic setting. The method was extended to the paraperspective setting (Poelman *et al.* 1997). Triggs and Sturm applied the factorization method to the projective setting (Sturm *et al.* 1996; Triggs 1996).

The advantage of the factorization method for SFM can be ascribed to the SVD's denoising capacity, as will be analyzed in chapter 4. It states that, as the size of the matrix increases, the low-rank approximation matrix approaches the noise-free matrix. That is the underlying superiority of the factorization method: all the feature points are treated uniformly so that most of the noise can be suppressed if the size of the measurement matrix is large enough.

That is also the reason that the factorization method has attracted much attention from the computer vision community.

- The method has been extended to lines, conics under affine setting (Quan *et al.* 1996; Quan *et al.* 1997; Kahl *et al.* 1998; Kahl *et al.* 1999), and to planes under projective settings (Rother *et al.* 2002).
- In (Aguiar *et al.* 1999; Aguiar *et al.* 2003), it was shown that the factorization method can be reduced to a rank 1 problem.
- In (Morita *et al.* 1997), a sequential algorithm has been proposed to deal with the incoming frames, with little loss of the precision.
- The method has been extended to the case of the multi-body factorization (Costeira *et al.* 1995; Costeira *et al.* 1998; Gear 1998; Kanatani 2001) *etc.*
- The related rank 4 constraint has been employed in the optical flow computation (Irani 1999; Irani 2002).
- Scalar-weighted factorization has been studied in (Aguiar *et al.* 1999; Aguiar *et al.* 2000; Aguiar *et al.* 2003). Another more general setting addresses the directional uncertainty (Morris *et al.* 1998; Irani *et al.* 2000; Anandan *et al.* 2002).
- A few papers have appeared that deal with the missing-data problem in the factorization method (Shum *et al.* 1995; Jacobs 1997; Kahl *et al.* 1999; Jacobs 2001; Brandt 2002; Rother *et al.* 2002; Brand 2003; Guerreiro *et al.* 2003; Chen *et al.* 2004). We address this issue in chapter 5.
- A few attempts have been made to optimize the projective factorization (Sturm *et al.* 1996; Heyden 1997; Chen *et al.* 1999; Oliensis 1999; Mahamud *et al.* 2001; Mahamud *et al.* 2003).
- Robust statistics has been introduced in the context of the factorization method (Huynh *et al.* 2001; Aanaes *et al.* 2002).

3.2 The factorization method under orthographic settings

The critical idea about the factorization method (Tomasi *et al.* 1992) for the problem of structure from motion is that the measurement matrix is of rank 4 if it is noise free. From this property, the motion matrix and the structure matrix can be obtained by the SVD, up to an affine transformation. In the orthographic setting, the normalization constraints between the camera axes (row vectors of the motion matrix) are employed to solve the affine ambiguity of the motion/structure matrix.

3.2.1 Measurement matrix and its rank 4 property

Suppose that P feature points of a rigid object are observed in an F -view sequence by a moving camera (the rank 4 property still holds for the cases of a moving object by a static camera and both moving). Let $(x_{f,p}, y_{f,p})$ be the image position of the p^{th} feature point in the f^{th} frame. Then, the measurement matrix $\mathbf{W} \in R^{2F,P}$ is arranged as:

$$\mathbf{W} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,P} \\ \vdots & & \vdots \\ x_{F,1} & \cdots & x_{F,P} \\ y_{1,1} & \cdots & y_{1,P} \\ \vdots & & \vdots \\ y_{F,1} & \cdots & y_{F,P} \end{bmatrix} \quad (3.1)$$

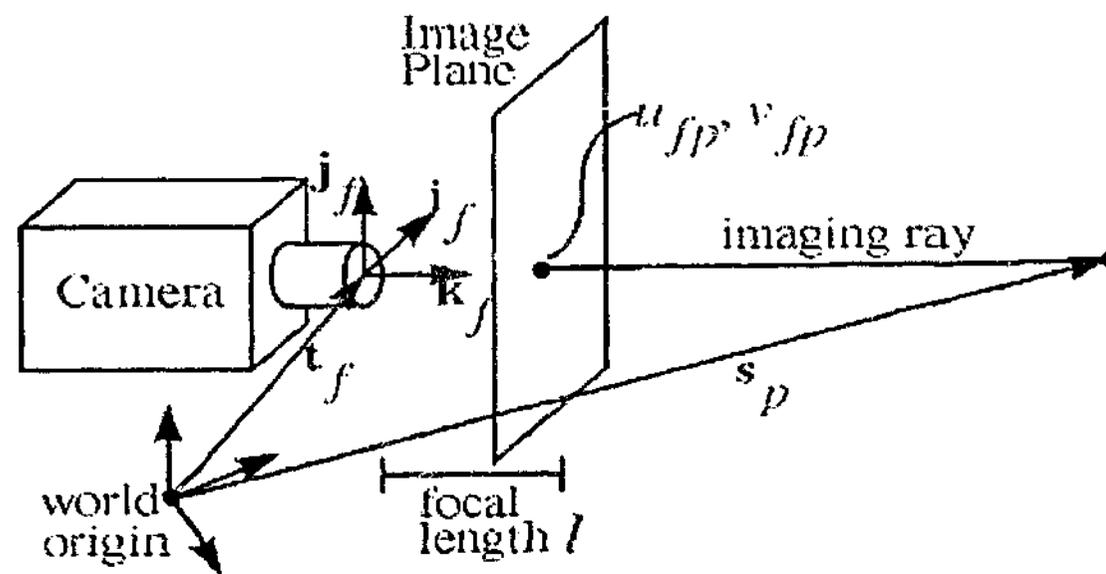


Figure 3.1: The coordinate systems: world system and camera system. (Excerpted from (Poelman *et al.* 1997))

Suppose the p^{th} feature point is represented by a vector $S_p = [x_p, y_p, z_p]^T$. And, suppose the camera orientation at frame f is represented by the orthonormal vectors i_f , j_f and k_f , and its focal center in the world coordinate is represented by t_f , as shown in Figure 3.1. Assuming an orthographic projection, we have:

$$\begin{cases} x_{f,p} = i_f \bullet (s_p - t_f) \\ y_{f,p} = j_f \bullet (s_p - t_f) \end{cases} \quad (3.2)$$

In matrix form, we have:

$$W = MS + T[1,1,\dots,1] \quad (3.3)$$

where

$$M = \begin{bmatrix} i_1^T \\ \vdots \\ i_F^T \\ j_1^T \\ \vdots \\ j_F^T \end{bmatrix} \quad (3.4)$$

$$S = [S_1, S_2, \dots, S_p] \quad (3.5)$$

$$T = [-i_1 \bullet t_1, \dots, -i_f \bullet t_f, -j_1 \bullet t_1, \dots, -j_f \bullet t_f]^T \quad (3.6)$$

If we place the origin of the world coordinate at the mass center of the object, the *registered measurement matrix* has the following form:

$$\tilde{W} = MS \quad (3.7)$$

where $\tilde{W} = W - \bar{w}[1,1,\dots,1]$ with $\bar{w} \in R^{2F,3}$ and $\bar{w}_k = \sum_{i=1}^P W_{k,i} / P$.

From (3.7), the registered measurement matrix has a rank of 3 at most, because M has 3 columns and S has 3 rows. This is the rank theorem in (Tomasi *et al.* 1992). Note, the measurement matrix usually has a rank of 4 because there exists the translation vector T , as in (3.3). Precisely, the measurement matrix lies in a rank-4 affine subspace.

By the SVD, the registered measurement matrix can be approximated by a rank 3 matrix: $\tilde{W}^3 = U\Sigma V^T$, where $U \in O^{2F,3}$, $V \in O^{P,3}$ and $\Sigma = \text{diag}\{\sigma_1, \sigma_2, \sigma_3\}$. At

this stage, the motion and structure matrices can be determined to an affine transformation:

$$\begin{cases} \hat{\mathbf{M}} = \mathbf{U}\Sigma^{1/2} \\ \mathbf{S} = \Sigma^{1/2}\mathbf{V}^T \end{cases} \quad (3.8)$$

where $\Sigma^{1/2} = \text{diag}\{\sqrt{\sigma_1}, \sqrt{\sigma_2}, \sqrt{\sigma_3}\}$.

3.2.2 Metric transformation

The decomposition of (3.8) is not unique, because the registered measurement matrix can be factorized into any other pairs $\{\hat{\mathbf{M}}\mathbf{A}, \mathbf{A}^{-1}\hat{\mathbf{S}}\}$, with a non-singular matrix \mathbf{A} . The correct \mathbf{A} can be determined by using the normalization constraints:

$$\begin{cases} \|\mathbf{i}_f\| = \|\mathbf{j}_f\| = 1 \\ \mathbf{i}_f \bullet \mathbf{j}_f = 0 \end{cases} \quad (3.9)$$

Suppose

$$\mathbf{L} = \mathbf{A}\mathbf{A}^T = \begin{bmatrix} l_1 & l_2 & l_3 \\ l_2 & l_4 & l_5 \\ l_3 & l_5 & l_6 \end{bmatrix} \quad (3.10)$$

The normalization constraints (3.9) can be rephrased in the following matrix form:

$$\mathbf{G}\mathbf{l} = \mathbf{c} \quad (3.11)$$

where

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}^T(\mathbf{i}_1, \mathbf{i}_1) \\ \vdots \\ \mathbf{g}^T(\mathbf{i}_F, \mathbf{i}_F) \\ \mathbf{g}^T(\mathbf{j}_1, \mathbf{j}_1) \\ \vdots \\ \mathbf{g}^T(\mathbf{j}_F, \mathbf{j}_F) \\ \mathbf{g}^T(\mathbf{i}_1, \mathbf{j}_1) \\ \vdots \\ \mathbf{g}^T(\mathbf{i}_F, \mathbf{j}_F) \end{bmatrix}_{3F \times 6}, \quad \mathbf{l} = \begin{bmatrix} l_1 \\ \vdots \\ l_6 \end{bmatrix}_{6 \times 1} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{2F \times 1} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{F \times 1} \end{bmatrix}_{3F \times 1} \quad (3.12)$$

$$\mathbf{g}^T(\mathbf{a}, \mathbf{b}) = [a_1b_1, a_1b_2 + a_2b_1, a_1b_3 + a_3b_1, a_2b_2, a_2b_3 + a_3b_2, a_3b_3] \quad (3.13)$$

\mathbf{l} can be obtained by the pseudo-inverse method:

$$\mathbf{l} = \mathbf{G}^+ \mathbf{c} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{c} \quad (3.14)$$

From (3.10), the matrix L has been obtained, and consequently, the Cholesky decomposition (Golub *et al.* 1996) of L gives A .

Note, the metric transformation A , obtained above, is not unique either, because any other $A' = AR$ with $RR^T = I_3$ still satisfies the normalization constraints (3.9). Thus, by the metric transformation, we obtain the motion and structure matrices up to a rotation transformation.

3.3 PCA-based face recognition

Another particularly active area of computer vision research, also employing subspace analysis, is that of PCA-based face recognition* (Turk *et al.* 1991; Hallinan 1994; Eipstein *et al.* 1995). A human face, in typical applications, must be recognized despite illumination changes between the target image (to be recognised) and the database of candidate images. It has been observed that: "the variations between the images of the same face due to illumination and viewing direction are almost larger than image variations due to change in face identity" (Moses *et al.* 1994). The issue of large illumination effects makes the problem of face recognition challenging (Belhumeur *et al.* 1997; Shashua 1997; Belhumeur *et al.* 1998; Georghiades *et al.* 1998; Jacobs *et al.* 1998; Georghiades *et al.* 2001).

In order to tackle this issue, PCA has been utilized to model the lighting variation in images; because it has been proved, experimentally (Hallinan 1994; Murase *et al.* 1994; Nene *et al.* 1994; Eipstein *et al.* 1995; Murase *et al.* 1995; Yuille *et al.* 1999) and theoretically (Basri *et al.* 1999; Ramamoorthi *et al.* 2001; Ramamoorthi 2002; Basri *et al.* 2003), that the possible images of the same Lambertian object,

* Here, we have to clarify the difference between the common PCA (Turk *et al.* 1991; Hallinan 1994; Eipstein *et al.* 1995) and linear subspace analysis (Belhumeur *et al.* 1997; Basri *et al.* 1999; Basri *et al.* 2003). In face recognition and related applications, several terminologies, like PCA (Turk *et al.* 1991), eigenface (Turk *et al.* 1991) and eigenimage (Hallinan 1994; Eipstein *et al.* 1995), have been used for such dimensionality reduction techniques. PCA (Turk *et al.* 1991; Hallinan 1994; Eipstein *et al.* 1995) works on the correlation matrix, where the mean of the images was first subtracted. While, in linear subspace analysis, we work directly on the original data (Belhumeur *et al.* 1997; Basri *et al.* 1999; Basri *et al.* 2003), without subtracting their mean. Recently, some theoretical analysis and experimental result prove that better performance can be obtained directly by using the linear subspace analysis, without subtracting the mean. In section 4.4, we analyse the performance of the linear subspace analysis, without subtracting the mean (Ramamoorthi 2002).

under different lighting conditions, approximately concentrate in a low-dimensional subspace, although the dimension of the image set for an object is “equal to the number of distinct surface normals” (Belhumeur *et al.* 1998).

- Experimental observations (Turk *et al.* 1991; Hallinan 1994; Eipstein *et al.* 1995) have helped firmly establish that the images of the same face, produced under different lighting conditions, approximately lie in a low-dimensional subspace.
- Similar approaches can be used in general object recognition and pose determination systems. A particularly influential example of such was the SLAM system (Murase *et al.* 1994; Nene *et al.* 1994; Murase *et al.* 1995), which captured the variations due to pose and illumination by a 20-dimensional (or less) subspace.
- Recently, it was proved, by using spherical harmonics, that “all Lambertian reflectance functions obtained with arbitrary distant light sources lie in close to a 9D linear subspace”: Basri and Jacobs (Basri *et al.* 1999; Basri *et al.* 2003) and Ramamoorthi and Hanrahan (Ramamoorthi *et al.* 2001; Ramamoorthi 2002).

Closely related to the second part of Chapter 4 is the generative model for learning object shape and albedo from multiple images (Yuille *et al.* 1999). In (Yuille *et al.* 1999), a few images, under different illumination conditions, can be employed to calculate the eigenvectors (or a low-dimensional subspace) by the SVD (Golub *et al.* 1996). Moreover, the surface consistency constraint is employed to reconstruct the surface up to a generalized bas relief (GBR) ambiguity (Belhumeur *et al.* 1999). The GBR reconstruction is beyond the scope of this thesis. In Chapter 4, we will analyze the noise effect on the subspace learning process.

Chapter 4

Analysis of noise effect in the SVD-based applications

As mentioned in chapter 1, linear subspace analysis (LSA) has become rather ubiquitous in the solution of a wide range of problems arising in pattern recognition and computer vision. The essence of these approaches is that certain structures are intrinsically (or approximately) low dimensional: for example, the factorization approach to the problem of structure from motion (SFM) and principal component analysis (PCA)-based approach to face recognition, as overviewed in chapter 3. In LSA, the singular value decomposition (SVD) is usually the basic mathematical tool. However, researchers have rather blindly used a SVD, without knowing the essential characteristics of its performance in the noise-corrupted environment. With the help of matrix perturbation theory, we present such an analysis here. First, the “*denoising capacity*” of the SVD is analysed. Second, we study the “*learning capacity*” of the LSA-based recognition system in a noise-corrupted environment. These results should help one to design more optimal systems in computer vision, particularly in tasks, such as SFM and face recognition. Our analysis agrees with certain observed phenomenon, and these observations, together with our simulations, verify the correctness of our theory.

A direct application is that we clarify some issues regarding an optimal learning strategy for face recognition. Further application of the theory in this chapter can be found in chapters 5 and 8.

This chapter is structured as follows. In section 4.1, we raise the questions concerning the noise in LSA. In section 4.2, we first present our results about this subject. In sections 4.3 and 4.4, our results are justified, with the matrix perturbation theory in chapter 2 as the main tool. In section 4.5, some simulation results are presented to testify to the correctness of our results and we explain some phenomena, observed by other researchers.

4.1 Issues to be addressed

Linear subspace analysis has found applications in many problems in computer vision and pattern recognition, where the high-dimensional representations of certain structures are intrinsically (or approximately) low dimensional. In this chapter we focus on two very prominent problems: Structure from Motion (SFM), and PCA-based face recognition, as overviewed in Chapter 3, although a whole host of other computer vision and pattern recognition tasks fall within the framework of our analysis.

4.1.1 Noise Effects

Despite such a plethora of applications where one expects, in principle, the measurements to be of low rank; it is widely understood that noise is inevitably introduced in the data. In the presence of noise, the matrix in question quickly becomes full rank. Thus, the matrix has to be fitted to its closest low-rank approximation. The SVD gives the best solution to this problem (Golub *et al.* 1996), measured by the Frobenius norm and 2-norm. The result is guaranteed to be optimal (Press *et al.* 1992) if the noise is i.i.d. Gaussian. Not surprisingly, therefore, the SVD has become a widely used tool. For example, the factorization method (Tomasi *et al.* 1992; Poelman *et al.* 1997) achieves a *Maximum Likelihood* affine reconstruction from multiple (>2) views, as pointed out in (Reid *et al.* 1996; Hartley *et al.* 2000).

From a related point of view, the low-rank approximation can be regarded as a “denoising” tool, where we refer to the measure of the sum of squared difference (SSD)* between the noise-corrupted matrix (or the “denoised” matrix) and the noise-free matrix. Compared with a noisy matrix that is always of full rank, its low-rank approximation matrix, obtained by SVD, is always closer to the noise-free matrix, i.e. the underlying ground truth. For example, the multiview subspace constraint was utilized to improve the accuracy of recovered homographies, especially for those that have small regions (Zelnik-Manor *et al.* 1999; Zelnik-Manor *et al.* 2002).

* In image denoising, we usually use the terminology of mean square error (MSE).

Thus, linear subspace approximation is sometimes a model simplification and sometimes a denoising process (and often both, simultaneously).

4.1.2 Performance Questions

4.1.2.1 Denoising capacity of SVD

Although SVD is widely employed to fit a large matrix to its low-dimensional subspace, little work has been done to analyze the performance of SVD in such noise-corrupted cases. It is well known (Golub *et al.* 1996) that one can, by SVD, obtain the best solution to the low-rank approximation, measured by 2-norm or Frobenius-norm. However, the meaning of “optimality” in the context of the noise-corrupted matrix is that the rank- r approximation matrix obtained by the SVD is the closest rank- r matrix to the noise-corrupted matrix. We do not know the capacity of SVD for separating the signal from the noise. Supposing the noise level is small enough, how much signal is retained by keeping the largest r components? Or, how much noise has been reduced by discarding the other components? In this sense, we are *blindly* using a SVD, *without knowing its denoising capacity*: how close is the low-rank approximation matrix to the noise-free matrix, or how close is the SVD-based subspace to the ground-truth subspace.

The lack of such performance analysis impedes the careful design of optimal systems. A natural issue arising is how to characterize the achieved accuracy with the growth in data (in the SFM context, this can be either through a growth in the number of frames analyzed, or by a growth in the number of features tracked). In the factorization approach to SFM, it is widely accepted that more frames produce more accurate result than a few (“few” typically being little more than 3) frames. It was even claimed (Thomas *et al.* 1999) that the 3D scene could be reconstructed to arbitrary accuracy given enough frames. *However, what is the gain of adding the data from one extra frame to a very large measurement matrix? What happens as the number of the frames approaches infinity? Can the 3D scene be truly reconstructed with arbitrary accuracy? Can such arbitrary accuracy be achieved only by the increase of the frames (while the number features do not increase)? Is an increase in the number of frames the most efficient way to obtain an increase in accuracy?*

In the example of SFM, as suggested above we can also possibly augment the number of feature points, or we can augment the number frames, or we can do both: i.e., both the row and the column of the matrix can grow towards the infinite in size. However, in a related problem, the matrix consisting of the homographies over two views, is restricted to a class of $m \times 9$ matrices (Zelnik-Manor *et al.* 1999; Zelnik-Manor *et al.* 2002). Such a matrix can only "grow" in one dimension, not both. We introduce some terminology to describe this difference: We call the matrix potentially-double-infinite if it has infinite rows and columns, and potentially-single-infinite for those who has constant rows (columns) and possibly infinite columns (rows). This raises another question: *What is the difference between these two types of matrices in terms of the precision that can be achieved?*

In summary, the first aim of this chapter is to analyze the *denoising* capacity of SVD, i.e., to identify the error that still resides in the low-rank approximation matrix and how this error relates to the growth of additional data.

4.1.2.2 Learning capacity of linear subspace analysis

Different questions, to those posed above, arise from the face recognition applications (including the object recognition and pose determination, and related applications). In the PCA-based face recognition approach, the eigenimage representation relies on a compact approximation of the large image database (or "training" set), by spanning this set (approximately) with a few orthogonal basis images. Such an approach attempts to capture and characterize the essential object or face features, and their variations in appearance under lighting and pose changes. Although the "illumination cone" (Belhumeur *et al.* 1998) (see also (Zhao *et al.* 1999)) can be obtained by as little as three images, the result is usually not accurate enough. Firstly, there is inevitably some noise in the images, like quantization error. Secondly, it is difficult to satisfy the conditions in proposition 3 in that paper (Belhumeur *et al.* 1998). Even if we can have three distinct light sources that can shed light on all the points of the surface, we cannot, in practice, exclude other light sources that cause attached or cast shadows on the subject. These considerations, plus (general) noise, have generally resulting in researchers trying to "learn" the eigenimages by a data reduction step applied to many "learning samples". Thus,

many learning samples were needed to produce a good basis, for example, 66 images were used for one object (Belhumeur *et al.* 1997). *What is the relationship between the learning capacity and the size of the learning samples?* Note, the learning process will be explained in section 4.4, and a more detailed description of such learning processes can be found in (Turk *et al.* 1991; Hallinan 1994; Belhumeur *et al.* 1997).

Understanding the error, still residing in the basis images, will hopefully help to design the recognition system. Accurate basis images are desired because the recognition algorithm relies on projecting the test image, to be identified, on the basis images. Note that the test image itself contains noise. Thus the noise in the LSA-based recognition system comes from two sources: one from the basis and the other from the test image. Do these two types of noise interfere with each other?

The second aim of this chapter is to present some theoretical analysis of the learning capacity of LSA-based recognition systems. Specifically, the error (measured by the sum of squared differences – SSD) of the LSA-based recognition system can be separated into two parts: one from the basis and the other from the test image, and we obtain some analytical results about their effects on the performance of the recognition system. We show that it is possible, theoretically, to design the optimal recognition system if we know the expectation of the test images.

4.2 Major results

4.2.1 Major results

Here, we present the major results of this chapter, by which we can answer the questions in the introduction. The justification of these results will be deferred until section 4.3 and section 4.4.

Result 1 (*Denoising capacity of SVD*): Suppose a matrix $A \in \mathbb{R}^{m,n}$ lies in a low-dimensional, r , subspace. It is corrupted by i.i.d. Gaussian noise producing another

matrix \mathbf{B} , which is directly observed. Then, the error that still resides in the rank- r approximation matrix, \mathbf{B}^r , is

$$E | B_{i,j}^r - A_{i,j} | = \sigma \sqrt{\frac{r(m+n) - r^2}{mn}} \quad (4.1)$$

if the noise level σ , compared with the signal level, is small enough. Specially, as $m, n \rightarrow \infty$, the rank- r approximation of \mathbf{B} approaches \mathbf{A} , i.e. $\mathbf{B}^r \rightarrow \mathbf{A}$; and if $n \equiv k$ ($k \geq r$) and $m \rightarrow \infty$,

$$E | B_{i,j}^r - A_{i,j} | \rightarrow \sigma \sqrt{\frac{r}{k}} \quad (4.2)$$

Result 2 (Learning capacity of LSA): For a rank- r LSA-based recognition system, the "error measure" (the SSD) comes from two independent sources: the noise in the basis images and the noise in the test image. Specifically, the expectation of the SSD, over the learning samples, is:

$$(m-r)\sigma_t^2 + (m-r)r\sigma_l^2/n \quad (4.3)$$

where m is the dimension of the object, n is the number of learning samples, σ_t , and σ_l are the noise levels, in the test image and the learning samples respectively (Suppose both σ_t and σ_l are small enough, compared with the signal level σ_s). Moreover, for a *random* test image set, (4.3) is *optimal* among the size- n learning sets; and the size- n learning set is *optimal iff* it has r equal singular values.

Result 1 and result 2 will be motivated in section 4.3 and section 4.4 respectively.

4.3 Denoising capacity of SVD

In this and subsequent sections, we analyze the performance of SVD-related applications, as promised and sketched in the introduction and in section 4.2: (a) the

denoising capacity of SVD; (b) and the *learning* capacity of LSA-based recognition system. We motivate our analysis by the perturbation expansion theory concerning singular values and singular vectors, as presented in chapter 2.

4.3.1 Case of distinct singular values

First, we consider the simplest case: a square matrix with a few distinct non-zero singular values. \mathbf{A} , \mathbf{B} , \mathbf{C} , and $\mathbf{\Sigma}$ are defined as in theorem 2 in chapter 2: \mathbf{A} is the signal matrix, \mathbf{N} is the i.i.d. Gaussian noise matrix (with zero mean and σ^2 variance), $\mathbf{B}=\mathbf{A}+\mathbf{N}$, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{C} = \mathbf{U}^T\mathbf{N}\mathbf{V}$. Note \mathbf{C} is still an i.i.d. Gaussian noise matrix (with zero mean and σ^2 variance). Further, define $\mathbf{\Omega} = \mathbf{C} + \mathbf{\Sigma}$. Then,

$$\mathbf{B} = \mathbf{U}\mathbf{\Omega}\mathbf{V}^T \quad (4.4)$$

$\{\lambda_i\}$, $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$, defined as (2.11-2.13) in chapter 2, are the right and the left singular vectors of \mathbf{B} respectively; $\{\lambda'_i\}$, $\{\mathbf{x}'_i\}$ and $\{\mathbf{y}'_i\}$, defined as (2.14-2.16) in chapter 2, are the right and the left singular vectors of $\mathbf{\Omega}$ respectively. Obviously, from (4.4),

$$\mathbf{y}_i = \mathbf{U}\mathbf{y}'_i \text{ and } \mathbf{x}_i = \mathbf{V}\mathbf{x}'_i \quad (4.5)$$

And, also from (4.4), the singular values of \mathbf{B} , $\{\lambda_i\}$, are same as the corresponding singular values of $\mathbf{\Omega}$, $\{\lambda'_i\}$.

Suppose that the noise-free matrix \mathbf{A} should have a rank of r , i.e. $\mathbf{A} = \sum_{i=1}^r \kappa_i \mathbf{U}_i \mathbf{V}_i^T$.

Combining (2.2) in chapter 2, (4.4) and (4.5), the closest rank- r approximation of \mathbf{B} is

$$\mathbf{B}^r = \sum_{i=1}^r \lambda_i \mathbf{y}_i \mathbf{x}_i^T = \mathbf{U} \left(\sum_{i=1}^r \lambda'_i \mathbf{y}'_i \mathbf{x}'_i{}^T \right) \mathbf{V}^T = \mathbf{U}\mathbf{\Omega}^r \mathbf{V}^T \quad (4.6)$$

Then

$$\|\mathbf{B}^r - \mathbf{A}\|_F^2 = \left\| \sum_{i,j} (\Omega'_{i,j} - \Lambda_{i,j}) \mathbf{U}_i \mathbf{V}_j^T \right\|_F^2 \quad (4.7)$$

where $\Lambda \in R^{m \times m}$, $\Lambda_{i,j} = 0$ if $(i,j) \notin \{(1,1), (2,2), \dots, (r,r)\}$ and $\Lambda_{i,i} = \kappa_i$ for $(i = 1, \dots, r)$. Due to the mutual orthonormality among any $U_i V_j^T$, we have the following formula:

$$\|B^r - A\|_F^2 = \|\Omega^r - \Lambda\|_F^2 \quad (4.8)$$

According to the perturbation theory in chapter 2, the first order perturbation of $\lambda_i' y_i' x_i'^T$ (to see the definition of $\{\lambda_i'\}$, $\{x_i'\}$ and $\{y_i'\}$ in (2.14-2.16), in chapter 2), for example $\lambda_1' y_1' x_1'^T$, is:

$$\lambda_1' y_1' x_1'^T \approx \begin{bmatrix} \lambda_1' & \lambda_1' a^T \\ \lambda_1' b & 0 \end{bmatrix} \approx \begin{bmatrix} \kappa_1 + C_{1,1} & \kappa_1 a^T \\ \kappa_1 b & 0 \end{bmatrix} \quad (4.9)$$

where $a = \left[\frac{\kappa_2 C_{2,1} + \kappa_1 C_{1,2}}{\kappa_1^2 - \kappa_2^2} \quad \dots \quad \frac{\kappa_r C_{r,1} + \kappa_1 C_{1,r}}{\kappa_1^2 - \kappa_r^2} \quad \frac{C_{1,r+1}}{\kappa_1} \quad \dots \quad \frac{C_{1,m}}{\kappa_1} \right]^T$, and

$b = \left[\frac{\kappa_1 C_{2,1} + \kappa_2 C_{1,2}}{\kappa_1^2 - \kappa_2^2} \quad \dots \quad \frac{\kappa_1 C_{r,1} + \kappa_r C_{1,r}}{\kappa_1^2 - \kappa_r^2} \quad \frac{C_{r+1,1}}{\kappa_1} \quad \dots \quad \frac{C_{m,1}}{\kappa_1} \right]^T$. Note, 2-order

and higher-order terms have been dropped. Similarly, the first-order perturbations of $\lambda_i' y_i' x_i'^T$, for $(i = 2, \dots, r)$, can be obtained.

By combining such results as (4.9), it is easy to obtain

$$\Omega^r - \Lambda = Y \quad (4.10)$$

where $Y = \begin{bmatrix} C_{1,1} & \dots & C_{1,r} & C_{1,r+1} & \dots & C_{1,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ C_{r,1} & \dots & C_{r,r} & C_{r,r+1} & \dots & C_{r,m} \\ C_{r+1,1} & \dots & C_{r+1,r} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ C_{m,1} & \dots & C_{m,r} & 0 & \dots & 0 \end{bmatrix}$

$$E \|B^r - A\|_F^2 = E \|Y\|_F^2 = \sum E Y_{i,j}^2 = (2rm - r^2) \sigma^2 \quad (4.11)$$

$$E |B_{i,j}^r - A_{i,j}| = \sigma \frac{\sqrt{2rm - r^2}}{m} \quad (4.12)$$

Obviously, (4.12) is a special case of (4.1) for square matrices, where $n=m$.

We note that the fact about the zero-block in Y in (4.10) holds in the context of the first-order perturbation theory. Actually, there are second and higher order terms in the zero-block of Y and they are not exactly zeroes. However, they are near zeroes as the noise level approaches zero; or precisely, they are much smaller, compared with other items, like $C_{i,j}$ for $i \leq r$ or $j \leq r$. We can testify to this fact by the following simple example in Matlab. Suppose $A \in R^{10,10}$ is a signal matrix, with all zero entries except $A_{1,1} = A_{2,2} = A_{3,3} = 100$. $C \in R^{10,10}$ is a noise matrix, generated as $randn(10)$ in Matlab. The observed signal matrix is $B = A + C$. The matrix in Table 4.1 is a typical example of the approximation error, $B^3 - A$, between A and the rank 3 approximation of B , B^3 . Please note the zero-block in Y (highlighted) is much smaller than other entries, although they are not exactly zeroes.

Table 4.1: A typical example of the approximation error, $B^3 - A$.

1.188	1.1857	-0.1289	-0.823	0.337	0.1243	-0.4911	-1.0207	0.467	-1.5508
2.2023	1.0554	-0.6569	-0.2284	0.8511	0.0955	0.8644	1.5689	-0.3891	-0.6872
-0.9865	-1.4727	1.1689	1.0365	-0.5205	-0.5185	-0.0025	0.063	-0.714	-1.9757
0.5235	-0.0745	0.4441	-0.0005	0.0012	0.0018	0.0032	0.0062	0.0005	0.0165
-0.3597	1.1914	0.2415	-0.0025	-0.0078	0.0005	-0.012	-0.0224	0.0079	0.0076
-0.2397	0.0212	1.2484	-0.0152	0.0074	0.0069	-0.0012	-0.0034	0.0102	0.021
-0.0112	1.1295	1.3475	-0.0115	-0.0025	0.006	-0.0097	-0.0184	0.014	0.0345
1.0295	1.3365	-0.9429	0.0221	-0.0204	-0.0077	-0.0066	-0.01	-0.0066	0.0072
0.9273	0.255	-0.0308	0.0087	-0.0057	-0.0016	0.0023	0.0054	-0.0036	0.0158
0.3838	-0.9499	0.677	-0.0064	0.0106	0.0041	0.0101	0.0184	-0.0005	0.0126

4.3.2 Case of multiple singular value

As in theorem 4 in chapter 2, suppose the first k ($k \leq r$) singular values of A are same. Following the notation in theorem 4 in chapter 2, we similarly have, as done in section 4.3.1:

$$\mathbf{B}' = \sum_{i=1}^r \lambda_i y_i x_i^T = (\mathbf{U}\mathbf{U}') \left(\sum_{i=1}^r \lambda_i y_i' x_i'^T \right) (\mathbf{V}\mathbf{V}')^T = (\mathbf{U}\mathbf{U}') \Omega'' (\mathbf{V}\mathbf{V}')^T = \mathbf{U} \Omega' \mathbf{V}^T \quad (4.13)$$

By the same techniques as in section 4.3.1, the first-order perturbation of Ω'' has the following form (please note the similar form between (2.32, 2.33) and (2.15, 2.16) and the fact that the up-left $k \times k$ submatrix of Ω' , $\Omega'_{1k,1k}$, is a diagonal matrix.):

$$\Omega'' \approx \begin{bmatrix} \Omega'_{1,1} & \cdots & \Omega'_{1,r} & \Omega'_{1,r+1} & \cdots & \Omega'_{1,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Omega'_{r,1} & \cdots & \Omega'_{r,r} & \Omega'_{r,r+1} & \cdots & \Omega'_{r,m} \\ \Omega'_{r+1,1} & \cdots & \Omega'_{r+1,r} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Omega'_{m,1} & \cdots & \Omega'_{m,r} & 0 & \cdots & 0 \end{bmatrix} \quad (4.14)$$

Then,

$$\begin{aligned} \Omega' &= \mathbf{U}' \Omega'' \mathbf{V}'^T = \begin{bmatrix} \mathbf{F} & & & & & \\ & \mathbf{I}_{m-k} & & & & \\ & & \begin{bmatrix} \Omega'_{1,1} & \cdots & \Omega'_{1,r} & \Omega'_{1,r+1} & \cdots & \Omega'_{1,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Omega'_{r,1} & \cdots & \Omega'_{r,r} & \Omega'_{r,r+1} & \cdots & \Omega'_{r,m} \\ \Omega'_{r+1,1} & \cdots & \Omega'_{r+1,r} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Omega'_{m,1} & \cdots & \Omega'_{m,r} & 0 & \cdots & 0 \end{bmatrix} & & & \\ & & & & & & \mathbf{E}^T \\ & & & & & & & \mathbf{I}_{m-k} \end{bmatrix} \\ &= \begin{bmatrix} \Omega_{1,1} & \cdots & \Omega_{1,r} & \Omega_{1,r+1} & \cdots & \Omega_{1,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Omega_{r,1} & \cdots & \Omega_{r,r} & \Omega_{r,r+1} & \cdots & \Omega_{r,m} \\ \Omega_{r+1,1} & \cdots & \Omega_{r+1,r} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Omega_{m,1} & \cdots & \Omega_{m,r} & 0 & \cdots & 0 \end{bmatrix} = \Lambda + \mathbf{Y} \end{aligned}$$

where Λ and \mathbf{Y} are same as those in (4.10), and \mathbf{E} and \mathbf{F} are defined in theorem 4 in chapter 2. Obviously, the same result, as (4.12), has been obtained.

4.3.3 Extension to the rectangular matrix

As stated in section 2.2.3.1 in chapter 2, we only have to consider the first r largest singular values. Thus, in the cases of rectangular matrices, the perturbation theory concerning the singular values/vectors still holds and the performance analysis, in section 4.3.1 and section 4.3.2, can be easily extended to the rectangular matrices. Here, we only present the final result, omitting the tedious mathematical deduction, which is almost same as that in section 4.3.1 and section 4.3.2. Suppose the signal matrix, \mathbf{A} , and noise matrix, \mathbf{N} , lie in $R^{m,k}$ ($m, k \geq r$). Other conditions stay same as in section 4.3.1.

$$\mathbf{\Omega}^r - \mathbf{\Lambda} = \mathbf{Y}$$

$$\text{where } \mathbf{Y} = \begin{bmatrix} C_{1,1} & \cdots & C_{1,r} & C_{1,r+1} & \cdots & C_{1,k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ C_{r,1} & \cdots & C_{r,r} & C_{r,r+1} & \cdots & C_{r,k} \\ C_{r+1,1} & \cdots & C_{r+1,r} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ C_{m,1} & \cdots & C_{m,r} & 0 & \cdots & 0 \end{bmatrix}$$

$$E \|\mathbf{B}^r - \mathbf{A}\|_F^2 = E \|\mathbf{Y}\|_F^2 = \sum E Y_{i,j}^2 = (rm + rk - r^2) \sigma^2 \quad (4.15)$$

$$E |B_{i,j}^r - A_{i,j}| = \sigma \sqrt{\frac{rm + rk - r^2}{mk}} \quad (4.16)$$

which is the same as (4.1). As $m \rightarrow \infty$, while k is a constant, $E |B_{i,j}^r - A_{i,j}| \rightarrow \sigma \sqrt{\frac{r}{k}}$, a non-zero constant. *As suggested by (4.2), it is impossible to reconstruct 3D scene to arbitrary accuracy by the factorization method using an affine camera model, by only increasing the number of the frames (while keeping the number of the feature points unchanged).* This contrasts with the claim that 3D scene could be reconstructed to arbitrary accuracy given enough frames (Thomas *et al.* 1999). However, we recognize the need for caution, our setting is not exactly the same as that in the paper (Thomas *et al.* 1999), where the perspective model was adopted.

4.4 Learning capacity of LSA-based recognition system

In this section, we analyze the performance of LSA-based recognition systems when the test image is correctly identified. Under such an assumption, there is still some error, as stated in the introduction, because of the noise in the basis images and the noise in the test image. In the following, we analyze the effect of these noise on the recognition system (also by the means of first-order perturbation theory).

Before we motivate the performance analysis of the LSA-based recognition system, we present a simple description of the LSA-based face recognition algorithm (Belhumeur *et al.* 1997; Georghiades *et al.* 1998; Georghiades *et al.* 2001). It consists of two steps: the off-line learning stage and the on-line recognition stage. In the learning stage, the image basis is obtained this way: a set of learning images for one face is arranged as a learning matrix A so that each image is regarded as one column of the learning matrix A . Suppose the face image has a dimension of m , and n learning samples are collected. $A \in R^{m \times n}$. The r ($r \ll m$ and $r \leq n$) basis images can be obtained as the first r left singular vectors of A , which correspond to the r largest singular values. In the on-line recognition stage, a test image is projected on the r basis images and its distance to the image basis is used for recognition.

4.4.1 Perturbation of the basis images

First, we analyze the learning stage, by using the matrix perturbation theory in chapter 2. By SVD, the low-dimension subspaces, $U^r = [y_1, y_2, \dots, y_r]$ and $V^r = [x_1, x_2, \dots, x_r]$, as defined in theorem 2 in chapter 2, are obtained. In some cases, such as in face recognition, the consequent step is contingent on an accurate basis. Here, we only study the subspace U^r : $U^r = UH$, where

$$\mathbf{H} = \begin{bmatrix}
 1 & \frac{\kappa_2 C_{1,2} + \kappa_1 C_{2,1}}{\kappa_2^2 - \kappa_1^2} & \dots & \frac{\kappa_r C_{1,r} + \kappa_1 C_{r,1}}{\kappa_r^2 - \kappa_1^2} \\
 \frac{\kappa_1 C_{2,1} + \kappa_2 C_{1,2}}{\kappa_1^2 - \kappa_2^2} & 1 & \vdots & \vdots \\
 \vdots & \dots & \ddots & \frac{\kappa_r C_{r-1,r} + \kappa_{r-1} C_{r,r-1}}{\kappa_r^2 - \kappa_{r-1}^2} \\
 \frac{\kappa_1 C_{r,1} + \kappa_r C_{1,r}}{\kappa_1^2 - \kappa_r^2} & \dots & \frac{\kappa_{r-1} C_{r,r-1} + \kappa_r C_{r-1,r}}{\kappa_{r-1}^2 - \kappa_r^2} & 1 \\
 \frac{C_{r+1,1}}{\kappa_1} & \frac{C_{r+1,2}}{\kappa_2} & \dots & \frac{C_{r+1,r}}{\kappa_r} \\
 \vdots & \vdots & \ddots & \vdots \\
 \frac{C_{m,1}}{\kappa_1} & \frac{C_{m,2}}{\kappa_2} & \dots & \frac{C_{m,r}}{\kappa_r}
 \end{bmatrix} \quad (4.17)$$

Note: From (4.17), we can roughly see that, for different singular vectors $\{\mathbf{U}_i\}$, their perturbations $\{y_i\}$ have been corrupted to a different extent, which depends on their *strength* (more formally, on their corresponding singular values). If $m \gg r$, the corruption comes mostly from $\{\mathbf{U}_i \mid (i > r)\}$. Obviously, the corruption in y_i ($i \leq r$) is approximately inversely proportional to its corresponding singular value, κ_i . Thus, y_1 can be considered *cleanest*, while y_r the *dirtiest*. In section 4.4.2, we will return to this point when the projection error is analyzed.

Furthermore, to decompose \mathbf{H} into: $\mathbf{H} = \mathbf{E} + \mathbf{F} + \mathbf{G}$, where $\mathbf{E} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{(m-r) \times r} \end{bmatrix}$

$$\mathbf{F} = \begin{bmatrix}
 0 & \frac{\kappa_2 C_{1,2} + \kappa_1 C_{2,1}}{\kappa_2^2 - \kappa_1^2} & \dots & \frac{\kappa_r C_{1,r} + \kappa_1 C_{r,1}}{\kappa_r^2 - \kappa_1^2} \\
 \frac{\kappa_1 C_{2,1} + \kappa_2 C_{1,2}}{\kappa_1^2 - \kappa_2^2} & 0 & \vdots & \vdots \\
 \vdots & \dots & \ddots & \frac{\kappa_r C_{r-1,r} + \kappa_{r-1} C_{r,r-1}}{\kappa_r^2 - \kappa_{r-1}^2} \\
 \frac{\kappa_1 C_{r,1} + \kappa_r C_{1,r}}{\kappa_1^2 - \kappa_r^2} & \dots & \frac{\kappa_{r-1} C_{r,r-1} + \kappa_r C_{r-1,r}}{\kappa_{r-1}^2 - \kappa_r^2} & 0 \\
 0 & 0 & \dots & 0 \\
 \vdots & \vdots & \ddots & \vdots \\
 0 & 0 & \dots & 0
 \end{bmatrix} \quad (4.18)$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \vdots & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \\ \frac{C_{r+1,1}}{\kappa_1} & \frac{C_{r+1,2}}{\kappa_2} & \cdots & \frac{C_{r+1,r}}{\kappa_r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{C_{m,1}}{\kappa_1} & \frac{C_{m,2}}{\kappa_2} & \cdots & \frac{C_{m,r}}{\kappa_r} \\ \frac{C_{r+1,1}}{\kappa_1} & \frac{C_{r+1,2}}{\kappa_2} & \cdots & \frac{C_{r+1,r}}{\kappa_r} \end{bmatrix} \quad (4.19)$$

4.4.2 Projection of a new test image on the basis images

The underlying noise-free subspace $\mathbf{U}' = \mathbf{U} \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{(m-r) \times r} \end{bmatrix} = \mathbf{U}\mathbf{E}$. Suppose a noise corrupted test image \mathbf{p} to be identified is observed, and the underlying truth is \mathbf{q} : $\mathbf{q} = \mathbf{U}\mathbf{f}$ and $\mathbf{p} = \mathbf{U}(\mathbf{f} + \mathbf{g})$. Because $\mathbf{q} \in \mathbf{U}'$, only the first r components of \mathbf{f} are possibly non-zeroes, i.e. $\mathbf{f} = [f_1, f_2, \dots, f_r, 0, \dots, 0]^T$. In practice, the noise-corrupted test image has to be projected on the noise-corrupted basis in the recognition system because the noise free basis is always unknown. More formally, the projection error of \mathbf{p} on \mathbf{U}'' is used:

$$\begin{aligned} \mathbf{p} - \mathbf{U}''\mathbf{U}''^T\mathbf{p} &= \mathbf{U}(\mathbf{f} + \mathbf{g}) - \mathbf{U}(\mathbf{E} + \mathbf{F} + \mathbf{G})(\mathbf{E} + \mathbf{F} + \mathbf{G})^T\mathbf{U}^T\mathbf{U}(\mathbf{f} + \mathbf{g}) \\ &\approx \mathbf{U}[\mathbf{g} - \mathbf{E}\mathbf{E}^T\mathbf{g} - (\mathbf{E}\mathbf{F}^T + \mathbf{E}\mathbf{G}^T + \mathbf{F}\mathbf{E}^T + \mathbf{G}\mathbf{E}^T)\mathbf{f}] \\ &= \mathbf{U}[\mathbf{g} - \mathbf{E}\mathbf{E}^T\mathbf{g} - (\mathbf{E}\mathbf{G}^T + \mathbf{G}\mathbf{E}^T)\mathbf{f}] \\ &= \mathbf{U}[\mathbf{g}' - \mathbf{G}\mathbf{f}'] \end{aligned} \quad (4.20)$$

where \mathbf{g}' has same components as \mathbf{g} , except its first r zeroes, i.e. $\mathbf{g}' = [0, 0, \dots, 0, g_{r+1}, \dots, g_m]^T$. And $\mathbf{f}' = [f_1, f_2, \dots, f_r]^T$. Note, in (4.20), the 2-order and higher-order terms have been dropped: \mathbf{F} , \mathbf{G} , and \mathbf{g} can possibly approach $\mathbf{0}$. From (4.20),

$$\mathbf{p} - \mathbf{U}''\mathbf{U}''^T\mathbf{p} = \mathbf{U}[\mathbf{g}', C'_1, C'_2, \dots, C'_r][\mathbf{1}, -\mathbf{h}^T]^T \quad (4.21)$$

$$\|\mathbf{p} - \mathbf{U}''\mathbf{U}''^T\mathbf{p}\|_F = \|[\mathbf{g}', C'_1, C'_2, \dots, C'_r][\mathbf{1}, -\mathbf{h}^T]^T\|_F \quad (4.22)$$

where $C'_i = [0, \dots, 0, C_{r+1,i}, C_{r+2,i}, \dots, C_{m,i}]^T$ and $\mathbf{h} = [f_1/\kappa_1, \dots, f_r/\kappa_r]^T$.

We suppose the basis is obtained from n learning samples, i.e., the learning matrix is $\mathbf{A} \in R^{m \times n}$, and each entry of \mathbf{A} has energy of σ_s^2 , and is corrupted with i.i.d. Gaussian noise with energy of σ_t^2 . It is also assumed that the test image has energy of σ_s^2 and is corrupted with noise of σ_t^2 . $\sum_{i=1}^r \kappa_i^2 \cong mn\sigma_s^2$, $\sum_{i=1}^m \sum_{j=1}^n C_{i,j}^2 \cong mn\sigma_t^2$, $\sum_{i=1}^r f_i^2 \cong m\sigma_s^2$, and $\sum_{i=1}^m g_i^2 \cong m\sigma_t^2$. $\|\mathbf{g}'\|_F \cong \sqrt{m-r}\sigma_t$, and $\|\mathbf{C}'_i\|_F \cong \sqrt{m-r}\sigma_t$. Due to the independence among $\{\mathbf{g}', \{\mathbf{C}'_i \mid i=1, \dots, r\}\}$, (4.22) becomes

$$\|\mathbf{p} - \mathbf{U}''^r \mathbf{U}''^{rT} \mathbf{p}\|_F^2 \cong (m-r)\sigma_t^2 + (m-r)\sigma_t^2 \sum_{i=1}^r \frac{f_i^2}{\kappa_i^2} \quad (4.23)$$

$$\|\mathbf{p}\|_F^2 \cong m(\sigma_s^2 + \sigma_t^2) \quad (4.24)$$

Obviously, from (4.23), the projection error is contingent on the relationship between $\{f_i\}$ and $\{\kappa_i\}$. From \mathbf{G} in (4.19), and (4.23), it can be concluded that the basis \mathbf{y}_1 that corresponds to the largest singular value is the *cleanest*, and that the basis \mathbf{y}_r that corresponds to the least singular value is the *dirtiest*. The *cleanness* of the j^{th} basis \mathbf{y}_j , here, is measured by the projection error, in (4.23), which is introduced by the j^{th} unit-norm basis image. For a random test image, the best and worst performance is:

$$(m-r)\sigma_t^2 + (m-r)\sigma_t^2 \frac{\sum_{i=1}^r f_i^2}{\kappa_1^2} \leq \|\mathbf{p} - \mathbf{U}''^r \mathbf{U}''^{rT} \mathbf{p}\|_F^2 \leq (m-r)\sigma_t^2 + (m-r)\sigma_t^2 \frac{\sum_{i=1}^r f_i^2}{\kappa_r^2} \quad (4.25)$$

$$(m-r)\sigma_t^2 + (m-r)\sigma_t^2 \frac{m\sigma_s^2}{\kappa_1^2} \leq \|\mathbf{p} - \mathbf{U}''^r \mathbf{U}''^{rT} \mathbf{p}\|_F^2 \leq (m-r)\sigma_t^2 + (m-r)\sigma_t^2 \frac{m\sigma_s^2}{\kappa_r^2} \quad (4.26)$$

where $\kappa_r^2 \leq \frac{mn\sigma_s^2}{r} \leq \kappa_1^2$. Define, furthermore, $\kappa_i^2 = c_i mn\sigma_s^2$:

$$(m-r)\sigma_t^2 + \frac{m-r}{nc_1} \sigma_t^2 \leq \|\mathbf{p} - \mathbf{U}''^r \mathbf{U}''^{rT} \mathbf{p}\|_F^2 \leq (m-r)\sigma_t^2 + \frac{m-r}{nc_r} \sigma_t^2 \quad (4.27)$$

4.4.3 Performance analysis over the learning samples

We have given the best and the worst performance analysis of the recognition system. Next, we want to analyze the average performance of the system when we

test the basis on the whole learning examples, i.e. all the images that are used to obtain the basis images.

$$E_{q \in \{A, j=1, \dots, n\}} \left\| \mathbf{p} - \mathbf{U}^r \mathbf{U}^{rT} \mathbf{p} \right\|_F^2 = (m-r)\sigma_i^2 + (m-r)\sigma_i^2 \sum_{i=1}^r \frac{E_{q \in \{A, j=1, \dots, n\}} f_i^2}{\kappa_i^2} \quad (4.28)$$

From (2.2) in chapter 2,

$$\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_n] = [\mathbf{U}_1, \dots, \mathbf{U}_r] \text{diag}(\kappa_1, \dots, \kappa_r) [\mathbf{V}_1, \dots, \mathbf{V}_r]^T = [\mathbf{U}_1, \dots, \mathbf{U}_r] [\kappa_1 \mathbf{V}_1, \dots, \kappa_r \mathbf{V}_r]^T$$

Surprisingly,

$$\sum_{q \in \{A, j=1, \dots, n\}} f_i^2 = \|\kappa_i \mathbf{V}_i\|_F^2 = \kappa_i^2 \quad \text{and} \quad E_{q \in \{A, j=1, \dots, n\}} f_i^2 = \frac{\kappa_i^2}{n} \quad (4.29)$$

Then

$$E_{q \in \{A, j=1, \dots, n\}} \left\| \mathbf{p} - \mathbf{U}^r \mathbf{U}^{rT} \mathbf{p} \right\|_F^2 \cong (m-r)\sigma_i^2 + (m-r)r\sigma_i^2/n \quad (4.30)$$

It can be easily proved that $(m-r)\sigma_i^2 + (m-r)r\sigma_i^2/n$ is the expectation for any test sets when the r largest singular values of the learning matrix \mathbf{A} are equivalent. Moreover, from (4.33), this is also the best expectation performance over a *random* sample set, where the *randomness* means that $E f_i^2$ in (4.23) should be statistically equivalent.

From this formula, (4.30), we can see clearly the effects of all the parameters in the recognition system. Given that the noise in the learning samples and in the test image, compared with the signal, is small, the performance can be regarded to be independent of the signal level. As m approaches a very large number, compared with r , the SSD is almost linearly dependent on m . As the number of the learning samples, n , increases, the recognition system improves: the error from the basis images decreases, and as n approaches infinite, the error from the basis images approaches zero. However, the error from the test image cannot be reduced except by having a cleaner image.

Another measure, used in the recognition system, is the angle between the test image and the basis images:

$$\frac{\|p - U''U''^T p\|_r^2}{\|p\|_r^2} = \frac{(m-r)\sigma_i^2 + (m-r)r\sigma_i^2/n}{m(\sigma_s^2 + \sigma_i^2)} \xrightarrow{m \rightarrow \infty} \frac{\sigma_i^2 + r\sigma_i^2/n}{\sigma_s^2 + \sigma_i^2} = \frac{\sigma_i^2}{\sigma_s^2 + \sigma_i^2} + \frac{r\sigma_i^2}{n(\sigma_s^2 + \sigma_i^2)} \quad (4.31)$$

Supposing $m \gg r$, the angle is independent of the size of the object, and depends on the energy level of the signal and the noise (in the learning samples and in the test image). As the size of the learning samples, n , increases, the system improves: the error from the basis images approaches zero and the error from the test image gradually dominates in the total error.

4.4.4 The optimal learning set

Suppose that the expectation of the test images, i.e. $\{f_i^2\}$, in (4.23), is known. How should we design the recognition system: specifically, how to select the learning samples, so that the system, concerning the expectation, has the best performance? Obviously, only the second term in (4.23) is dependent on the learning samples. The problem is:

$$\min \sum \frac{f_i^2}{\kappa_i^2}, \text{ subject to } \sum \kappa_i^2 = C \quad (4.32)$$

$\sum \kappa_i^2 = C$ means that, when the dimension, m , and the size, n , of the learning samples is large enough, the signal energy, $\sum \kappa_i^2$, should be approximately $mn\sigma_s^2$. By using a Lagrange multiplier, the minimum can be obtained iff

$$\frac{f_i}{\kappa_i^2} \equiv \text{Cons} \quad (4.33)$$

From (4.33), we can draw such a conclusion, however it is a little surprising, that the basis images obtained from the n samples of \mathbf{A} are not optimal when the test image set is also $\{\mathbf{A}_i\}$. The reason is that, the basis, corresponding to the largest singular value, is *overlearned* in the *learning* process: from (4.33), the optimal *learning ability*, κ_i^2 , should be proportional to f_i , while κ_i^2 is actually proportional to f_i^2 , as in (4.29).

4.5 Simulation results

Here, we have to note that it is very difficult to have real data with high precision ground truth. Thus, in this section, we present some *simulations* to verify result 1 and result 2.

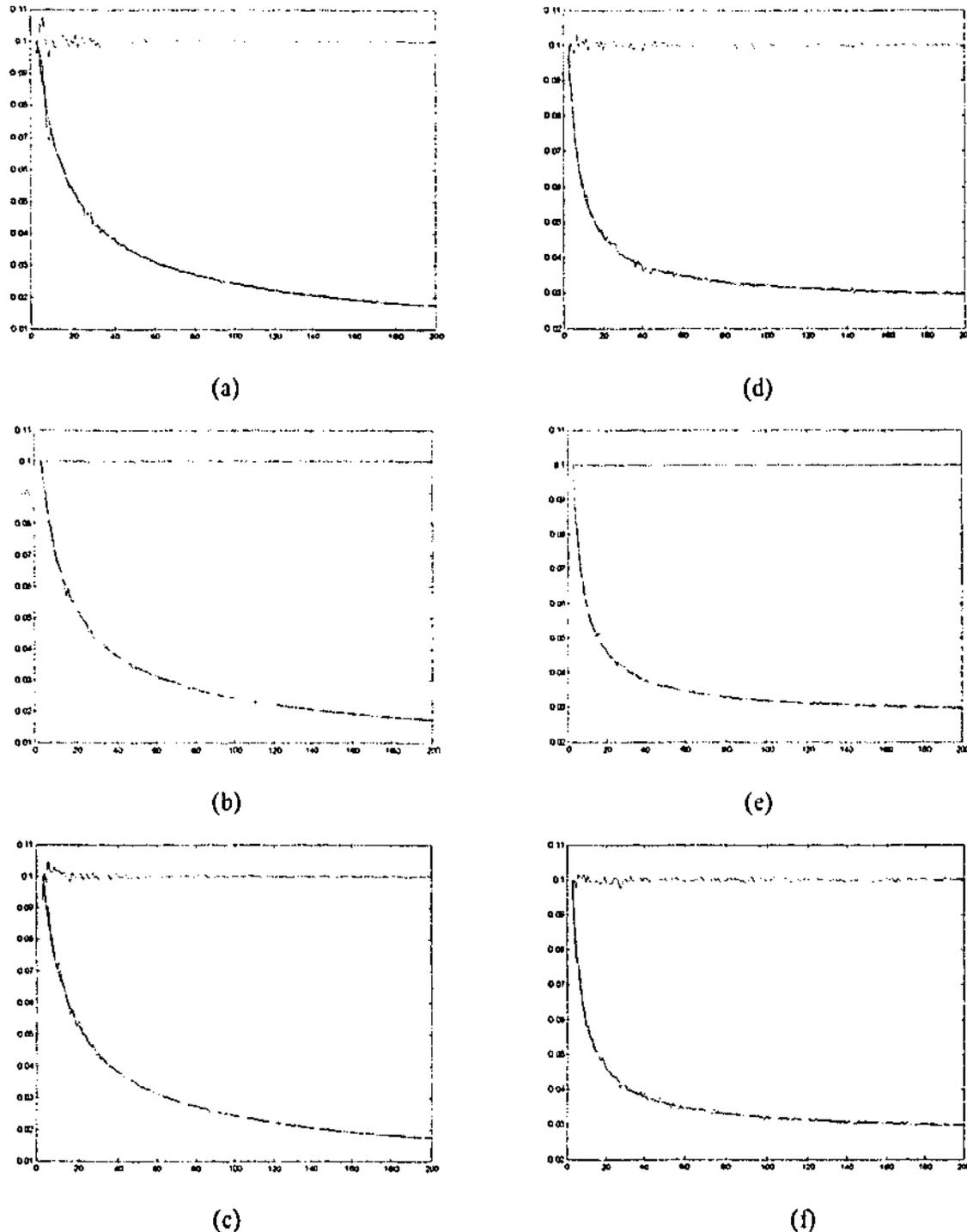


Figure 4.1: The average error that still resides in the approximation matrix. The abscissa denotes the number of the rows of the matrices, and the error is on the ordinate. (a-c) are for the square matrices, and (d-f) are for the rectangular matrices, which have a constant, 40, columns. There are three curves in every sub-figure: the (approximately) straight curve in the upper part denotes the original noise in the noise corrupted matrix, and the smooth/unsmooth curves are the expectation/actual error in the approximation matrix respectively. In (a) and (d), the signal and the noise are randomly generated. In (b) and (e), the noise levels are normalized, so that the average energy in each entry of the matrices is 0.01. In (c) and (f), the signal matrices have 3 equivalent singular values, while the energy level remains same.

4.5.1 Simulation of the denoising capacity of SVD

In a recent paper, an experimental result related the SVD's *denoising* performance has been reported (Chen *et al.* 2004). In that example, noise with amplitude of $1.5/40=0.037$ still resides in the approximation matrix: where the noise-free 40×40 matrix, with a rank of 3, had been corrupted with zero-mean-and-0.01-variance Gaussian noise. From result 1 we have derived, the value should be 0.038. That this is pretty close to the result in (Chen *et al.* 2004), confirms the theory present here.

To provide further evidence, we have carried out our own simulations. Here, we work on a set of rank-3 matrices. For square matrices, the size of the matrices increases from 3 to 200; while for rectangular matrices, the number of the columns remains unchanged, staying at 40. The noise level is 0.1. In Matlab notation, $\mathbf{M} = \text{randn}(\text{rows}, 3) * \text{randn}(3, \text{columns}) + 0.1 * \text{randn}(\text{rows}, \text{columns})$ is the noise-corrupted matrix. Figure 4.1 shows the simulation results of SVD's denoising performance, compared with the expectation from result 1. It can be easily observed that the expected curve almost coincides with the simulation result. In contrast with Figure 4.1 (d-f) (rectangular matrices), the curves in Figure 4.1 (a-c) (square matrices) can be observed to continue towards zero error, while the error for the rectangular matrices changes little after the number of the rows increases to 20 or 40.

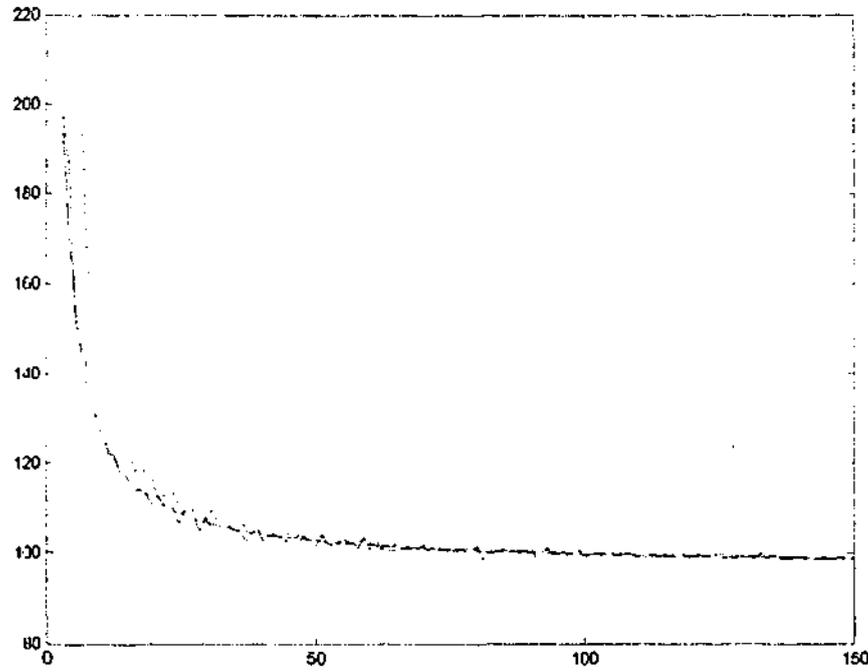
4.5.2 Simulation of the *learning* capacity for LSA-based recognition

In this section, we present some simulation results concerning the SSD performance of the LSA-based recognition system, as stated in section 4.4. Suppose we work on a set of rank 3 subspaces but in a dimension of 100. In this section, the parameters are set as follows: $m=100$, $r=3$, $\sigma_s = 100$, and $\sigma_t = \sigma_r = 1$. First, the SSD performance of a set of basis images is analyzed, over two test sets: the learning set, from which the basis images are obtained, and another random set where its 3 singular values have been artificially equalized. Obviously, as the learning sample size approaches infinite, the SSD, over two sets, approaches a stable value, as shown in Figure 4.2-a. Over the learning set, the performance, denoted by solid curve, almost coincides with the expectation from (4.30), denoted by dashed curve. Over the random set, the performance is denoted by dotted curve. Because the 3 singular values of the random

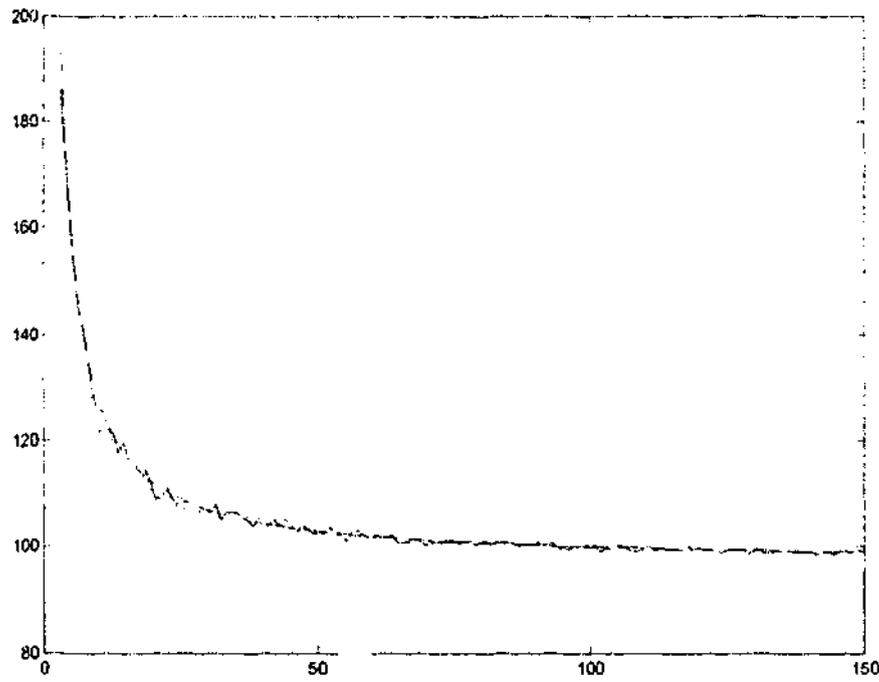
test set have been artificially equalized, the best performance over this random set can be obtained only if the learning set has 3 equal singular values, from (4.33). However, the random learning set always has 3 distinct singular values. Thus, the performance over the random test set is worse than the optimal curve, denoted by dashed curve, especially for the small-size learning samples; in fact, the performance for the recognition system is very bad, at 5,771.6, 788.1 and 588.1 respectively, when the learning sample sizes are only 3, 4 and 5. In order to make the curves clear, these points have been omitted in Figure 4.2-a.

Conversely, next, we first have a random test set, and show the performance of different learning sets (different basis images): an optimal learning set, which complies with (4.33), and a random learning set, who has 3 equal singular values. For the random learning set, with 3 equal singular values, its performance, denoted by the solid curve, can be expected to coincide with the expectation (4.30), denoted by the dashed curve, as shown in Figure 4.2-b. Obviously, the optimal learning set, complying with (4.33), has a better performance than the random learning set, especially for small learning sizes.

Note, if the learning set and the test set are *truly* randomly generated, it probably has a very bad SSD performance, especially for a small-size learning set. For example, the r^{th} basis image may be very *dirty*, because the r^{th} singular value of the learning set is comparatively small; while most of the energy of the test image probably comes from this basis image. In such cases, the error from the basis images, especially from the r^{th} basis image, will dominate the total error, as can be seen from (4.23).

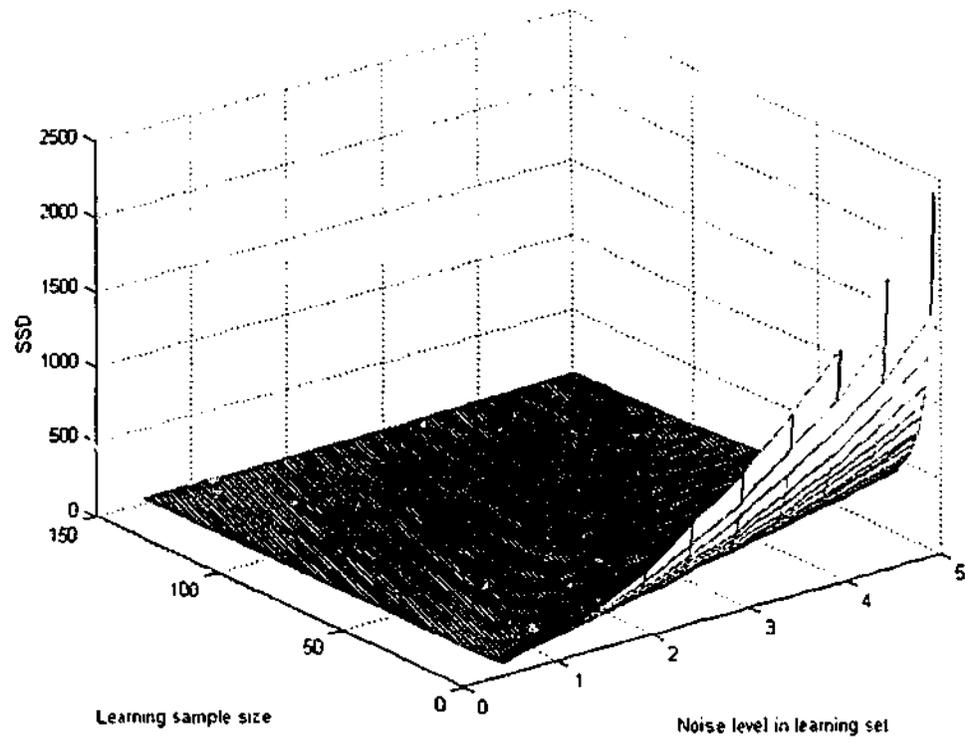


(a)

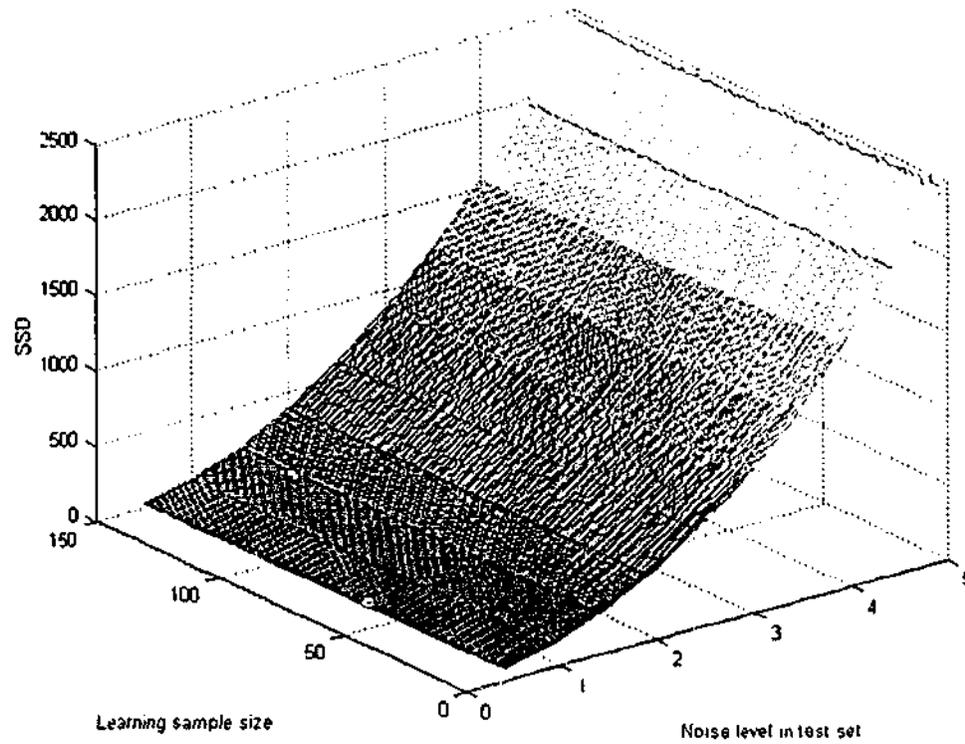


(b)

Figure 4.2, The dependency of SSD on the size of learning samples.(a) for a learning set, over two test sets: (solid) the learning set from which the basis images are obtained, and (dotted) another random set that has 3 equal singular values; (b) for a test set, by two learning sets: (dotted) the optimal learning set and (solid) another random learning set that has 3 equal singular values. In both subfigures, the dashed curves denote the expectation from (4.30).

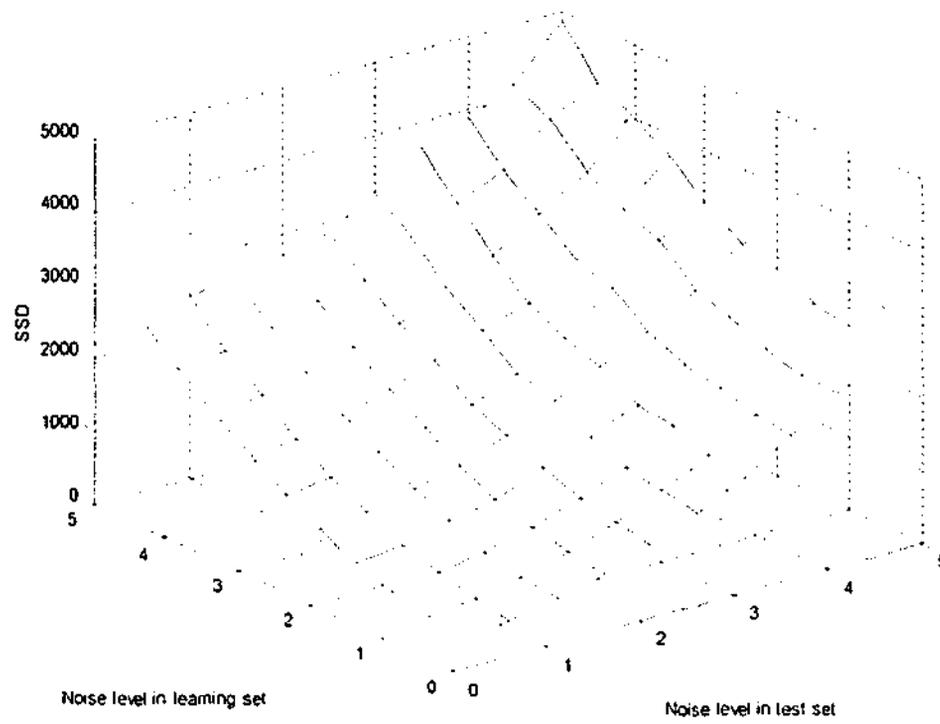


(a)

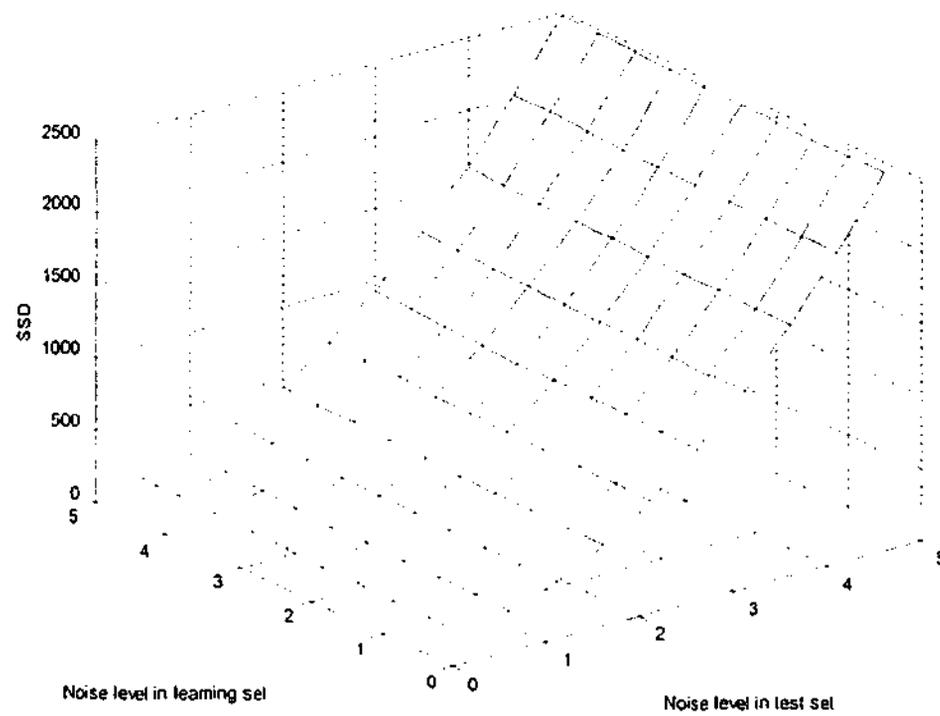


(b)

(to be continued)



(c)



(d)

Figure 4.3: The effects of three parameters in (4.23) on SSD. To see the description in the text.

In Figure 4.3, we show the effects of the three parameters in (4.30), the size of the learning samples, n , the noise level in the learning set, σ_l , and the noise level in the

test set, σ_t ; on SSD when the recognition system works over the *learning* samples. Figure 4.3(a) shows the performance of SSD when the noise level in *test* image is 0.5 (very small). It can be easily observed: the *square* dependency on the noise level in the learning set and the decreased effects of the noise in the learning set as the learning size increases. Figure 4.3 (b) shows the performance of SSD when the noise level in *learning* samples is 0.5 (very small). It can be easily observed: the *square* dependency on the noise level in the test set and its effect is almost independent of the learning size. Figure 4.3(c) and (d) show the effect of the noise levels of the learning set and the test set when the learning sizes are 3 and 125 respectively. When the learning size is 3, the noise in learning set has almost a same effect on SSD as the noise in test set; when the learning size is 125 ($\gg 3$), the noise in learning set can be almost neglected if it is not much higher than that in the test set.

4.5.3 Relationship with some experimental observations

Here, we can explain such phenomena previously reported in the computer vision literature, by using the analysis in section 4.3 and section 4.4. For example, in SFM, the root-mean-square error of the recovered shape with respect to the true shape was reported in (Morita *et al.* 1997). Fig. 6 in that paper (Morita *et al.* 1997) shows that the error approaches a constant value after the number of the frames increases to 20 or 40, as almost coincides with the result 1, in section 4.2 or the Figure 4.1 in section 4.5.1.

Another two observations are related to result 2, in section 4.2. In (Basri *et al.* 2003), it was reported that no significant deterioration of the performance was found for LSA-based face recognition, if the images were subsampled by 16×16 squares, which means that, m , the number of the rows of \mathbf{A} , decreases by $1/256$. However, the reduced m is still very large, about 1000 ($\gg 4$ or 9). We can find the explanation from (4.31): the performance, measured by the angle between the test image and the basis images, is almost independent of m if $m \gg r$.

The last, but not the least, (maybe even the most important), observation was that "recognition of an object under a particular lighting and pose can be performed

reliably provided the object has been previously seen under similar circumstances” (Georghiades *et al.* 2001). A very reliable explanation can be found from (4.23) and (4.33). For a test image, if it or its similar cases have been observed in the learning samples, its $\{f_i^2\}$ will probably have a *good* relationship with $\{\kappa_i^2\}$, i.e., for a larger κ_i^2 , f_i^2 is also larger, and vice versa. More formally, if (4.33) holds, the recognition system has a best performance. However, for a test image, which is produced under very different lighting conditions from those in the learning set, its $\{f_i^2\}$ probably has very *bad* relationship with $\{\kappa_i^2\}$. If most of its energy comes from the *dirtiest* basis, which corresponds to the r^{th} singular value of the learning matrix, from (4.23), the recognition error is probably very large. This not only explains the drawback of PCA-based face recognition, pointed out in (Georghiades *et al.* 2001), but also gives a possible solution, as suggested by (4.33). For a *random* test set, the best learning samples should be selected this way: to equalize the first r largest singular values as possible. However, we do not present any specific strategies for this *open*, and probably promising, issue.

4.6 Conclusion

The main contribution of this chapter is the presentation of a theoretical analysis of SVD-based low rank projections: specifically the *denoising* capacity of SVD (where we characterized the error that still resides in the SVD-*denoised* matrix) and the *learning* capacity of LSA-based recognition systems (where we showed that the projection error can be decomposed into two independent sources, one from the test image and the other from the basis image). The work presented in chapters 5 and 8 directly exploits these results.

Chapter 5

Recovering the Missing Components in a Large Noisy Low-Rank Matrix: Application to SFM

In computer vision, it is common to require operations on matrices with “missing data”, for example because of occlusion or tracking failures in the Structure from Motion (SFM) problem. Such a problem can be tackled, allowing the recovery of the missing values, if the matrix should be of low rank (when noise free). The filling in of missing values is known as imputation. Imputation can also be applied in the various subspace techniques for face and shape classification, on-line “recommender” systems, and a wide variety of other applications.

However, iterative imputation can lead to the “recovery” of data that is seriously in error. In this chapter we provide a method to recover the most *reliable* imputation, in terms of deciding when the inclusion of extra rows or columns, containing significant numbers of missing entries, is likely to lead to poor recovery of the missing parts. Although the proposed approach can be equally applied to a wide range of imputation methods, we address only the SFM problem in this chapter. The performance of the proposed method is compared with Jacobs’ and Shum’s methods for SFM.

The work presented in this chapter has been published in (Chen et al. 2004).

5.1 Introduction

Several problems in computer vision (and beyond) can be reduced to fitting a large matrix to its closest low-rank approximation: the factorization method under affine models of Structure from Motion (SFM) (Tomasi et al. 1992; Poelman et al. 1997; Kahl et al. 1999; Kanatani 2001), optical flow estimation in multi-frame video

(Irani 1999; Irani 2002), subspace constraints in face recognition and indexing, pose determination, data mining and a plethora of related problems (e.g., customer modelling and recommender systems (Sarwar et al. 2000; Brand 2003)).

In this chapter, we restrict our application to the structure from motion in an affine camera setting, although this is to make the problem concrete rather than to exploit any special structure of that problem. Indeed we do *not* use any features of the problem formulation that is specific to the particular application (see section 5.1.1) so we will generically say that the matrix \mathbf{M} (of dimension $m \times n$ and with real number entries) should be (without noise) of rank $r \ll \min\{m, n\}$. A consequence of the matrix being of rank r is that it can be factored into \mathbf{RS} for real rank- r matrices \mathbf{R} of size $m \times r$ and \mathbf{S} of size $r \times n$, and vice versa. For the SFM problem, we are of course interested in *particular* factors (the factorization is not unique because for any invertible matrix \mathbf{G} of size $r \times r$ we have $\mathbf{RS} = (\mathbf{RG})(\mathbf{G}^{-1}\mathbf{S})$). However, for other problems we are not interested in any of the factors *per se* but are interested in the projection onto a low rank matrix to reduce noise, to fill in missing data, or extrapolate to as yet uncollected data. For example, we may wish to exploit the low rank constraint to assist in the feature point-matching problem (predicted search ranges) or to extrapolate tracks.

In most real world problems, noise is inevitably introduced in the data. In the presence of noise, the measurement matrix quickly becomes full-rank. Thus, the matrix has to be projected upon its low-rank approximation \mathbf{M}^r minimising mean squared error (using Frobenius norm):

$$\|\mathbf{M} - \mathbf{M}^r\|_F^2 \quad (5.1)$$

The singular value decomposition (SVD) gives the best solution to this problem (Golub et al. 1996): $\mathbf{M} = \mathbf{UDV}^T$, $\mathbf{M}^r = \mathbf{UD}^r\mathbf{V}^T$ where \mathbf{D}^r is obtained by setting to 0 all of the singular values except the r largest ones. This is classical and is the starting point of the original factorization method for SFM, and hence for many of its variants.

We could equivalently seek the rank- r factors explicitly in the formulation. That is, finding \mathbf{R} of size $m \times r$ and \mathbf{S} of size $r \times n$, that minimize

$$\|\mathbf{M} - \mathbf{RS}\|_F^2 \quad (5.2)$$

In such cases, one can side-step directly computing the "clean" \mathbf{M}' (the reprojected points in SFM terminology).

The issues to solve, other than computational efficiency issues, include: how to deal with missing values, and how to deal with large amounts of data or data that is arriving sequentially. We will focus here on the first problem, and an algorithm is presented in section 5.4.3.

5.1.1 Missing-data problem in SFM

In SFM, one starts from the mathematical relationship between the measurement matrix \mathbf{M} (coordinates of features tracked through frames), the object-camera motion matrix \mathbf{R} , and the structure/shape matrix \mathbf{S} . In the non-degenerate cases, and assuming an affine camera, the measurement matrix, should be exactly of rank 4. However, one can exploit the special structure: the "registered" measurement matrix, formed by subtracting the centre of mass of the image points from their coordinates, which should be of rank 3 (Tomasi et al. 1992; Poelman et al. 1997; Kahl et al. 1999; Kanatani 2001), and one can even reduce the problem to a rank-1 problem (Aguiar et al. 2003).

Regardless of what formulation, in terms of rank, the SVD cannot be directly used if some of the data are unavailable. This issue has been regarded (Jacobs 1997; Jacobs 2001; Rother et al. 2002) as the major drawback of the factorization method.

Attempts to apply a subspace projection approach, in the presence of missing data, can be divided into two categories:

- 1 Those that attempt to "fill in" (or *impute*) the missing values:
 - a. The seminal approach of Tomasi and Kanade (Tomasi et al. 1992) where the filling in is called "*hallucination*". In their somewhat heuristic approach to the missing data sub-problem, a full submatrix (no missing

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM

entries) is first decomposed by the factorization method, and then the initial solution grows by one row or by one column at a time, hallucinating missing data. The final estimate is then refined by employing a steepest descent minimization method on a Least Squares fitting criterion (equation 5.2) $\|M - RS - C\|_F^2$ where the inclusion of C makes the adjustment for the registration.

- b. Jacobs' method (Jacobs 1997; Jacobs 2001) treated each column, with some missing entries, as an affine subspace, and solved the problem by obtaining the intersection of all the quadruple (in practice, a large selection of) affine subspaces. Unknown entries are recovered by finding, for each column, the least squares regression onto this subspace.

2 Methods that directly obtain the factors – thus not imputing the measurement matrix (directly)

e.g., Shum's method (Shum et al. 1995) and Guerreiro and Aguiar's work (Guerreiro et al. 2003). Though Shum's method was not originally formulated for SFM (see section 5.1.2) Jacobs (Jacobs 1997; Jacobs 2001) suggested that it could be applied to the SFM problem. We note that Shum's formulation uses data weighting to incorporate confidence measures, an elaboration not essential to our exposition. In essence, the method iteratively solves coupled least squares problems for the factors starting from the formulation of equation 5.2 but modifying the Frobenius norm so that only entries for measured data are involved, and adding the weights as mentioned previously. Since the formulation is bilinear in the factors, one can hold one factor constant and solve a linear least square problem for the other factor. Thus the missing data are only indirectly imputed (one can "reproject" the recovered structure onto the images). See chapter 6.

Tomasi and Kanade's approach to the problem of occlusion (Tomasi et al. 1992) has the following disadvantages: needing to start from a complete submatrix (it is a NP-hard problem of finding the largest complete submatrix), asymmetric usage of

the data, and error propagation, as pointed out by Jacobs (Jacobs 1997; Jacobs 2001).

The greatest advantage of Jacobs' method lies in the fact that it does not need to start from a complete submatrix. Ideally, for a generic problem, all the quadruple affine subspaces should be utilized in order to obtain a good result. In practice, a selection of the affine subspaces is needed. However, in the severe noise case, using only a small portion of the affine subspaces may produce unsatisfactory results. Intrinsically, Jacobs' linear approach can be employed in any missing-data problems under low-rank constraint; however, better performance for SFM problem can be obtained, because some "outlier" detection strategies are used, by incorporating the specialty of the SMF problem; while, for a general low-rank problem, the performance of the generic algorithm proved to be far away from the optimal solution, especially when there is a lot of missing data.

One drawback of Shum's approach is its dependence on an initial matrix, although a random initial matrix works when the percentage of the missing data is low and the data is not highly corrupted by noise. Even taking Jacobs' result as its initial point, Shum's approach still tends to diverge when there is a lot of missing data, especially for the generic low-rank problems.

Recently, by combining Jacobs' method (Jacobs 1997; Jacobs 2001) with the projective factorization method of Sturm & Triggs (Sturm et al. 1996), Martinec et al. (Martinec et al. 2002) solved the missing-data problem under the perspective model. Various geometric constraints (Heyden et al. 1998; Kahl et al. 1999; Brandt 2002), have also been employed to cope with the missing-data problem. For example, Heyden and Kahl (Heyden et al. 1998; Kahl et al. 1999) proposed to use "closure constraints" for affine construction, where the missing-data problem can be naturally handled. They noted that Jacobs' method could be regarded to be "dual" to the closure constraints. It should also be noted that the missing-data problem in SFM could be efficiently solved by an incremental SVD (Brand 2002). Our own method for solving this problem is to be found in section 5.4.3.

5.1.2 Other missing data problems under low rank constraint

Low rank based imputation is so commonly useful that it is not surprising that many variations have appeared in the literature. Many applications are quite far removed from SFM: e.g., DNA prediction (Troyanskaya et al. 2001), or in recommender system (Sarwar et al. 2000; Brand 2003). Yet these studies share the same intrinsic nature: missing-data problem under low-rank constraints.

The approach used in DNA prediction (Troyanskaya et al. 2001) employs a "SVDimpute" algorithm that bears a superficial similarity to our approach. The starting point of that approach is to fill in the missing values with row averages, then to use the SVD to rank r -project, then regress the missing values against the spanning vectors of the SVD, the process then being re-iterated until convergence. The first potential drawback of these imputation methods is that, the initial values for the starting point are rather arbitrary. Such limits its application to the cases where only a few data are missing (Sarwar et al. 2000; Brand 2003). Secondly (Troyanskaya et al. 2001), only one missing component is updated at a time – an inefficiency. More importantly, as will be covered in the Appendix, such a strategy does not impute with minimal distance to the "current" subspace. Thus convergence cannot be ensured. Indeed, the same criticisms as have been levelled at Tomasi and Kanade apply: strong dependence on the starting matrix and the imputation order (Brand 2002; Brand 2003). In addition, the iterative imputation method has the possibility of exhibiting "bad behaviour" (see Appendix), i.e. the estimate goes further from the underlying optimal solution as the iteration proceeds. However, such an important issue was overlooked in (Troyanskaya et al. 2001).

In a recommender system, the low rank constraint is supposed to capture customer preferences and it needs to be continually updated. However, it would be very computationally expensive to update the system online by traditional SVD. Brand (Brand 2002; Brand 2003) proposed an incremental SVD to efficiently do this work, making the online updating possible. In what Brand calls bootstrapping (Brand 2003), he re-orders the matrix to have a dense submatrix in the top left corner and incrementally adds rows and columns using incremental SVD updating

routines. Incremental update is also desirable in SFM problems (Brand 2002), but it is beyond the scope of the present chapter.

5.1.3 Contributions of this chapter

The main contribution of this chapter is that we provide a means of determining which parts of the matrix should be used in the iterative imputation/recovery process. In the SFM context, this corresponds to deciding which tracks and/or which frames (typically the former) should be exploited in the iterative recovery process. Intuitively, the gain, on the one hand, of using more data (rows and/or cols) is balanced by the fact that extra rows and cols carry more missing entries. Rows or columns that have almost all entries missing are not likely to bring much extra information and the extra degrees of freedom can make the recovery less stable. Incorporation of data with more missing values can cause the solution to “wander” away from the true solution.

As a second contribution, we present an iterative imputation strategy and prove its weak convergence. Although falling short of a theoretical guarantee, the weak convergence, together with our mechanism of precluding the “wandering” of the iterative approach, ensures the iteration to the optimal solution in almost every case. This will be demonstrated by experiments.

5.1.4 Overview of the chapter

In section 5.2, we first state the general missing-data problem under low-rank constraint, using an objective function that is subtly different from the one in Shum’s method. In section 5.3, we analyze the central idea, used in the imputation approach (Sarwar et al. 2000; Troyanskaya et al. 2001; Brand 2003), i.e., to fill in the missing data so that the complete vector has a minimal distance to a known low-rank subspace. Then, we propose a new iterative method of recovering the missing data in a large low-rank matrix; and prove its weak convergence. In section 5.4, based on the analysis of the *denoising capacity* of the SVD in chapter 4, we propose a criterion determining whether it is worth incorporating the incomplete vectors in the iteration. In section 5.5, we experimentally compare the

algorithm with Jacobs' and Shum's methods. In the Appendix, we discuss some aspects of the iterative method, including its convergence, the "wandering" issue, a bootstrapping strategy that provides a partial solution to the "wandering issue" (hinting at a more complete solution), and the relation to other approaches.

5.2 The definition of the problem and its nonlinear nature

5.2.1 The problem

A large matrix $\mathbf{M} \in R^{m \times n}$, which should have a low rank r , is corrupted with noise (assumed to be i.i.d. Gaussian), and has missing entries. The problem is to recover these missing entries and to minimize the approximation error between the recovered matrix, $\hat{\mathbf{M}}$, and its closest rank- r approximation, $\hat{\mathbf{M}}^r$:

$$\min \|\hat{\mathbf{M}} - \hat{\mathbf{M}}^r\|_F^2 \quad (5.3)$$

subject to $\hat{M}_{i,j} = M_{i,j}$ if $M_{i,j}$ is observed. In other words, we seek to minimize the difference between the imputed matrix $\hat{\mathbf{M}}$ (where the missing values have been recovered but the matrix *has not been de-noised*) and the closest rank- r approximation of the imputed matrix $\hat{\mathbf{M}}^r$ (now imputed and de-noised).

Note: The minimization objective is different from that in Shum's approach (Shum et al. 1995), where the objective is to *recover the matrix factors* that minimize the *re-projection error of the "non-missing" data*, i.e. the sum of the square of the difference between known elements in the incomplete matrix and the corresponding elements in the new recovered matrix, which is exactly of low-rank. Moreover Shum's formulation incorporates *weighted errors* – an elaboration that can be extremely effective if one has error covariance estimates that can be exploited. Weighted error norms are beyond the scope of this chapter (however see chapter 6) and so we express Shum's formulation as:

$$\min \|\mathbf{M} - \hat{\mathbf{R}}\hat{\mathbf{S}}\|_{F_{\text{non-missing}}}^2 \quad (5.4)$$

In essence, (5.4) predisposes one to directly seek the factors, and to perform imputation and de-noising together. This suggests different implementation strategies but the solutions to both formulations should be equivalent. Of course,

given different implementation strategies, the stability and convergence properties can differ.

5.2.2 Non-linearity of the problem

Obviously, Shum's formulation (equation 5.5.4) is non-linear: in fact it is bi-linear in the factors \mathbf{R} and \mathbf{S} . Here, we show the intrinsic non-linearity of our formulation (equation 5.5.3).

Suppose $\mathbf{M} \in R^{m,n}$. Its closest rank- r matrix, measured by the Frobenius norm, is

$$\mathbf{M}^r = \mathbf{U}^r \boldsymbol{\Sigma}^r (\mathbf{V}^r)^T = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \text{ with } \|\mathbf{M} - \mathbf{M}^r\|_F^2 = \sum_{i=r+1}^p \sigma_i^2 \text{ (Golub et al. 1996),}$$

where $p = \min(m, n)$ and $\{\sigma_i^2\}$ are the non-descending eigenvalues of $\mathbf{M}^T \mathbf{M}$.

Suppose \mathbf{M} has some missing entries $\{M_{i,j} \mid (i,j) \in \Xi\}$, where

$\Xi = \{(i,j) \mid M_{i,j} \text{ is unknown, } 1 \leq i \leq m, 1 \leq j \leq n\}$. $\mathbf{E}_{i,j} \in R^{m,n}$, has all zero entries,

except a one at (i,j) . Let the recovered matrix be $\hat{\mathbf{M}}$, $\hat{\mathbf{M}} = \bar{\mathbf{M}} + \sum_{(i,j) \in \Xi} k_{i,j} \mathbf{E}_{i,j}$, where

$$\bar{M}_{i,j} = \begin{cases} M_{i,j} & (i,j) \notin \Xi \\ 0 & (i,j) \in \Xi \end{cases}. \text{ The characteristic polynomial of } \hat{\mathbf{M}}^T \hat{\mathbf{M}}, p(\lambda), \text{ is a high-}$$

order polynomial of λ and $k_{i,j}$. The equation, $p(\lambda) = 0$, has n non-negative roots

for any $\{k_{i,j}\}$, because $\hat{\mathbf{M}}^T \hat{\mathbf{M}}$ is positive semi-definite. The problem reduces to

finding $\{\hat{k}_{i,j}\}$, which minimizes the sum of the least $n-r$ roots of the equation,

$p(\lambda) = 0$. This is a nonlinear problem.

Consider a simple case, $\mathbf{M} \in R^{10,10}$ with a missing entry $M_{1,1}$. Suppose \mathbf{M} should

be of rank 4, if it were noise free and had no missing entries. Its characteristic

polynomial, $p(\lambda, t)$, where t denotes the missing entry, is of the form:

$$p(\lambda, t) = \lambda^{10} + f_2(\lambda)t^2 + f_1(\lambda)t + f_0(\lambda) = \lambda^{10} + \sum_{i=0}^9 \lambda^i g_i(t), \text{ where } f_i(\lambda) = \sum_{j=0}^{i-2} f_{i,j} \lambda^j$$

and $g_i(t) = \sum_{j=0}^{i-2} g_{i,j} t^j$, and $f_{i,j}$ and $g_{i,j}$ are determined by \mathbf{M} . This equation is

nonlinear and the problem of minimizing the sum of the least δ roots is very complicated. If there are many missing entries in the matrix, the problem appears intractable from this point of view.

5.3 An iterative imputation method

In this section, an iterative algorithm, based on the imputation principle, is proposed, and we prove a weak convergence of the iterative algorithm.

5.3.1 Minimization of the distance of a vector with missing entries to a known subspace

The key starting point is to “grow” a complete matrix by adding rows or columns, filling in those missing entries in the new rows or columns. Without loss of generality, we consider only the case of column-wise growth of the complete matrix. Thus suppose we have a complete matrix, $\mathbf{M} \in R^{m,n}$, which should be of rank r ($r \leq m, n$) if it were noise-free; and another vector $\mathbf{x} \in R^m$, with missing components. Ideally, $[\mathbf{M}, \mathbf{x}]$ should be also of rank r if both of them were noise-free and complete. Suppose the first k ($k \leq m - r$) components of \mathbf{x} (i.e., $x_{1:k}$) are missing (swapping rows if necessary). The imputation method finds a linear combination of column vectors in \mathbf{M} , fitting \mathbf{x} the best (Troyanskaya et al. 2001; Brand 2002):

$$\hat{x}_{1:k} = \mathbf{U}_1 (\mathbf{U}_2^T \mathbf{U}_2)^{-1} \mathbf{U}_2^T \mathbf{x}_{k+1:m} \quad (5.5)$$

where, by SVD, the rank- r projection of \mathbf{M} is

$$\mathbf{M}^r = \mathbf{U} \text{diag}(\mathbf{s}) \mathbf{V}^T = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} \text{diag}(\mathbf{s}) \mathbf{V}^T, \text{ and } \mathbf{U}_1 \text{ is the upper } k \text{ rows of } \mathbf{U} \text{ and } \mathbf{U}_2 \text{ is}$$

the rest of \mathbf{U} .

Intuitively: $\hat{\mathbf{x}}$ is the closest point to the subspace $\text{Span}(\mathbf{U})$. Because this property is crucial in proving the convergence in section 5.3.3, we give a formal proof here.

Theorem 1: The estimate $\hat{\mathbf{x}}$, obtained from (5.5), is the closest point to the subspace $\text{Span}(\mathbf{U})$.

Proof: For any estimate, $\tilde{\mathbf{x}}$, suppose $\mathbf{U}^T \tilde{\mathbf{x}} = \tilde{\mathbf{c}}$:

$$\|\tilde{\mathbf{x}} - \mathbf{U}\mathbf{U}^T \tilde{\mathbf{x}}\|_F^2 = \|\tilde{\mathbf{x}} - \mathbf{U}\tilde{\mathbf{c}}\|_F^2 = \|\tilde{\mathbf{x}}_{1:k} - \mathbf{U}_1 \tilde{\mathbf{c}}\|_F^2 + \|\mathbf{x}_{k+1:m} - \mathbf{U}_2 \tilde{\mathbf{c}}\|_F^2 \geq \|\mathbf{x}_{k+1:m} - \mathbf{U}_2 \hat{\mathbf{c}}\|_F^2$$

where the equality holds *iff* $\tilde{\mathbf{c}}$ is the LS solution $\hat{\mathbf{c}}$ for $\mathbf{U}_2 \mathbf{c} = \mathbf{x}_{k+1:m}$ and $\tilde{\mathbf{x}}_{1:k} = \mathbf{U}_1 \hat{\mathbf{c}}$. QED

Note: Although the solution by (5.5) is optimal in terms of the distance between the vector with missing data and the known subspace, it is not true for the new subspace of $[\mathbf{M}, \hat{\mathbf{x}}]$; because the new subspace depends not only on \mathbf{M} , but also on $\hat{\mathbf{x}}$.

5.3.2 An iterative algorithm for the problem (*Iter*)

In this subsection, we present an iterative algorithm (called *Iter*) to solve the nonlinear problem defined in section 5.2.1. Though *Iter* performs well in the vast majority of cases, it does not *always* converge to a good solution. Hence this core algorithm will be improved in section 5.4.

Algorithm (*Iter*)

(i) *Starting from a complete submatrix:* Suppose, w.l.o.g., that \mathbf{M} , after some row and column exchanges, has a block representation: $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$, where all entries in \mathbf{A}

are known, and some entries in \mathbf{B} , \mathbf{C} , and \mathbf{D} are missing. For example, permute columns so that columns with least missing values are on the left and permute rows so that rows with least missing values are towards the top. We do not need the largest submatrix – any \mathbf{A} of size $2r \times 2r$ or larger will do.

(ii) *Initialization - growing a complete sub-matrix:* (a) *Column-wise filling.* First consider the submatrix $[\mathbf{A} \ \mathbf{B}]$. Recover $\hat{\mathbf{B}}$ from \mathbf{A} by equation 5.5 and obtain

$$\begin{bmatrix} \mathbf{A} & \hat{\mathbf{B}}_1 & \mathbf{B}_2 \\ \mathbf{C} & \mathbf{D}_1 & \mathbf{D}_2 \end{bmatrix}, \text{ where the missing entries in } \hat{\mathbf{B}}_1 \text{ have been recovered and the}$$

missing entries in \mathbf{B}_2 cannot be recovered. Note: this induces a split of submatrix

D. (b) *Row-wise filling*. Similarly, recover $[C \ D_1]$ from $[A \ \hat{B}_1]$, and obtain

$$\begin{bmatrix} A & \hat{B}_1 & B_2 \\ \hat{C}_1 & \hat{D}_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}. \text{ Note: after sub-step (b), } \begin{bmatrix} A & \hat{B}_1 \\ \hat{C}_1 & \hat{D}_{11} \end{bmatrix} \text{ is now } A, \text{ the complete}$$

submatrix, in the block representation of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$; $\begin{bmatrix} B_2 \\ D_{12} \end{bmatrix}$, $[C_2 \ D_{21}]$ and D_{22}

now are B , C and D respectively.

After sub-step (a), check whether all the missing entries have been recovered. If so, terminate the initialization step and go to the iteration step; if not, go to sub-step (b). After sub-step (b), check for completion again. If all the entries have been recovered, go to the iteration step. If not, check the following condition: Is the number of the non-recovered entries before sub-step (a) the same number as after sub-step (b)? If so, the missing entries in B , C , and D cannot be recovered. If the number of non-recovered entries decreases, continue the initialization step (a) by regarding the recovered entries as "non-missing". (Note: although growing the complete submatrix to obtain the initial complete matrix, as described here, is somewhat iterative, we prefer to view this as an initialisation step to the refinement iterations that follow in the next step.)

After this initialization procedure, we obtain a recovered matrix \hat{M}_1 , which is complete; and we prepare for the iterative stage by setting a convergence measure $d_0 = \infty$.

(iii) *Iteration – refining the complete matrix*: From \hat{M}_i , obtain its closest rank- r approximation by SVD: $\hat{M}'_i = U_i \Sigma_i V_i^T$. Compute the rank- r approximation error

$$d_i = \|\hat{M}'_i - \hat{M}_i\|_F. \text{ If}$$

$$d_{i-1} - d_i < \varepsilon \tag{5.6}$$

terminate the iteration; else, from U_i , recover the missing entries in B , C , and D

by (5.5), and obtain \hat{B}_{i+1} , \hat{C}_{i+1} and \hat{D}_{i+1} . Set $\hat{M}_{i+1} = \begin{bmatrix} A & \hat{B}_{i+1} \\ \hat{C}_{i+1} & \hat{D}_{i+1} \end{bmatrix}$.

5.3.3 The convergence of the iterative algorithm

In this section, we prove a weak convergence of the iterative imputation algorithm above (thus the algorithm is independent of the initial matrix when the matrix has not been badly corrupted by the noise or by the missing data - as experimentally verified).

Theorem: The iterative algorithm above converges to a local minimum.

Proof: Suppose \mathbf{m} is an arbitrary column of \mathbf{M} , and its estimates are $\hat{\mathbf{m}}_i$ and $\hat{\mathbf{m}}_{i+1}$ at the i^{th} and the $(i+1)^{\text{th}}$ iteration steps, respectively.

$$\begin{aligned} \|\hat{\mathbf{M}}_i^r - \hat{\mathbf{M}}_i\|_F^2 &= \sum_{\text{all } \mathbf{m}} \|\hat{\mathbf{m}}_i - \mathbf{U}_i \mathbf{U}_i^T \hat{\mathbf{m}}_i\|^2 \\ &\geq \sum_{\text{all } \mathbf{m}} \|\hat{\mathbf{m}}_{i+1} - \mathbf{U}_i \mathbf{U}_i^T \hat{\mathbf{m}}_{i+1}\|^2 \\ &\geq \sum_{\text{all } \mathbf{m}} \|\hat{\mathbf{m}}_{i+1} - \mathbf{U}_{i+1} \mathbf{U}_{i+1}^T \hat{\mathbf{m}}_{i+1}\|^2 = \|\hat{\mathbf{M}}_{i+1}^r - \hat{\mathbf{M}}_{i+1}\|_F^2 \end{aligned}$$

The first inequality is from theorem 1, and the second from the SVD theorem (Golub et al. 1996). **QED**

Note: There are many ways to detect/characterise convergence. Another condition for the convergence, not so rigorous as (5.6), is to check the variation of the missing entries, i.e.

$$\|\hat{\mathbf{M}}_{i+1} - \hat{\mathbf{M}}_i\|_F < \epsilon' \quad (5.7)$$

Condition (5.7) is easier to check. However, condition (5.7) is stronger than (5.6), and it may happen that condition (5.7) fails to indicate convergence. The cases, non-convergent measured by (5.7), are described as *divergent* in section 5.5.2 and section 5.5.3.

5.4 SVD's denoising capacity vs. missing data

Vectors, with only a few "non-missing" components, may cause the iteration to "wander away" from the true solution. Moreover, even if the optimal solution,

defined as in (5.3), can be obtained^{*}, we experimentally find that these recovered vectors may degrade the accuracy that might have been gained from the other reliable data, alone. We have experimented, with some success with various strategies to detect and rectify this (see Appendix), however the true solution will be found in a closer analysis of the de-noising process. By analyzing the SVD's denoising capacity in chapter 4, we present a criterion to decide whether it is worth incorporating an incomplete vector into the iteration.

5.4.1 SVD's denoising capacity and its extension to an incomplete matrix

In chapter 4, with the tool of the matrix perturbation theory (Wilkinson 1965), the SVD's *denoising capacity* is analyzed, in terms of the size of the matrix, the noise level, and the underlying rank (equations (4.1) and (4.2) in chapter 4).

The advantage of the SFM factorization method can be ascribed to the SVD's denoising capacity. From (4.1) in chapter 4, we can see, as the size of the matrix increases, the low-rank approximation matrix approaches the noise-free matrix. That is the underlying superiority of the factorization method when applied to a complete matrix: all the feature points are treated uniformly so that most of the noise can be suppressed if the size of the measurement matrix is large enough.

However, SVD is not directly applicable when there is some missing data in the matrix. A possible solution is to first recover the missing data, using for example the iterative imputation method above; then to SVD the recovered matrix. However, when there are a lot of missing components, a vector with only a few "non-missing" components, might degrade the accuracy obtainable from the other reliable data. Yet, using only a small complete submatrix may not achieve optimal de-noising ability – clearly there is a trade-off here. This is illustrated in Figure 5.1: as the missing percentage increases, the performance deteriorates.

^{*} With synthetic data, or real data with artificial occlusion, it is, of course, easy to check for divergence and to assess how badly the solution has been degraded by the addition of one or more columns with large missing data and/or large amounts of noise.

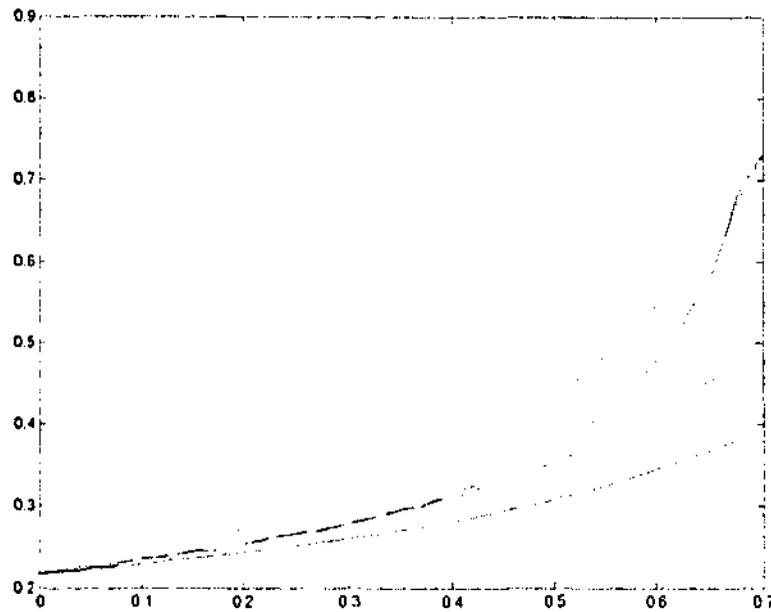


Figure 5.1: The optimal performance of the iterative algorithm. The abscissa is the missing percentage, ρ ; and the ordinate is RMS error for the iterative algorithm. Four curves are drawn: the upper dotted one is $(1-\rho)^{-1}$, the lower solid one $(1-\rho)^{-1/2}$, the dashed one in the middle is $(1-\rho)^{-0.7}$, and the solid one in the middle is the optimal performance of the proposed iterative algorithm (section 5.3.2).

A natural question arises: is it possible to find a submatrix, complete or incomplete, which is more reliable than the whole matrix? From (4.1) in chapter 4, the denoising capacity of the SVD is dependent on the ratio between $(m+n-r)r^*$ and mn : the former is the number of the independent elements of the low-rank matrix (Shum et al. 1995), and the latter is the number of the variables in the matrix. From this fact, we postulate that the incomplete matrix approximately has similar “denoising capacity”.

Hypothesis 1 (the denoising capacity of the incomplete matrix): Suppose there are p ($p \geq (m+n-r)r$) “non-missing” components in a matrix \mathbf{B} , and each row (column) has at least r “non-missing” components. The best estimate of \mathbf{B} , $\hat{\mathbf{B}}$, should have the following property:

$$E |\hat{B}_{i,j} - A_{i,j}| = \sigma \sqrt{\frac{r(m+n)-r^2}{p}} = \sigma \sqrt{\frac{r(m+n)-r^2}{mn}} \sqrt{\frac{mn}{p}} = \sigma \sqrt{\frac{r(m+n)-r^2}{mn}} \sqrt{\frac{1}{1-\rho}} \quad (5.8)$$

* Please note that in Shum’s formulation (Shum et al. 1995) the mean is also considered, so in that case there are $(m+n-r)r+n$ independent variables.

where ρ is the percentage of the missing data.

Compared with the denoising capacity of the complete matrix, the error in the incomplete matrix should increase by $\sqrt{\frac{1}{1-\rho}}$ as a function of the missing percentage. The RMS error index of the iterative algorithm approximately follows $(1-\rho)^{-0.7}$ * (see Figure 5.1), when the percentage is less than 0.5 - not exact agreement but still useful.

We employ (5.8) as a criterion as to whether it is worth incorporating a vector, with missing data, into the iteration. For an incomplete matrix with a rank of r , all of whose columns and rows have at least r "non-missing" components, we define its *unreliability* as the ratio between the number of its independent variables and the number of non-missing components:

$$c = \frac{r(m+n) - r^2}{p} \quad (5.9)$$

Thus, we propose to use the following strategy: first, use the iterative algorithm in section 5.3.2, to recover the most *reliable* incomplete sub-matrix, which has the minimal unreliability ratio; then, project other columns (rows) on it, if required, using the imputation method. Specifically for SFM, our strategy is: first reconstruct the 3D scene and the cameras by the factorizing *the most reliable measurement matrix* (obtained by the algorithm in section 5.4.3); then to estimate the positions of *other* feature points and *other* camera matrices, using the techniques in (Tomasi et al. 1992).

5.4.2 The minimal unreliability ratio in SFM

It is an NP-hard problem to find the submatrix that has the minimal unreliability ratio. Here, we propose a simple approach: to iteratively exclude the vector(s), which has the least "non-missing" components among the retained submatrix, until

* The exponent may vary in different settings: with different-size matrix or with different underlying rank. However, the optimal performance is generally better than $(1-\rho)^{-1}$.

the unreliability ratio beings to increase. Obviously, only a local minimum can be obtained, in general. However, in many cases, such as the SFM problem and the recommender system, we usually have a *thin* matrix, i.e., it has a large width or height. In the following discussion, we suppose w.l.o.g. we have an incomplete matrix whose width is much larger than its height. We then *sort the columns* so that the columns with the least missing entries are towards the left. Now we simply must find a “cut” point, beyond which to exclude unreliable columns. Indeed, if we restrict the exclusion to columns, the optimal property can be proved. Without loss of generality, suppose $n \gg m \gg r$, and the non-missing number in the i^{th} column, k_i , is descending, i.e., $k_i > k_{i+1}$ for $1 \leq i < n$. The unreliability ratio of the submatrix M_l (the left l columns of M), is:

$$c_l = (m + l - r)r / \sum_{i=1}^l k_i \quad (10)$$

We only need to prove: $c_l > c_{l+1} \Rightarrow c_{l-1} > c_l$ and $c_l < c_{l+1} \Rightarrow c_{l+1} < c_{l+2}$. That is, the curve c_l has one minimum. The first can be easily proved:

$$c_l > c_{l+1} \Leftrightarrow c_l > \frac{r}{k_{l+1}} \Rightarrow c_l > \frac{r}{k_l} \Leftrightarrow c_{l-1} > c_l .$$

Please note c_l, r, k_l are positive numbers. The second fact can be similarly proved.

5.4.3 Algorithm (*IterPart*)

In this section, we propose another algorithm, which still uses *Iter*, in section 5.3.2, at its core.

Use quick cull of cols(rows) that are not reasonable to iteratively impute (section 5.4.2).

- 1 Use the “sweeping” initialization of the core algorithm (section 5.3.2). This could be augmented with a bootstrapping strategy (Appendix), but such appears to be unnecessary in all of our experiments.
- 2 Use error norm monitored iteration of the core algorithm to convergence (i.e., the iteration step in *Iter*, in section 5.3.2).
- 3 Finally, recover the “hopeless” (the entries not recovered by 1-3), if one really must, with another approach – e.g. Tomasi-Kanade. These portions may not

be recovered well but at least they will be somewhat recovered and they will not pollute the accuracy of the previously recovered portion.

In short, *IterPart* is an improvement on *Iter* that benefits from our heuristic approach to deciding which of the entries are worth recovering directly. The difference is in step 1 (to predetermine where the core of reliable information is likely to be) and step 4 (optional recovery of those parts that are likely to be unreliable). This approach tends to converge more often than other methods.

5.4.4 Discussion

IterPart (section 5.4.3) performs almost the same as *Iter* (section 5.3.2), when there are only a few missing components. Suppose the matrix is very large: $n \gg m \gg r$. Then, the unreliability ratio for the complete matrix is about r/m . Thus, if each column (or a row) has less than r (or nr/m) missing components, the whole matrix is the most reliable one; i.e., *IterPart* is the same as *Iter*. Moreover, if the missing percentage is comparatively low, both of them are expected to have similar performance, as will be validated by experiments.

When there are a lot of missing components, *IterPart* should perform better than the *Iter*. Generally, each column (row) in the most reliable submatrix has more than $2r$ non-missing components; because the most reliable matrix would generally have an unreliability ratio less than 0.5 . If the matrix can be recovered, there should be $(m+n-r)r$ non-missing components at least, i.e., the unreliability should be less than 1 . The unreliability ratio decreases as a result of the cutting processes. The vectors with only r non-missing components are retained in the most reliable matrix *only* if the whole incomplete matrix has an unreliability ratio of 1 .

We also note that *Iter* has a risk of divergence, even when employing an additional "bootstrapping" strategy outlined in the Appendix. *IterPart* generally, does not have such problems, even without the aid of bootstrapping, as will be demonstrated by experiments.

5.5 Experiments

In this section, we compare the performance of eight approaches: *Iter* (proposed in section 5.3.2), its variant: *IterPart* (proposed in section 5.4.3), Jacobs's three methods: (*rankr* "Jacobs1", *rankrsfm* "Jacobs2", and *rankrsfm_tpose* "Jacobs3"), and Shum's method, also with three variants: *Shum1*, *Shum2*, *Shum3* (starting from Jacobs' methods above). We use rank4 versions of Jacobs routines, sidestepping the erroneous centroid subtraction in the presence of missing data (Heyden et al. 1998; Kahl et al. 1999). We present 4 groups of experiments, two using synthetic data, another from the *box* sequence, which was also used by Jacobs (Jacobs 1997; Jacobs 2001), and the other from the *dinosaur* sequence, which is somewhat more challenging.

In the first group of experiments, we concern the convergence of the core iterative algorithm, *Iter*. In the other groups of experiments, we focus on stability since *Iter*, *Shum1*, *Shum2*, and *Shum3* have almost the same performance *when they converge*. *IterPart* has a very small risk of divergence. It should be very stable because only the most reliable submatrix is used in the iteration, where each row (column) *generally* has more than $2r$ non-missing components and $r+1$ *at least*. Indeed, no divergence case has been found in all 20,000 cases we examined (20-noise-level \times 10-level-of-missing-percentage \times 100-times repetition).

5.5.1 Only one unknown entry

Consider a matrix $\tilde{\mathbf{M}} \in R^{10 \times 10}$, whose rank is 3. $\tilde{\mathbf{M}}$ is corrupted with Gaussian noise (zero mean and unit variance) producing \mathbf{M} , which is observed. Specifically, in Matlab notation, $\tilde{\mathbf{M}} = \text{randn}(10,3) \times \text{randn}(3,10) \times 5$ and $\mathbf{M} = \tilde{\mathbf{M}} + \text{randn}(10,10)$. Suppose a single element, $M_{1,10}$, is unknown.

In this experiment, in order to evaluate the algorithm, we also search the neighborhood of the candidate solution, by perturbing the estimated value, $\hat{M}_{1,10}$. We compute the distances of 200 perturbed matrices, $\tilde{\mathbf{M}}^i$, respectively to their rank-3 approximations, $\hat{\mathbf{M}}^3$, where $\hat{M}_{1,10}$ takes one of 200 values centred around

$\hat{M}_{1,10}$, i.e., $\hat{M}_{1,10}^i = \text{step} \times i + \hat{M}_{1,10}$ for $i = -100:-1, 1:100$. When the step is small (e.g., 0.1), we search a small area: while a large step (e.g., 3) is used to search a large area. Figure 5.2 shows two of these experiments, one of which is denoted by the solid curves and the other by the dotted curves. Two curves in the lower part are from the experiment using a smaller step and the other two curves from the larger step. The horizontal axes are the step numbers in the above recipe for generating the perturbations: the point 0 is the solution obtained by the iterative algorithm. Note: thus the scales of the upper and lower graphs are different – the lower curves are in fact an expanded part of the upper curves. From the smaller steps, the solution appears to be a local minimum. From the larger step, we may see other local minimums or maximums.

Thus we can see examples of the iteration behaviour: suppose, for example, that the initial value of $M_{1,10}$ in the matrix corresponding in the solid-curve example is assigned the value $\hat{M}_{1,10} + 3 \times 80$, which is shown as the star, “*”, on the solid curve. Starting from here, the algorithm cannot find the correct solution. *Even worse, when the convergence condition is criterion (5.7), the iterations will proceed to the infinite if there is no other local minimum in the right part, i.e., if the convergence condition is $\|\hat{M}_{i+1} - \hat{M}_i\|_F < \varepsilon'$, defined in (5.7), the algorithm will not converge. However, it does converge under the condition of $d_{i+1} - d_i < \varepsilon$ and stops somewhere. Those cases, non-convergent measured by (5.7), are described non-convergent in sections 5.5.2 and 5.5.3.*

We have run the experiments 10,000 times, and in all of them we found good solutions, which can be regarded as the global minimum. First, the recovered data is closer to the noise-free data than the noise-corrupted one. Secondly, the distance of the noise-corrupted matrix to its rank-3 approximation is almost same as the solution by the algorithm. Thirdly, compared with the other 200 perturbed matrices selected in a large or small area around the solution produced by our method, that solution is the best one, as shown in Figure 5.2. It has to be admitted that such sampling strategy can never totally rule out the existence of other better solutions within the sampling area. However, the optimal solution, if it is not the one

obtained by our approach, must lie beyond the large area because of the smoothness of the objective function (as observed in Figure 5.2) and it is not meaningful in practice.

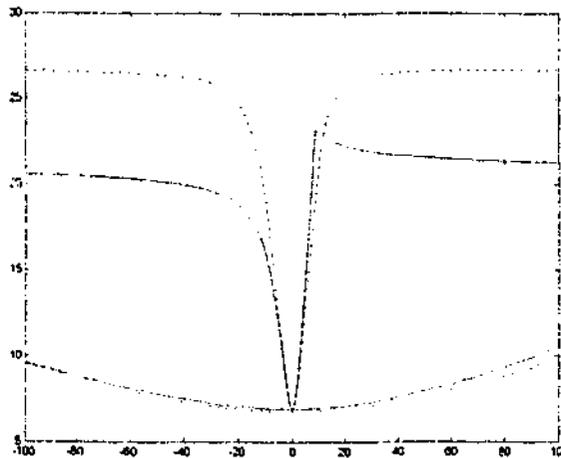


Figure 5.2: Two 10-by-10 examples with one unknown entry

5.5.2 Synthetic data in a 8-frame-and-40-point sequence

As in (Heyden et al. 1998; Kahl et al. 1999), all the synthetic image data is generated this way: the 3D feature points are uniformly distributed in a cube, within $[-500,500]*[-500,500]*[-500,500]$ units; the cameras are placed around 1000 units far away from the origin. Thus, the 2D image size is about $500*500$. Then, different levels of Gaussian noise, from 1 to 20, are added into the 2D feature points. Because the proposed algorithm has to start from a complete sub-matrix, we suppose that the first $8*8$ sub-matrix is always non-missing and the missing entries are then randomly distributed in the rest of the matrix. In addition, in order to have a recoverable incomplete matrix, we make sure that each row/column of the incomplete has 4 non-missing entries at least. The simulation repeats 100 times for each setting.

The experimental results under noise level of 1, 5, 10, 15 and 20, are shown in Figure 5.4. Please note, we do not include those *divergent* cases for the approaches of *Iter*, *Shum1*, *shum2*, and *shum3* (if the RMS of any iterative algorithms has a magnitude of 3 times or more than the noise level, the algorithm is regarded

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM

divergent); because the divergent cases would require a greatly expanded RMS axis. Figure 5.3 depicts the convergence rate for the iterative algorithms. Since the convergence rate is strongly dependent on the missing percentage, we only compare the average convergence rates (over different noise levels) for the same missing percentage.

We can see, from Figure 5.4, that the proposed iterative algorithm (*Iter*) has almost the same performance as Shum's, and that these four curves (*Iter* and 3 version of Shum's) merging into the second lowest trace. Another conclusion is that the more stable variant of our method (*IterPart*) shows its superiority when there is a lot of missing data, performing much better than *Iter* and *Shum*, as expected from section 5.4. Of Jacobs' methods, the *rankrsfm* performs best, good enough to be the initial point for the iterative algorithms. Note, *rankrsfm_tpose* is much worse than *rankr*. Though the three versions of Shum's algorithm (starting from the three versions of Jacobs as their initial matrix) perform identically with *Iter* when they converge, Figure 5.3 shows that *Iter* generally converges at least as reliably. Note, the improved algorithm, *IterPart* converges 100% of the experiments.

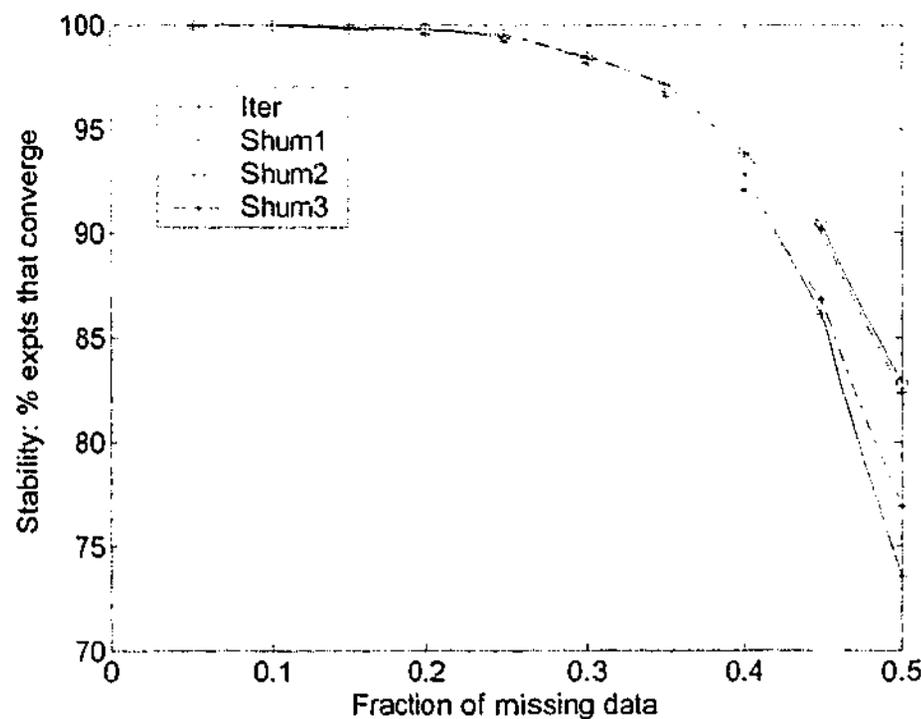
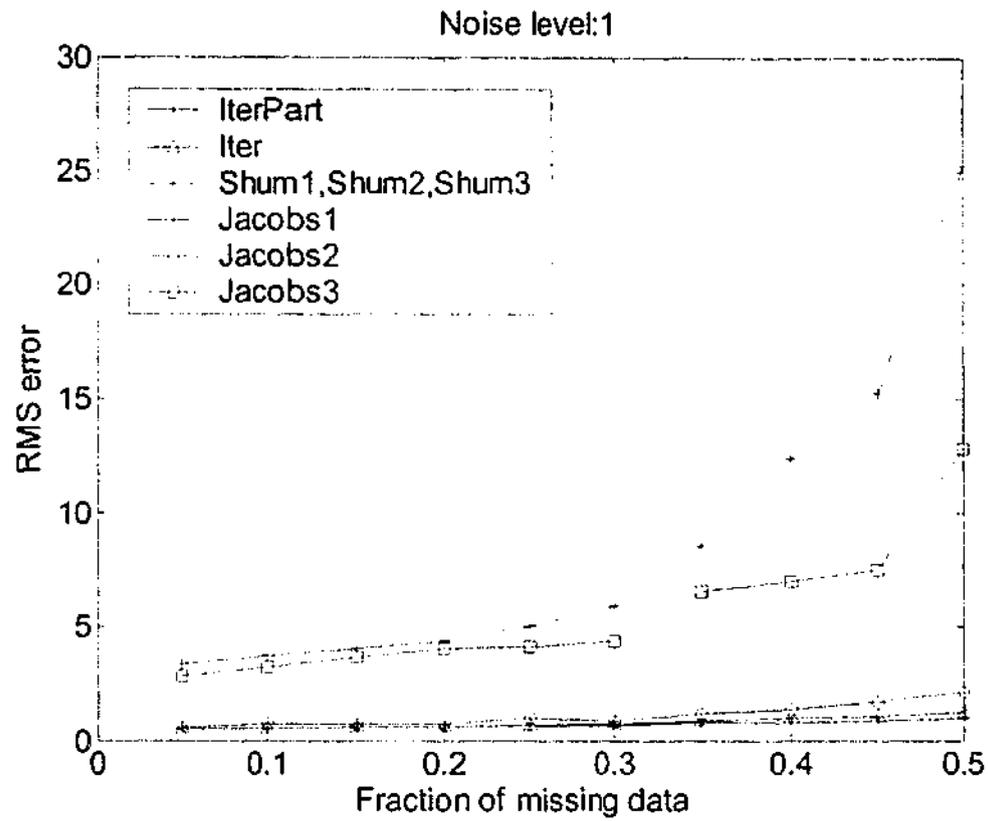
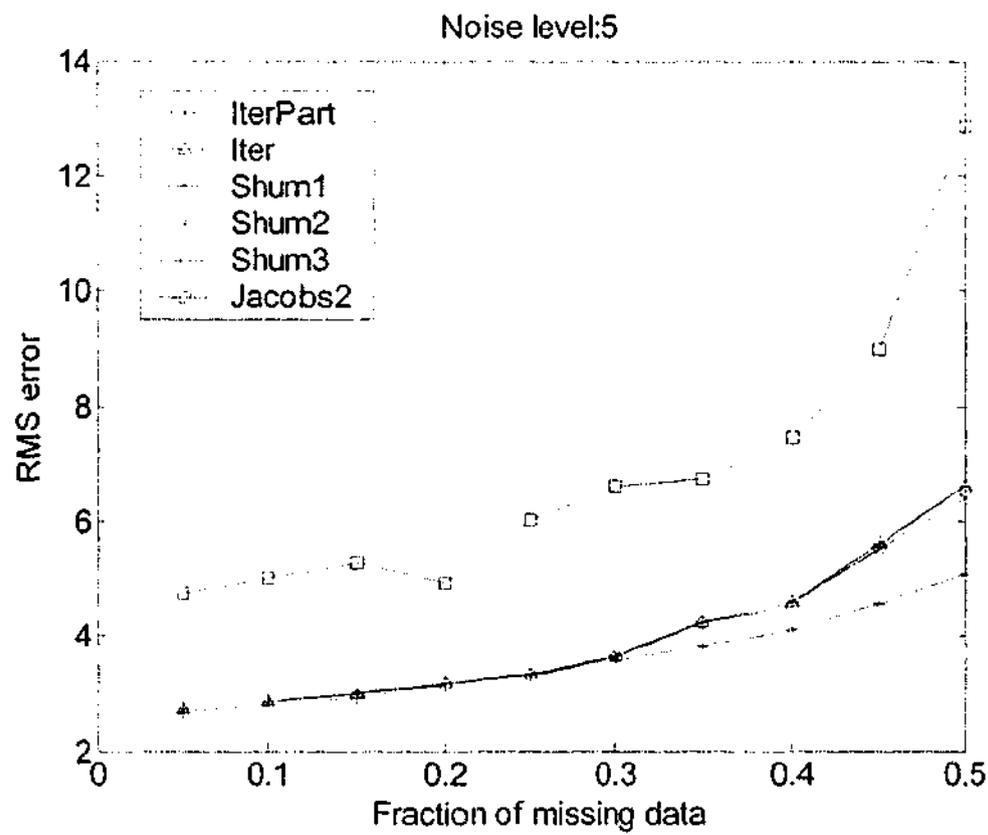


Figure 5.3: The convergence rate of four iterative methods against the missing entry fraction. Dotted curve with plus (+): *Iter*; solid curve with circle: *Shum+rankrsfm*; dotted curve with star (*): *Shum+rankrsfm_tpose*; and solid curve with plus (+): *Shum+rankr*.

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM



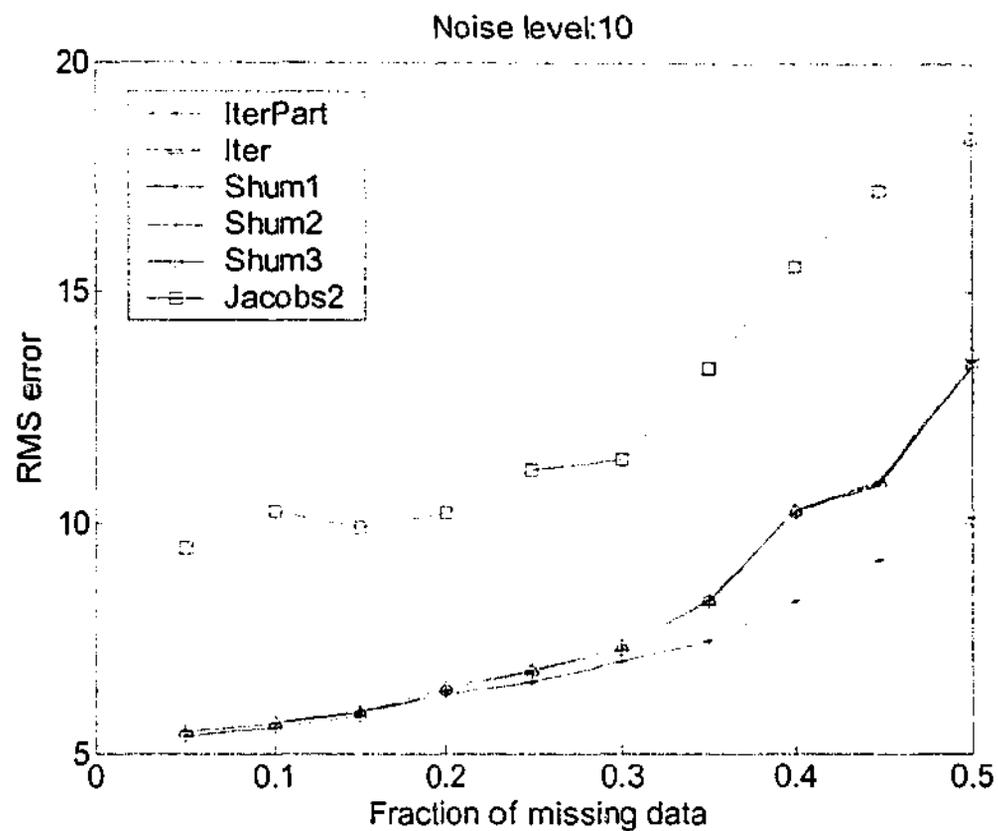
(a)



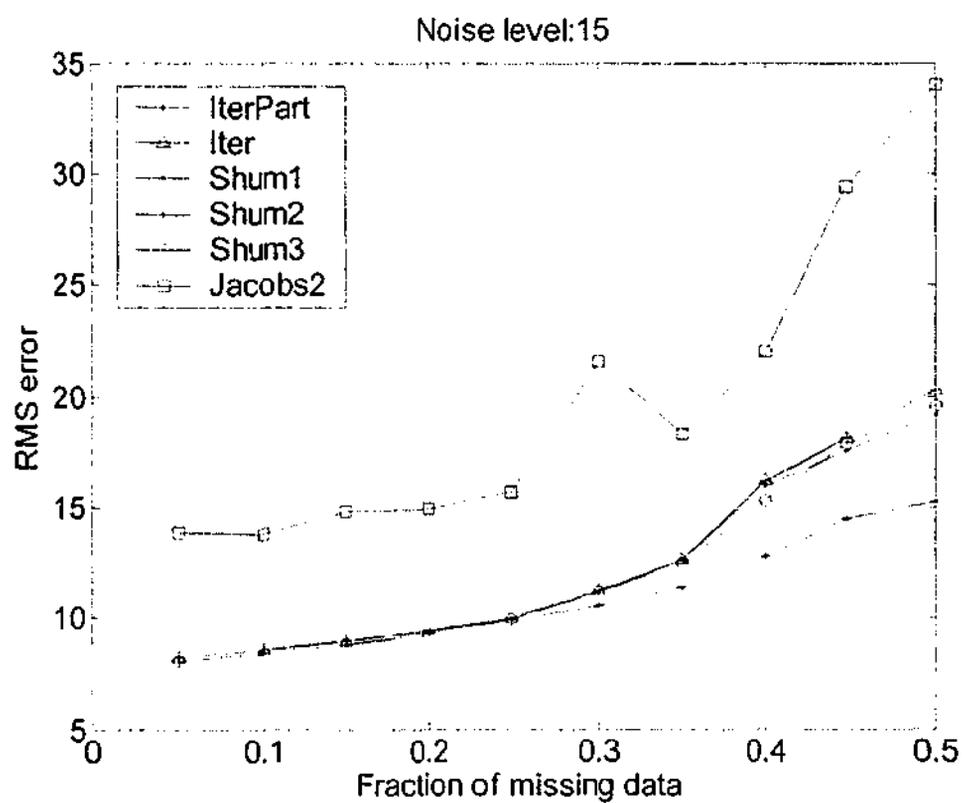
(b)

(To be continued)

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM



(c)



(d)

(To be continued)

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM

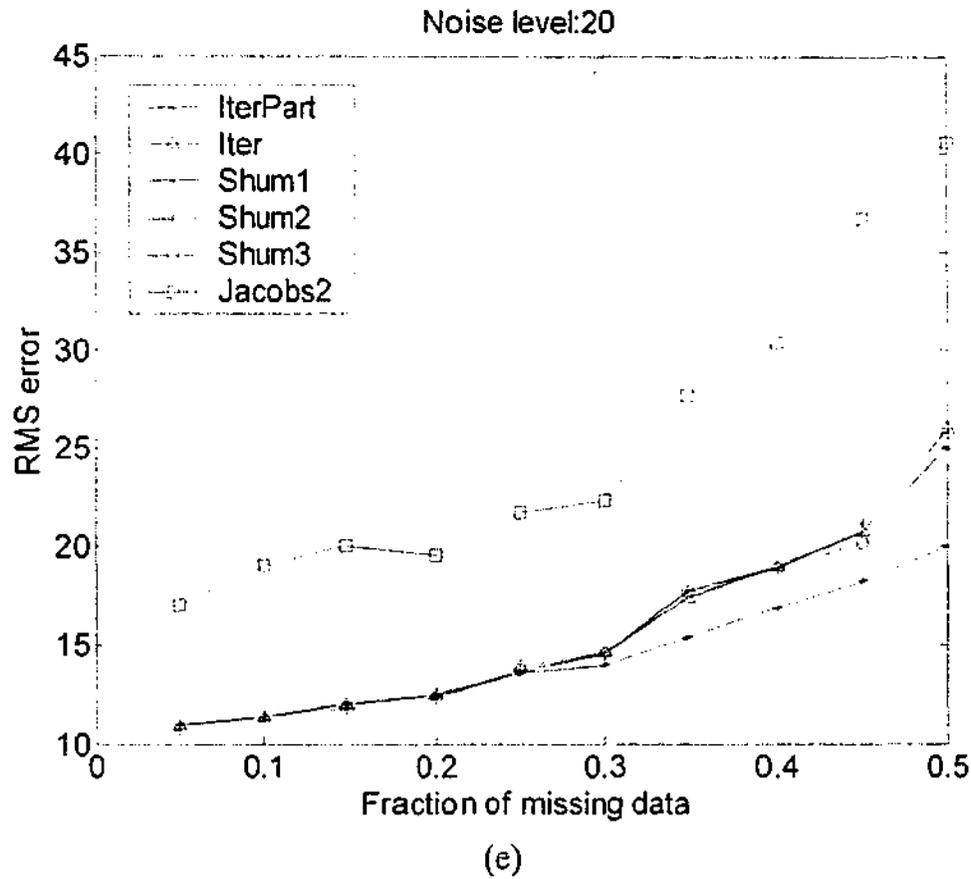


Figure 5.4: The reprojection RMS error of the eight methods, as described in the beginning of section 5.5. The abscissa is the missing percentage, and the ordinate is the reprojection RMS error. In (a), we depict all eight methods when the noise level is only 1 (From the best to the worst, they are **IterPart**, **Iter** (and 3 **Shums**), **rankrsfm**, **rankr**, and **rankrsfm_tpose**); while, in (b-e), with noise levels of 5, 10, 15 and 20, respectively, only six methods: three versions of Shum's method, **Iter**, **IterPart**, and the best Jacobs' method ("**rankrsfm**"), are depicted, in order to make the comparison visible. **IterPart** is the best, and **rankrsfm** is the worst one, and the other four have almost the same performance.

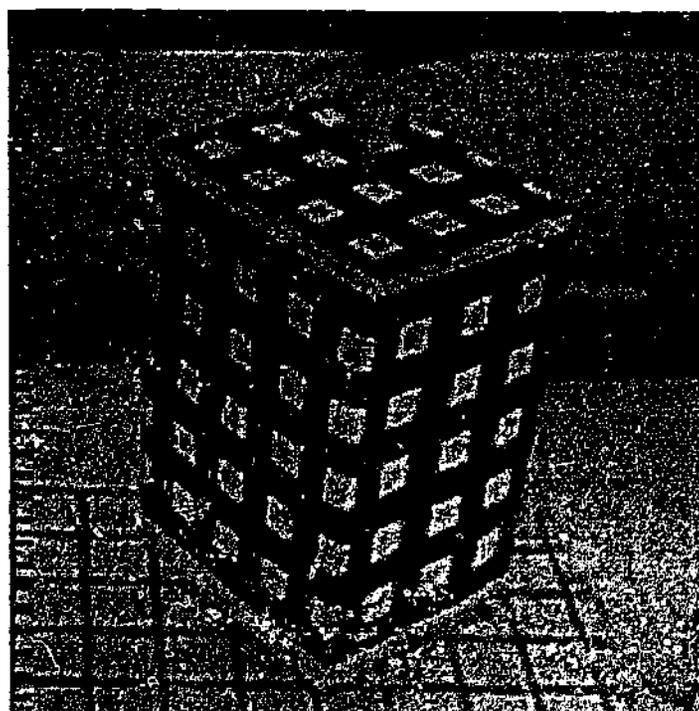
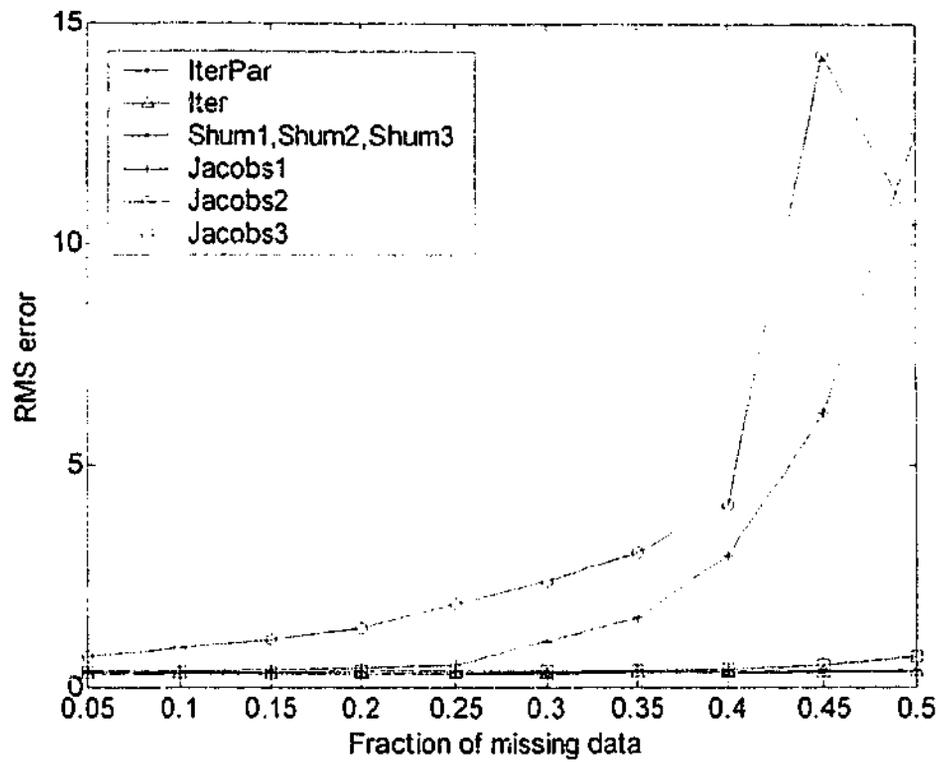


Figure 5.5: One frame of the box sequence.

5.5.3 Box sequence

Here, to test the algorithms on real data, we use the box video, which was used in (Jacobs 1997; Jacobs 2001). The sequence consists of 40 feature points across 8 frames. One frame is shown in Figure 5.5. As in section 5.5.2, we suppose that 8 points in 4 frames are available. This 8×8 submatrix is randomly selected. We then randomly occlude (consider as missing) the other feature points. For this example, as shown in Figure 5.6, the five methods have almost the same performance when they converge: *Iter*, *IterPart*, *Shum1*, *Shum2*, and *Shum3*. Their convergence rate is shown in Figure 5.7.



(a)

Figure 5.6: The performance of the RMS reprojection error by the eight methods on the box sequence are depicted: triangle (Δ) for five approaches (**Iter**, **IterPart**, and 3 Shum approaches), circle (o) for rankrsm_tpose, star (*) for rankr and cross(+) for rankrsm. Note: Five approaches (**Iter**, **IterPart**, and 3 Shum approaches) have almost the same performance so those five curves merged into one curve at the bottom.

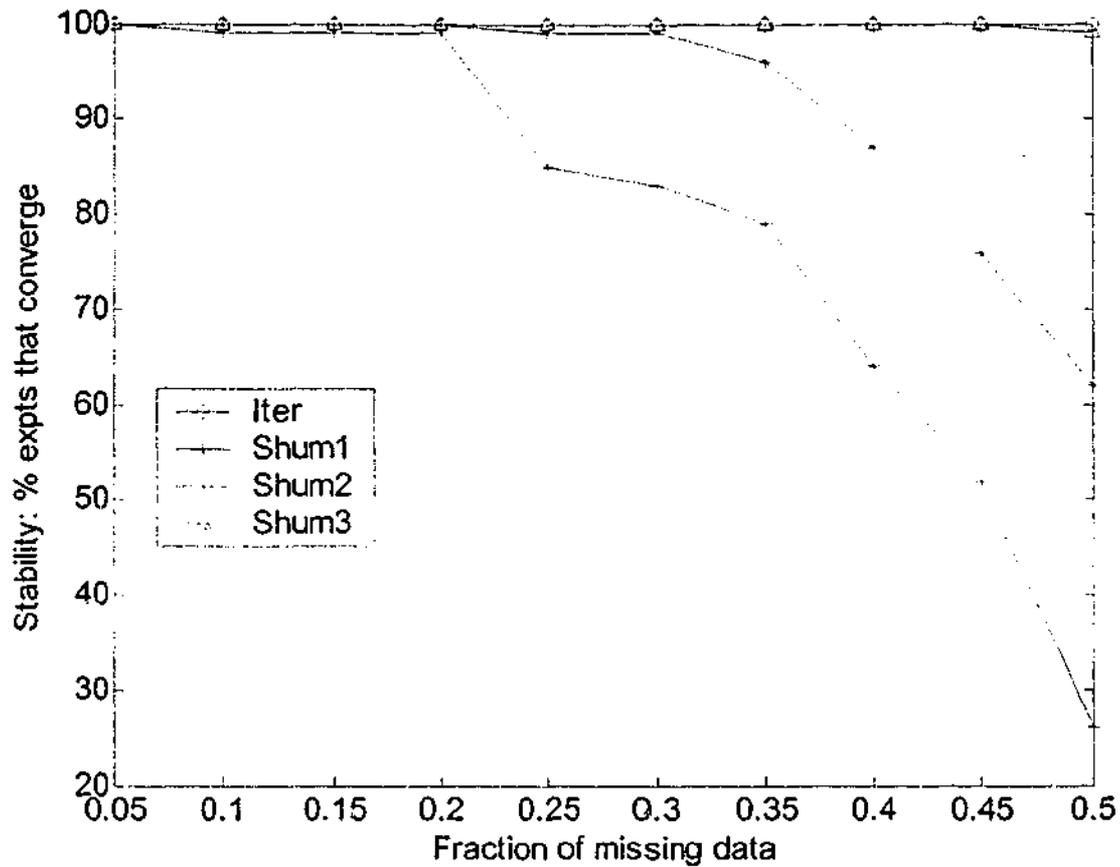


Figure 5.7: The convergence rate of the four iterative methods are depicted: circle (o) for Iter, triangle (Δ) for Shum+rankrsfm_tpose, star (*) for Shum+ rankr and cross(+) for Shum+ rankrsfm.

5.5.4 Dinosaur sequence

Here, we present an example where some data is truly missing (i.e., not artificially occluded to simulate missing data). 4983 feature points were tracked over the 36-frame “dinosaur” sequence (Fitzgibbon et al. 1998), and the 20th frame is shown in Figure 5.8, where the feature points are denoted by symbol “+”. The feature points, extracted by the Harris interest operator (Harris et al. 1988), were obtained from Oxford (<http://www.robots.ox.ac.uk/~vgg/data/>). Over the dinosaur sequence, about 90.84% data is missing; and the mask of the tracked feature points is shown in Figure 5.9, where a black pixel in (i, j) means the i^{th} feature point (in abscissa) is tracked in the j^{th} frame (in ordinate) and a grey pixel denotes the occlusion/missing data. Under the assumption of the affine camera, the measure matrix should lie in a four-dimension subspace. However, in this example, the perspective factor is not negligible, and the four-dimension subspace does not fit the feature points well

even without other noise. Thus, the model error, as well as the error introduced in the feature extraction, makes it a challenging task to recover these missing feature points. We note that the projective model was used, by Martinec (Martinec et al. 2002), to recover the dinosaur sequence. It is beyond the scope of this chapter to tackle such a setting, but we find that our results, even in the inferior affine setting, are approximately same, at least as far as one can determine from gross statistics, as Martinec's results (Martinec et al. 2002).

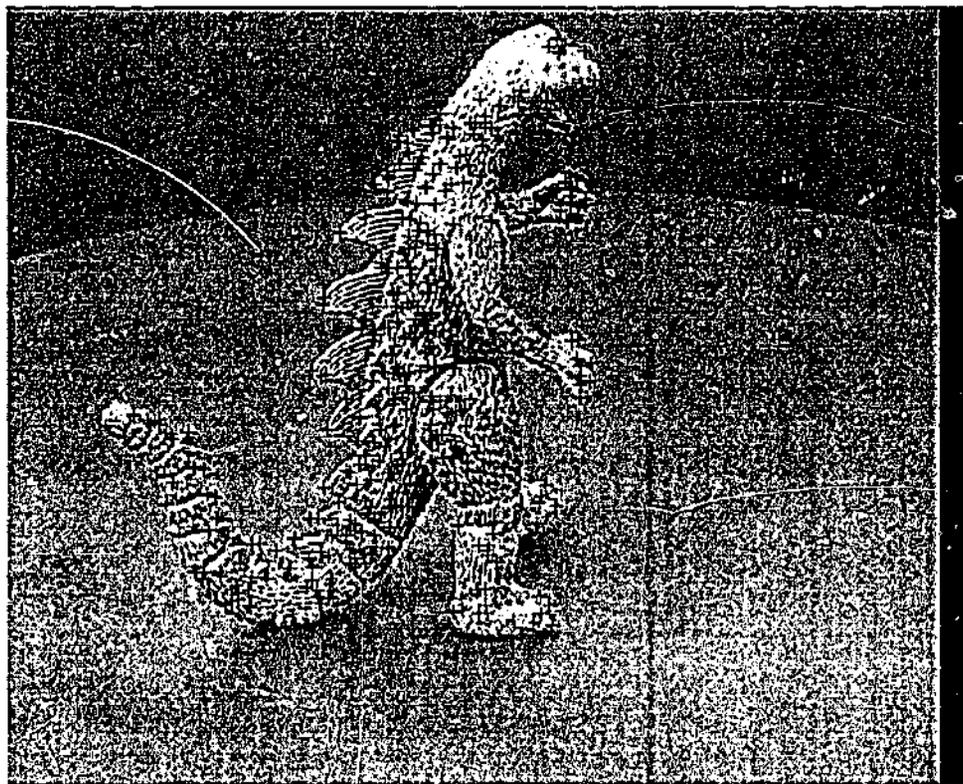


Figure 5.8: The 20th frame of the dinosaur sequence



Figure 5.9: The missing data (grey) and measured data (black) for the dinosaur sequence.

The core iterative algorithm (*Iter*) fails on the total sequence because of too much missing data and strong noise. By excluding the vectors with a few non-missing components (*IterPart*), the most reliable matrix has 36 frames and 336 feature points, with an unreliability ratio of 0.2892 , where each point has been tracked over more than 6 (>6) frames, and each frame tracked more than 20 feature points. We compare all algorithms using this same subset of "reliable" data.

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM

First, by the core iterative method in section 5.3.2, we reconstruct the 336 (“most reliable”) feature points, as shown in Figure 5.10, where about 77% data is missing. The result by Jacobs’ method under affine camera, as shown in Figure 5.11 (a), is unsatisfactory. When the initial result is not accurate enough, Shum’s approach tends to diverge, or become trapped in a local minimum, as shown in Figure 5.11 (b-c). The recovered tracks by the proposed method *Iter* are shown in Figure 5.11 (d). (Which is, of course, the same as that by *IterPart* since we have pruned.)

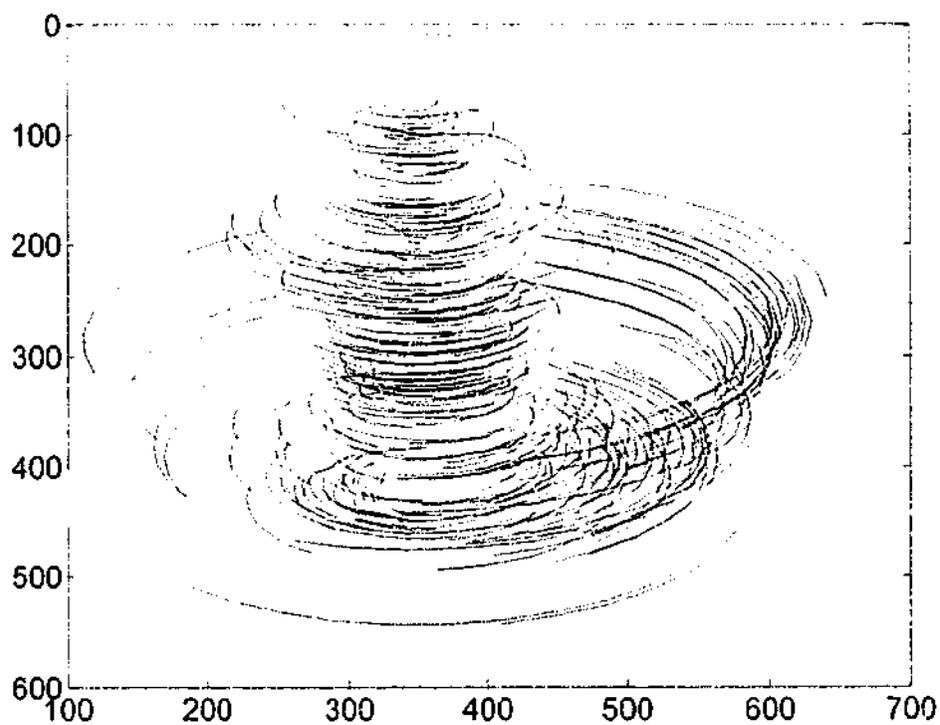
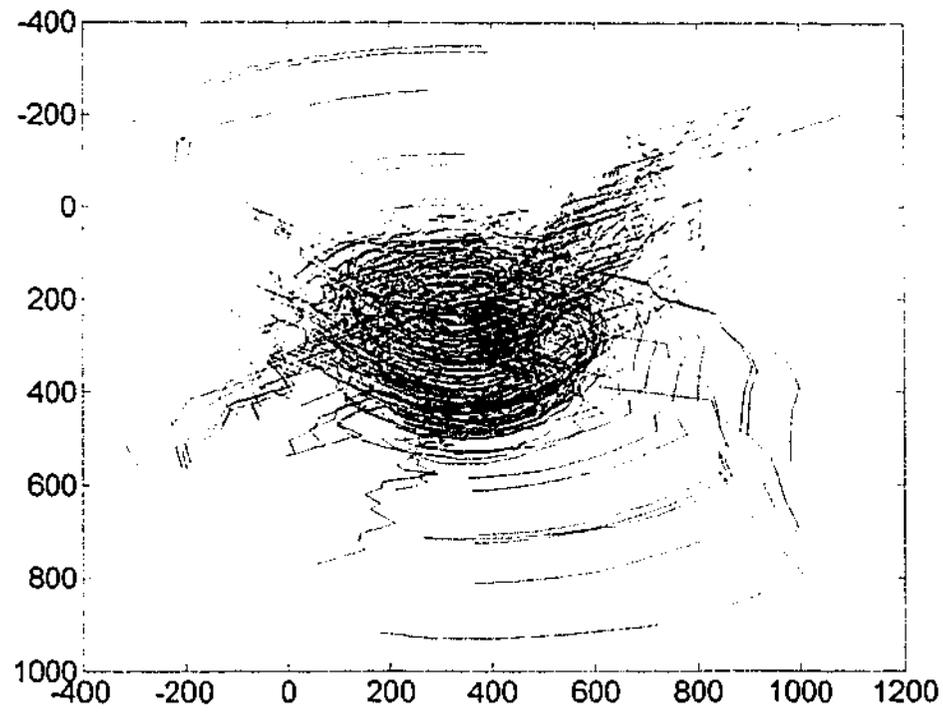
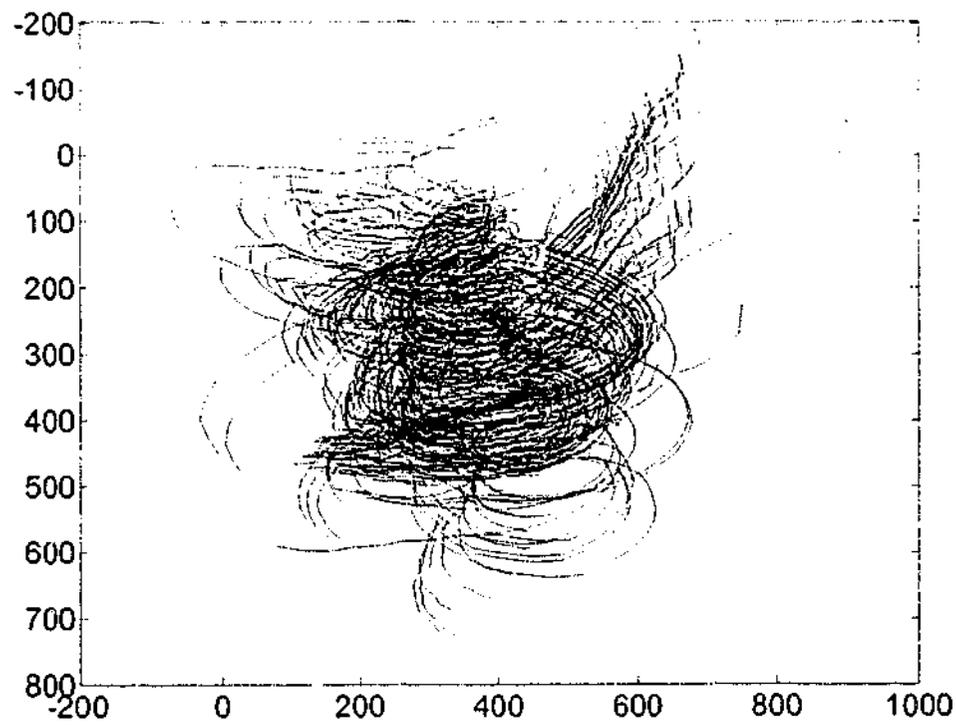


Figure 5.10: The 336 tracked feature points over 36 frames

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM



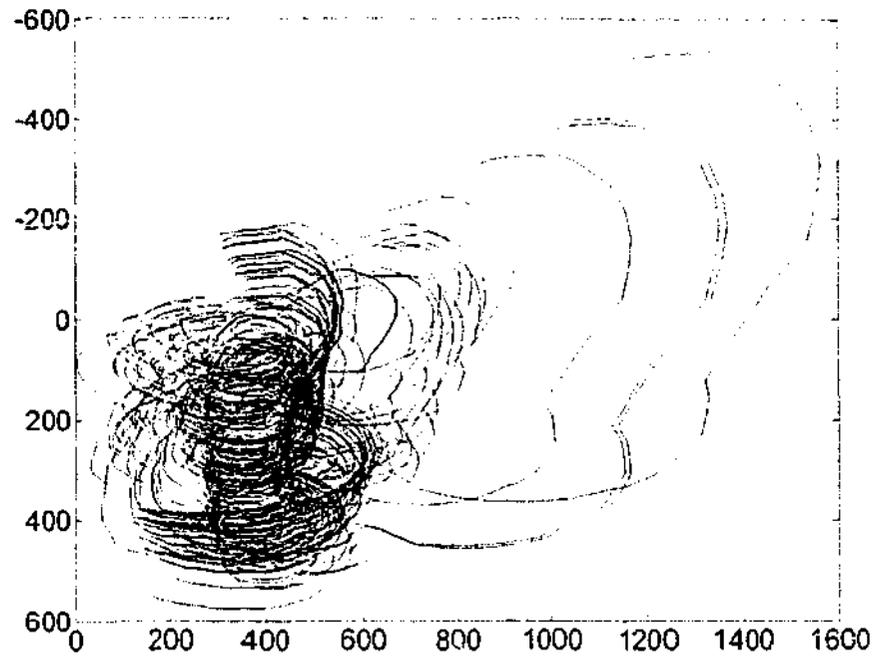
(a)



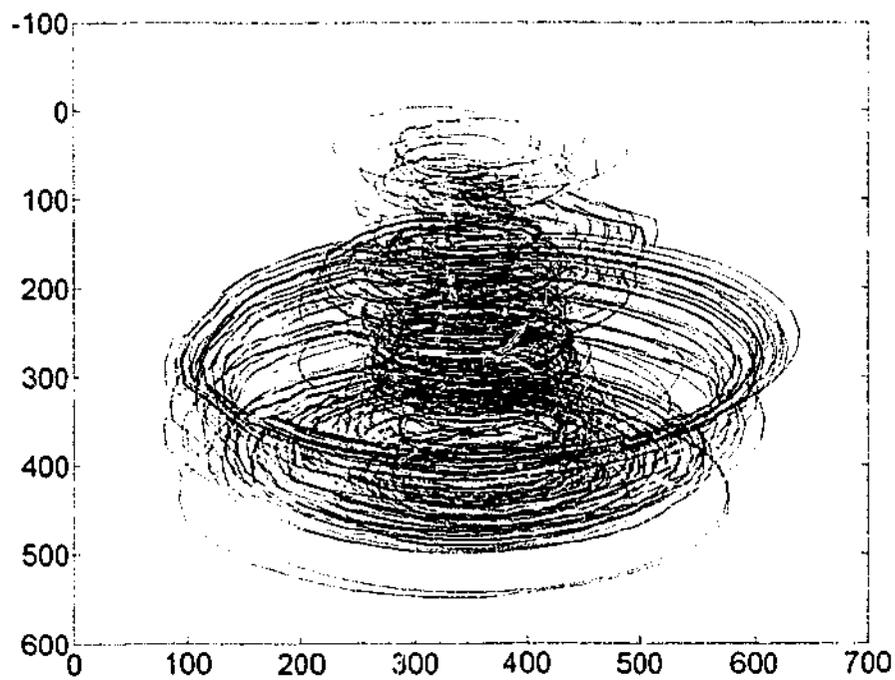
(b)

(to be continued)

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM



(c)



(d)

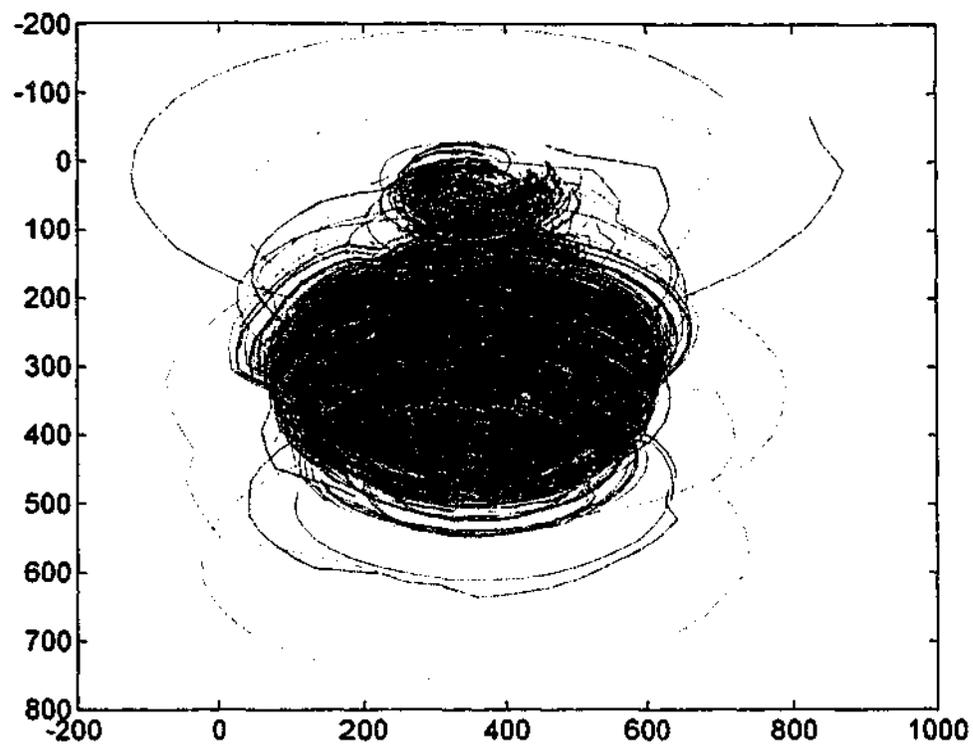
Figure 5.11: The 336 recovered tracks by the Jacobs' and Shum's and the proposed methods: (a) Jacobs, (b) Shum's result after 100 iterations, (c) Shum's result after 400 iterations, (d) The 336 recovered tracks by Iter.

Chapter 5: Recovering the missing components in a large noisy low-rank matrix: Application to SFM

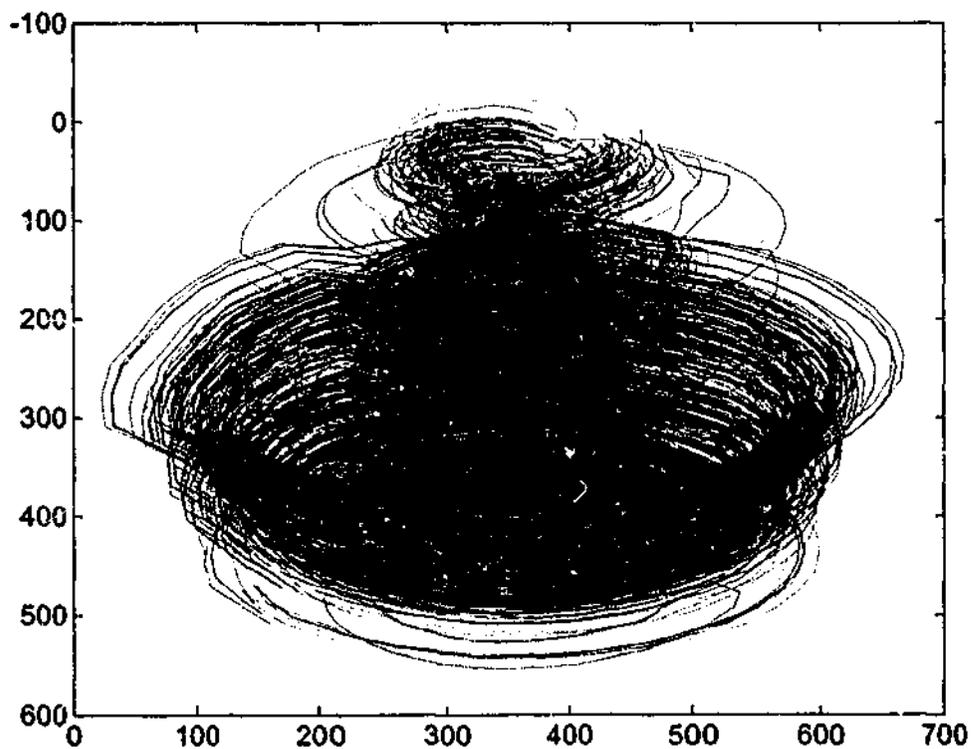
By combining Jacobs' method (Jacobs 1997; Jacobs 2001) and Sturm and Triggs' projective factorization method (Sturm et al. 1996), a good result over the whole sequence was reported (Martinec et al. 2002): the mean reprojection error per image point, measured by pixels, was reported as 1.76 pixels, and the maximal reprojection error was reported as 73.9 pixels. The mean error and maximal error were reported as 0.64 and 41.5 pixels (respectively) after bundle adjustment.

However, the above indexes (mean error and maximal error) may be sometimes misleading in assessing the performance of the algorithms as we demonstrate here. Using the stable variant of the proposed iterative method (*IterPart*), we conducted some experiments over two selections of the data: a) the whole 4983 feature points, and b) with only the 2683 feature points that were tracked over more than 2 frames. (2300 feature points were tracked only over 2 frames in the dinosaur sequence!) Our results of the reprojection tracks, for 4983 and 2683 feature points respectively, are shown in Figure 5.12 (a-b). Obviously, the result from 2683 features is much better than that from 4983 features. The recovered tracks should be approximately elliptical, because the sequence was taken while the dinosaur was on a rotating turn-table (Fitzgibbon et al. 1998). Note: all the wild recovered tracks in the first experiment are from the 2300 feature points which have been tracked over only 2 frames – thus the likely reason for such sensitive behaviour in Figure 5.12 (a) is that some feature points are tracked only over 2 frames (any noise in these features is likely to be influential). Contrast the visual quality with the impression conveyed by the mean/maximal error for 4983 and 2683 features, which are respectively 1.8438/72.4467 and 2.4017/72.4467 pixels; obviously these measures alone are misleading since the reconstruction from the case with only 2683 features scores worse although it has no wild recovered tracks. In fact, the mean/maximal error for the 2300 feature points tracked over only 2 frames is only 0.4088/ 7.8093 pixels. Since we only have the measures, as reported by Martinec (Martinec et al. 2002), it is not clear whether his results may have included such wild (and wrong) recovered tracks.

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM



(a)



(b)

Figure 5.12: The recovered tracks over 36 frames, (a) for 4983 points, and (b) for 2683 points.

5.6 Conclusion

The main contribution of this chapter is the development of a criterion one can use to recover the most *reliable* submatrix – i.e., to decide which parts of a matrix contain too many missing values to be included in the imputation. We also propose an iterative algorithm to employ the above criterion to the problem of missing data in a large low-rank matrix and we prove its convergence. In the cases, where the matrix has been badly corrupted by the missing data, the approach we propose is superior to other approaches. We avoid the NP-hard problem of finding the largest complete submatrix, as one does not need to start with a very large complete submatrix in our approach. Due to the convergence (toward the optimal solution, as demonstrated by the experiments), one can expect to arrive at the same solution even when starting from different complete sub-matrices.

As a result of our work, we also draw to the attention of the reader a salutary message regarding the use of simple error measures in making decisions about the superiority of one algorithm over another. It may be the case, as we demonstrated, that an approach with several very bad tracks, scores better than a method with generally very good tracks. Some care must be taken in assessing the contributions of studies that report only a single such measure (see also appendix A.4).

Appendices

A.1 “Bad-behaviour” and a bootstrapping strategy

As noted in the introduction, for the proposed core iterative imputation method (*Iter*) only the convergence to a local minimum is proved. The worse case scenario is that, some components “wander away” from the underlying ground truth as the iteration proceeds. We call this phenomenon “bad behaviour”. Some vectors have polluted the first r components and the remaining data cannot “correct” the values that have “wandered”. By an example (F. De la Torre et al. 2003), it has been shown that, if one data (an outlier) has 10 times the energy as the sum of the rest of the data, the outlier becomes the first principal component, and the first and the

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM

second original principal components become the other two principal components, approximately. Such a fact can be easily proved by the matrix perturbation theory (Wilkinson 1965), by regarding the outlier as the signal matrix and the original signal matrix as the perturbation.

If we followed the algorithm *Iter*, outlined in section 5.3.2, we may observe this *in a few cases* (although *very rare*, it *does* occur), when the percentage of the missing data is very high. The problem is with the initialization step. In the bad cases, the initialization step in the algorithm usually needs a few loops to obtain a complete initial matrix. However, no refinement is made on the newly increased submatrix before it continues to absorb other columns/rows.

In practice, such a phenomenon can be easily detected. From experiments, we found that the energy in $\|\hat{\mathbf{M}}_{i+1} - \hat{\mathbf{M}}_i\|_F^2$ concentrates in a few missing-values, mostly in one or two columns (rows). Having detected the likely “wandering”, we can attempt to “purify” the matrix in the initialization phase or in the iteration phase. We can first regress these bad (one or two) columns (or rows) against the other columns (or rows), and then continue the iteration. Another is to restart the algorithm: in the initialization step, using those columns (or rows) that do not produce such bad behaviour producing a partially complete matrix; then regress the columns (rows) with bad behaviour against the partially complete matrix before re-starting the iteration. The second strategy, experimentally, performs better than the first.

The “afterward” bootstrapping strategy is not ideal because of its time consuming. Generally, the wandering-away behaviour occurs with those columns, with only r (or slightly more than r) “non-missing data”; because the noise in such cases can be influential, especially when the subspace is ill-conditioned. For other columns, with only a few missing components, the imputation method of (5.5) is intrinsically an overdetermined system; therefore, it can resist noise to some extent, and consequently, it is unlikely that the wandering-away behaviour occurs with these vectors.

Thus, we propose the following bootstrapping strategy* to overcome the wandering-away: to recover those columns (rows) with fewer missing values first, i.e., to recover the more stable vectors in the inner initialization loops. In order to reduce the computation loops in the initialization step, we suggest that only those columns (rows), with more than (or equal to) $2r$ non-missing data, should be incorporated into the complete submatrix, by using the imputation method (equation 5.5).

Such a strategy raises another issue: in some cases, the complete submatrix stops increasing because no incomplete vector has more than (or equal to) $2r$ “non-missing” values. In such cases, one can temporarily relax the constraint of requiring $2r$ “non-missing values” – using columns (rows) with $2r-1$ “non-missing values” (even as low as r if need be) to break the impasse and then resume with the more conservative demand of at least $2r$ “non-missing values”.

This bootstrapping strategy can increase the robustness of the algorithm, especially when there are a lot of missing components; while it only incurs a little computation overhead—one or two more loops in the initialization step. However, we have found a similarly motivated procedure that makes this bootstrapping largely redundant (section 5.4).

A.2 Revisiting the objective function in (5.5.3)

As stated in section 5.2.1, our objective function is subtly different from that, used in Shum’s approach (Shum et al. 1995). However, under the strong convergence condition (5.7), the error index for the missing components, $\sum_{(i,j) \in \Xi} (\hat{M}_{i,j} - \hat{M}_{i,j}^r)^2$, where $\Xi = \{(i, j) \mid M_{i,j} \text{ is unknown}\}$, approaches zero during the iterations. Thus, the objective function of (5.3), under the convergence condition of (5.7), is effectively same as Shum’s objective function. It will also be proved by experiments that virtually the same solution is obtained by our method in section

* In (Brand 2003), a similar bootstrapping strategy was employed to make the imputation method robust.

5.3.2 and by Shum's method, *providing* both of them converge. In practice our approach converges far more reliably.

A.3 Difference from the imputation in (Troyanskaya et al. 2001)

The iterative algorithm in section 5.3.2 (*Iter*) has been loosely anticipated by the method in (Troyanskaya et al. 2001). Here, we describe the difference between the proposed algorithm in section 5.3.2 and the algorithm in (Troyanskaya et al. 2001). As noted in the introduction, the iterative imputation method in (Troyanskaya et al. 2001) can not be shown to converge, although the iteration may stop after a few loops. The problem with the method in (Troyanskaya et al. 2001) lies in its updating procedure in the iteration. In (Troyanskaya et al. 2001), even if there is more than one missing component in one column, only one missing data is updated at a time; by regarding all other components known, including other missing data that has been estimated. Thus, k applications of updating are needed for a column, where k components are missing. Note: if every incomplete vector has only one missing entry (an entirely unlikely event) then the method is same as *Iter*, outlined in section 5.3.2. However, if there is more than one missing component the two are not equivalent and any method that can only recover one missing entry at a time raises the question: which imputation order should be taken? After some components have been updated, should their old or new values be employed in the sequential estimation for other missing components? Generally, for any sequential updating, a different estimate from that, by (5.5), would be obtained, i.e., the estimate in (Troyanskaya et al. 2001) does not have the nice property that it is the closest point to the current subspace. Consequentially, no convergence can be promised in the iterative method in (Troyanskaya et al. 2001).

A.4 RMS and Re-projection error

Generally, the root mean square (RMS) of the reprojection error is used to evaluate the performance of the reconstruction algorithm. However, the reprojection error index, in the real data sequence, may be misleading unless we have the ground truth. We illustrate the reason for our cautionary note here.

Chapter 5: Recovering the missing components in a large noisy low-rank matrix: Application to SFM

In chapter 4, it is proved that as the size of the matrix approaches infinite, its low-rank approximation approaches the underlying noise-free matrix. Consequently, for a very large matrix, if we compare its low-rank approximation with the noise-corrupted matrix, the residuals are approximately the added noise; yet if we compare with the ground truth (the uncorrupted matrix) the error should be around 0. From Figure 5.13, we can observe this point: a series of synthetic measurement matrices ($\tilde{\mathbf{M}}$) are generated and i.i.d. Gaussian noise (0-mean-and-1-variance) is added, observing \mathbf{M} . The reprojection error, compared with $\tilde{\mathbf{M}}$ and \mathbf{M} , is depicted by the dashed curve and the solid curve, respectively. We also compare the rank-4 approximation of \mathbf{M} , \mathbf{M}^4 , with $\tilde{\mathbf{M}}$ and \mathbf{M} , the error is depicted by the dot-with-star curve and the dotted curve, respectively. Obviously, the RMS indexes, against \mathbf{M} (upper traces – “observed/noise corrupted” data), are misleading, in evaluating the performance. If we use the RMS error against the noise corrupted measurement matrix, the reconstruction error also increases as the size of the matrix increases (upper two traces); contrasting with an accepted fact that more frames produce more accurate reconstruction (Morita et al. 1997; Thomas et al. 1999). In contrast, the lower two traces (using “ground truth”) show the correct trend.

Because of this, we mainly rely on the synthetic data in evaluating the performance of the algorithms. In addition, please note that we use a different reprojection error index, from that in Jacobs’ paper: in our work the RMS error is obtained over the whole sequence, including those artificially occluded points. It makes little difference in most cases; however, the occluded points should be included in the evaluation, if possible, because in some pathological cases, we can find the reprojection error for the non-missing data is comparatively small, while that for the whole data is very large. In section 5.5.2, we can easily find such a case: with 50% data missing and a noise level of 10, where the RMS error for the non-missing entries is only 7.2098, while the RMS error for the artificially missing entries/all entries is 57.2773/38.2693, by the iterative algorithms.

Chapter 5: Recovering the missing components in a large noisy low-rank matrix:
Application to SFM

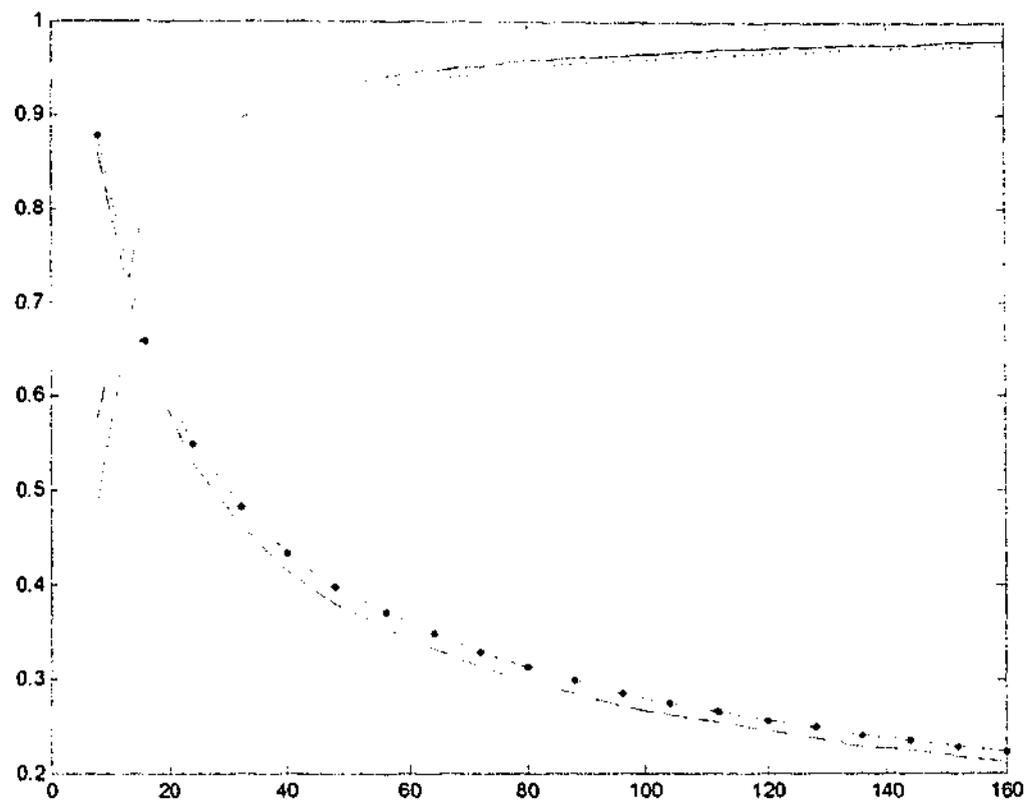


Figure 5.13: The RMS errors in the low-rank approximation matrix and the reprojection RMS error: the abscissa is the size of the square measurement matrix and the ordinate is the RMS error. The dotted with star curve and the dotted one is the RMS errors of the rank-4 approximation matrix, compared with the noise-free matrix and the noise-corrupted matrices, respectively. The dashed one and the solid one denote the reprojection RMS errors, compared with the noise-free matrix and the noise-corrupted matrices, respectively.

Chapter 6

Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting

The theory presented up until now has assumed that the noise in the measurement matrix is uncorrelated — a setting where SVD-based methods are optimal ways to “denoise”. In this chapter, we study the parameter estimation problem in a general heteroscedastic linear system, by putting the problem in the framework of the bilinear approach to low-rank matrix approximation. The ellipse fitting problem is studied as a specific example of the general theory. Despite the impression given in the literature, the ellipse fitting problem is still unsolved when the data comes from a small section of the ellipse. Although there are already some good approaches to the problem of conic fitting, such as FNS and HEIV, convergence in these iterative approaches is not ensured, as pointed out in the literature. Another limitation of these approaches is that they can't model the correlations among different rows of the “general measurement matrix”. Our method, of employing the bilinear approach to solve the general heteroscedastic parameter estimation problem, overcomes these limitations: it is convergent and can cope with a general heteroscedastic problem. However, experimental results show that *none* of the methods investigated, including ours, can be considered adequate for fitting data from a small arc of the ellipse.

6.1 Introduction

Parameter estimation in a heteroscedastic system has become an active subject, in order to overcome the difficulties of the total least squares (TLS) method (Huffel *et al.* 1991), as can be found in (Leedan *et al.* 1999; Chojnacki *et al.* 2000; Leedan *et al.* 2000; Matei *et al.* 2000; Matei *et al.* 2000; Chojnacki *et al.* 2003; Chojnacki

et al. 2004). Another active research topic is to employ the bilinear approach to calculate the low-rank approximation of a large matrix in some challenging environments (Shum *et al.* 1995; Morris *et al.* 1998; Mahamud *et al.* 2001; Guerreiro *et al.* 2003; Vidal *et al.* 2004), where the traditional SVD (Golub *et al.* 1996) does not work or its solution is not optimal. Here, in this chapter, we apply the bilinear approach to solve the parameter estimation problem in a general heteroscedastic environment. First, we review the work on these two research topics.

6.1.1 Parameter estimation in a heteroscedastic system

Many parameter estimation problems can be reduced to the following linear form:

$$\mathbf{w}^T(\mathbf{x})\theta = 0 \quad (6.1)$$

$\mathbf{w}(\mathbf{x})$ are $n \times 1$ carriers of the observed quantity \mathbf{x} , for example, a prominent problem in computer vision: conic fitting. We will study the conic fitting problem in section 6.4.

Suppose m different quantities \mathbf{x}_i ($i = 1, 2, \dots, m$) are observed. We arrange the carriers as a general "measurement matrix" $\mathbf{W} \in R^{m,n}$:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}^T(\mathbf{x}_1) \\ \mathbf{w}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{w}^T(\mathbf{x}_m) \end{bmatrix} \quad (6.2)$$

Without loss of generality, suppose $m \geq n$. If not, (6.2) is an underdetermined system. If \mathbf{W} is noise free, it is rank deficient, with a rank of $n-1$. However, it quickly becomes full rank, due to noise. Many optimization approaches and their associated objective functions have been proposed to solve this parameter estimation problem, as can be found in a comprehensive survey (Zhang 1997). Among them, a straightforward solution to (6.1) is the right singular vector of \mathbf{W} , associated with the least singular value. Such a solution is usually called as the TLS estimate (Huffel *et al.* 1991), because it minimizes the following objective function:

$$\sum \mathbf{0}^T \mathbf{w}(x) \mathbf{w}^T(x) \mathbf{0} / \|\mathbf{0}\|^2 \quad (6.3)$$

It is also the maximum likelihood (ML) estimate if the noise/uncertainty in the carriers \mathbf{w} (not the observed qualities \mathbf{x}) is i.i.d. Gaussian.

However, the assumption of i.i.d. Gaussianity usually does not hold, especially in the system (6.1), because the carriers are transformed quantities of the observed data. Even if the noise in \mathbf{x} can reasonably be assumed to be i.i.d. Gaussian, the uncertainty in the carriers \mathbf{w} often loses this property. The violation of the i.i.d. Gaussianity makes the problem challenging to the TLS method. For example, a biased estimate is obtained by the TLS method, if the noisy points come from a segment of the conic, as testified experimentally (Leedan *et al.* 1999; Leedan *et al.* 2000) and proved theoretically (Kanatani 1994; Kanatani 1996).

In order to overcome the difficulties, introduced by the non-i.i.d. Gaussianity, Kanatani analyzed this problem from a geometric statistics view and devised the renormalization method (Kanatani 1993; Kanatani 1994; Kanatani 1996). The idea behind this is to approximately equalize the noise in all carriers. Other general approaches to this heteroscedastic problem include HEIV (Leedan *et al.* 1999; Leedan *et al.* 2000; Matei *et al.* 2000; Matei *et al.* 2000) and FNS (Chojnacki *et al.* 2000; Chojnacki *et al.* 2004). In the HEIV model, the covariance matrix \mathbf{C}_i between the carriers in \mathbf{w}_i is first obtained from a linearization process, then, the parameters are estimated by minimizing the Mahalanobis distance:

$$\sum_{i=1}^m (\mathbf{w}_i - \mathbf{w}_{i_0})^T \mathbf{C}_i^{-1} (\mathbf{w}_i - \mathbf{w}_{i_0}) \quad (6.4)$$

where \mathbf{C}^{-1} is the pseudo inverse of \mathbf{C} and \mathbf{w}_{i_0} is the underlying ground truth of \mathbf{w}_i . This minimization problem is reduced to a generalized eigenproblem, where the generalized eigenvector, associated with the least eigenvalue, is needed. In the FNS method, an approximated maximum-likelihood (AML) objective function is employed. It is also reduced to a generalized eigenproblem. In (Chojnacki *et al.* 2004), it has been proved that these two approaches, HEIV and FNS, are intimately related.

6.1.2 Bilinear approach to the low-rank matrix approximation

The SVD is the basic tool for calculating the low-rank matrix approximation. The principle behind the SVD (Golub *et al.* 1996) states that any matrix, $\mathbf{W} \in R^{m,n}$, can be decomposed into

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \quad (6.5)$$

where $\mathbf{U} \in O^{m,m}$, $\mathbf{V} \in O^{n,n}$ and $\Sigma = \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_p\} \in R^{m,n}$, with $p = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$. An important fact is that one can easily construct \mathbf{W}^r , the closest rank r approximation of \mathbf{W} , measured by 2-norm or Frobenius-norm, as (Golub *et al.* 1996):

$$\mathbf{W}^r = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (6.6)$$

Specifically,

$$\|\mathbf{W} - \mathbf{W}^r\|_2 = \sigma_{r+1} \quad (6.7)$$

$$\|\mathbf{W} - \mathbf{W}^r\|_F = \sqrt{\sum_{j=r+1}^n \sigma_j^2} \quad (6.8)$$

From the optimality measured by the Frobenius-norm, the estimate by (6.6) is also the ML estimate (Press *et al.* 1992; Reid *et al.* 1996; Hartley *et al.* 2000), if the noise in the matrix \mathbf{W} is i.i.d. Gaussian.

However, the SVD method does not work on an incomplete matrix (with missing data). Moreover, the solution by (6.6) is not optimal if the noise in \mathbf{W} does not obey the i.i.d. Gaussian model. Some effort has been devoted to the missing data problem (Jacobs 1997; Kahl *et al.* 1998; Kahl *et al.* 1999; Jacobs 2001; Guerreiro *et al.* 2003; Chen *et al.* 2004) and the heteroscedastic noise problem (Aguar *et al.* 1999; Aguiar *et al.* 2000; Irani *et al.* 2000; Anandan *et al.* 2002; Aguiar *et al.* 2003). Another promising approach to these problems, as mentioned in chapter 5, is the bilinear approach (Shum *et al.* 1995; Guerreiro *et al.* 2003; Vidal *et al.* 2004)¹, where one tries to fit \mathbf{W} as the product:

¹ In (Vidal *et al.* 2004), the bilinear approach is called the PowerFactorization method.

$$\mathbf{RS} \quad (6.9)$$

with $\mathbf{R} \in R^{m,r}$ and $\mathbf{S} \in R^{r,n}$. To do so, one iteratively updates \mathbf{R} and \mathbf{S} , by alternately holding \mathbf{S} and \mathbf{R} constant, respectively:

$$\mathbf{R}_{new} = \min_{\mathbf{R}} \|\mathbf{W} - \mathbf{RS}\|_F \quad (6.10)$$

$$\mathbf{S}_{new} = \min_{\mathbf{S}} \|\mathbf{W} - \mathbf{RS}\|_F \quad (6.11)$$

The objective function in (6.10) and (6.11) can be reformulated as:

$$\|\mathbf{W} - \mathbf{RS}\|_F^2 = \sum_i \|\mathbf{w}'_i - \mathbf{r}'_i \mathbf{S}\|_F^2 = \sum_i \|\mathbf{S}^T (\mathbf{r}'_i)^T - (\mathbf{w}'_i)^T\|_F^2 \quad (6.12)$$

$$\|\mathbf{W} - \mathbf{RS}\|_F^2 = \sum_i \|\mathbf{R} \mathbf{s}_i - \mathbf{w}_i\|_F^2 \quad (6.13)$$

where \mathbf{w}'_i is the i^{th} row of \mathbf{W} and \mathbf{r}'_i is the i^{th} row of \mathbf{R} , and \mathbf{s}_i and \mathbf{w}_i are the i^{th} columns of \mathbf{S} and \mathbf{W} , respectively². If the noise in \mathbf{W} is i.i.d. Gaussian, \mathbf{r}'_i in (6.12), or \mathbf{s}_i in (6.13), can be separately calculated as the least squares (LS) solution, which minimizes

$$\|\mathbf{S}^T (\mathbf{r}'_i)^T - (\mathbf{w}'_i)^T\|_F^2 \quad (6.14)$$

$$\|\mathbf{R} \mathbf{s}_i - \mathbf{w}_i\|_F^2 \quad (6.15)$$

This way, each step of the iteration is reduced to solving a linear system:

$$\mathbf{Ax}=\mathbf{b} \quad (6.16)$$

with the LS solution as:

$$\hat{\mathbf{x}} = \mathbf{A}^{-} \mathbf{b} \quad (6.17)$$

More details can be found in (Shum *et al.* 1995), and we will revisit this point in section 6.3. In this bilinear approach to the low-rank approximation, the missing data problem can be naturally coped with, and a scalar-weighted uncertainty can also be incorporated (Shum *et al.* 1995). Moreover, this bilinear approach can be further developed to incorporate directional uncertainty (Morris *et al.* 1998) (although the measurement matrix was assumed to be complete in (Morris *et al.*

² In the following, a matrix is usually denoted by a bold capital letter, eg \mathbf{W} . Its i^{th} column is denoted by \mathbf{w}_i and its i^{th} row is denoted by \mathbf{w}'_i .

1998), the method can be naturally extended to the missing data problem, with directional uncertainty.)

6.1.3 The issues to be studied and the organization of this chapter

In HEIV and FNS, only the correlation among the carriers in $w(x_i)$ can be dealt with, although this is the most common case in practice. In this chapter, we will consider the general case, where the uncertainties in different carriers $\{w(x_i)\}$ are correlated. To do so, we rephrase the general heteroscedastic parameter estimation problem into the framework of the bilinear approach. Then, to make our theory concrete, we consider a specific computer vision task: conic fitting.

In section 6.2, we formulate the parameter estimation problem with an objective function which is subtly different from (6.4), and then we rephrase this problem in the framework of the low-rank matrix approximation. In section 6.3, we present our bilinear approach to the problem of the low-rank approximation in the heteroscedastic system. In section 6.4, we study the specific computer vision task: conic fitting, including the issue of noise level estimation. In section 6.5, our results, with comparison with other competing approaches, are presented.

6.2 The parameter estimation problem

6.2.1 Objective function to be minimized

Temporarily, we suppose that the noise model in the carriers $\{w(x_i)\}$ is known. More precisely, the correlated Gaussian model, with covariance matrix $C \in R^{m \times m}$, is employed to characterize the uncertainties among the vectorized carriers $\text{vecl}\{w(x)\}$, where

$$\text{vecl}\{w(x)\} = \begin{bmatrix} w(x_1) \\ w(x_2) \\ \vdots \\ w(x_m) \end{bmatrix} \in R^{m \times 1} \quad (6.18)$$

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting

Please note that the covariance matrix C is symmetric and positive semi-definite, and can be factorized into $C = \sum_{i=1}^m \sigma_i \mathbf{u}_i \mathbf{u}_i^T$, with $\sigma_i \geq 0$.

We start by defining the following modified Mahalanobis distance as the objective function to be minimized:

$$\min_{\mathbf{1}} (\text{vec}\{\mathbf{w}(\mathbf{x})\} - \mathbf{1})^T C^+ (\text{vec}\{\mathbf{w}(\mathbf{x})\} - \mathbf{1}) \quad (6.19)$$

where $C^+ = \sum_{i=1}^m \frac{\mathbf{u}_i \mathbf{u}_i^T}{\sigma_i}$. The vector $\mathbf{1} = \begin{bmatrix} \mathbf{1}_1 \\ \mathbf{1}_2 \\ \vdots \\ \mathbf{1}_m \end{bmatrix} \in R^{m \cdot n}$ in (6.19), with $\mathbf{1}_i \in R^{n \cdot 1}$, is

associated with a rank $n-1$ matrix $\mathbf{L} = \begin{bmatrix} \mathbf{1}_1^T \\ \mathbf{1}_2^T \\ \vdots \\ \mathbf{1}_m^T \end{bmatrix} \in R^{m \cdot n}$.

In plain language, the minimization of the objection function (6.19) is to obtain a rank $n-1$ approximation matrix, which has the shortest modified Mahalanobis distance to the general measurement matrix. If the uncertainties in the general measurement matrix are Gaussian, i.i.d. or correlated, the minimizer of the (6.19) is the ML estimate, as will be shown in section 6.3.

Assume that $\hat{\mathbf{1}}$ is the solution of the system of (6.19), and it has an associated rank $n-1$ matrix $\hat{\mathbf{L}}$. *The solution of the system of (6.1) is taken as the right singular vector of $\hat{\mathbf{L}}$, associated with the least singular value.*

If the uncertainties in different carriers $\mathbf{w}(\mathbf{x}_i)$ and $\mathbf{w}(\mathbf{x}_j)$ for $i \neq j$ are independent, the objective function (6.19) can be formulated as

$$\sum_{i=1}^m (\mathbf{w}_i - \mathbf{w}_{i_0})^T C_i^+ (\mathbf{w}_i - \mathbf{w}_{i_0}) \quad (6.20)$$

where \mathbf{C}_i is the covariance matrix for $w(x_i)$ and the matrix, $\begin{bmatrix} \mathbf{w}_{1o}^T \\ \mathbf{w}_{2o}^T \\ \vdots \\ \mathbf{w}_{mo}^T \end{bmatrix}$, has a rank of $n-1$.

Despite the similarity between (6.20) and (6.4), which is the objective function of the HEIV method, please note the difference between them. First, w_{io} is the assumed underlying ground truth, in (6.4). In contrast, in (6.20), w_{io} can be characterized by the property that its associated matrix has a rank of $n-1$. We are deliberately projecting onto the "nearest" rank $n-1$ matrix as the *starting* point of our bilinear approach to the heteroscedastic problem. Second, the modified Mahalanobis distance is employed in (6.20). In contrast, the Mahalanobis distance is employed in (6.4). They are identical if the covariance matrix is non-singular. However, there is a difference in cases, where the covariance matrix is singular, i.e., some singular values of the covariance matrix are zeroes. Obviously, if σ_i is zero, and (6.19) or (6.20) are not mathematically meaningful. It will become clear in section 6.3.1.1, that, in such cases, this can be reduced to an equality constrained LS problem (Golub *et al.* 1996). In contrast, (6.4) can be reduced to a LS problem.

6.3 The bilinear approach to the heteroscedastic parameter estimation

Although we have reformulated the bilinear approach in a simple mathematical language, as the linear system (6.16), the case becomes complicated if the uncertainty model in \mathbf{W} is not i.i.d. Gaussian. In order to simplify the development of the solution to the low-rank approximation in a general heteroscedastic system, we first consider the case of (6.20), where the uncertainties between different carriers $w(x_i)$ and $w(x_j)$ for $i \neq j$ are independent.

6.3.1 Update of \mathbf{R}

In case of (6.20), the uncertainties between different rows of \mathbf{W} are assumed to be independent, so we can separately update each row of \mathbf{R} , as in (6.14). The updating of each row of \mathbf{R} equals to solving the linear system (6.16), where the uncertainties in \mathbf{b} are modeled as correlated Gaussian noise, with a covariance matrix of \mathbf{C} . Note that, \mathbf{A} is \mathbf{S}^T , and \mathbf{b} is the transform of the i^{th} row of \mathbf{W} , $(\mathbf{w}'_i)^T$; and the estimated $\hat{\mathbf{x}}$ would be the transform of the i^{th} row of \mathbf{R} , $(\mathbf{r}'_i)^T$, which is to be updated.

The minimization objective function in the linear system (6.16) is:

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} (\mathbf{Ax} - \mathbf{b})^T \mathbf{C}^{-1} (\mathbf{Ax} - \mathbf{b}) \quad (6.21)$$

Suppose $\mathbf{C} = \mathbf{U} \text{diag}(d_1, d_2, \dots, d_n) \mathbf{U}^T$. Define

$\mathbf{Q} = \text{diag}(1/\sqrt{d_1}, 1/\sqrt{d_2}, \dots, 1/\sqrt{d_n}) \mathbf{U}^T$. The solution to (6.21) is:

$$\hat{\mathbf{x}} = (\mathbf{QA})^+ \mathbf{Qb} \quad (6.22)$$

Proof: We arrange the minimization objective function in (6.21) as:

$$(\mathbf{Ax} - \mathbf{b})^T \mathbf{Q}^T \mathbf{Q} (\mathbf{Ax} - \mathbf{b}) = (\mathbf{QAx} - \mathbf{Qb})^T (\mathbf{QAx} - \mathbf{Qb})$$

Obviously, (6.22) is the solution to the above objective function, and consequently, is the solution of (6.21).

It will become clear in section 6.3.1.1, that the uncertainties in \mathbf{Qb} are i.i.d. Gaussian and the solution in (6.22) is the ML estimate, if the uncertainties in \mathbf{b} are Gaussian.

6.3.1.1 Case with zero singular values in the covariance matrix \mathbf{C}

As we note in section 6.2.1, the modified pseudo inverse of the covariance matrix \mathbf{C} does not make sense if \mathbf{C} has some zero singular values. However, there is usually a constant carrier in (6.1), i.e., this component is noise free. Consequently, \mathbf{C} has, at least, a zero singular value. Here, we study this case and present our solution to this problem.

First, we study the covariance matrix of the transformed \mathbf{b} , $\mathbf{U}^T \mathbf{b}$.

$$\text{cov}(\mathbf{U}^T \mathbf{b}) = \mathbf{U}^T \text{cov}(\mathbf{b}) \mathbf{U} = \mathbf{U}^T \mathbf{C} \mathbf{U} = \text{diag}(d_1, d_2, \dots, d_n) \quad (6.23)$$

(6.23) means that the coupled uncertainties in \mathbf{b} have been decoupled in the transformed $\mathbf{U}^T \mathbf{b}$. If all $d_i \neq 0$ and the coupled uncertainties in \mathbf{b} are Gaussian, the uncertainties in $\mathbf{Q} \mathbf{b}$ are i.i.d. Gaussian, and (6.22) can be taken as the ML estimate.

Any zero d_i in (6.23) means that the \mathbf{u}_i -direction component of \mathbf{b} , $\mathbf{u}_i^T \mathbf{b}$, has no uncertainties or noise. Without loss of generality, we suppose the last k d_i for $i = n - k + 1, \dots, n$ are zeroes. Define

$$\mathbf{A}_1 = \text{diag}(1/\sqrt{d_1}, 1/\sqrt{d_2}, \dots, 1/\sqrt{d_{n-k}}) \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_{n-k}^T \end{bmatrix} \mathbf{A}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{u}_{n-k+1}^T \\ \mathbf{u}_{n-k+2}^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} \mathbf{A},$$

$$\mathbf{b}_1 = \text{diag}(1/\sqrt{d_1}, 1/\sqrt{d_2}, \dots, 1/\sqrt{d_{n-k}}) \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_{n-k}^T \end{bmatrix} \mathbf{b} \quad \text{and} \quad \mathbf{b}_2 = \begin{bmatrix} \mathbf{u}_{n-k+1}^T \\ \mathbf{u}_{n-k+2}^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} \mathbf{b}.$$

Now, it is clear that the uncertainties in \mathbf{b}_1 are i.i.d. Gaussian, and that \mathbf{b}_2 is noise free. Thus, the optimal estimate of (6.21) should be the solution of the following constrained minimization problem:

$$\min_{\mathbf{A}_2 \mathbf{x} = \mathbf{b}_2} \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1 \quad (6.24)$$

(6.24) is an equality constrained least squares problem, and its solution can be found in (Golub *et al.* 1996) (see the appendix).

Now, it is clear that our objective function in (6.19) or (6.20) makes sense if we adopt the interpretation of $0/0 = 0$. More importantly, the solution of (6.24) is the optimal solution of (6.21).

In contrast, if we employ (6.4) as the objective function, as in the HEIV method, in zero-singular-value cases, (6.21) will be reduced to a simple LS problem: $\min \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1$, without the equality constrained $\mathbf{A}_2 \mathbf{x} = \mathbf{b}_2$. Obviously, the

objective function of (6.4) can't be employed in such cases. In order to overcome this difficulty in (Leedan *et al.* 1999; Leedan *et al.* 2000), the constant component is not included in (6.4), and has been separately considered from the other columns.

6.3.2 Update of S

Because the uncertainties in the columns of \mathbf{W} , w_i and w_j for $(i \neq j)$ in (6.13), are not independent, the minimization process can't be separately dealt with (as (6.15) does not apply). We have to jointly solve a matrix equation: $\mathbf{AX}=\mathbf{B}$. Note that, \mathbf{A} is \mathbf{R} , and \mathbf{B} is \mathbf{W} ; and that \mathbf{X} is the \mathbf{S} , which is to be updated.

Fundamentally, we abstract (6.13) as the following minimization problem.

Suppose $\mathbf{A} = \begin{bmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_2 \\ \vdots \\ \mathbf{a}'_m \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{b}'_1 \\ \mathbf{b}'_2 \\ \vdots \\ \mathbf{b}'_m \end{bmatrix}$, with $\mathbf{a}'_i \in R^{1,n}$ and $\mathbf{b}'_i \in R^{1,r}$. Suppose $\mathbf{b}'_i{}^T$ is

corrupted with correlated Gaussian noise with \mathbf{C}_i covariance matrix, which can be factorized into $\mathbf{C}_i = (\mathbf{U}^i) \text{diag}(d_1^i, d_2^i, \dots, d_r^i) (\mathbf{U}^i)^T$. And, the uncertainties in $\mathbf{b}'_i{}^T$ are independent of those in $\mathbf{b}'_j{}^T$ for $j \neq i$. Thus, the ML estimation is to solve the following minimization problem:

$$\mathbf{X} = \min_{\mathbf{X} \in R^{n,r}} \sum_{i=1}^m (\mathbf{a}'_i \mathbf{X} - \mathbf{b}'_i) \mathbf{C}_i^{-1} (\mathbf{a}'_i \mathbf{X} - \mathbf{b}'_i)^T \quad (6.25)$$

We suppose that $d_i^j \neq 0$. If not, we can convert the problem to an equality constrained least squares problem (6.24), as in section 6.3.1.1. Define $\mathbf{\Omega}_i = \text{diag}(1/\sqrt{d_1^i}, 1/\sqrt{d_2^i}, \dots, 1/\sqrt{d_r^i}) (\mathbf{U}^i)^T$. The correlated uncertainties in \mathbf{B} can be decoupled by:

$$\begin{bmatrix} \mathbf{a}'_1 \mathbf{X} \mathbf{Q}_1^T \\ \mathbf{a}'_2 \mathbf{X} \mathbf{Q}_2^T \\ \vdots \\ \mathbf{a}'_m \mathbf{X} \mathbf{Q}_m^T \end{bmatrix} = \begin{bmatrix} \mathbf{b}'_1 \mathbf{Q}_1^T \\ \mathbf{b}'_2 \mathbf{Q}_2^T \\ \vdots \\ \mathbf{b}'_m \mathbf{Q}_m^T \end{bmatrix} \quad (6.26)$$

(6.26) equals to the following linear system:

$$\begin{bmatrix} \mathbf{Q}_1 \otimes \mathbf{a}'_1 \\ \mathbf{Q}_2 \otimes \mathbf{a}'_2 \\ \vdots \\ \mathbf{Q}_m \otimes \mathbf{a}'_m \end{bmatrix} \text{vec}(\mathbf{X}) = \begin{bmatrix} \text{vec}(\mathbf{b}'_1 \mathbf{Q}_1^T) \\ \text{vec}(\mathbf{b}'_2 \mathbf{Q}_2^T) \\ \vdots \\ \text{vec}(\mathbf{b}'_m \mathbf{Q}_m^T) \end{bmatrix} \quad (6.27)$$

where \otimes denotes the Kronecker product of two matrices, and, for a matrix \mathbf{X} with

$$r \text{ columns, } \text{vec}(\mathbf{X}) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix}. \quad (6.27) \text{ comes from the property of the Kronecker}$$

product: $\text{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\text{vec}(\mathbf{X})$. The uncertainties in the right side of (6.27) have already been made i.i.d. Gaussian. Thus, the optimal solution to $\text{vec}(\mathbf{X})$, and consequently \mathbf{X} , can be obtained by the LS estimation.

Note the solution of $\text{vec}(\mathbf{X})$ from (6.27), and consequently \mathbf{X} , minimizes the objective function in (6.25).

Proof: we arrange the minimization objective function in (6.25) as:

$$\begin{aligned} & \sum_{i=1}^n (\mathbf{a}'_i \mathbf{X} - \mathbf{b}'_i) \mathbf{Q}_i^T \mathbf{Q}_i (\mathbf{a}'_i \mathbf{X} - \mathbf{b}'_i)^T \\ &= \sum_{i=1}^n (\mathbf{a}'_i \mathbf{X} \mathbf{Q}_i^T - \mathbf{b}'_i \mathbf{Q}_i^T) (\mathbf{a}'_i \mathbf{X} \mathbf{Q}_i^T - \mathbf{b}'_i \mathbf{Q}_i^T)^T \\ &= \sum_{i=1}^n [(\mathbf{Q}_i \otimes \mathbf{a}'_i) \text{vec}(\mathbf{X}) - \text{vec}(\mathbf{b}'_i \mathbf{Q}_i^T)] [(\mathbf{Q}_i \otimes \mathbf{a}'_i) \text{vec}(\mathbf{X}) - \text{vec}(\mathbf{b}'_i \mathbf{Q}_i^T)]^T \end{aligned}$$

The uncertainties in $\text{vec}(\mathbf{b}'_i \mathbf{Q}_i^T)$ are i.i.d. Gaussian, so, the ML solution to (6.27) is the LS estimate. The LS estimate of $\text{vec}(\mathbf{X})$ in (6.27) minimizes the above objective function, and consequently, the related \mathbf{X} minimizes the objective function in (6.25). \square

6.3.2.1 Constant column in the measurement matrix

Assume that there is a constant column in the general measurement matrix, as will be found in the conic fitting, i.e. $\mathbf{M} = [\mathbf{M}', \mathbf{1}]$. In such cases, we can single out the constant column in the above updating of \mathbf{S} . The last column of \mathbf{S} , s_r , can be calculated as:

$$\hat{s}_r = R^{-1} \quad (6.28)$$

Note, in the updating of each row of R , we convert the minimization problem of (6.21) as an equality constrained LS estimation problem of (6.24). Consequently, $\hat{r}'_r S = [\hat{r}'_r, 1]$. It is clear that the approximated measurement matrix $\hat{R}S$, after each updating of R , has an exact constant column 1. This means that $1 \in \text{span}(\hat{R})$. So, (6.28) holds without any error, i.e. $R\hat{s}_r = 1$.

6.3.2.2 Discussion of the convergence of the bilinear approach

In sections 6.3.1 and 6.3.1.1, we studied the updating of R , where the objective function is (6.20). Because of the assumed independence among the uncertainties in different rows of W in (6.2), we can separately update each row of R , minimizing the associated part in the sum of (6.20). In sections 6.3.2 and 6.3.2.1, we have jointly updated S in order to incorporate the correlation among different columns of W . Thus, the objective function in (6.20) decreases after each updating step of R or S . From these observations, we can see that the bilinear approach converges, in contrast to the lack of proof of convergence of the HEIV or FNS methods.

6.3.3 A more general update

In the general case of (6.10) or (6.11) what we need to do is to solve a matrix equation:

$$AX=B \quad (6.29)$$

where $A = \begin{bmatrix} a'_1 \\ a'_2 \\ \vdots \\ a'_m \end{bmatrix} \in R^{m,r}$, $B = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_m \end{bmatrix} \in R^{m,n}$, $X \in R^{r,n}$. In a general heteroscedastic

case, the uncertainties in B are characterized by the covariance matrix C for the vectorized $\text{vecl}(B) = [b'_1 \ b'_2 \ \dots \ b'_m]^T \in R^{mn,1}$. The objective function to be minimized is:

$$\hat{X} = \min_X (\text{vecl}(AX - B))C^+ (\text{vecl}(AX - B))^T \quad (6.30)$$

Similarly, C can be factorized as $C = U \text{diag}(d_1, d_2, \dots, d_m) U^T$, and define $Q = \text{diag}(1/\sqrt{d_1}, 1/\sqrt{d_2}, \dots, 1/\sqrt{d_m}) U^T$.

First, we convert the equation of $\mathbf{a}'_i \mathbf{X} = \mathbf{b}'_i$ to $(I_n \otimes \mathbf{a}'_i) \text{vec}(\mathbf{X}) = \mathbf{b}'_i$. Then, $\mathbf{A}\mathbf{X}=\mathbf{B}$ can be rewritten as:

$$\begin{bmatrix} I_n \otimes \mathbf{a}'_1 \\ I_n \otimes \mathbf{a}'_2 \\ \vdots \\ I_n \otimes \mathbf{a}'_m \end{bmatrix} \text{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{b}'_1 \\ \mathbf{b}'_2 \\ \vdots \\ \mathbf{b}'_m \end{bmatrix} \quad (6.31)$$

and further as

$$Q \begin{bmatrix} I_n \otimes \mathbf{a}'_1 \\ I_n \otimes \mathbf{a}'_2 \\ \vdots \\ I_n \otimes \mathbf{a}'_m \end{bmatrix} \text{vec}(\mathbf{X}) = Q \begin{bmatrix} \mathbf{b}'_1 \\ \mathbf{b}'_2 \\ \vdots \\ \mathbf{b}'_m \end{bmatrix} \quad (6.32)$$

The uncertainties in the right side of (6.32) have been i.i.d. Gaussian if the uncertainties in C are Gaussian. Thus, (6.32) can be solved by the LS method, and the associated \mathbf{X} can be obtained.

6.3.4 Discussion of the optimality

From the above sections, we can see that the optimal solution, at least a local optimal solution, is iteratively obtained, if we evaluate the estimate using the objective functions in (6.19) or (6.20). However, it is not the ML estimate if the uncertainties in \mathbf{W} (or in \mathbf{b} and \mathbf{B}) are not Gaussian. Because of this, we assumed the uncertainties in \mathbf{W} (or in \mathbf{b} and \mathbf{B}) are Gaussian when we referred to the ML estimate above.

6.4 Application in conic fitting

As can be observed in section 6.3 and in the objective function in (6.19) or (6.20), the crux of our bilinear approach to the heteroscedastic low-rank approximation, and consequently of the heteroscedastic parameter estimation problem, is to obtain

the covariance matrix of the carriers w . In this section, we will study this issue, by taking the conic fitting problem as a specific example. This case has been analyzed in (Leedan *et al.* 1999; Leedan *et al.* 2000), where the covariance matrix of the carriers was obtained.

As in (Leedan *et al.* 1999; Chojnacki *et al.* 2000; Leedan *et al.* 2000; Chojnacki *et al.* 2004), we also assume that each component of the observed x is corrupted with i.i.d. and σ^2 -variance Gaussian noise, and consequently, that the uncertainties in different carriers are independent.

6.4.1 Covariance matrix in the conic fitting

A conic is characterized by the following constraint:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (6.33)$$

The carriers in (6.33) are x^2 , xy , y^2 , x , y , and 1. By the linearization, we reformulate (6.33) in the form of (6.1):

$$[x_i, y_i, x_i y_i, x_i^2, y_i^2, 1][d, e, b, a, c, f]^T = 0 \quad (6.34)$$

The conic fitting problem is to estimate the parameters, a , b , c , d , e and f , from a few (at least 6), noisy points.

We can neglect the constant component, by using the techniques suggested in section 6.3.1.1 and 6.3.2.1. So, we only need study the uncertainty model for the first five carriers $[x, y, xy, x^2, y^2]$.

As in (Leedan *et al.* 1999; Leedan *et al.* 2000), we employ the following covariance matrix to characterize the uncertainties in $[x, y, xy, x^2, y^2]$:

$$\sigma^2 \begin{bmatrix} 1 & 0 & y_o & 2x_o & 0 \\ 0 & 1 & x_o & 0 & 2y_o \\ y_o & x_o & x_o^2 + y_o^2 + \sigma^2 & 2x_o y_o & 2x_o y_o \\ 2x_o & 0 & 2x_o y_o & 4x_o^2 + 2\sigma^2 & 0 \\ 0 & 2y_o & 2x_o y_o & 0 & 4y_o^2 + 2\sigma^2 \end{bmatrix} = \sigma^2 \tilde{C}_o \quad (6.35)$$

where the subscript "o" denotes the underlying ground truth of the associated quantity. In practice, the ground truth is unknown and we use the observed quantities. So, in the following, we do not use the symbol "o" in the subscript.

If we drop the σ^2 in $\tilde{\mathbf{C}}$ in (6.35), we have:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & y & 2x & 0 \\ 0 & 1 & x & 0 & 2y \\ y & x & x^2 + y^2 & 2xy & 2xy \\ 2x & 0 & 2xy & 4x^2 & 0 \\ 0 & 2y & 2xy & 0 & 4y^2 \end{bmatrix} = \mathbf{D}\mathbf{D}^T \quad (6.36)$$

where

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ y & x \\ 2x & 0 \\ 0 & 2y \end{bmatrix} \quad (6.37)$$

From (6.36), \mathbf{C} has a rank of 2. Supposing $\mathbf{C} = \mathbf{U}\text{diag}(d_1, d_2, 0, 0, 0)\mathbf{U}^T$, we have

$$\text{span}(\mathbf{u}_1, \mathbf{u}_2) = \text{span}(\mathbf{D}) \quad (6.38)$$

If the x and y coordinates are much larger than the noise in them (this is true in most points), it would hold that

$$\text{span}(\mathbf{u}_1, \mathbf{u}_2) \approx \text{span}(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2) \quad (6.39)$$

where $\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2$ are the singular vectors of $\tilde{\mathbf{C}}$, associated with the two largest singular values. This can be obtained from the matrix perturbation theory, by regarding the terms of σ^2 in $\tilde{\mathbf{C}}$ as some perturbation.

Suppose we observe x, y , with noise ε_x and ε_y in them, respectively. The uncertainties in the carriers $[x, y, xy, x^2, y^2]$, introduced by ε_x and ε_y , are:

$$\varepsilon = \begin{bmatrix} x_o + \varepsilon_x \\ y_o + \varepsilon_y \\ (x_o + \varepsilon_x)(y_o + \varepsilon_y) \\ (x_o + \varepsilon_x)^2 \\ (y_o + \varepsilon_y)^2 \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \\ x_o y_o \\ x_o^2 \\ y_o^2 \end{bmatrix} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ y_o \varepsilon_x + x_o \varepsilon_y + \varepsilon_x \varepsilon_y \\ 2x_o \varepsilon_x + \varepsilon_x^2 \\ 2y_o \varepsilon_y + \varepsilon_y^2 \end{bmatrix} \quad (6.40)$$

And, ε can be expressed as:

$$\varepsilon = \mathbf{D}[\varepsilon_x, \varepsilon_y]^T + [0, 0, \varepsilon_x \varepsilon_y, \varepsilon_x^2, \varepsilon_y^2]^T \quad (6.41)$$

From (6.36), (6.38), (6.39), (6.40) and (6.41), the first order uncertainties are modeled by the covariance matrix \mathbf{C} , and consequently approximately by the first two singular vectors of $\tilde{\mathbf{C}}$, associated with the two largest singular values. This property will be used in the noise level estimation.

6.4.2 Noise level estimation

As can be observed in (6.35), the noise level, σ , in the observed qualities is needed in obtaining the covariance matrix of the carriers. Because the second order terms of uncertainties in the carriers are not Gaussian, we only use their first order uncertainties in estimating the noise level in the observed data. Taking the conic fitting as an example, the first order uncertainties are $\mathbf{D}[\varepsilon_x, \varepsilon_y]^T$.

First, we have the following fact

$$[\varepsilon_x, \varepsilon_y] \mathbf{D}^T (\mathbf{D}\mathbf{D}^T)^{-1} \mathbf{D}[\varepsilon_x, \varepsilon_y]^T = \varepsilon_x^2 + \varepsilon_y^2 \quad (6.42)$$

where $\mathbf{D}^T (\mathbf{D}\mathbf{D}^T)^{-1} \mathbf{D} = \text{diag}(1,1)$.

From (6.36), (6.38), (6.39), (6.40) and (6.41), the rank 2 approximation of the covariance matrix $\tilde{\mathbf{C}}$, is approximately \mathbf{C} if the x and y coordinates are much larger than the noise level: $\tilde{\mathbf{C}}^2 \approx \mathbf{C} = \mathbf{D}\mathbf{D}^T$. Moreover, the uncertainties captured by the 2 largest singular vectors of $\tilde{\mathbf{C}}$, are approximately $\mathbf{D}[\varepsilon_x, \varepsilon_y]^T$. Combining these observations and (6.42), we employ the following estimate for the noise level:

$$\sqrt{\frac{1}{2m} \sum_{i=1}^m \mathbf{e}'_i \tilde{\mathbf{C}}_i^{-2} \mathbf{e}'_i{}^T} \quad (6.43)$$

where $\tilde{\mathbf{C}}_i^{-2}$ is the pseudo inverse of the rank-2 approximation matrix of $\tilde{\mathbf{C}}_i$ in (6.35), and \mathbf{e}'_i is the i^{th} row of the error matrix \mathbf{E} , $\mathbf{E} = \mathbf{W} - \hat{\mathbf{R}}\hat{\mathbf{S}}$, with $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ as the current estimates in the bilinear approach (6.9). Note, in the calculation of the error matrix \mathbf{E} , the constant column in \mathbf{W} is not included.

6.5 Experimental results

In this section, we conduct experiments on the conic fitting, to validate the correctness of our general theory in section 6.3. With this aim, we mainly compare our approach with other competing approaches to this problem: including FNS (Chojnacki *et al.* 2000), HEIV (Leedan *et al.* 1999; Leedan *et al.* 2000), KAN (Kanatani 1994; Kanatani 1996) and the *constrained* TLS method (Fitzgibbon *et al.* 1999). The method in (Fitzgibbon *et al.* 1999) is a specific implementation of the TLS method (Huffel *et al.* 1991), for the conic fitting problem, as pointed out in (Leedan *et al.* 2000), **in particular it enforces that the solution is an ellipse.**

It has been established in (Chojnacki *et al.* 2004), that the HEIV and the FNS are intimately related, with only different numerical solution; and it has also experimentally proved that both of them have almost same performance, where the AML objective function is employed as a criterion. The following experiments suggest that HEIV performs better than FNS in the more challenging problems, for example, where the points distribute in a small portion (e.g., a quarter) of the ellipse; although they have almost same performance in other mildly challenging settings, for example, where the points are from an half ellipse. We do not know the reason for this difference in performance.

In all the experiments, we use the following setting: the true ellipse has a major axis of size 100 and a minor axis of size 50. Two factors have much influence on the estimates of, almost all, the methods mentioned above: the noise level and the span of the points. All the methods produce good estimates, indeed estimates

which are almost same, if the points span the whole ellipse. Because of this, we do not run experiments on the whole ellipse.

6.5.1 Noise level=2 over a half ellipse

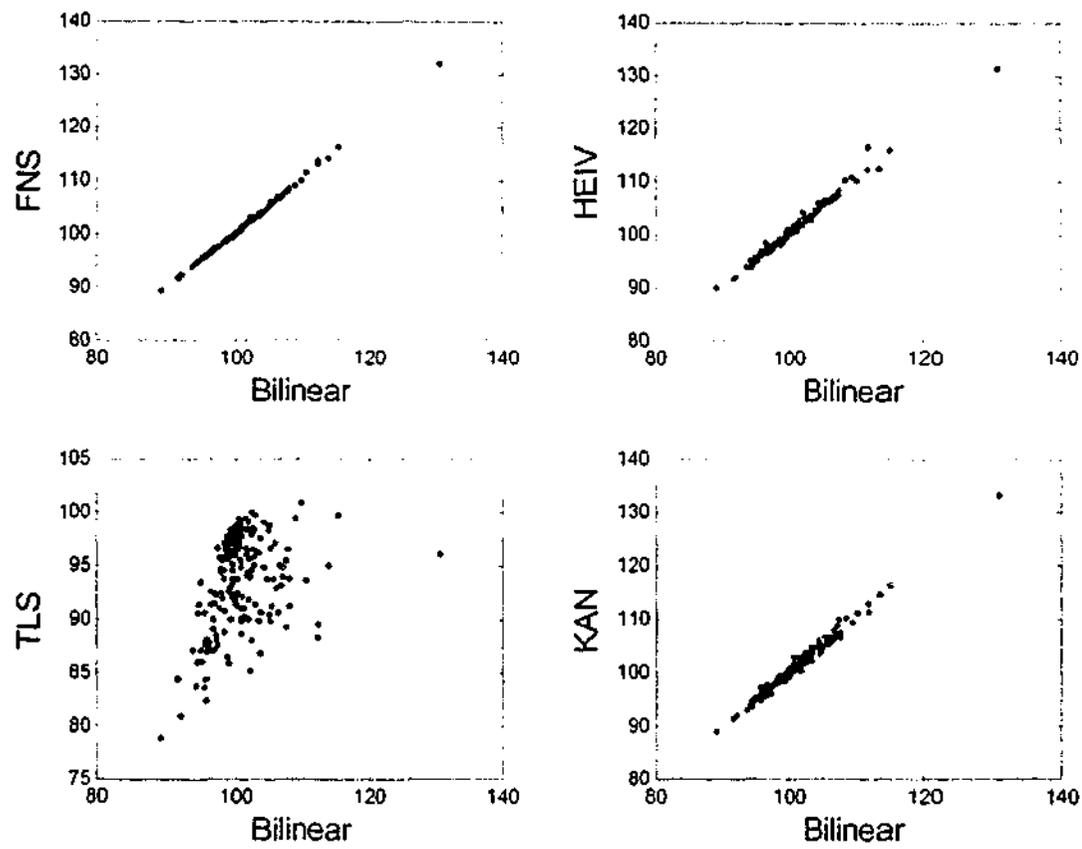
First, we conduct the experiments in the following setting. 100 points are randomly generated in a half ellipse, where the segment of the half ellipse is also randomly selected. Then, i.i.d. Gaussian noise, with noise level of 2, is added to the 100 points. The experiment is repeated 200 times (with different random samples). Surprisingly, our bilinear approach performs almost identically to HEIV, FNS and KAN, all of which perform better than TLS.

Table 6.1: The statistics of the estimated major length, minor length, x and y coordinates of the center, and the angle between the major axis and the horizontal axis. The ground truth is listed in the first row. For every method, its mean, with its standard deviation in the brackets, is listed in each row. *Noise =2 over 1/2 ellipse*

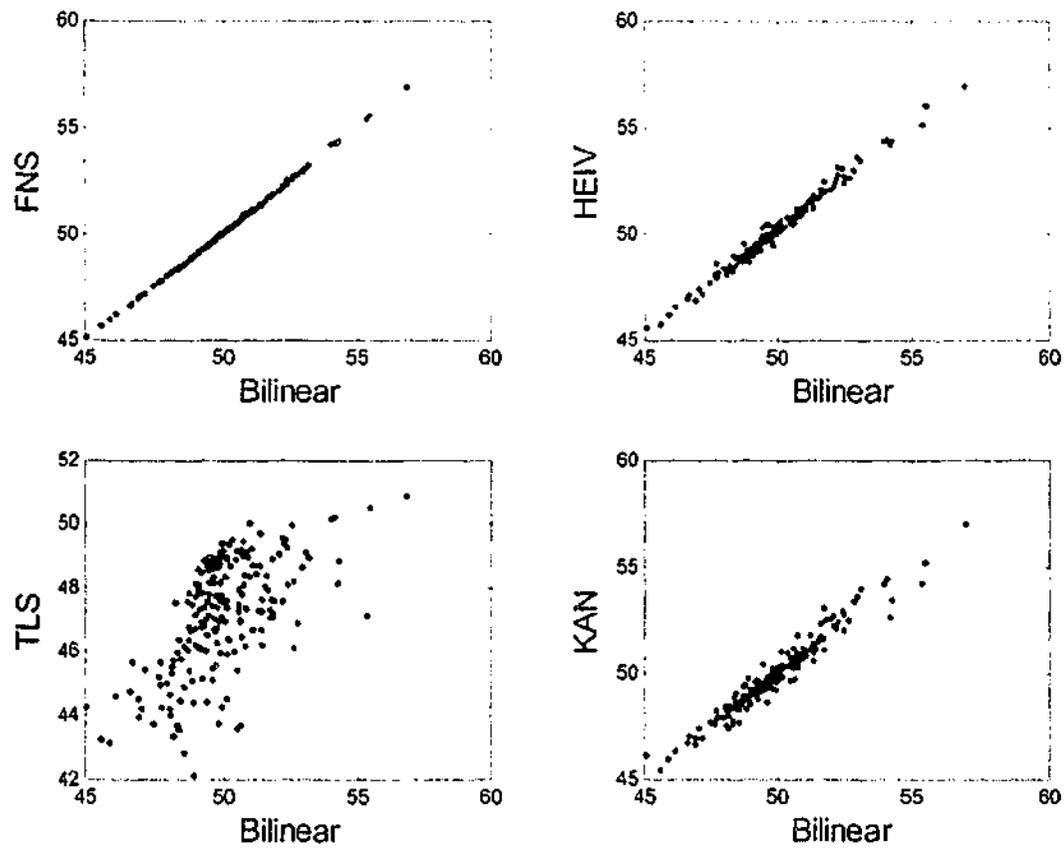
	Major(100)	Minor(50)	Center X(0)	Center Y(0)	Angle(0)
Bilinear	100.6332 (4.3768)	49.9533 (1.6987)	-0.1382 (4.3744)	0.1164 (1.7272)	0.0218 (1.2630)
FNS	100.7461 (4.5155)	49.9794 (1.7131)	-0.1540 (4.5255)	0.1194 (1.7441)	0.0176 (1.2746)
HEIV	100.8303 (4.4745)	50.1564 (1.6956)	-0.1783 (4.4852)	0.1182 (1.7454)	0.0235 (1.2878)
KAN	100.6214 (4.5938)	49.9326 (1.7042)	-0.1670 (4.5990)	0.0903 (1.7457)	0.0328 (1.2966)
TLS	93.6993 (4.3251)	47.1748 (1.7675)	0.2508 (7.5858)	-0.3264 (3.3746)	0.4267 (2.6456)

We experimentally find that the error in the five parameters above is not independent. The error in the coordinates of the center and the orientation angle are strongly dependent on the estimates of the major length and the minor length. If both the major length and the minor length are correctly estimated, the other three parameters are probably close to the ground truth. Because of this observation, we mainly resort to the major length and the minor length in the evaluation of the methods, in the following. In Figure 6.1, we show the performance of the methods, in contrast with that of the Bilinear. We can observe a strong linear correlation between the four approaches: HEIV, FNS, KAN and Bilinear.

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting



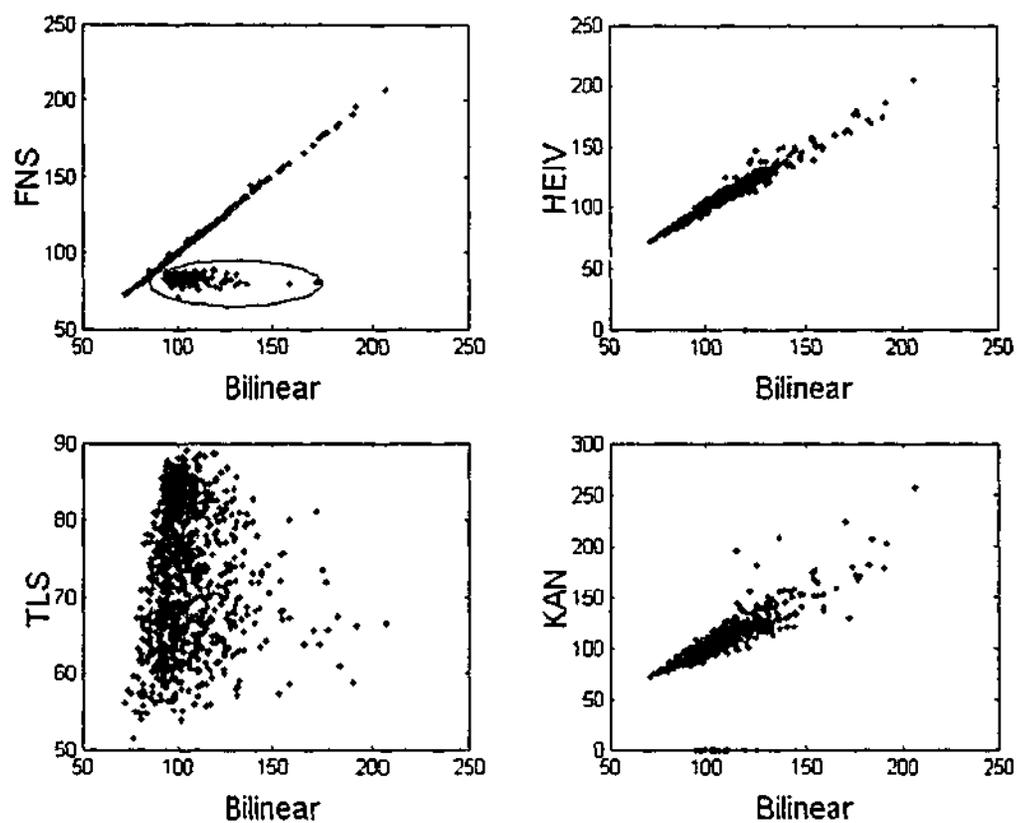
(a)



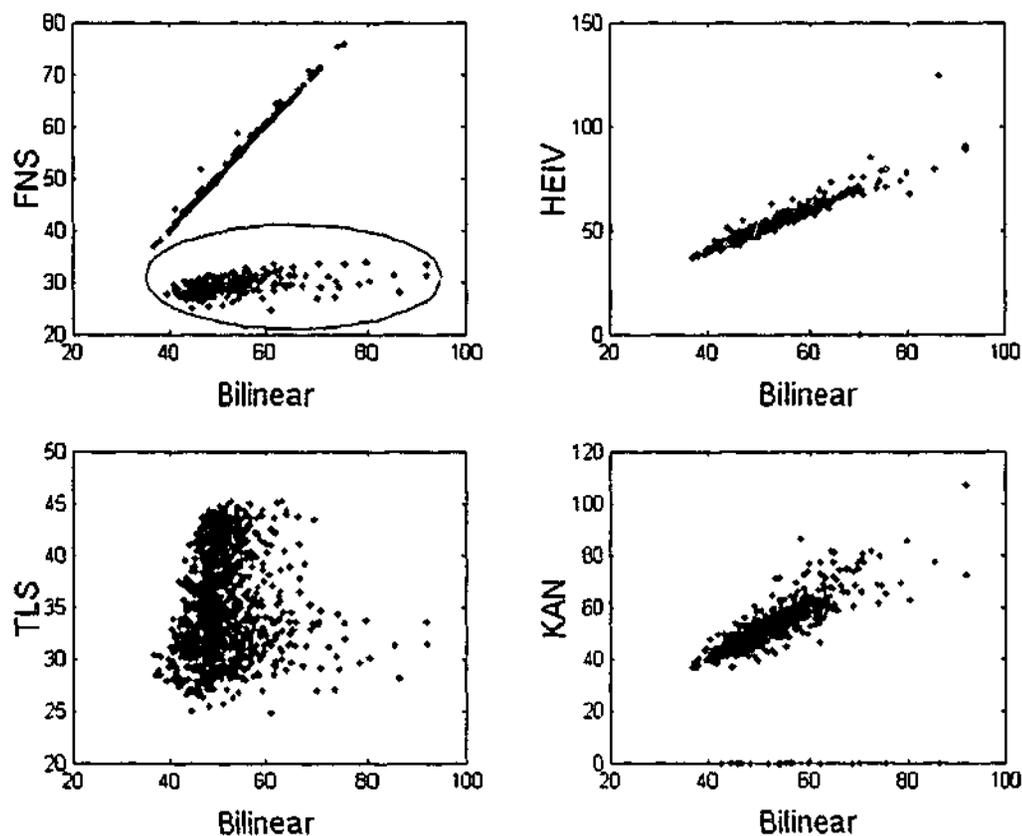
(b)

Figure 6.1: The comparison of the bilinear approach, with the FNS, HEIV and KAN, TLS. (a) is the estimated major length and (b) is the estimated minor length.

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting



(a)



(b)

Figure 6.2: See the caption of Figure 6.1. $Noise = 2$ over $3/8$ ellipse. The only other difference is that the TLS method is not included. In two of the graphs, there are a significant number of "outlier" results that we have highlighted by drawing an enclosing boundary around them.

6.5.2 Noise level=2 over 3/8 ellipse

The next experiment differs from the above experiment in that now the points are randomly generated from a random 3/8 ellipse. The noise level is still 2. In order to present a better comparison, we repeated the experiment 1000 times to obtain the statistics (listed in Table 6.2).

Table 6.2: See the caption of Table 6.1. *noise=2 over 3/8 ellipse*

	Major(100)	Minor(50)	Center_X(0)	Center_Y(0)	Angle(0)
Bilinear	103.6820 (16.0952)	51.0717 (6.6344)	0.4578 (16.0868)	0.2702 (6.9479)	-0.0461 (3.2988)
FNS	100.3527 (17.4383)	46.9251 (9.4859)	0.4511 (16.4996)	0.2104 (9.7672)	0.0434 (3.6045)
HEIV	103.7216 (15.5261)	51.4157 (6.8837)	0.3804 (15.4628)	0.3439 (7.2849)	-0.0313 (3.3207)
KAN	104.4603 (18.5451)	51.2461 (7.3436)	0.3138 (18.5808)	0.0375 (7.8171)	-0.0042 (3.5893)
TLS	73.8776 (9.0869)	35.7235 (5.0653)	0.1016 (28.0569)	-0.2730 (13.6420)	-0.2007 (10.5892)

However, taken alone, the statistics in Table 6.2 don't adequately reflect the performance of the methods. Consider also Figure 6.2 and Table 6.3. We find that the FNS method performs much worse than the HEIV, KAN and Bilinear approaches in some cases, as can be observed in Figure 6.2. (Note, although there are a few cases in the circles in Figure 6.2, where the Bilinear, HEIV and KAN also produce "bad" estimates; in many cases, the Bilinear, HEIV and KAN produce "good" estimates, as can be observed in Table 6.3). The problem with the FNS method is that there is no guarantee of convergence. Because of the lack of convergence, in some cases, the FNS stops in the first iteration step, and consequently, its estimate is the same as the initial estimate, which we chose as the TLS estimate for initializing FNS. (This also accounts for the fact that the FNS method produces almost 100% ellipses in the following experiments, which are even more challenging. Note, the TLS always produces an ellipse because the constraint $4de - c^2 = 1$ is enforced in the TLS method.)

To summarize: it is difficult to evaluate the approaches in this setting. This is because one approach scores better in a few cases, while another approach scores

better in other different cases. Moreover, as we will see in the more challenging experiments below, some estimates are wildly wrong, (they may not even be ellipses - except when the special TLS method is employed, enforcing the elliptic constraint). For these reasons, the statistics above, by themselves, can not reliably reflect the performance of the approaches. Worse, the wildly wrong estimates make the statistics, like mean, misleading in assessing the performance of the methods. For example, from the mean of the major length, the FNS is the best method. However, if we examine the figures in Figure 6.2 in detail, we find that FNS actually is worse than the HEIV, KAN and Bilinear methods.

In order to present a meaningful comparison, we mainly resort to the following statistics: for a method, how often does it produce good estimate? As in Figure 6.2, we only use the estimated major length and the estimated minor length in evaluating the performance, because the accuracy of other parameters is strongly dependent on the accuracy of the lengths. More precisely, we regard an estimate as "good" if the error of the estimated major, and minor, lengths fall short of 10% or 20% of the true lengths. In this example, we regard the estimate good if its major length lies in [90,110] or [80,120] and if its minor length lies in [45,55] or [40,60]. The KAN and FNS methods perform a little worse than HEIV and the proposed Bilinear method.

To provide an indication of our measure, two examples of "good" estimates are shown in Figure 6.3. The good estimates by these four methods are shown in Figure 6.4.

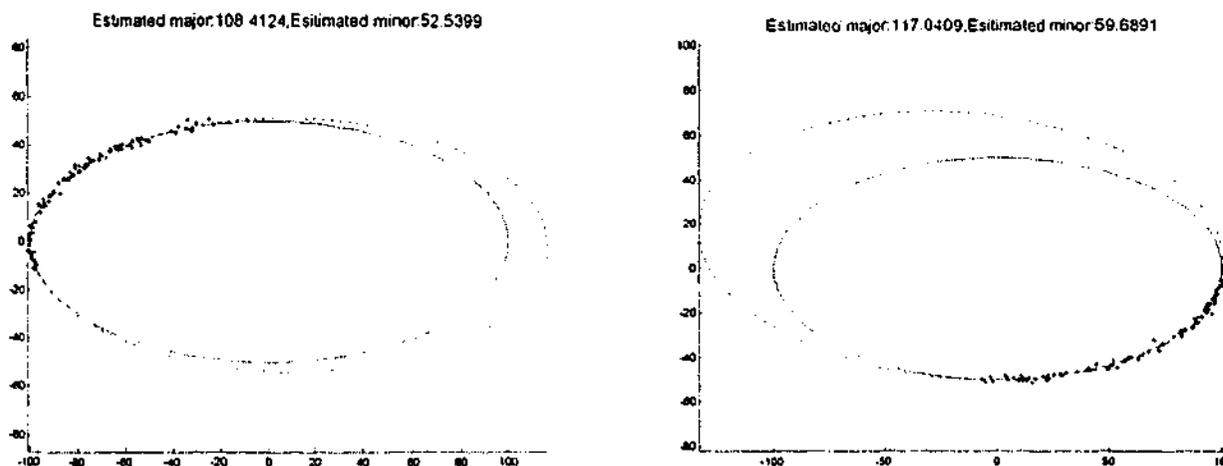


Figure 6.3: Two example of "good" estimates, falling in the 10% and 20% range.

Table 6.3: The "good" estimates for $noise=2$ over $3/8$ ellipse. See the definition of "good" estimate in the text.

	10%	20%	Other ellipses	Non-ellipse
Bilinear	532	825	175	0
HEIV	526	820	179	1
KAN	470	773	205	22
FNS	444	686	314	0
LS	0	1	999	0

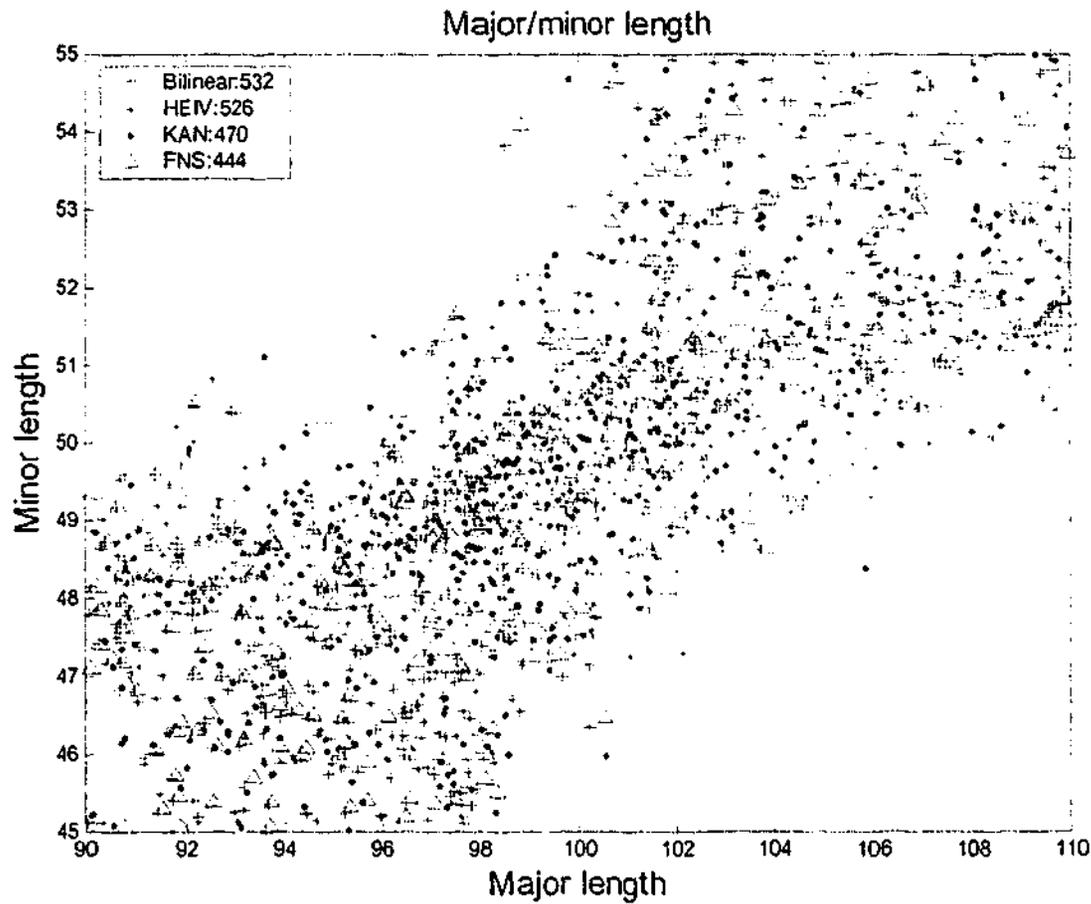


Figure 6.4: The "good" estimates in 1000 trials of the Bilinear, FNS, HEIV and KAN approaches for $noise=2$ over $3/8$ ellipse. The number after the approaches in the legend is how often the associated approach produces "good" estimates in 1000 trials.

6.5.3 Noise level=1 over a quarter ellipse

In this experiment the noise level is 1 and the points are from a quarter of the ellipse. We also run 1000 trials for this setting. As we discussed above, the statistic of mean and standard deviation are not good indexes for comparing. We only list how often the approaches succeed in producing "good" estimates in 1000 trials.

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting

Table 6.4: The “good” estimates for $noise=1$ over $1/4$ ellipse. See the definition of “good” estimate in the text.

	10%	20%	Other ellipses	Non-ellipse
Bilinear	229	421	569	10
HEIV	222	425	553	22
KAN	188	383	481	136
FNS	125	246	752	2
TLS	0	0	1000	0

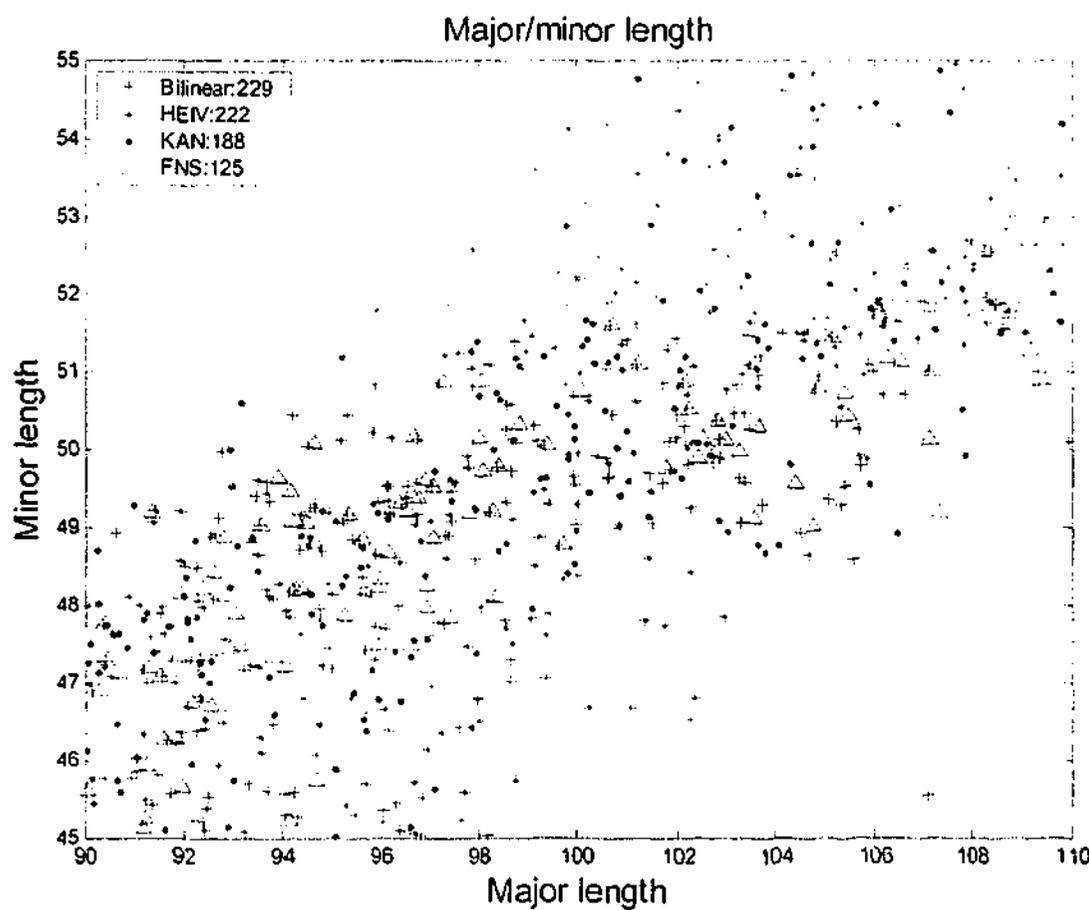


Figure 6.5: The “good” estimates in 1000 trials of the Bilinear, FNS, HEIV and KAN approaches for $noise=1$ over $1/4$ ellipse. The number after the approaches in the legend is how often the associated approach produces “good” estimates in 1000 trials. The “good” estimates are defined by the 10% range.

Although there is a strong linear correlation between the results produced by the HEIV, FNS, KAN and Bilinear methods in some settings, as can be observed in Figure 6.1 and Figure 6.2; they actually have quite different performance in this more challenging environment.

Note: even though the HEIV and the bilinear methods seem to have a similar performance, in terms of the statistics in Table 6.3 and Table 6.4, they actually have different outputs in many cases. For example, although the Bilinear approach and the HEIV approach produce a similar result, in terms of how often they produce "good" estimates; there are only 142 cases, where both approaches simultaneously produce "good" estimates, falling in the 10% range. This means that, in 80 cases, while the HEIV result falls in the range of 10%, the Bilinear does not. On the other hand, the Bilinear approach produces good estimates in 87 cases, where the HEIV approach does not.

We also comment that, due to the moderately high failure rate, none of these approaches can't be regarded as a solution to the conic fitting problem in the most challenging forms (data over a small arc of the ellipse only).

6.5.4 Noise level=2 over a quarter ellipse

In this last experiment the noise level is 2 and the points are from a quarter of the ellipse. As in section 6.5.3, we only list how often the approaches succeed in producing "good" estimates in 1000 trials.

Table 6.5: The "good" estimates for *noise=2 over 1/4 ellipse*. See the definition of "good" estimate in the text.

	10%	20%	Other ellipses	Non-ellipse
Bilinear	88	211	663	126
HEIV	75	179	577	244
KAN	29	75	313	612
FNS	10	26	965	9
TLS	0	0	1000	0

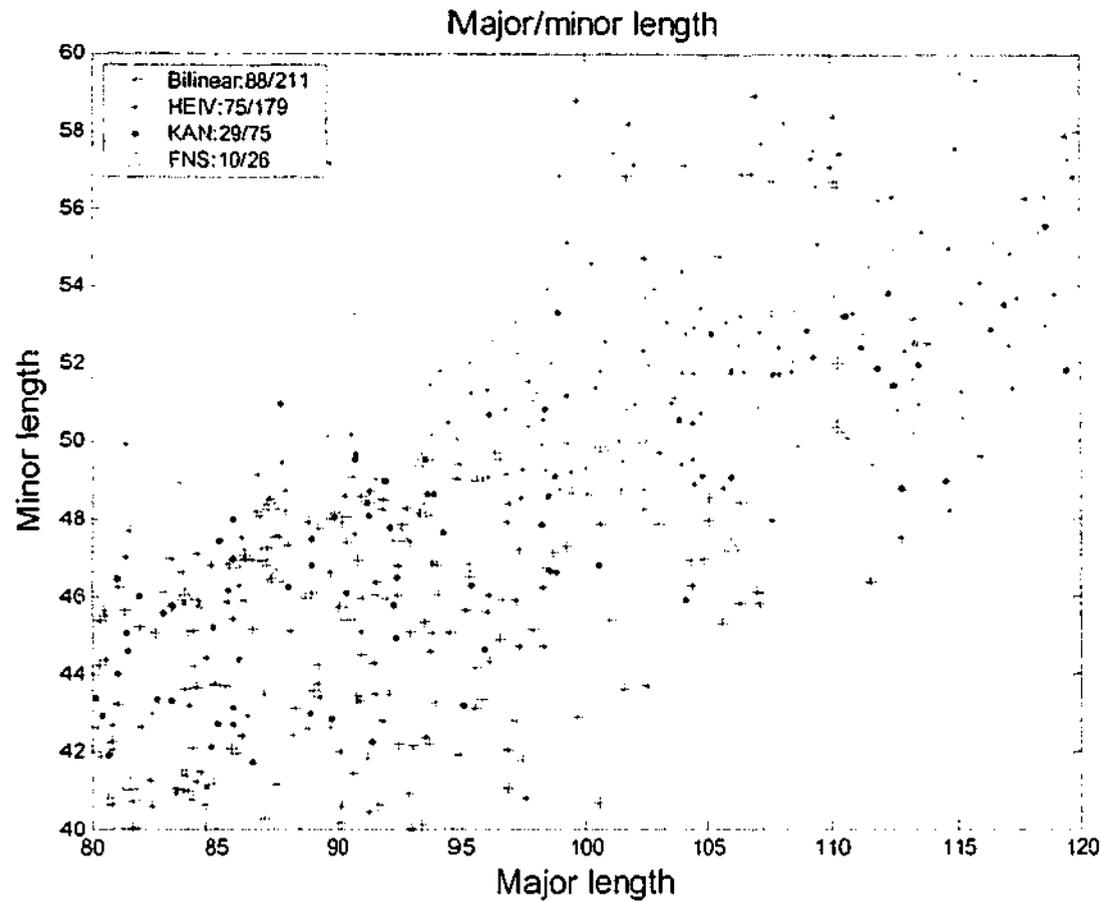


Figure 6.6: The “good” estimates in 1000 trials of the Bilinear, FNS, HEIV and KAN approaches for *noise=2 over 1/4 ellipse*. The numbers after the approaches in the legend are how often the associated approach produces “good” estimates in 1000 trials. The “good” estimates are defined by the 10% and 20% range, respectively.

We remark that, only on 19 or 92 cases, out of the 1000 trials, both HEIV and Bilinear produce “good” estimates, in terms of the 10% or 20% ranges, respectively.

6.5.5 Comments on the experimental results

Although convergence to the ML estimate, at least a local optimal estimate, can be ensured in the proposed bilinear approach, as discussed in section 6.3.4, the results are not so good as expected. From the Table 6.4 and Table 6.5, the bilinear approach can't be regarded as a good solution to the problem, where the points only span a quarter of the ellipse; because it has only about 10-20% success rate of “good” estimates.

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting

There are two possible reasons for this. First, as suggested in section 6.3.4, only a local optimal solution can be ensured in the iteration process. If the initial estimate deviates far from the global optimal estimate, the iteration is possibly trapped in other local minimal estimates. In our experiments, we took the TLS result (Fitzgibbon *et al.* 1999) as the initial estimate for the bilinear approach. However, it has been suggested in (Leedan *et al.* 2000) that the TLS result is not “adequate to be used as an initial solution.” Also note that we take the TLS result as the initial estimate for FNS approach. This possibly accounts for the fact that FNS performs much worse than HEIV in the challenging settings, as shown in Table 6.4 and Table 6.5.

The second reason, possibly, is due to the specific nature of the conic fitting problem. Also as discussed in section 6.3.4, the optimal solution, measured by (6.19) or (6.20), does not imply the ML estimate, because the ML optimality applies only when the uncertainties in the general measurement matrix \mathbf{W} are Gaussian. As we analysed in section 6.4.1, the second order uncertainties are not Gaussian. Strictly, even if we obtain the optimal solution, measured by (6.19) or (6.20), it is not the ML estimate.

We also have to remark that, although that the bilinear approach outperforms other competing approaches, as can be observed in experiments above; we do not claim that the proposed bilinear approach can replace the other approaches because of two facts. First, as pointed out in the experiments, although the results, by FNS, HEIV, and KAN show a strong correlation with the result by the proposed bilinear approach, there are many situations where one method succeeds and other methods fail. There is no clear “safe-bet” in this regards.

Second, the computation complexity of the proposed bilinear approach impedes its practical applications especially when the points only span a quarter ellipse. In the challenging settings, as in sections 6.5.3, 6.5.4, the bilinear approach suffers from its low convergence rate. The bilinear approach in (Shum *et al.* 1995; Gear 1998; Vidal *et al.* 2004) is a variant of the orthogonal iteration method in (Golub *et al.* 1996), whose convergence rate is the ratio between the $(r+1)^{th}$ singular value and

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting

the $(r+1)^{th}$ singular value. It is not clear what governs the convergence rate in the heteroscedastic setting.

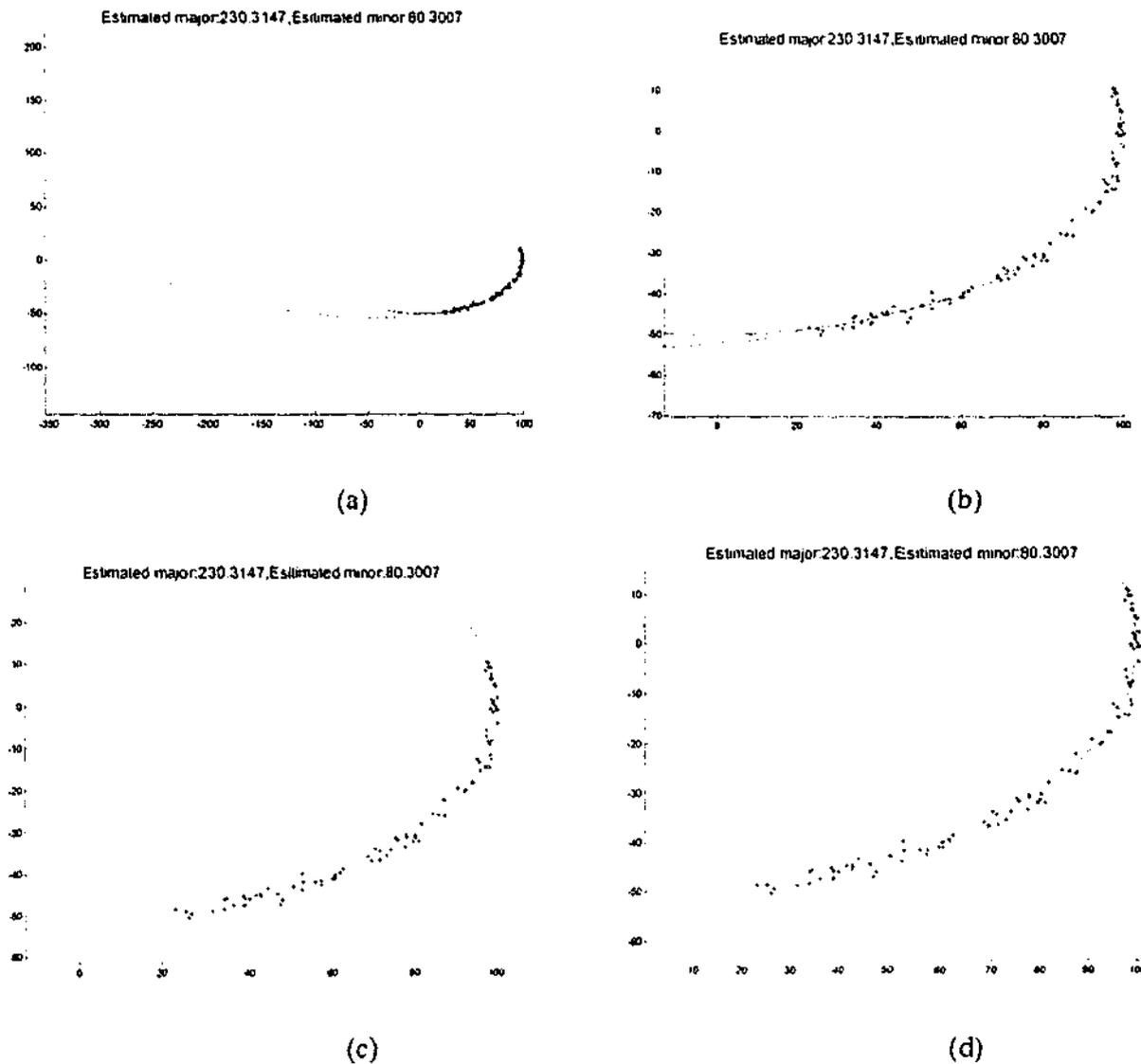


Figure 6.7: A bad estimate, with details. The solid ellipse is the ground truth, and the dotted ellipse is the estimated ellipse. The dots, '.', are the noisy feature points.

6.5.6 Question raised

We have stated that the approaches, including FNS, HEIV, KAN and the proposed bilinear approach, can't be regarded to a good solution to the conic fitting problem, when the points only span a small arc of the ellipse. Here, we highlight aspects of the problem from another point of view. Figure 6.7.a shows a "bad" estimate, whose estimated major and minor lengths are 230 and 80, respectively. In terms of the estimated parameters, this estimated ellipse is wildly wrong, because it is far from the truth. However, from Figure 6.7.b, we can find that the estimated ellipse

fits the points very well. If we separate the fitted ellipse and the underlying ground truth, it is very difficult to decide which one fits the noisy points better, as shown in Figure 6.7.c and Figure 6.7.d. This suggests that in many cases it may actually be unreasonable to expect the true ellipse to be recovered in such extreme cases of only a small fraction of the ellipse containing data.

6.6 Conclusion

In this chapter, we present a general theory of the parameter estimation problem in a heteroscedastic linear system. This theory suggests a bilinear solution method which we implemented and tested. The method was shown to perform relatively well, and, for ellipse fitting where the data covers a large fraction of the ellipse, the results are good. However, *none* of the methods investigated, including ours, can be considered adequate for fitting data from a small arc of the ellipse. As we illustrated in our concluding section, it is perhaps true that in at least some of the cases where the methods fail, it is unreasonable to expect any method to produce the “true” solution. However, we have no way of making such a notion precise and for testing the “reasonableness” of the task.

In the latter stage of this work, the author became aware of work that tries to project onto a low rank subspace by optimization on the Grassman manifold. See (Lu et al. 1997; Manton et al. 2003). It would be interesting to use such methods on this problem.

Appendix: Equality Constrained Least Squares

The equality constrained least squares problem is as:

$$\min_{\mathbf{A}_2 \mathbf{x} = \mathbf{b}_2} \|\mathbf{A}_1 \mathbf{x} - \mathbf{b}_1\| \quad (\text{A.6.1})$$

where $\mathbf{A}_1 \in R^{m,n}$, $\mathbf{A}_2 \in R^{p,n}$, $\mathbf{b}_1 \in R^{m,1}$, and $\mathbf{b}_2 \in R^{p,1}$.

Without loss of generality, assume $\text{rank}(\mathbf{A}_2) = p$ and $p < n$. Let $\mathbf{Q}^T \mathbf{A}_2^T = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ be the QR factorization of \mathbf{A}_2^T , where \mathbf{R} is a $p \times p$ upper triangle matrix. Set

Chapter 6: Bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting

$A_1 Q = [P_1, P_2]$ and $Q^T x = \begin{bmatrix} y \\ z \end{bmatrix}$, where $y \in R^{p_1}$, $z \in R^{n-p_1}$. With these

transformations, (A.6.1) becomes

$$\min_{R^T y = b_2} \| P_1 y + P_2 z - b_1 \| \quad (\text{A.6.2})$$

where the vector \hat{y} can be determined from the constraint $R^T y = b_2$. (A.6.2)

becomes

$$\min_z \| P_2 z - (b_1 - P_1 \hat{y}) \| \quad (\text{A.6.3})$$

which is an unconstrained LS problem. The solution to the equality constrained LS problem (A.6.1) is:

$$\hat{x} = Q \begin{bmatrix} \hat{y} \\ \hat{z} \end{bmatrix} \quad (\text{A.6.4})$$

Chapter 7

Orthographic projection of distances: a low-dimensional approximation

7.1 Introduction

The distance between two feature points in an image is a coordinate-free quality; however, it has seldom been used in computer vision or pattern recognition tasks because it rapidly changes as the view varies. In this chapter, we investigate one property of a quantity we call the “*distance vector*”, which consists of the distance among the pairs of feature points under orthographic projection. We prove that the distance vectors under different views approximately lie in a linear subspace with a dimension of 6.

7.2 Distance vector, spherical harmonics, and linear subspace

It has been proved, by exploiting spherical harmonic expansions, that the gray scale images of a Lambertian object approximately lie in a 9D linear subspace (Basri *et al.* 1999; Ramamoorthi *et al.* 2001; Ramamoorthi 2002; Basri *et al.* 2003). Similarly, we will prove here that the “distance vectors” of a rigid object approximately lie in a 6D linear subspace, and the 6D subspace accounts for at least 99.76% of the variability of the distance vector.

7.2.1 Definition of distance vector (distance matrix)

For m feature points in an image, there are $C_m^2 = m(m-1)/2$ line segments joining all pairs of such points. We use a symmetric $m \times m$ matrix, \mathbf{D} , with all zeros in its diagonal entries, to represent all the segments between features. $D_{i,j}$ is the length of the line segment between the i^{th} feature point and the j^{th} feature point. We arrange the $m(m-1)/2$ entries, $D_{i,j}$ with $i > j$, of the upper-half matrix as a vector,

calling it the *distance vector*. The order of the entries of the distance vector does not make difference in the subsequent sections. What counts is that we arrange all the distance vectors in different views in the same order.

7.2.2 Spherical harmonics

Spherical harmonics for the functions on the surface of the unit sphere have the same role as the Fourier basis for functions in the plane. Spherical harmonics are defined as:

$$Y_{n,m}(\theta, \phi) = \sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}} P_{n,|m|}(\cos\theta) e^{im\phi} \quad (7.1)$$

where $P_{n,m}$ is the associated Legendre functions, with

$P_{n,m}(z) = \frac{(1-z^2)^{m/2}}{2^n n!} \frac{d^{n+m}}{dz^{n+m}}(z^2-1)$ for $n = 0, 1, 2, \dots$ and $-n \leq m \leq n$. $Y_{n,m}$ for $-n \leq m \leq n$ is the n^{th} order spherical harmonics, and there are $2n+1$ spherical harmonics in the n^{th} order.

Spherical harmonics form an orthogonal basis for functions on the surface of the unit sphere. A function on the unit sphere can be decomposed into a series of the spherical harmonics:

$$f(\theta, \phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^n f_{n,m} Y_{n,m}(\theta, \phi) \quad (7.2)$$

where $f_{n,m}$ is the strength of the harmonics of Y_{nm} :

$$f_{n,m} = \iint f(\theta, \phi) Y_{n,m}(\theta, \phi) \sin(\theta) d\theta d\phi \quad (7.3)$$

7.2.3 Spherical harmonics for the *sine* function

As was used in (Basri *et al.* 1999; Ramamoorthi *et al.* 2001; Ramamoorthi 2002; Basri *et al.* 2003), the *sine* function can be approximately decomposed into the a few spherical harmonics. Particularly, the harmonics expansion of *sine* function can be decomposed into:

$$\sin(\theta) = \sum_{n=0}^{\infty} k_{2n} Y_{2n,0} \quad (7.4)$$

where $k_{2n} = \frac{\pi}{2^n n!} \sum_{i=n}^{2n} \{(-1)^i C_{2n}^i P_{2i}^{2n} \sum_{j=0}^{i-n} \{(-1)^j C_{i-n}^j \prod_{k=1}^{j+1} \frac{2k-1}{2k}\}\}$. The calculation of k_{2n} is arranged in the appendix.

Please note that all the odd order harmonics $Y_{2n+1,m}$ are eliminated. The reason is that the *sine* function is symmetric with $\pi/2$ and all odd order terms are anti-symmetric with $\pi/2$. Another fact is that, because the *sine* function is circular symmetric, only the zonal harmonics (i.e. $m=0$ in equation (7.1)) exist in the expansion of (7.4), i.e., $f_{n,m} = 0$ in (7.3) for $m \neq 0$.

The amplitudes up to the 20^{th} order spherical harmonic are plotted in Figure 7.1. Specifically, $k_0 = \pi\sqrt{\pi}/2$, $k_2 = -\pi\sqrt{5\pi}/16$ and $k_4 = -3\pi\sqrt{\pi}/128$. (Note $k_{2n+1} = 0$.) The approximation up to the second/fourth order accounts for 99.76/99.96% of the energy of the *sine* function --- $\frac{8}{3}\pi$:

$$\int_0^{2\pi} \int_0^\pi \sin^2 \theta \sin \theta d\theta d\phi = \frac{8}{3}\pi \quad (7.5)$$

Figure 7.2 shows the close agreement between the *sine* function and the spherical harmonic approximations up to the second and the fourth order.

7.2.4 Low dimensionality of the distance vectors

Under the orthographic camera, the length between two points in a view is the product of its length in the 3D world and $\sin(\theta)$, where θ is the angle between the optical axis of the camera and the 3D line. Suppose each point on the unit sphere surface represents the direction of one unit-length segment in 3D space. Consequently, its distance vector is the function $\sin(\theta)$ when the optical axis of the orthographic camera coincides with the z direction. From (7.4), this distance vector can be approximately spanned up to the second or the fourth order, with 2 or 3, respectively, orthogonal zonal harmonics.

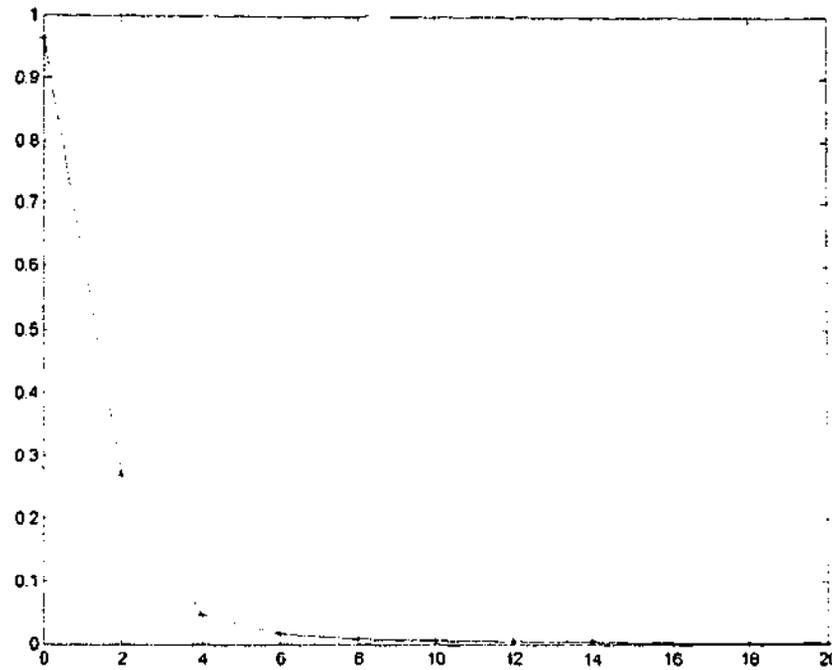


Figure 7.1: The amplitudes of the spherical harmonics up to the tenth order.

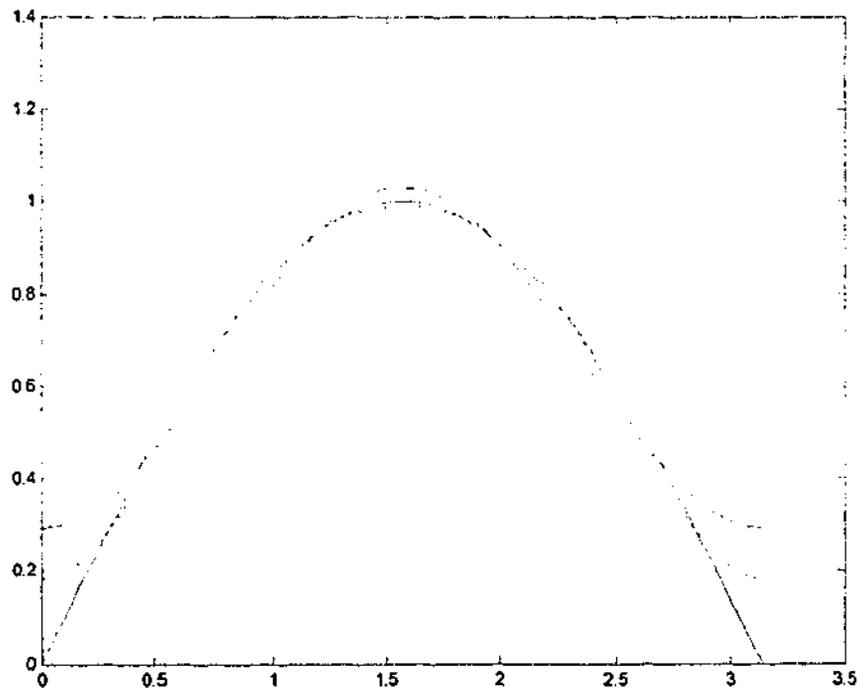


Figure 7.2: The approximations of the *sine* function. The solid curve is the *sine* function, the dashed/dotted curves for the second/fourth order approximation.

When the view is taken from other direction, other non-zonal harmonics are not zeroes any more. Viewing from other direction has the same effect on the spherical harmonics as a rotation. In (Ramamoorthi *et al.* 2001), it has been shown that the rotation of the harmonics only *mix* the energy between the same order harmonics

and that the total energy in one order does not change. *Precisely, a rotation of a harmonics is a linear combination of all spherical harmonics within the same order.* This fact has also been pointed out in (Basri *et al.* 1999; Basri *et al.* 2003). Thus, when viewed from other directions from the z direction, the *sine* function on the surface of the unit sphere can be approximated by up to the $2n^{\text{th}}$ order with $\sum_{i=0}^n (1+4i) = (2n^2 + 3n + 1)^{\text{th}}$ harmonics; particularly for the second or fourth order, respectively with 6 or 15 harmonics. Because the rotation only mixes the energy of the harmonics with the same order, these 6 or 15 harmonics still accounts for 99.76/99.96% of the energy of the rotated *sine* function.

7.2.5 *Sine* function on a sparse set of points

Of course, only some sparse directions are needed for an object with some feature points, instead of the whole sphere. Moreover, line segments in 3D world have different lengths. However, for a sparse set of directions, the property of the low-dimensionality still applies. Indeed, a slightly better approximation can be expected for the sparse case, as is validated by the experiments.

We arrange the distance matrix this way: the distance vector, with the same order, for each view takes a column. Two cases are considered as examples: 100 or 20 points are randomly generated and then are randomly orthographically projected upon 1000 views. Next, we apply SVD (Golub *et al.* 1989) on the distance matrix. The energy distribution for 100 points and 20 points is listed in Table 7.1, comparing with the theoretical values for the sphere.

The data in Table 7.1 confirms our claim that the low-dimensionality of the distance vectors still holds for a sparse set of directions. The first 6 singular vectors of the distance matrix account for 99.79% or 99.80% of the energy, compared with 99.76% for the *sine* function on the surface of the unit sphere.

Another property can be observed, from this example, that the difference between the second and the sixth singular values decreases as the feature points increases, as shown in Figure 7.3. It can be expected that the singular values in the same

order are approximately same when enough feature points averagely distribute in 3D space.

Table 7.1: Energy distribution for the sphere and a sparse set of direction.

Simulation			Sphere	
	100 points	20 points		
1 st	93.47%	93.24%	0 th	92.53%
2 nd ---6 th	6.33%	6.55%	2 nd	7.23%
7 th ---15 th	0.17%	0.18%	4 th	0.20%

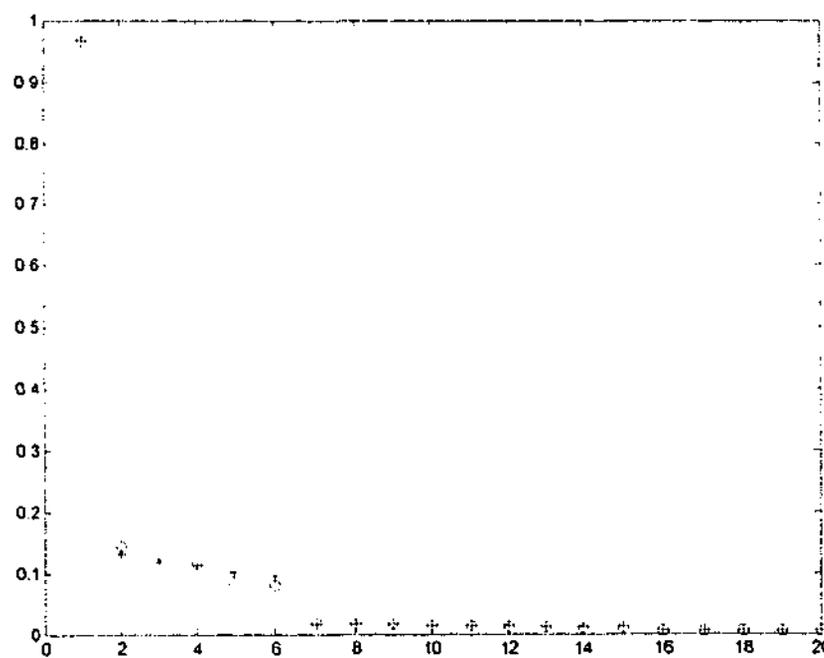


Figure 7.3: The singular values of the distance matrices with 20 (denoted by circles "o") or 100 (denoted by crosses "+") feature points

7.3 Discussion and conclusion

As discussed in previous chapters of this thesis, the low-dimensional subspace analysis has been widely employed in computer vision tasks, like the factorization method for structure from motion and the PCA (or the linear subspace approach) to the face recognition problem. In this chapter, we prove a new low-dimensional property that a 6D subspace accounts for at least 99.76% of the variability of the distance vector.

Closely related to this approximately 6D property of the distance vector is the fact that the measurement matrix lies in a 4D subspace, as discussed in chapter 3. At this stage, we cannot demonstrate that the new property established here is directly useful for computer vision tasks. However, it does seem likely that such a remarkable feature can be exploited.

Appendix

Here, we derive the analytical formula for the decomposition of the *sine* function. According to (7.3), the coefficient of the $2n^{\text{th}}$ order harmonics, k_{2n} , is:

$$k_{2n} = 2\pi \int_0^\pi \sin^2 \theta Y_{2n,0}(\theta) d\theta \quad (7.6)$$

where $Y_{2n,0}(\theta) = \sqrt{\frac{(2n+1)}{4\pi}} P_{2n}(\cos \theta)$. We can express $P_k(t)$ as:

$$P_n(t) = \frac{1}{2^n n!} \sum_{i=n}^{2n} (-1)^i C_i^{2n} P_{2i}^{2i-2n} t^{2i-2n} \quad (7.7)$$

The following formula can be easily obtained:

$$\int_0^\pi \cos^{2k} \theta \sin^2 \theta d\theta = \sum_{j=0}^k (-1)^j C_j^{2k} \int_0^\pi \sin^{2(j+1)} \theta d\theta \quad (7.8)$$

$$\int_0^\pi \sin^2 \theta d\theta = \pi / 2 \quad (7.9)$$

$$\begin{aligned} \int_0^\pi \sin^{2(j+1)} \theta d\theta &= \int_0^\pi (2j+1) \sin^{2j} \cos^2 \theta d\theta \\ &= (2j+1) \int_0^\pi \sin^{2j} \theta d\theta - (2j+1) \int_0^\pi \sin^{2(j+1)} \theta d\theta \\ &= \frac{2j+1}{2j+2} \int_0^\pi \sin^{2j} \theta d\theta \\ &= \pi \prod_{k=1}^{j+1} \frac{2k-1}{2k} \end{aligned} \quad (7.10)$$

Combining (7.6), (7.9) and (7.10), we have obtained

$$k_{2n} = \frac{\pi}{2^n n!} \sum_{i=n}^{2n} \{(-1)^i C_i^{2n} P_{2i}^{2i-2n} \sum_{j=0}^{i-n} \{(-1)^j C_j^{i-n} \prod_{k=1}^{j+1} \frac{2k-1}{2k}\}\} \quad (7.11)$$

Chapter 8

Subspace-based face recognition: outlier detection and a new distance criterion

8.1 Introduction

Illumination effects, including shadows and varying lighting, make the problem of face recognition challenging. Experimental and theoretical results show that the face images under different illumination conditions approximately lie in a low-dimensional subspace, hence principal component analysis (PCA) or low-dimensional subspace techniques have been used, as overviewed in Chapter 3.

The contributions of this chapter are: (a) In section 8.3, we propose a new error distance for the subspace-based recognition problem. This is based on the new theory about the *learning capacity* in low-rank subspace approaches, presented in chapter 4. (b) In order to remove points not following the reflectance model, we employ the iterative reweighted least square (IRLS) technique (section 8.4) to detect the pixels that do not obey the dimension-3 subspace constraint, such as eyeballs. The experiments on the Yale-B face database show the effectiveness of the new techniques.

The work presented in this chapter has been published in (Chen et al. 2004).

8.2 Lambertian reflection and low dimension subspace

It has been experimentally observed that the set of images, observed under different lighting conditions, can be considered as lying in a low dimension

subspace (of the set of all possible images of a certain person). This fact can be theoretically justified so that, depending upon the complexity of the image model, we can demonstrate that the dimension of the subspace is as low as 3-9 (in this chapter, we use 7 dimension subspace to model the attached shadow.) This section overviews the background in the context of face recognition.

8.2.1 Lambertian reflectance and 3-dimensional subspace

The images of a Lambertian object can be approximately modeled by a 3-dimensional subspace if the light source lies at infinity and there is neither attached shadow nor cast shadow (Shashua 1997; Georghiades *et al.* 2001). Following (Bellhumeur *et al.* 1998), for any point p on a Lambertian surface, illuminated by an infinite light source, its intensity can be described by

$$I(p) = a(p)\mathbf{n}(p)^T \mathbf{s} = \mathbf{b}(p)^T \mathbf{s} \quad (8.1)$$

where $a(p)$ (a scalar) is the albedo* at position p , $\mathbf{n}(p)$ (a 3-vector) is the inward normal of the surface at position p , and \mathbf{s} (a 3-vector) is the direction of the light. Let $\mathbf{B} \in R^{n \times 3}$ be a matrix where each row is $\mathbf{b}(p)^T$. The illumination subspace can be generated by:

$$L = \{x \mid x = \mathbf{B}\mathbf{s}, \forall \mathbf{s} \in R^3\} \quad (8.2)$$

The images without shadows are a subset of L . The set of all images, the non-negative orthant, is defined as:

$$L_0 = \{x \mid x = \max(\mathbf{B}\mathbf{s}, 0), \forall \mathbf{s} \in R^3\} \quad (8.3)$$

A general subspace-based algorithm for the face recognition is (Georghiades *et al.* 2001):

(a) Training stage. Arrange the training samples (images of the same face under varying light conditions) as the training matrix, each column of which is an image.

* The albedo factor can be neglected, because it only scales the associated row of \mathbf{B} by the factor of a . The training matrix, which consists of the face images without shadow, should be of rank 3, no matter whether the albedo at different positions is the same or different. We do not refer to the albedo in the following.

By SVD (Golub *et al.* 1989), the 3 basis images are derived from the 3 singular vectors that correspond to the 3 largest singular values.

(b) Recognition stage. Calculate the distance of the test image to the 3-dimensional subspace that is spanned by the 3 basis images. The target is selected as that which has the shortest distance.

As we discuss in the next section, the Lambertian assumption, particularly excluding shadows, is a bit unrealistic and one usually uses slightly larger dimensional subspaces (with obvious changes to the algorithm described above).

8.2.2 Attached shadow and low-dimensional subspace

The 3-dimensional constraint does not hold when there is a shadow. Intrinsically, the dimension of the image set for an object is “equal to the number of distinct surface normals”. However, it has been proved, experimentally and theoretically, that the image set approximately lies in a low-dimensional subspace (Turk *et al.* 1991; Hallinan 1994; Eipstein *et al.* 1995). It has been experimentally proved that images with shadow can be approximately modeled by 5 ± 2 eigenimages (Eipstein *et al.* 1995). Moreover, an important theoretical proof shows that the images of a Lambertian object can be approximately modeled by a 9-dimensional subspace if there is no cast shadow (Basri *et al.* 1999; Ramamoorthi *et al.* 2001; Ramamoorthi 2002; Basri *et al.* 2003). Based on this 9-dimension theory, 9 points of light for face recognition were optimally determined (Lee *et al.* 2001).

8.2.3 Generation of the image basis from synthetic images

One does not want to use more images than necessary in constructing a training set. It has been shown (Lee *et al.* 2001) that for a single face approximately 9 well-chosen lighting directions are optimal. However, the result in (Lee *et al.* 2001) was not good enough in practice. A reliable approach to obtain the image basis is to calculate them from a large amount of training images, for example 80-120 training images (Georghiadis *et al.* 2001). Although a large set of images is unwieldy, a possible solution to this problem is to use the synthetic images, as in (Georghiadis *et al.* 2001). In this chapter, we also employ this strategy to obtain the image basis.

Taking the Yale-B face database as an example, the procedure of generating the image basis in the training stage is (the image basis for each person is separately generated in our approach), as follows:

Training algorithm:

- (a). Obtain the illumination subspace L in (8.2) from more than 2 images that have no shadow. Here, we use the 7 images in "subset 1" in Yale B face database as the training samples.
- (b). Generate the synthetic images that are illuminated by a light at infinity, using equation (8.3). Note: the nonlinear operation of setting to zero the negative pixel values moves the resulting images outside of the 3 dimensional subspace.
- (c). Calculate the approximate low-dimensional subspace from the simulated images, by SVD (Golub *et al.* 1989). The 7D basis image of person 1 and person 4 are shown in Figure 8.4.

As an important variation, we employ the iterative least squares procedure as an "outlier" detection strategy when we calculate the illumination subspace in sub-step (a), because not all the pixels of a face can be approximately Lambertian, for example the eyeballs and eyebrows. This *outlier detection strategy* will be presented in section 8.4.

In the simulation of the possible images in sub-step (b), we only consider the effect of the attached shadow, as the "Cones-attached" in (Georghiades *et al.* 2001). However, we don't need to reconstruct the face to a generalized Bas-Relief (GBR) transformation (Belhumeur *et al.* 1999). Instead, we "randomly" generate the synthetic images, because an arbitrary "linear" combination of the three basis images can be an image, illuminated by a light with unknown direction (Georghiades *et al.* 2001). It should be noted that the negative pixels in the synthetic images have to be set as zeroes. The attached shadow can be modeled this way, while the cast shadow cannot be modeled. Although the direction of the light s is randomly generated, we set the energy of first basis image to half as that of the other two basis images, in order to model the shadow effect better.

In calculating the basis images in sub-step (c), which accounts for the attached shadow, we find that the 7-dimensional subspace performs slightly better than the 9-dimensional subspace. Crucially, in our approach, the “outliers” detected in sub-step (a) are not included in calculating the distances of the test image t to the 7-dimensional subspaces. In more detail, suppose we are calculating the distance of the test image I_t from the subspace of the i^{th} training image set

$$\frac{\|e_{t,i}\|_F}{\|I\|_F} \quad (8.4)$$

where $e_{t,i}$ is the projection error calculated by ignoring (masking out) those pixels in I_t , in the same positions as we detected outliers in i^{th} training set. That is, when calculating the distance of the test image to each training image set we will exclude different parts of the test image according to the “mask” of the relevant training images.

8.3 Learning capacity of low-dimensional subspace and a new distance criterion

Few people have properly estimated where the noise in the training and recognition processes resides. In chapter 4, based on the matrix perturbation theory (Wilkinson 1965; Stewart *et al.* 1990), the learning capacity of the low-dimensional linear subspace has been studied. The theory states that the distance of a new test vector to the estimated low-dimensional subspace comes from two sources: one source is the noise in the training samples and another in the noise in the test image.

Suppose we work on the m dimensional vectors and n sample vectors are available in the training stage. The training samples can be arranged as an $m \times n$ matrix L , taking each sample as a column. Ideally, this training matrix should be of low rank r if there were no noise in the samples. For example, the training matrix, consisting of the Lambertian images without shadow, is of rank 3. By the SVD (Golub *et al.* 1989), the training matrix can be decomposed into $L = U\Sigma V^T$. U and V are $m \times r$

and $n \times r$ matrices with orthogonal columns, respectively, and Σ is a $r \times r$ diagonal matrix: $diag\{\kappa_1, \kappa_2, \dots, \kappa_r\}$. The test image \mathbf{p} should lie in the subspace $span\{\mathbf{U}\}$, i.e., $\mathbf{p} = \mathbf{U}[f_1, f_2, \dots, f_r]$, if it were noise free. Ideally, if both the training samples and the test image were noise free, the projection error of \mathbf{p} on the subspace $span\{\mathbf{U}\}$ is 0, i.e., $\|\mathbf{p} - \mathbf{U}\mathbf{U}^T\mathbf{p}\| = 0$. However, noise inevitably exists in the training samples \mathbf{L}' and in the test image \mathbf{p}' . Consequently, the noise free subspace \mathbf{U} cannot be obtained: another r -dimension subspace \mathbf{U}' is actually obtained.

From the theory in section 4.4.2 (specifically, equation 4.23 in chapter 4), the following formula describes the effects of the noise (σ_i) in the training images and the noise (σ_t) in the test image, on the "error measure" (SSD, the sum of the squared difference):

$$\|\mathbf{p}' - \mathbf{U}'\mathbf{U}'^T\mathbf{p}'\|_F^2 = (m-r)\sigma_t^2 + (m-r)\sigma_i^2 \sum_{i=1}^r \frac{f_i^2}{\kappa_i^2} \quad (8.5)$$

where σ_t and σ_i , small enough compared with the signal level, are the noise levels for the test image and the training samples respectively.

Since some error is introduced by the noise in the training samples, this part of the error in (8.5) should be subtracted in the recognition stage. More formally, suppose the new test image \mathbf{p}' has a distance of d to the r -dimensional subspace $span\{\mathbf{U}'\}$. From (8.5), we take the following distance as the criterion for the classification:

$$\sqrt{\max(d^2 - (m-r)\sigma_i^2 \sum_{i=1}^r \frac{f_i^2}{\kappa_i^2}, 0)} \quad (8.6)$$

The estimation of the noise level σ_i in the training samples will be discussed in the following section.

8.3.1 Noise level estimation

In this section, we explain how to estimate the noise level σ_i in (8.6). In our synthetic generation of the image basis, we first estimate the noise level in the actual images. Then, we use these estimates to calculate the estimates for the noise

levels in the synthetic images (generated by a linear combination of basis images, as per (b) step in section 8.2.3).

From (Bishop 1998), the maximal likelihood (ML) estimate of the noise level in an r -dimensional illumination subspace is as follows:

$$\sqrt{\frac{1}{m(m-r)} \sum_{i=r+1}^m \kappa_i^2} \quad (8.7)$$

where κ_i is the i^{th} singular value of the actual training matrix. It should be noted that the estimate in (8.7) is calculated from the outlier-detected training matrix (i.e., after removing outliers).

We calculate the total noise energy of the synthetic training matrix and regard the root mean of that energy as the noise level.

8.4 Non-Lambertian pixel detection

Although the human faces can be approximately modeled as Lambertian, some part are obviously non-Lambertian, for example the eyeballs and eyebrows. Moreover, some parts of the true training samples that are in the shadow do not obey the 3-dimension constraint. In order to obtain an accurate 3-dimensional illumination subspace, we should exclude these abnormal pixels.

Here, we employ a variant of the iterative reweighted least square (IRLS) (Rousseeuw *et al.* 1987) as the "outlier" detection strategy: the weight is either 1 or 0. IRLS works by iteratively fitting the model to the data: essentially, we adjust the weight of each data, according to its residual; then, a new model is obtained by minimizing the reweighted least square. More details can be found in (Rousseeuw *et al.* 1987).

More specifically, we retain those data whose residual is less than 3 times of the noise scale and prune the other data. Thus, a general IRLS iteratively works this way:

- (i) to estimate the scale from the residual of the retained data.

(ii) if there is some "outliers", whose residual is larger than 3 times* of the scale, to prune these data and go to sub-step (i); else, terminate the iteration.

Because we work on low-dimensional subspaces, the general 1/0 IRLS can not be directly applied to detect the non-Lambertian pixels. In particular, we have to define the residual for a pixel, which in fact is an n -dimensional vector if we work on n training images.

Suppose an $m \times n$ training matrix consists of n training images, each of which has m pixels. First, calculate the r -dimensional subspace by SVD (Golub *et al.* 1989). Second, calculate the residual matrix, by subtracting the r largest components from each column (each image). Third, calculate the 2-norm of all row (a row corresponds to the same pixel in different images) vectors and regard them as the residual for the corresponding pixels. The scale can be estimated as the root mean square of the residuals of the retained pixels. The detected mask for the non-Lambertian pixels are displayed in Figure 8.3, where the black pixels denote the non-Lambertian ones. The detected outliers do not appear to be perfect - certainly this part could be improved. However, we can observe that, from Table 8.2, this imperfect outlier strategy actually improves the face recognition performance. Also from Table 8.1, the contribution of this outlier detection strategy can be confirmed by the change of the ratio between the third and the fourth largest singular values.

8.4.1 Performance evaluation of the outlier detection strategy

For a Lambertian object, the fourth singular value of the training matrix should be zero if the training matrix is noise free. Due to noise, this does not hold. Here, we employ the ratio between the third singular value and the fourth singular value as the index to how well the matrix can be approximated by another rank-3 matrix. The larger the ratio, the better the approximation. Because we use the Yale B face database in our experiment (in section 8.5), the ratio, respectively for these 10 persons in Figure 8.1, is listed in Table 8.1. The row labeled "original" denotes the training matrices before the outlier detection and the row of "outlier detected"

* The factor 3 is somewhat arbitrary. A factor in the range 2.7 to 3 is often used. This is based on including a certain (large) fraction of inliers to a Gaussian distribution (e.g., 99%).

denotes the training matrices that do not include the detected outliers. Clearly, the outlier detection has improved the ratios. The detected outlier masks are shown in Figure 8.3.

Table 8.1: The ratio between the third singular value and the fourth singular value.

Person	1	2	3	4	5	6	7	8	9	10
original	3.761	2.460	4.216	3.967	3.050	2.265	3.261	5.364	3.923	2.159
outlier detected	4.335	4.209	5.949	4.967	5.992	2.582	4.348	7.990	5.344	3.109

8.5 Experimental results

In this section, we report our results, comparing with that in (Georghiades *et al.* 2001; Lee *et al.* 2001). As in (Georghiades *et al.* 2001), we also carry out the face recognition experiments on the Yale-B face database, which consists of 10 persons. This face database can be employed to study the pose estimation and the illumination effects. For each person, pictures were taken from 9 poses, and at each pose, 64 different illumination conditions were used. However, we only study the illumination effects on the face recognition; thus we only use the 64 frontal images. These 64 images are divided into 5 subsets of 7/12/12/14/19 pictures respectively. From “subset 1” to “subset 5”, there is more and more shadow in the pictures. Two images are shown in Figure 8.2 for each subset. In fact, the pictures in “subset 5” are almost indiscernible, as shown in Figure 8.2, and no result has previously been reported on this subset. We follow (Georghiades *et al.* 2001) in cropping, centering and resizing the images. The 10 persons are shown in Figure 8.1.

By employing the new strategies outlined in this chapter, we obtain a good performance on this subset, up to 92.1% correct. From Table 8.2, we can see that the outlier detection strategy contributes more than the new distance criterion to the improvement in our approach. We can also see that both strategies, adopted together, improve more than each strategy employed separately. Compared with the dimension-7 subspace, the dimension-9 subspace performs a little worse, as also can be observed from Table 8.2. Although the experimental setting suffers

from the small size of the Yale-B face database, we can still observe the contributions of our new strategies by comparing with other competing approaches (Georghiades *et al.* 2001; Lee *et al.* 2001). Our purpose is not to claim that we have a complete and foolproof method - rather to show that the two proposals can lead to gains in performance.

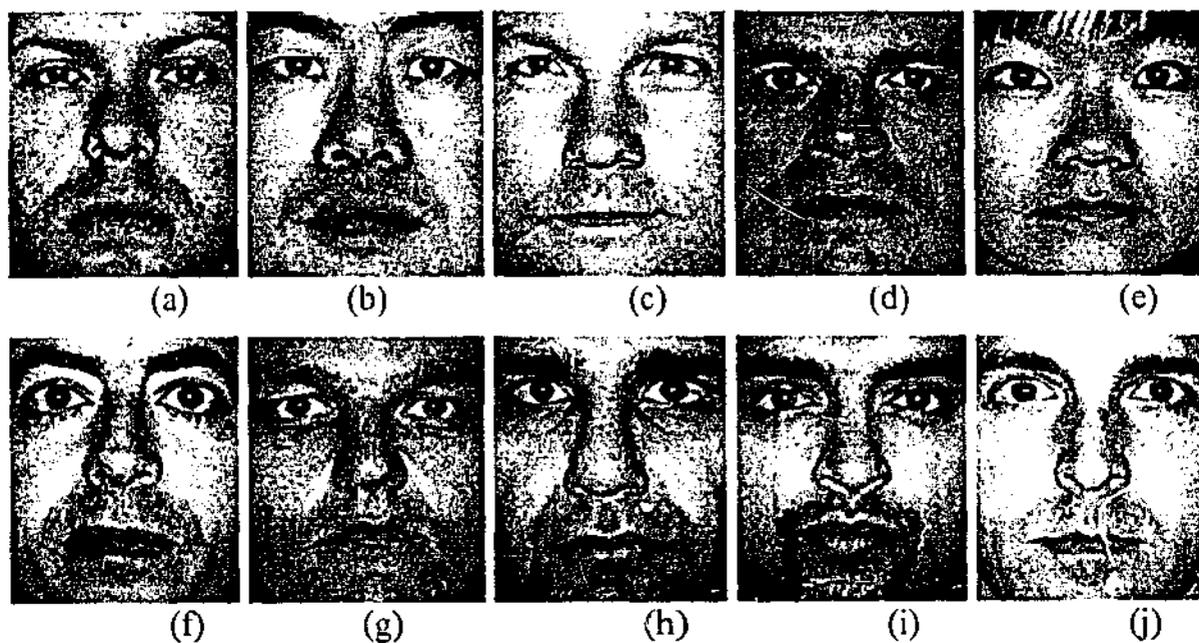


Figure 8.1: 10 persons in Yale-B face database.

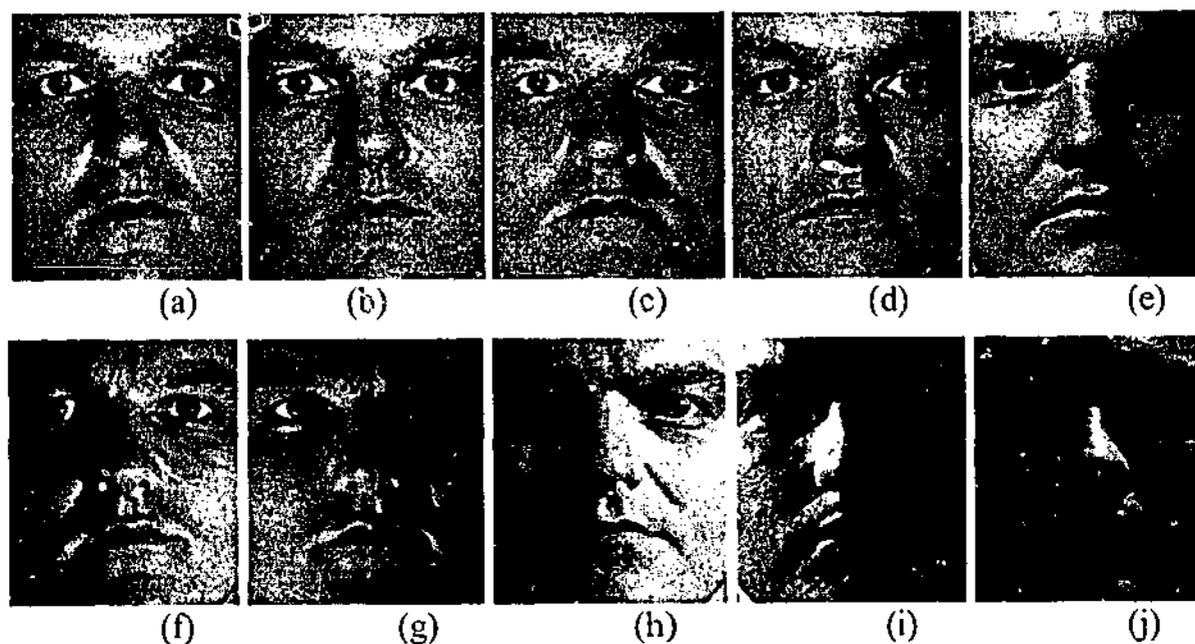


Figure 8.2: Different images under different illumination conditions, for person 7 in Figure 8.1. (a) and (b) from subset 1; (c) and (d) from subset 2; (e) and (f) from subset 3; (g) and (h) from subset 4; and (i) and (j) from subset 5.

Chapter 8: Subspace-based face recognition: outlier detection and a new distance criterion

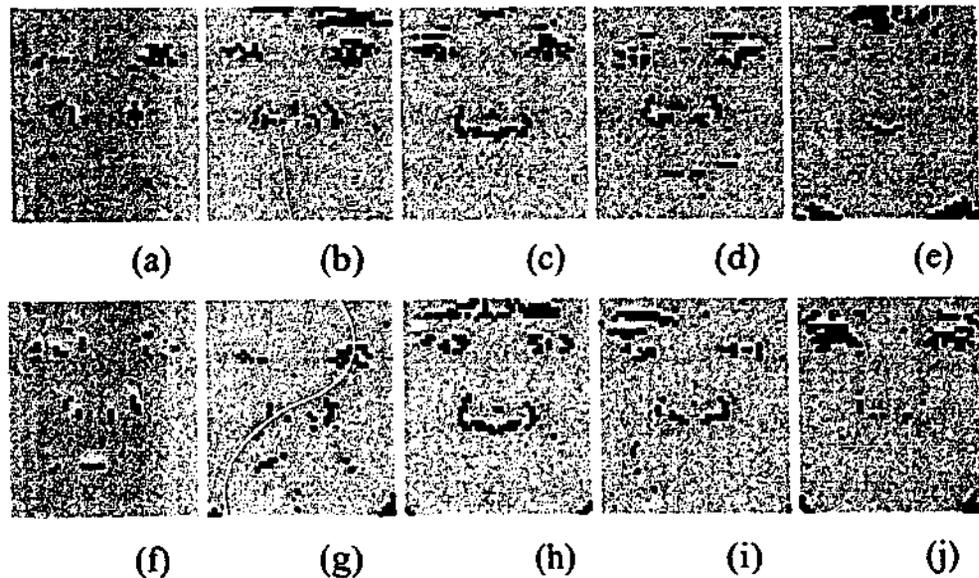


Figure 8.3: Mask for the outliers, which do not obey the 3-dimensional constraint. The black pixels denote the outliers. From (a) to (j), the masks correspond to person 1 to person 10 in Figure 8.1.

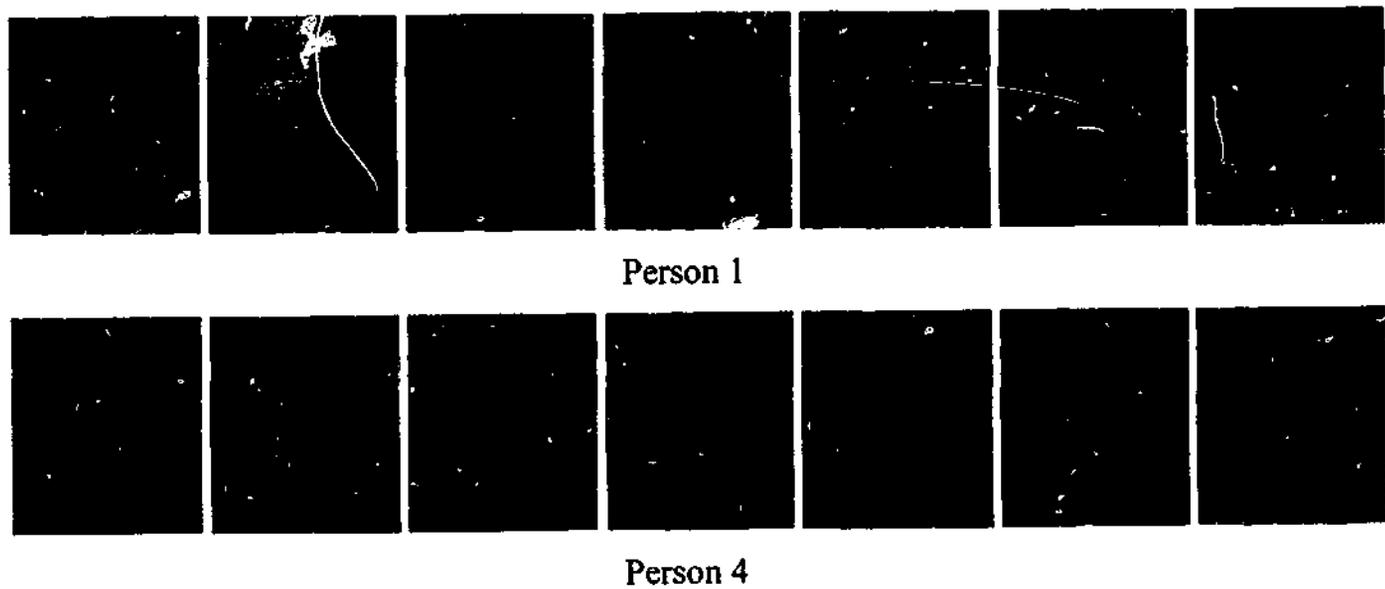


Figure 8.4: The 7D image basis of person 1 and person 4 in the Yale-B face database. Note that green denote positive pixels and red for negative, and that the detected pixels (outliers) in Figure 8.3 are excluded in these basis images, shown as black.

Table 8.2: Comparison of the error classification rate on Yale-B face database. In the bottom four rows, the numbers in the brackets are the dimensions of the subspace, used in our approach. Note that a method employing only our outlier detection strategy (row 5) performs better than previous methods (rows 1-3). Likewise, only using our new distance criterion (row 6), we can do better than previous methods except 9PL (row 3). However, both improvements combined (row 7) performs better than previous methods (rows 1-3). Method 4, Cones-cast, achieves the same flawless recognition rate on subsets 1-4; however, since they did not report an error rate for the most challenging subset 5, we can't say whether our method definitely performs better. However, it is perhaps salient that none of the cited previous methods attempted such difficult images.

	Method	Subset 1-3	Subset 4	Subset 5
1	Linear subspace (Georghiadis <i>et al.</i> 2001)	0	15	/
2	Cones-attached (Georghiadis <i>et al.</i> 2001)	0	8.6	/
3	9PL (Lee <i>et al.</i> 2001)	0	2.8	/
4	Cones-cast (Georghiadis <i>et al.</i> 2001)	0	0	/
5	Outlier detection (dim 7)	0	2.1	11.0
6	New distance (dim 7)	0	5	15.8
7	Outlier detection+ New distance (dim 7)	0	0	7.9
8	Outlier detection+ New distance (dim 9)	0	1.4	10.5

8.6 Conclusion

In this chapter, we introduce two new techniques for subspace-based face recognition: outlier detection and the exploitation of a new distance-based criterion. Without reconstructing the 3D scene, the standard subspace approach, augmented with the new techniques described here, proves to be at least comparable to Cones-cast, where the cast shadow has to be detected and consequently demands the GBR reconstruction (Belhumeur *et al.* 1999). Moreover, by the new techniques, a good performance can be obtained on "subset

5", which is the most challenging in the Yale-B face database and on which no performance has been reported before. While we can see much room for improvement in the implementation of our outlier detection strategy, we have shown that, even when only partially successful in identifying outliers from the subspace model, it can be responsible for a significant improvement in accuracy. Our new distance criteria, as it properly recognizes and adjusts for noise in the training set, is also shown to be useful as a separate component, as well as in conjunction with the outlier detection strategy.

Chapter 9

Conclusion and future directions

9.1 Summary of the contributions of this thesis

In this thesis, we mainly studied the linear subspace analysis in computer vision applications, from a statistical view. The contributions of this thesis can be categorized into three aspects: (1) a theoretical analysis of the i.i.d. Gaussian noise effect on the linear subspace analysis; (2) an investigation of heteroscedastic parameter fitting; and (3) practical applications of improved algorithms to structure from motion and face recognition problems. In this chapter, we briefly expand upon these contributions before we suggest some avenues of future work.

Theoretical contributions to the linear subspace analysis:

- Using the matrix perturbation expansion theory, we derived an explicit formula for the *denoising capacity* of the approximated (low rank) matrix, in terms of the noise level, the sizes of the measurement matrix and the dimensionality.
- By using the first-order matrix perturbation theory, we also derived an explicit formula for such a reprojection error measure. This error comes from two independent sources: one source is the noise in the training samples and another in the noise in the test image.

Practical contributions to the computer vision tasks:

- In the missing data problem under low rank constraint, we presented a criterion to recover the most *reliable* submatrix, in terms of deciding when the inclusion of extra rows or columns, containing significant numbers of missing entries, is likely to lead

to poor recovery of the missing parts. We applied this algorithm in the structure from motion problem

- We proposed a new error distance for the subspace-based recognition problem. This is based on the new theory about the *learning capacity* in low-rank subspace approaches.
- In face recognition, we employed the iterative reweighted least square (IRLS) technique to detect the pixels that do not following the Lambertian reflectance model.

In this thesis, we also studied other rank-constraint problems.

- We proved that the distance vectors under different views approximately lie in a linear subspace with a dimension of 6.
- We studied the parameter estimation problem in a general heteroscedastic linear system, by putting the problem in the framework of the bilinear approach to low-rank matrix approximation.

9.2 Directions of future work

In terms of theory, we only studied the *i.i.d. Gaussian* noise effect on the *linear* subspace analysis. It would be interesting to extend the theories in chapter 4 to the following two problems. First, it is worth studying the effect of the heteroscedastic noise in the low-rank approximation. For example, the conic fitting problem can be reduced to a rank-5 approximation of a $p \times 6$ matrix, where p is the number of points. How does the number of the points ($p \geq 6$) affect the precision of the estimation? Second, the interesting problems (e.g., conic fitting and fundamental matrix estimation) are essentially non-linear problems, although one often formulates them as a linear estimation problem. It would be interesting to investigate how the linearization affects the precision of the solution.

We only studied the illumination effects on the face recognition problem. More specifically, we employed a 7 dimensional subspace to model the attached shadow. A possible extension of this issue is to deal with the cast shadow in the face

recognition problem. Some papers about this issue have appeared in a recent conference, Europe Conference on Computer Vision 2004, such as (Frolova et al. 2004; Zhou et al. 2004).

Several recent papers (Lu et al. 1997; Mühlich et al. 2001; Manton et al. 2003) suggest different avenues of investigation for dealing with heteroscedastic rank-constrained problem. It should be interesting to use these approaches to ellipse fitting and contrast and compare.

Finally, we must acknowledge that what we have done is incomplete. For example, we studied "imputation" (filling in missing data) in an i.i.d. setting but not in a heteroscedastic setting. This thesis has made significant contributions on many fronts but a complete theory is still missing. Moreover, it has to be admitted that the algorithms described here, while making a contribution in several ways, fail to present the complete solution. Much work remains to be done.

References

- Aanaes, H., R. Fisker, et al. (2002). "Robust factorization." *IEEE Trans Pattern Analysis and Machine Intelligence* 24(9): 1215-1225.
- Aguiar, P. M. Q. and J. M. F. Moura (1999). Factorization as a rank 1 problem. *Proc. Conf. Computer Vision and Pattern Recognition*, 178-184.
- Aguiar, P. M. Q. and J. M. F. Moura (2000). Weighted factorization. *ICIP*, 549-552.
- Aguiar, P. M. Q. and J. M. F. Moura (2003). "Rank 1 weighted factorization for 3D structure recovery: algorithms and performance analysis." *IEEE Trans Pattern Analysis and Machine Intelligence* 25(9): 1134-1149.
- Anandan, P. and M. Irani (2002). "Factorization with uncertainty." *Int'l J. Computer vision* 49(2/3): 101-116.
- Basri, R. and D. W. Jacobs (1999). Lambertian Reflectance and Linear Subspaces. *Proc. Int'l Conf. Computer Vision*, 383-390.
- Basri, R. and D. W. Jacobs (2003). "Lambertian Reflectance and Linear Subspaces." *IEEE Trans Pattern Analysis and Machine Intelligence* 25(2): 218-233.
- Belhumeur, P., D. J. Kriegman, et al. (1999). "The Bas-Relief Ambiguity." *Int'l J. Computer vision* 35(1): 33-44.
- Belhumeur, P. N., J. Hespanha, et al. (1997). "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection." *IEEE Trans Pattern Analysis and Machine Intelligence* 19(7): 711-720.

- Belhumeur, P. N. and D. Kriegman (1998). "What is the Set of Images of an Object under all Possible Illumination Conditions?" *Int'l J. Computer vision* 28(3): 245-260.
- Biennu, G. and L. Kopp (1983). "Optimality of high-resolution array processing using eigensystem approach." *IEEE Trans Acoustics, Speech, and Signal Processing* 31(5): 1235-1248.
- Bishop, C. M. (1998). *Bayesian PCA*. NIPS, MIT Press, 382-388.
- Brand, M. (2002). Incremental Singular Value Decomposition of Uncertain Data with Missing Values. *Proc. European Conf. Computer Vision*, 707-720.
- Brand, M. (2003). Fast online SVD revisions for lightweight recommender systems. *SIAM 3rd International Conference on Data Mining*.
- Brandt, S. (2002). Closed-Form Solutions for Affine Reconstruction under Missing Data. *Statistical Methods for Video Processing Workshop*, in conjunction with ECCV02, 109-114.
- Chen, P. and D. Suter (2003). "An analysis of linear subspace approaches for computer vision and pattern recognition." Technical Report MECSE-6-2003, Monash University, Clayton 3800, Australia, 2003. <http://www.ds.eng.monash.edu.au/techrep/reports/2003/MECSE-6-2003.pdf>.
- Chen, P. and D. Suter (2004). "Subspace-based face recognition: outlier detection and a new distance criterion." Accepted by *International Journal of Pattern Recognition and Artificial Intelligence* (Special issue of ACCV 2004).
- Chen, P. and D. Suter (2004). Subspace-based face recognition: outlier detection and a new distance criterion. *Asian Conf. Computer Vision* 2004.830-835.

- Chen, P. and D. Suter (2004). "Recovering the missing components in a large noisy low-rank matrix: Application to SFM." To appear in *IEEE Trans Pattern Analysis and Machine Intelligence* 26(8).
- Chen, Q. and G. Medioni (1999). Efficient iterative solution to m-view projective reconstruction problem. *Proc. Conf. Computer Vision and Pattern Recognition*, 55-61.
- Chien, Y. T. and K. S. Fu (1967). "On the generalized Karhunen Loeve expansion." *IEEE Trans Inform. Theory* 13(3): 518-520.
- Chojnacki, W., M. J. Brooks, et al. (2000). "On the fitting of surfaces to data with covariances." *IEEE Trans Pattern Analysis and Machine Intelligence* 22(11): 1294-1303.
- Chojnacki, W., M. J. Brooks, et al. (2003). "Revisiting Hartley's normalized eight-point algorithm." *IEEE Trans Pattern Analysis and Machine Intelligence* 25(9): 1172 - 1177.
- Chojnacki, W., M. J. Brooks, et al. (2004). "From fns to heiv: a link between two vision parameter estimation methods." *IEEE Trans Pattern Analysis and Machine Intelligence* 26(2): 264-268.
- Costeira, J. and T. Kanade (1995). A multi-body factorization method for motion analysis. *Proc. Int'l Conf. Computer Vision*, 1071 - 1076.
- Costeira, J. and T. Kanade (1998). "A multibody factorization method for independently moving objects." *Int'l J. Computer vision* 29(3): 159-179.
- Eipstein, R., P. Hallinan, et al. (1995). 5±2 eigenimages suffices: An empirical investigation of low-dimensional lighting models. *Proc. IEEE Workshop Physics-Based vision*, 108-116.

- F. De la Torre and M. J. Black (2003). "A framework for robust subspace learning." *IJCV* 54: 117-142.
- Fitzgibbon, A., G. Cross, et al. (1998). Automatic 3D Model Construction for Turn-Table Sequences, 3D Structure from Multiple Images of Large-Scale Environments. LNCS 1506, 155-170.
- Fitzgibbon, A., M. Pilu, et al. (1999). "Direct least square fitting of ellipses." *IEEE Trans Pattern Analysis and Machine Intelligence* 21(5): 476-480.
- Frolova, D., D. Simakov, et al. (2004). Accuracy of Spherical Harmonic Approximations for Images of Lambertian Objects under Far and Near Lighting. *Proc. European Conf. Computer Vision*.574-587.
- Gear, C. W. (1998). "Multibody grouping from motion images." *Int'l J. Computer vision* 29(2): 133-150.
- Georghiades, A., P. N. Belhumeur, et al. (2001). "From Few to Many: Generative Models of Object Recognition." *IEEE Trans Pattern Analysis and Machine Intelligence* 23(6): 643-660.
- Georghiades, A., D. Kriegman, et al. (1998). Illumination cones for recognition under variable lighting: faces. *Proc. Conf. Computer Vision and Pattern Recognition*, 52-58.
- Gold, S. and A. Rangarajan (1996). "A graduated assignment algorithm for graph matching." *IEEE Trans Pattern Analysis and Machine Intelligence* 18(4): 377-388.
- Golub, G. H. and C. F. V. Loan (1996). *Matrix Computations*. Baltimore, Johns Hopkins University Press.

- Guerreiro, R. F. C. and P. M. Q. Aguiar (2003). Estimation of Rank Deficient Matrices from Partial Observations: Two-Step Iterative Algorithms. *Energy Minimization Methods in Computer Vision and Pattern Recognition*.
- Hallinan, P. (1994). A low-dimensional representation of human faces for arbitrary lighting conditions. *Proc. Conf. Computer Vision and Pattern Recognition*, 995-999.
- Harris, C. J. and M. Stephens (1988). A combined corner and edge detector. *Proc. Alvey Vision Conference*, 147-151.
- Hartley, R. and A. Zisserman (2000). *Multiple view geometry in computer vision*, Cambridge University Press.
- Heyden, A. (1997). Projective structure and motion from image sequences using subspace methods. *Scandinavian Conference on Image Analysis*, 963-968.
- Heyden, A. and F. Kahl (1998). Reconstruction from Affine Cameras using Closure Constraints. *ICPR*, 47-50.
- Huffel, S. V. and J. Vandewalle (1991). *The total least squares problem: computational aspects and analysis*, SIAM.
- Huynh, D. Q. and A. Heyden (2001). Outlier detection in video sequences under affine projection. *Proc. Conf. Computer Vision and Pattern Recognition*, 695-701.
- Irani, M. (1999). Multi-frame optical flow estimation using subspace constraints. *Proc. Int'l Conf. Computer Vision*, 626-633.
- Irani, M. (2002). "Multi-frame correspondence estimation using subspace constraints." *Int'l J. Computer vision* 48(3): 173-194.

- Irani, M. and P. Anandan (2000). Factorization with uncertainty. Proc. European Conf. Computer Vision, 539-553.
- Jacobs, D. (1997). Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. Proc. Conf. Computer Vision and Pattern Recognition, 206-212.
- Jacobs, D. (2001). "Linear fitting with missing data for structure-from-motion." Computer Vision and Image Understanding 82: 57-81.
- Jacobs, D., P. N. Belhumeur, et al. (1998). Comparing Images Under Variable Illumination. Proc. Conf. Computer Vision and Pattern Recognition, 610-617.
- Kahl, F. and A. Heyden (1998). Structure and motion from points, lines and conics with affine cameras. Proc. European Conf. Computer Vision, 327-341.
- Kahl, F. and A. Heyden (1999). "Affine structure and motion from points, lines and conics." Int'l J. Computer vision 33(3): 163-180.
- Kanatani, K. (1993). "Unbiased estimation and statistical analysis of 3-D rigid motion from two views." IEEE Trans Pattern Analysis and Machine Intelligence 15(1): 37 - 50.
- Kanatani, K. (1994). "Statistical bias of conic fitting and renormalization." IEEE Trans Pattern Analysis and Machine Intelligence 16(3): 320 - 326.
- Kanatani, K. (1996). Statistical Optimization for Geometric Computation: Theory and Practice, Amsterdam: Elsevier.
- Kanatani, K. (2001). Motion segmentation by subspace separation and model selection. Proc. Int'l Conf. Computer Vision, 301-306.

- Kung, S. Y. and K. S. Arun (1987). "SVD algorithms for linear system approximation and spectrum estimation." *Advances in Statistical Signal Processing 1*: 203-250.
- Lee, K., J. Ho, et al. (2001). Nine points of light acquiring subspaces for face recognition under variable lighting. *Proc. Conf. Computer Vision and Pattern Recognition*, 519-526.
- Leedan, Y. and P. Meer (1999). Estimation with bilinear constraints in computer vision. *Proc. Int'l Conf. Computer Vision*, 733-738.
- Leedan, Y. and P. Meer (2000). "Heteroscedastic Regression in Computer Vision: Problems with Bilinear Constraint." *Int'l J. Computer vision* 37(2): 127-150.
- Lu, W. S., S. C. Pei, et al. (1997). "Weighted low-rank approximation of general complex matrices and its application in the design of 2-D digital filter." *IEEE Trans Circuits Syst.* 144: 650-655.
- Mahamud, S., M. Hebert, et al. (2003). "Provably-Convergent Iterative Methods for Projective Structure from Motion." Submitted to *Int'l J. Computer vision*.
- Mahamud, S., M. Hebert, et al. (2001). Provably-Convergent Iterative Methods for Projective Structure from Motion. *Proc. Conf. Computer Vision and Pattern Recognition*, 1018-1025.
- Manton, J. H., R. Mahony, et al. (2003). "The geometry of weighted low-rank approximations." *IEEE Trans Signal Processing* 51(2): 500-514.
- Mardia, K. V. (1979). *Multivariate analysis*. New York, Academic Press.
- Martinec, D. and T. Pajdla (2002). Structure from many perspective images with occlusion. *Proc. European Conf. Computer Vision*, 355-369.

- Maslov, S. and Y. C. Zhang (2001). "Extracting Hidden Information from Knowledge Networks." *Physical Review Letters* 87(24): 248701.
- Matei, B. and P. Meer (1999). Optimal rigid motion estimation and performance evaluation with bootstrap. *ICPR*, 345.
- Matei, B. and P. Meer (2000). A general method for Errors-in-Variables problems in computer vision. *Proc. Conf. Computer Vision and Pattern Recognition*, 18-25.
- Matei, B. and P. Meer (2000). Reduction of bias in maximum likelihood ellipse fitting. *ICPR*, 794-798.
- Moore, B. C. (1981). "Principal component analysis in linear systems, controllability, observability, and model reduction." *IEEE Trans Automatic Control* 26(3): 17-32.
- Morita, T. and T. Kanade (1997). "A sequential factorization method for recovering shape and motion from image streams." *IEEE Trans Pattern Analysis and Machine Intelligence* 19(8): 858-867.
- Morris, D. D. and T. Kanade (1998). A unified factorization algorithm for points, line segments and planes with uncertainty models. *Proc. Int'l Conf. Computer Vision*, 696-702.
- Moses, Y., Y. Adini, et al. (1994). Face recognition: The problem of compensating for changes in illumination direction. *Proc. European Conf. Computer Vision*, 286-296.
- Mühlich, M. and R. Mester (2001). Subspace Methods and Equilibration, *Computer Vision. Scandinavian Conference on Image Analysis*

- Murase, H. and S. K. Nayar (1994). "Illumination planning for object recognition using parametric eigenspaces." *IEEE Trans Pattern Analysis and Machine Intelligence* 16(12): 1219-1227.
- Murase, H. and S. K. Nayar (1995). "Visual learning and recognition of 3-D objects from appearance." *Int'l J. Computer vision* 14: 5-24.
- Nene, S. A., S. K. Nayar, et al. (1994). Software library for appearance matching (SLAM). ARPA Image understanding workshop.
- Oliensis, J. (1999). Fast and accurate self-calibration. *Proc. Int'l Conf. Computer Vision*, 745-752.
- Poelman, C. and T. Kanade (1997). "A paraperspective factorization method for shape and motion recovery." *IEEE Trans Pattern Analysis and Machine Intelligence* 19(3): 206-219.
- Pratt, W. K. (1975). *Digital image processing*. New York, Wiley.
- Press, W. H., S. A. Teukolsky, et al. (1992). *Numerical recipes in C*, Cambridge University Press.
- Quan, L. and T. Kanade (1996). A factorization method for affine structure from line correspondences, *Proc. Conf. Computer Vision and Pattern Recognition*, 803 - 808.
- Quan, L. and T. Kanade (1997). "Affine structure from line correspondences with uncalibrated affine cameras." *IEEE Trans Pattern Analysis and Machine Intelligence* 19(8): 834 - 845.
- Ramamoorthi, R. (2002). "Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian Object." *IEEE Trans Pattern Analysis and Machine Intelligence* 24(10): 1322-1333.

- Ramamoorthi, R. and P. Hanrahan (2001). "On the relationship between radiance and irradiance: Determining the illumination from images of a convex Lambertian object." *Journal of the Optical Society of America (JOSA A)* 18(10): 2448-2459.
- Reid, I. D. and D. W. Murray (1996). "Active tracking of foveated feature clusters using affine structure." *Int'l J. Computer vision* 18(1): 41-60.
- Rother, C. and S. Carlsson (2002). Linear multi view reconstruction with missing data. *Proc. European Conf. Computer Vision*, 309-324.
- Rother, C., S. Carlsson, et al. (2002). Projective Factorization of Planes and Cameras in Multiple Views. *ICPR*, 737-740.
- Rousseeuw, P. J. and A. M. Leroy (1987). *Robust regression and outlier detection*. New York, Wiley.
- Sarwar, B. M., G. Karypis, et al. (2000). Application of dimensionality reduction in recommender system -- A case study. *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*.
- Schmidt, R. O. (1979). "Multiple emitter location and signal parameter estimation." *IEEE Trans Antenna and Propog.* 34(3): 276-280.
- Shashua, A. (1997). "On photometric issues in 3D visual recognition from a single 2D image." *Int'l J. Computer vision* 21(1/2): 99-122.
- Shum, H., K. Ikeuchi, et al. (1995). "Principal component analysis with missing data and its applications to polyhedral object modeling." *IEEE Trans Pattern Analysis and Machine Intelligence* 17(9): 854-867.
- Stewart, G. W. and J. G. Sun (1990). *Matrix perturbation theory*, Academic press.

- Sturm, P. and B. Triggs (1996). A factorization based algorithm for multi-image projective structure and motion. Proc. European Conf. Computer Vision, 709-720.
- Thomas, J. I. and J. Oliensis (1999). "Dealing with noise in multiframe structure from motion." Computer Vision and Image Understanding 76(2): 109-124.
- Tomasi, C. and T. Kanade (1992). "Shape and motion from image streams under orthography: A factorization method." Int'l J. Computer vision 9(2): 137-154.
- Triggs, B. (1996). Factorization methods for projective structure and motion. Proc. Conf. Computer Vision and Pattern Recognition, 845 - 851.
- Troyanskaya, O., M. Cantor, et al. (2001). "Missing value estimation methods for DNA microarrays." Bioinformatics 17: 520-525.
- Turk, M. and A. Pentland (1991). "Eigenfaces for recognition." J. Cognitive Neuroscience 3(1): 71-96.
- Vidal, R. and R. Hartley (2004). Motion segmentation with missing data using PowerFactorization and GPCA. Proc. Conf. Computer Vision and Pattern Recognition.
- Wilkinson, J. H. (1965). The algebraic eigenvalue problem. Oxford, Clarendon Press.
- Yuille, A. L., D. Snow, et al. (1999). "Determining generative models of objects under varying illumination: shape and albedo from multiple images using SVD and integrability." Int'l J. Computer vision 35(3): 203-222.
- Zelnik-Manor, L. and M. Irani (1999). Multi-View Subspace Constraints on Homographies. Proc. Int'l Conf. Computer Vision, 710-715.

Zelnik-Manor, L. and M. Irani (2002). "Multi-View Subspace Constraints on Homographies." *IEEE Trans Pattern Analysis and Machine Intelligence* 24(2): 214-223.

Zhang, Z. (1997). "parameter estimation techniques: A tutorial with application to conic fitting." *Image and Vision Computing* 15: 59-76.

Zhao, L. and Y. H. Yang (1999). "Theoretical analysis of illumination in PCA-based vision systems." *Pattern recognition* 32: 547-564.

Zhou, S. K., R. Chellappa, et al. (2004). Characterization of Human Faces under Illumination Variations Using Rank, Integrability, and Symmetry Constraints. *Proc. European Conf. Computer Vision*.588-601

Errata and Amendments

- p IX, para 3, line 4: Replace “following” with “follow”.
- p IX, para 4, line 1: Replace “rank-constraint” with “rank-constrained”.
- p IX, para 4, line 2: Replace “approximately lie in a linear subspace with a dimension of 6.” with “lie in a space that is well approximated by a 6-dimensional linear subspace.”
- p 1, para 2, line 6: Replace “signal” with “matrix”.
- p 2, para 1, line 3: Add “(Aguiar *et al.* 2002; Sun *et al.* 2001)” after “frames”.
- p 2, para 3, line 3: Insert “(SFM)” between “motion” and “and”.
- p 3, para 1, line 3: Insert “Guerreiro *et al.* 2002; Maruyama *et al.* 1999” before “), etc”.
- p 3, para 2, line 4: Replace the last sentence with “The iterative methods have the advantage that they are driven by minimizing a global cost function. In contrast, the non-iterative ones are clearly suboptimal, although they are generally fast and stable.”
- p 3, para 4, line 1: Add “, i.e., each data point having a different covariance matrix)” after “heteroscedastic noise”
- p 4, para 2, line 2: Insert “Wiberg 1976” after “2004”.
- p 6, para 3, line 4: Replace “important in since” with “important since”.
- p 6, the second last para: Replace “approximately lie in a linear subspace with a dimension of 6.” with “lie in a space that is well approximated by a 6-dimensional linear subspace.”
- p 10, para 1, line 3: “ V ” in equation (2.1) should be bold as “ \mathbf{V} ”.
- p 10, para 2, line 2: Delete “first” between “the” and “ n ”.
- p 10, para 2, lines 4&6: Replace λ_i^2 with σ_i^2 .
- p 11, para 2, line 4: Add “be” between “will” and “produced”.
- p 13, para 1, the second last line: add “the” between “is” and “same”
- p 13, para 2, line 1: Replace “Theory 1” with “Theorem 1”.
- p 15, the last para: Explanation: “large” in line 2 means that the singular values have to be much larger than the noise level in the matrix in order to make (2.14-2.16) approximately hold. For a small singular value, the singular vectors (values), calculated from (2.14-2.16), may have a large error.
- p 16, para 3, line 2: Replace “eigenvalues” with “eigenvectors”.
- p 16, para 3, line 3: Replace “same” with “equal”.
- p 22 para 2 line 3: Insert “to surface patches (Guerreiro *et al.* 2002; Maruyama *et al.* 1999),” before “and to planes”.
- p 28, para 1, the fourth last line: Replace “more optimal” with “better”.
- p 42, para 1, line 2&3; p 44, para 1, line 4: “the energy of a matrix entry” denotes the quantity $\frac{\|\mathbf{M}\|_F^2}{mn}$, where \mathbf{M} is the matrix, and m and n are its sizes.
- p 44, between (43) and (44), the energy of a matrix, \mathbf{M} , denotes its norm $\|\mathbf{M}\|_F^2$.
- p 47, in the fourth last line; and p 52, line 8: the energy of a vector (an image), \mathbf{v} , denotes its norm $\|\mathbf{v}\|_F^2$.
- p 50, In the legend of Fig. 4.3, replace “To see” with “See”.
- p 51, In the second last para, the second last line, replace “angel” with “angle”.
- p 67, In the legend of Fig. 5.1, replace “the missing percentage” with “the fraction of missing data”.

- p 73: Add "For an explanation of the traces, see the text in page 72." to the legend of Fig. 5.2.
- p 78, para 1, line 3: Add a sentence: "The data of the tracked feature points was downloaded from Jacobs' homepage." before "One frame".
- p 126, as the concluding sentence to section 7.2.3, add "It should be noted that one could question whether an expansion in terms of other basis functions may be preferred; however, investigation of this question is beyond the scope of this thesis."
- p 140: Replace "Different images" with "Images".
- p 148: Replace "2004" in the second last reference with "2004a".
- p 148: Replace "2004" in the last reference with "2004b".
- p 149: Replace "2004" in the first reference with "2004c".
- p 22, para 8, Replace "Chen *et al.* 2004" with "Chen *et al.* 2004c".
- p 46, para 1, Replace "Chen *et al.* 2004" with "Chen *et al.* 2004c".
- p 53, para 3, Replace "Chen *et al.* 2004" with "Chen *et al.* 2004c".
- p 96, the last para, Replace "Chen *et al.* 2004" with "Chen *et al.* 2004c".
- p 131, para 3, Replace "Chen *et al.* 2004" with "Chen *et al.* 2004a; Chen *et al.* 2004b"

Insert the following references between p 147-158:

- Aguiar, P. M. Q. and J. M. F. Moura (1998). Video representation via 3D shaped mosaics. IEEE International Conference on Image Processing ICIP'98.
- Aguiar, P. M. Q. and J. M. F. Moura (2001). "Three-dimensional modeling from two-dimensional video." IEEE Transactions on Image Processing 10(10).
- Aguiar, P. M. Q. and J. M. F. Moura (2002). 3D rigid structure from video: What are "easy" shapes and "good" motions? IEEE Multimedia signal processing workshop.
- Guerreiro, R. F. C. and P. M. Q. Aguiar (2002). 3D Structure from video streams with partially overlapping images. IEEE International Conference on Image Processing ICIP'02.
- Maruyama, M. and S. Kurumi (1999). Bidirectional optimization for reconstructing 3D shape from an image sequence with missing data. IEEE International Conference on Image Processing ICIP'99.
- Sun, Z., V. Ramesh, et al. (2001). "Error characterization of the factorization method." Computer Vision and Image Understanding 82: 110-137.
- Wiberg, T. (1976). Computation of principle components when data are missing. 2nd Symposium computational statistics.

Additional comments on chapter 6 In chapter 6, there are several references to the assumption of Gaussian noise. In fact, this assumption is only necessary to obtain the *ML* result in section 6.3.4. For the remainder of the chapter, a weaker assumption (zero mean noise) suffices.

The following explanation may help the reader to interpret equation (6.4): The noiseless data points $\mathbf{W}_o = [\mathbf{w}_{1o}, \mathbf{w}_{2o}, \dots, \mathbf{w}_{no}]$ are "nuisance parameters". That is, they are not directly of interest. $\hat{\mathbf{W}}_o = [\hat{\mathbf{w}}_{1o}, \hat{\mathbf{w}}_{2o}, \dots, \hat{\mathbf{w}}_{no}]$ is the rank $n-1$ matrix that minimizes the objective function in (6.4), subject to the constraint $\hat{\mathbf{\theta}}^T \hat{\mathbf{W}} = \mathbf{0}^T$, where $\hat{\mathbf{\theta}}$ is the estimate of the practical parameters of interest ($\mathbf{\theta}$).