Robust Estimation in Computer Vision: Optimisation Methods and Applications

by

Trung Thanh Pham

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering, Computer and Mathematical Sciences
School of Computer Science

May 2014
This dissertation is dedicated to my parents,
my brothers and my sisters
for their unconditional love and endless support.
## Contents

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Declaration

Publications

Acknowledgements

1 Introduction

1.1 Robust Estimation in Vision & Challenges 3

1.2 Research Contributions 6

1.3 Thesis Outline 7

2 Optimisation Algorithms for Robust Estimation

2.1 Robust Fitting Criterion 10

2.1.1 Goodness of Fit 10

2.1.2 Model Complexity 10

2.1.3 Other Requirements 11

2.1.4 Objective Functions 11

2.2 Optimisation Algorithms for Model Selection 12

2.2.1 Tabu Search 13

2.2.2 Branch and Bound 15

2.2.3 Linear Programming Relaxation 17

2.2.4 Message Passing 18

2.2.5 Graph-cuts and Expansion Move Algorithm 19

2.2.6 Quadratic Programming 21

2.2.7 Discussion 23
## Contents

5.1.1 Fundamental Matrix .......................................................... 72
5.1.2 Multiple Fundamental Matrices for Dynamic Scenes .................. 73
5.1.3 Homography Matrix .......................................................... 73
5.1.4 Multiple Homography Matrices ............................................. 74
5.1.5 Estimating Image-to-image Transformations ............................ 74

5.2 Two-view Motion Segmentation ............................................... 75
5.2.1 Problem Statement .......................................................... 75
5.2.2 Estimation Methods .......................................................... 76
5.2.3 Experimental Design .......................................................... 76
5.2.3.1 Datasets ................................................................. 76
5.2.3.2 Benchmarking ........................................................... 77
5.2.3.3 Error Metric ............................................................. 77
5.2.4 Results ............................................................................. 78

5.3 Two-view Multi-homography Detection ...................................... 79
5.3.1 Problem Statement .......................................................... 79
5.3.2 Datasets ................................................................. 80
5.3.3 Results ................................................................. 80

5.4 Augmented Reality using Homography Mappings .......................... 82
5.4.1 Offline Preprocessing .......................................................... 84
5.4.2 Online Multiple Homography Detection .................................. 85
5.4.3 Plane Matching and Image Augmentation ............................... 85

5.5 Summary ................................................................. 86

6 Fitting with Geometric Priors .................................................. 91
6.1 High-Level Priors in Vision ...................................................... 92
6.2 Robust Fitting with Geometric Priors ....................................... 94
6.2.1 Energy Function ............................................................ 94
6.2.2 Energy Normalisation ....................................................... 96
6.2.3 Energy Minimisation ......................................................... 96
6.2.4 Interacting Geometric Costs vs. Category Costs ....................... 100

6.3 Experiments and Applications ................................................ 101
6.3.1 Two-view Multi-homography Estimation ............................... 101
6.3.2 Vanishing Point Detection in a Manhattan World. .................. 104
6.3.3 Circular Object Detection in Images .................................... 110
6.3.3.1 Fitting Circles ........................................................ 111
6.3.3.2 Fitting Disks ........................................................... 112
6.3.4 Planar Surface Reconstruction from 3D Point Clouds ............... 114
6.3.5 Tuning Parameter Sensitivity .............................................. 115

6.4 Summary ................................................................. 116

7 Conclusions and Future Work .................................................. 119
7.1 Summary of Contributions ...................................................... 119
7.2 Future Research Directions .................................................... 121
Contents

Bibliography 123
List of Figures

1.1 Typical computer vision applications which rely on robust estimation. . . . 2
1.2 An example of multiple circle fitting data. ................................. 4

2.1 An illustration of the over-fitting issue. ................................. 11
2.2 Robust parameter estimation using a global optimisation approach. . . . 13
2.3 A binomial solution tree for Branch-and-Bound. ........................ 16
2.4 Factor graph construction for multi-model fitting ........................ 19
2.5 The exponential increase of the number of required hypotheses against
the outlier ratio. .............................................................. 26
2.6 The exponential increase of the number of required hypotheses against
the size of the minimal subsets. ............................................. 26
2.7 Different types of “all-inlier” minimal subsets for the line and circle fitting. 28

3.1 Global optimisation using a simulated annealing approach. ............... 39
3.2 2D point cloud generated from 5 lines. .................................. 48
3.3 The evolution of data similarities. .......................................... 49
3.4 The evolution of ARJMC as the number of iterations increases. .......... 49
3.5 The evolution of the objective function value as ARJMC progresses. .... 49
3.6 The evolution of the estimated number of structures as ARJMC progresses. 50
3.7 Performance comparison between ARJMC and FLOSS on a motion seg-
mentation problem. ............................................................. 51

4.1 Line hypotheses fitted on a minimal and larger-than-minimal subset. ..... 55
4.2 A conceptual illustration of the RCM method on a multiple line fitting
problem. ........................................................................... 56
4.3 Demonstrating the RCM hypothesis generator on a multi-homography
fitting problem. ............................................................... 56
4.4 Qualitative comparison of hypotheses progressively generated using RANSAC,
Multi-GS and RCM on a multiple line fitting problem. ................... 63
4.5 Matrix of edge probabilities for the multiple line fitting task. .............. 65
4.6 Multi-homography detection on Raglan Castle. ............................. 67
4.7 Evolution of objective value by Algorithm 4.1 using different sampling
schemes. ........................................................................... 68
4.8 The probability of hitting all-inlier subsets for different sampling methods
as Algorithm 4.1 progresses. ................................................. 68
4.9 Evolution of objective value by Algorithm 4.1 using different sampling
schemes with local refinement and exploration. ............................. 69

5.1 Two-view geometry and fundamental matrix. ............................... 72
List of Figures

5.2 Two-view geometry and homography matrix. 73
5.3 Correspondence extraction between two views. 75
5.4 Comparing efficiency on motion segmentation on a subset of the data. 80
5.5 Comparing efficiency on homography detection on a subset of the data. 82
5.6 Using our RCMSA fitting method in an augmented reality application. 83
5.7 Preprocessing for an automatic augmented reality application. 84
5.8 Associating planes and estimated homographies. 85
5.9 Qualitative motion segmentation on breadcartoychips dataset. 87
5.10 Qualitative motion segmentation on cubebreadtoychips dataset. 87
5.11 Qualitative motion segmentation on breadtoycar dataset. 88
5.12 Qualitative motion segmentation using biscuitbookbox dataset. 88
5.13 Qualitative multi-homography detection using Johnsona dataset. 89
5.14 Qualitative multi-homography detection using Merton College 1 dataset. 89
5.15 Qualitative multi-homography detection using Unionhouse dataset. 90
5.16 Two examples where RCMSA fails to recover all the structures on multi-homography detection. 90

6.1 Two examples of computer vision applications where geometric priors should be effectively exploited to produce plausible results. 92
6.2 Graph construction for the expansion move energy. 100
6.3 Multiple homography estimation with geometric constraints. 104
6.4 Approximated residual from an edge to a vanishing point. 106
6.5 Quantitative vanishing point detection results using manually extracted lines. 107
6.6 Quantitative vanishing point detection results using automatically extracted lines. 108
6.7 Qualitative vanishing point detection results using automatically extracted lines. 109
6.8 Circular object detection using an image of diatom. 111
6.9 Circular object detection using an image of cells. 112
6.10 Counting flamingoes via disk model fitting. 113
6.11 Counting cells via disk model fitting. 114
6.12 Fitting planes to Pozzoveggiani 3D point cloud. 116
6.13 Fitting planes to Piazza Dante point cloud. 117
6.14 Fitting planes to Merton College 3 3D point cloud. 117
List of Tables

2.1 Different local-and-global search strategies for hypothesis sampling. . . . . 29

5.1 Summary statistics (median) over 50 repetitions on fundamental matrices
detection. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79

5.2 Summary statistics (median) over 50 repetitions on multi-homography
detection. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81

6.1 Quantitative results on two-view multi-homography estimation with geo-
metric constraints. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
List of Algorithms

2.1 Global optimisation based multi-structure robust estimation . . . . . . . . 12
2.2 Expansion move algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 21
2.3 RANSAC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
2.4 The guided-MLESAC sampling algorithm . . . . . . . . . . . . . . . . . . 28
2.5 The LO-RANSAC sampling algorithm . . . . . . . . . . . . . . . . . . . . 30
2.6 The NAPSAC sampling algorithm . . . . . . . . . . . . . . . . . . . . . . 31
2.7 The Multi-GS guided sampling algorithm . . . . . . . . . . . . . . . . . . . 32

3.1 Metropolis-Hastings algorithm. . . . . . . . . . . . . . . . . . . . . . . . . 37
3.2 Monte Carlo Markov Chain simulated annealing for geometric fitting. . . 40
3.3 Reversible jump mixture of kernels MCMC . . . . . . . . . . . . . . . . . . 40

4.1 Graph cuts based annealing optimisation for geometric fitting. . . . . . . 59
4.2 Updating Edge Probabilities in RCM . . . . . . . . . . . . . . . . . . . . . 64

6.1 Iterative hypothesis sampling and fitting for homography estimation. . . 103
Abstract

Robust parameter estimation is an important area in computer vision that underpins many practical applications. Typically, the task is to estimate a generic model from unstructured observations, where the model and observed data may vary depending on the specific applications. In most cases, computer vision data inherently contains noisy measurements, multiple instances (structures) of a model, and outliers (i.e., points that do not belong to any structures). Unfortunately, standard techniques such as Least Squares (LS), Least Median Squares (LMS) are not robust to such kind of data.

Over the past decades, much research effort in computer vision has been devoted to proposing more robust and efficient estimators. Among those, the estimators based on global optimisation have attracted considerable attention and increasingly showed promising results. However these optimisation based methods still are faced with a number of issues. First, for tractability these robust techniques optimise robust objective functions over a collection of randomly sampled hypotheses using combinatorial methods. The trouble is that the adequacy of the hypothesis set could not be asserted prior to the optimisation, so the overall estimation could be misleading. In addition, the process of randomly sampling the hypothesis set is very time-consuming, especially for high-order models and complex data, thus generally decreasing the fitting efficiency. Moreover, to ease the optimisation, outliers are often assumed to distribute uniformly in the data space, and measurement noise is assumed to approximately follow a Gaussian distribution. Unfortunately, such assumptions are not always valid in practice.

The research conducted in this thesis follows the global optimisation approach, and makes three distinct contributions to the robust estimation field. First, we propose a novel fitting approach that simultaneously samples hypotheses and optimises the robust objective functions, such that the under- or over- hypothesis sampling issue can be avoided. In effect, our fitting approach can effectively minimise the wastage of the hypothesis sampling and objective optimisation. The second contribution is an unconventional sampling method based on Random Cluster Model (RCM) for rapidly generating accurate hypotheses. The RCM sampling method is effectively integrated into a continuous sampling-and-fitting framework to provide the superior fitting efficiency. Finally, the thesis offers a new robust estimation framework which seamlessly considers high-level geometric priors during the parameter estimation to enhance the robustness against non-uniform outliers and non-Gaussian noise. We validate the performance of the robust methods presented in this thesis on various computer vision applications ranging from estimating motions, planar homographies in image sequences to detecting geometric objects in images and 3D point clouds.
Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University’s digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed:

Date:
Publications

In carrying out my postgraduate research, a number of papers were published, accepted or are currently under revision. The content of this thesis is partially based on those papers, as listed below:


Acknowledgements

I would like to gratefully acknowledge all of the people whose support and encouragement have helped me finish my PhD journey.

First and foremost, I would like to express the deepest appreciation to my supervisor, Prof. David Suter, who has brought me a passion for research, taught me the critical thinking and scientific writing skills. He has been very supportive since the days I contacted him for a postgraduate study. I am especially grateful to him for his incredible patience in discussing my research questions, reviewing my technical papers and particularly this thesis. I believe that without his supervision and constant help this dissertation would not have been possible.

I owe sincere thankfulness to my advisor, Dr. Tat-Jun Chin, for his invaluable help and guidance through the rough road to finish this thesis. He has given me constructive discussions about my research problems, taught me to question and debate the research techniques and results. He has enlightened me the fundamentals of writing and presentation. Without him, this thesis would never have been carried out.

I also very much appreciate Prof. Wojciech Chojnacki, A. Prof. Frank Neumann, Prof. Konrad Schindler and Dr. Jin Yu for the fruitful discussions and collaborative work. My gratitude is extended to Prof. Ba-Ngu Vo for sharing his priceless research and life experiences during my stay at the University of Western Australia. Many thanks also go to my mentor, Dr. Ben Appleton, who has shown me the fundamental tradeoff between industry and academia, and also kindly supervised me during my internship at Google.

I am grateful to the University of Adelaide for providing me the scholarship to pursue my PhD, to the school of computer science and the Australia Centre of Visual Technologies (ACVT) for a great research environment. In addition, I would like to express my sincere gratitude to all members in the school of computer science, especially my colleagues Dr. Hoi Sim Wong, Dr. Sakrapee Paisitkriangkrai, Guosheng Lin, Quoc-Huy Tran, Alvaro Parra Bustos, Julio Hernandez Zaragoza, Alex Chicowski, and Chao Zhang, whom I have had chances to talk and discuss all matters of research and life.

Last but not at least, this thesis would not have been existed without the unconditional love and endless support of my family. I dedicate this dissertation to my parents, whose love and hard-working have grown me up; to my brothers and sisters who have been always by my side.

Trung T. Pham
May, 2014
“The true sign of intelligence is not knowledge but imagination.”

Albert Einstein
Chapter 1

Introduction

Computer vision ultimately aims to mimic human visual perception. In particular, computer vision involves proposing systems that can semantically understand images perceived from cameras. While humans can visually understand the 3D world instantly and effortlessly, existing computer vision systems are still far from satisfactory. Nevertheless, there has been significant progress in computer vision over the past decades, and many interesting computer vision applications have successfully made real impact on our daily lives. Google Street View (Anguelov et al., 2010), for instance, is a remarkable computer vision product, which provides a real sense of navigation and exploration of the world for millions of users daily. Other notable examples include computer-generated imagery for film and video industries and image enhancements for digital cameras.

Towards understanding images, computer vision systems require algorithms/techniques that can efficiently and effectively extract useful information from vision data. Vision data could take many forms such as a single image, an image sequence or an image set; and what is extracted depends on the specific tasks. Typical examples include extracting locations, shapes and the quantity of geometric objects in an image (Duda and Hart, 1972; Lafarge et al., 2010); motions in a video sequence (Torr, 1998; Vidal et al., 2006; Schindler and Suter, 2006; Ozden et al., 2010); or projective transformations between images of the same scene (Vincent and Laganère, 2001; Zuliani et al., 2005). Figure 1.1 shows several examples. What makes computer vision challenging is that the relevant information to be extracted is usually encoded in a part of the given data, leaving the rest irrelevant. Unlike human visual perception, because of which we can easily recognise the cars in the street in spite of the presences of people, trees or buildings etc.; such an ability is not yet achieved in generic vision algorithms.

In many cases, the information of interest can be described by some form of parametric model. For example, geometric objects in an image can be described by line segments,
Chapter 1. *Introduction*

(a) Two-view motion segmentation.

(b) Planar surface detection.

(c) Vanishing point detection.

(d) Circular object detection.

**Figure 1.1:** Examples of typical computer vision applications which rely on robust estimation. (a) Two-view motion segmentation: given two views of a scene containing moving objects, the task is to estimate the motions between views as well as the number of motions (see Section 5.2, Chapter 5 for more details of this application). (b) Planar surface detection: given two views of a static scene containing some flat surfaces, the task is to detect the number of planes and the projective transformations between two views (see Section 5.3, Chapter 5 for more details). (c) Vanishing point detection: estimating vanishing directions in a single image (see Section 6.3.2, Chapter 6). (d) Circular object detection: detecting circular objects in an image via multiple-disk fitting (see Section 6.3.3, Chapter 6).
circles or rectangles; motions of cameras can be represented by transformation parameters (rotations and translations); trajectories of objects in a video can be modelled by linear subspaces (Rao et al., 2010). In such cases the information of interest can be extracted by estimating the underlying model parameters from the observed data using model fitting/estimation techniques.

The research conducted in thesis is devoted to innovating (in accuracy and efficiency) robust parameter estimation techniques, which contribute toward unlocking many practical computer vision applications.

1.1 Robust Estimation in Vision & Challenges

Parameter estimation is an interdisciplinary field. In the following paragraphs, we briefly revisit the developments of parameter estimation and expose the existing challenges encountered in computer vision.

A common early developed method for estimating the parameters $\theta$ of a model $\mathcal{M}$ from a set of data (observations) $\mathcal{X} = \{x_n\}_{n=1}^N$ was the Least Squares (LS) (Rousseeuw and Leroy, 1987). To find the optimal $\theta$ that best fits the data $\mathcal{X}$, the LS method minimises the total squared residuals of every observation $x_n$ from the model $\theta$ (for simplicity, $\theta$ also means an instance of model $\mathcal{M}$ with parameters $\theta$). That is

$$\theta^* = \arg \min_{\theta} \sum_{x_n \in \mathcal{X}} r(x_n, \theta)^2,$$

where $r(x_n, \theta)$ computes the residual of $x_n$ from $\theta$. When the model is linear, and the measurement noise in the data $\mathcal{X}$ is assumed to be normally distributed, the LS method provides an optimal estimate of $\theta$ with a closed-form solution.

Unfortunately, vision data rarely satisfies the assumption required by the LS method. Indeed, in most cases only a part of the data (inliers) obey the assumed model, leaving the rest as outliers. Consequently, the measurement noise is no longer assumed to be normally distributed, resulting in that the optimal estimate cannot be achieved by using the LS method. This motivates researchers, both in computer vision and statistics, to propose a new class of robust estimation techniques to deal with outliers. Under this context, robustness is the ability of an estimator to correctly uncover the model parameters regardless of the existence of outliers.

RANSAC (RANdom SAmple Consensus) (Fischler and Bolles, 1981) is the most popular robust technique proposed in computer vision. Instead of minimising the total squares
Chapter 1. *Introduction*

Inliers
Uniform outliers
Pseudo outliers
Cluttered outliers

Figure 1.2: An example of multiple circle fitting data, where the data contains noisy inliers from multiple circles, uniformly distributed outliers and cluttered outliers.

of residuals, RANSAC maximises the following quantity

$$\theta^* = \arg \max_\theta \sum_{x_n \in \mathcal{X}} \delta(r(x_n, \theta) \leq \sigma),$$

where $\delta(.)$ is an indicator function (i.e., $\delta(1) = 1, \delta(0) = 0$), $\sigma$ is a noise threshold. Intuitively, RANSAC seeks a model $\theta$ that has the maximum number of supporting inliers — observations that belong to a model within a certain threshold $\sigma$. Consequently, outliers can be easily discarded by imposing the threshold $\sigma$.

Least Median of Squares (LMS) (Rousseeuw, 1984) is another robust method, which was developed in the statistics community. The robust objective function of LMS is

$$\theta^* = \arg \min_\theta \text{median}_{x_n \in \mathcal{X}} r(x_n, \theta)^2.$$  

The median function is used to alleviate the effects of outliers. Ideally, if the portion of outliers in the data does not exceed 50 percent, the solution of (1.3) is a robust estimate of $\theta$.

However, such robust objective functions (e.g., the indicator function in equation (1.2) or median function in equation (1.3)) are generally non-differentiable, thus analytical solutions are not available. For tractability, robust techniques often resort to hypothesis generation. Particularly, a large collection of model hypotheses $\mathcal{H} = \{\theta_m\}_{m=1}^M$ is generated, where each hypothesis is constructed from a randomly sampled data subset (see Section 2.3 in Chapter 2 for details), then a hypothesis $\theta \in \mathcal{H}$ that optimises the fitting criterion (e.g., equation (1.2)) is selected as the best estimate.

Besides being contaminated by outliers, vision data usually contains multiple populations (structures). For example, a scene of a building possibly contains multiple flat
surfaces (see Figure 1.1(b)); a video contains a variable number of moving cars (see Figure 1.1(a)). In such cases, inliers of a specific structure become pseudo outliers relative to other structures while the data points that can not be assigned to any structures are called gross outliers (see Figure 1.2). Consequently, the outlier rate exponentially increases with the number of structures, making the parameter estimation task much more difficult. Furthermore, the number of structures is unknown, and has to be estimated from the data.

To deal with multi-structure data, a commonly used strategy (Vincent and Laganière, 2001; Kanazawa and Kawakami, 2004) is to sequentially apply a single-structure robust technique (e.g., RANSAC) to the data — fit one structure, remove the corresponding inliers and repeatedly fit another structure. Unfortunately, such a strategy is suboptimal (Zuliani et al., 2005) since the inaccuracies in estimating the first structures will heavily affect the estimation of the remaining structures. Also, devising a stopping criterion (for determining the number of structures) is non-trivial.

Another direction (Toldo and Fusiello, 2008; Chin et al., 2010a) is to segment the data into separate populations before estimating the model parameters for each structure independently using standard methods (e.g., LS, LMS). The data clustering is based on pairwise data similarities (i.e., their potential to have been drawn from the same structure), which are “learnt” by analysing the data preferences towards a randomly sampled hypothesis set $\mathcal{H}$. Instead of clustering in the data space, other methods such as the Hough Transform (HT) (Duda and Hart, 1972), Randomised Hough Transform (RHT) (Xu et al., 1990), or AKSWH (Wang et al., 2012) attempted to analyse the distribution of the hypotheses in $\mathcal{H}$ directly to infer the model parameters — peaks in the parameter space correspond to the structures in the data. The success of all these methods relies on the quality of the sampled hypothesis set $\mathcal{H}$. Ideally, $\mathcal{H}$ should contain many “good” hypotheses on each valid structure and a relatively small portion of “bad” hypotheses so that the hypotheses can form “clusters” (for peak extraction), or the data similarities (for data clustering) can be estimated correctly. However, such a condition on $\mathcal{H}$ requires extremely large computational cost (see Section 2.3 for details), especially for high-dimensional models and complex data (e.g., noisy data that contains relatively large number of structures and outliers).

In addition to the above-mentioned methods, many multi-structure fitting methods (Torr, 1998; Li, 2007; Yu et al., 2011; Isack and Boykov, 2012) use a global optimisation approach to fit all the structures simultaneously. Such methods formulate the multi-structure fitting using some optimisation criterion, and infer the solution that best meets the criterion. The optimisation criterion usually includes a model selection criterion (e.g., the Akaike information criterion (AIC) (Akaike, 1974)) to estimate the number
of structures in the data. For tractability, these methods are implemented in two disjoint stages: First generate a discrete set of hypotheses \( \mathcal{H} \) which sample the parameter space, then select a subset that optimises the fitting criterion using optimisation techniques. Depending on the specific definitions of the fitting criteria, various optimisation techniques have been applied to select the best models, namely tabu search (Schindler and Suter, 2006), branch-and-bound (Thakoor and Gao, 2008), linear programming (Li, 2007), message passing (Lazic et al., 2009), graph cuts (Isack and Boykov, 2012), and quadratic programming (Yu et al., 2011). Since all the structures are estimated jointly, the methods have been proved to be more accurate than the sequential fit-and-remove approaches. Nevertheless, the outcome of the model selection stage will only be sensible if the hypothesis set \( \mathcal{H} \) is sufficient (e.g., at least one good hypothesis sampled on each structure). However the adequacy of \( \mathcal{H} \) might be not known prior to the optimisation, because enormous sampling effort is required before hitting good hypotheses. Thus the two-stage approaches are suboptimal. This dissertation follows the same line, and proposes more sophisticated hypothesis sampling and optimisation strategies to enhance fitting accuracy and efficiency.

Apparently, regardless of the robust approaches, random hypothesis generation (for preparing the hypothesis set \( \mathcal{H} \)) is crucial to robust geometric model fitting. The overall fitting efficiency and accuracy heavily depend on the quality of the hypothesis set \( \mathcal{H} \). Preferably, \( \mathcal{H} \) should be small, but includes many good hypotheses. However, we will show in Chapter 2 that the hypothesis sampling process consumes considerably computational resource, a cost which exponentially increases with the model complexity and outlier rate. Therefore, accelerating the sampling process is necessary.

Last but not least although the existing robust methods have successfully demonstrated their superior robustness to outliers (empirically up to 80% as reported in (Wang and Suter, 2004)), these methods implicitly assume that outliers distribute uniformly in the data space, and inlier noise follows a Gaussian distribution. Unfortunately such assumptions are not valid in many practical applications (see Figure 1.2). Devising robust techniques to deal with such cases is also a challenging task.

### 1.2 Research Contributions

The research conducted in this thesis follows the global optimisation approach and focuses on three main aspects: random hypothesis generation, model selection (optimisation strategies) and robustness against non-uniform outliers and non-Gaussian inlier noise. The major contributions in this thesis are:
1. A novel robust estimation approach that conducts hypothesis sampling and model selection simultaneously. Our approach effectively reduces the computational cost of the hypothesis sampling and model selection optimisation. We develop two algorithms to realise this idea. The first algorithm is based on a Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm presented in Chapter 3; and the second algorithm is constructed using a simulated annealing method based on graph-cuts presented in Chapter 4.

2. An unconventional sampling technique which generates hypotheses using clusters of data. Our sampling technique rapidly retrieves many good model hypotheses, and thus significantly improves the overall estimation accuracy and efficiency (see Chapter 4).

3. A general fitting framework that seamlessly incorporates global geometric priors into parameter estimation. The framework is highly robust, especially against non-uniform outliers and non-Gaussian inlier noise (see Chapter 6).

1.3 Thesis Outline

In Chapter 2, we review related robust estimation methods that follow the global optimisation approach. We also review techniques for hypothesis generation, which underpins almost all robust techniques in computer vision.

In Chapter 3, we introduce a new optimisation strategy for multi-structure model fitting. As opposed to the disjoint hypothesis-then-select approaches, our hypothesis-and-select method conducts hypothesis sampling and fitting alternately and continuously. We analyse the distinct properties of our fitting algorithm using both synthetic and real datasets. The proposed algorithm is shown to be more efficient than the two-stage approaches.

The fourth chapter addresses the issues of the hypothesis generation. We propose a novel hypothesis sampling technique based on arbitrary-sized data subsets, which aims to produce more accurate hypotheses using less computational cost. The superior efficiency of our sampling method over the state-of-the-art techniques is demonstrated through an extensive experiment. Furthermore, the chapter proposes a variant of the hypothesis-and-select fitting approach which effectively encapsulates the new sampling method to maximise the fitting efficiency and accuracy.

In Chapter 5 we apply our robust estimation methods proposed in Chapters 3 and 4 to three practical applications: motion segmentation, planner surface detection, and
augmented reality in video. Through such applications, we carry out a comprehensive benchmarking to evaluate the performances of our proposed methods against the state-of-the-art robust estimation methods.

The sixth chapter focuses on exploiting global geometric priors in robust estimation. We argue that the robustness will be significantly improved if robust methods take advantage of the geometric priors. Accordingly, we introduce a fitting objective function that effectively encodes geometric consistencies between structures. The function is sufficiently simple to be optimised efficiently over a discrete set of hypotheses by using a graph-cut method. We conduct experiments on various vision applications to validate the superior improvement in robustness of our proposed estimator over previous methods.

In the last chapter, we summarise the important contributions presented in this thesis, and offer a number of possible future directions.
Chapter 2

Optimisation Algorithms for Robust Estimation

Parameter estimation and optimisation are closely related areas. While optimisation seeks the global optimum of a given function, in parameter estimation we would like to find a set of model parameters that “best” explain a given set of observations according to some fitting criterion. In many cases, the fitting criterion can be defined as an objective function, and the parameter estimation can be solved by using optimisation techniques. This chapter reviews important parameter estimation objective functions and the corresponding optimisation algorithms proposed in computer vision.

Firstly, let us state the multi-structure parameter estimation problem formally. The input is a set of $N$ observations $X = \{x_n\}_{n=1}^{N}$ (e.g., a set of features matched across two views of a scene), which are drawn from a number of populations/structures (e.g., moving objects in the scene). Each structure can be represented by a geometric model (e.g., fundamental matrix). In this thesis, the structures are assumed to be instances of the same model type $M$. The structures can be of different sizes in terms of the number of supporting inliers. The data $X$ is usually contaminated by measurement noise and gross outliers, where the noise level for each structure, and the amount of outliers are not known as a priori. The goal of robust estimation is to recover the underlying structures $\Theta = \{\theta_k\}_{k=1}^{K}$ embedded in $X$, where each $\theta_k \in \mathbb{R}^d$ is the parameters of one model instance. The number of structures $K$ is unknown and must be estimated from the data.

In the subsequent sections, we discuss basic requirements for defining an optimal criterion for the unknown parameters $\Theta$, and optimisation strategies for finding $\Theta$ that best meets the fitting criterion.
2.1 Robust Fitting Criterion

2.1.1 Goodness of Fit

The foremost requirement in parameter estimation is that the estimated parameters $\Theta$ must well explain the input data $X$. This requirement is usually represented via a goodness of fit term, which typically measures the residuals between the model parameters $\Theta$ and the observations $X$. Ideally, the models $\Theta$ should pass through (or near) every data point $x_n \in X$, e.g., $g(x_n, \Theta) \approx 0, \forall x_n \in X$, where $g$ is some distance function. Consequently, to evaluate the models $\Theta$, we can rely on the goodness of fit value $D(X, \Theta) = \sum_{n=1}^{N} g(x_n, \Theta)$, i.e., smaller is better.

Notice that there are $K$ model instances in $\Theta$, hence implicitly we need to assign each $x_n$ to one of the instances before computing $g(x_n, \Theta)$. A trivial assignment is to designate $x_i$ to the closest structure. That is

$$g(x_n, \Theta) = \min_{\theta_k \in \Theta} r(x_n, \theta_k), \quad (2.1)$$

where $r(x_n, \theta_k)$ is the (absolute) residual of $x_n$ to the $k$-th structure $\theta_k$ in $\Theta$.

2.1.2 Model Complexity

In multi-structure robust estimation, only considering the goodness of fit will not result in a good estimate. In fact, since the number of valid structures $K$ in $\Theta$ is variable, the goodness of fit $D(X, \Theta)$ will equate to zero if $K$ is arbitrarily large, e.g., each point is assigned to a unique structure that passes through the point. This issue, often called “over-fitting”, happens when the estimated model parameters $\Theta$ is excessively complex — typically $\Theta$ contains too many structures (and hence too many parameters) relative to the number of observations. Figure 2.1 demonstrates the case where we either fit 1 line or 4 lines to a 2D point cloud.

Consequentially, another crucial requirement for $\Theta$ is the model complexity. In this context, the model complexity is proportional to the number of structures in $\Theta$. Intuitively, among many possible solutions with approximately the same goodness of fit, the solution with the smallest number of structures is preferred. Therefore, a good estimate also minimises the complexity of $\Theta$, denoted as $C(\Theta)$. For example, $C(\Theta) = K \times d$ counts the number of parameters used to define $\Theta$ ($K$ is the number of structures in $\Theta$, $d$ is the dimension of each structure $\theta$).
Chapter 2. *Optimisation Algorithms for Robust Estimation*

2.1.3 Other Requirements

Besides the goodness of fit and model complexity, there could be other useful requirements for $\Theta$ so that more plausible estimations can be achieved. For instance, the point segmentation induced by the estimated $\Theta$ should be spatially smooth (Isack and Boykov, 2012) (i.e., nearby points should agree with the same structure). Such a requirement can help improve the estimation accuracy in many applications where points belonging to a genuine structure tend to be spatially close. Another common example is that the structures in $\Theta$ should be diverse, i.e., each $\theta \in \Theta$ represents a distinct true structure. This requirement is useful, for example, when structures in the data are largely uneven (in term of population sizes). In such cases, if the measurement noise is high, the dominant structure tends to be explained by several approximately identical model instances. This issue can be prevented by the structure diversity, e.g., disallowing similar model instances appearing in the solution $\Theta$.

2.1.4 Objective Functions

Once all the necessary requirements for multi-structure model estimation have been identified, one would like to define a single optimal criterion (objective function) that fully captures all the required aspects. The simplest objective function is a linear combination of all the requirements. For example, one can define an objective function as

$$E(\Theta) = D(X, \Theta) + wC(\Theta),$$

(2.2)

where $w$ is a *regularisation parameter*, which balances the goodness of fit and model complexity. More examples of popularly used fitting criteria will be shown in Section 2.2.
In general, objective functions with more constraints (requirements) would lead to more accurate estimates, but are hard to optimise. On the other hand, simple functions (e.g., containing only the goodness of fit) can be optimised easily, but the estimated models are less accurate. Therefore, depending on the specific applications, practitioners should choose a suitable fitting function and the corresponding optimisation technique that maximises the benefit (e.g., estimation accuracy and efficiency).

In most cases, the objective functions used in robust estimation such as (2.2) are non-differentiable and contain many local optima, thus closed-form analytical solutions do not exist. For tractability, the existing robust estimators instead optimise the objective functions over a discrete hypothesis set \( \mathcal{H} \), under the assumption that \( \mathcal{H} \) covers all true structures embedded in the data. The hypothesis set \( \mathcal{H} \) can be constructed via Monte-Carlo sampling (see Section 2.3 for a review of sampling methods). As a result, the robust estimation becomes selecting an optimal \( \Theta \subset \mathcal{H} \) that optimises a pre-defined objective function. Afterwards, the selected \( \Theta \) can be used to identify inliers of each structure, and each \( \theta_k \) in \( \Theta \) can be improved by fitting a model to the associated inliers using the LS method. Algorithm 2.1 summarises the idea, and Figure 2.2 illustrates a line fitting example.

Algorithm 2.1: Global optimisation based multi-structure robust estimation.

1: Construct an objective function that encodes the fitting requirements.
2: Hypothesis sampling: Randomly sample a large collection of hypotheses \( \mathcal{H} \).
3: Model selection: Select a subset \( \Theta \subset \mathcal{H} \) that optimises the objective function.
4: Identify inliers of each structure using the selected \( \Theta \).
5: Improve \( \Theta \) by fitting models to the inliers using the LS method.
6: Return the estimate \( \Theta \).

Nevertheless, the selection of an optimal subset \( \Theta \subset \mathcal{H} \) is, in general, a NP-hard problem. Section 2.2 reviews existing approximation techniques for such a difficult model selection problem.

### 2.2 Optimisation Algorithms for Model Selection

The selection of model hypotheses \( \Theta \) can be done in various ways. For example, one can think of the model selection as an assignment problem, where points are assigned to models according to some criterion. More specifically, given a set of points \( \mathcal{X} = \{x_n\}_{n=1}^N \), and a set of model hypotheses \( \mathcal{H} = \{\theta_m\}_{m=1}^M \), the assignment can be modelled by a vector of auxiliary binary variables \( a = [a_{11}, a_{21}, \ldots, a_{nm}, \ldots, a_{NM}] \) with \( a_{nm} = 1 \) if point \( x_n \) is assigned to hypothesis \( \theta_m \), \( a_{nm} = 0 \) otherwise. Given a particular \( a \), the associated models can be inferred, i.e., \( \Theta = \{\theta_m \subset \mathcal{H} | \sum_{n=1}^N a_{nm} > 0 \} \). Therefore, the selection
Chapter 2. Optimisation Algorithms for Robust Estimation

(a) 2D point cloud.
(b) Sampled hypotheses.
(c) Selected hypotheses.

Figure 2.2: Robust parameter estimation using a global optimisation approach. Given the input data shown in (a), the goal is to fit multiple lines onto the data. The existing optimisation based fitting methods are implemented in two stages: first randomly sample a large collection of line candidates (b) then resort a combinatorial optimisation method to select a subset of lines that best explain the data (c). The final estimated lines can be obtained by fitting line models to the inliers using the LS method. Remark: in (c) the colors indicate the point memberships. Black points are detected outliers.

of $\Theta$ can be achieved by performing inference over the assignment variables $a$. Another possibility is using binary indicating variables $b = [b_1, b_2, \ldots, b_m, \ldots, b_M]$ with $b_m = 1$ if hypothesis $\theta_m$ is selected and $b_m = 0$ otherwise.

In this section, we review the representative objective function formulations and the associated inference algorithms that have been used in various multi-model estimation settings. We also discuss their advantages and limitations, which motivate the research conducted in this thesis.

2.2.1 Tabu Search

Tabu search is a meta-heuristic search algorithm used for mathematical optimisation. Tabu search employs a neighbourhood search procedure to locally move from a solution to a “better” solution. The moves are repeated until a “deemed” optimal solution is found or a stopping criterion is reached. As opposed to standard local search algorithms which usually get stuck in suboptimal regions, Tabu search avoids local optima by forbidding moves to points that have been already visited in the recent past. Intuitively, Tabu search encourages moves to unexplored regions so that better optima may be found. Many other heuristics for escaping from local optima are presented in (Reeves, 1993).

In (Stricker and Leonardis, 1995), Tabu search was applied for estimating geometric models such as: lines, ellipses, planes, cylinders, spheres from 2D and 3D point clouds.
Later, Schindler and Suter (Schindler and Suter, 2006) used Tabu search to solve the two-view motion segmentation problem, in which \( \mathcal{X} \) is a set of matches across two views, \( \mathcal{H} \) is a set of candidate motion models. Here we adopt the formulation presented in (Stricker and Leonardis, 1995) to explain the idea. Given \( \mathcal{X} = \{x_n\}^N_{n=1} \) and \( \mathcal{H} = \{\theta_m\}^M_{m=1} \), the best subset \( \Theta \) can be obtained by maximising the following objective function

\[
E(b) = b^T Q b,
\]

where \( b \) are binary indicating variables, \( Q = (q_{ij}) \) is a symmetric \( M \times M \) matrix. (Given an optimised \( b \), the corresponding subset is \( \Theta = \{\theta_m \subset \mathcal{H} | b_m = 1\} \).) The elements of \( Q \) are defined as:

\[
q_{ii} = w_1|S_i| - w_2D_{ii} - w_3d_i,
\]

\[
q_{ij} = -w_1|S_i \cap S_j| + \frac{w_2D_{ij}}{2} \quad \text{for } i \neq j,
\]

where \( w_1, w_2 \) and \( w_3 \) are weighting parameters, \( S_i \) is a set of supporting inliers of hypothesis \( \theta_i \) (assuming that the inlier noise scales are known), \( d_i \) is the dimension of \( \theta_i \), \( D_{ii} \) and \( D_{ij} \) are the fitting error terms, defined as:

\[
D_{ii} = \sum_{x \in S_i} r(x, \theta_i)^2,
\]

\[
D_{ij} = \max \left( \sum_{x \in S_i \cap S_j} r(x, \theta_i)^2, \sum_{x \in S_i \cap S_j} r(x, \theta_j)^2 \right).
\]

Intuitively, the diagonal terms (i.e., \( q_{ii} \)) prefer hypotheses that have large supports, low fitting errors and small number of model parameters while the off-diagonal terms (i.e., \( q_{ij} \)) encourage hypotheses that overlap as least as possible.

Maximising the objective \( (2.3) \) over \( b \) is an NP-hard problem. In (Stricker and Leonardis, 1995) and (Schindler and Suter, 2006), approximate solutions of \( (2.3) \) are sought using Tabu search. Specifically, starting from an arbitrary subset \( \Theta \) (or \( b \)), a list of moves to neighbours of \( \Theta \) is generated by either adding or removing one hypothesis to (from) \( \Theta \) (simply switching on or off elements of \( b \)). Then, based on a number of heuristics, the highest benefit move is chosen as the current best solution. For example, the moves that add (or remove) the recently used hypotheses are illegal and prohibited (so that the last local maxima are not visited repeatedly). Among the legal moves, the one with the highest value of the objective function is selected. If all the moves are illegal, the one that adds (or removes) the least frequently used hypothesis is chosen. (The frequency of a certain hypothesis measures how often that hypothesis has been involved in a move.)
Chapter 2. Optimisation Algorithms for Robust Estimation

The search is repeated until a stopping criterion is satisfied (e.g., no better solution is found after a given number of iterations).

Although Tabu search is simple, it is a meta-heuristic method. The performance (both computational complexity and solution quality) highly depends on the implementation. Also no guarantee on the solution quality can be given, as opposed to other optimisation methods such as Branch-and-Bound and graph-cuts, which will be reviewed in the next sections.

2.2.2 Branch and Bound

Branch and Bound (B&B) is a general algorithm for finding global optimal solutions of non-convex problems. There are two main steps: branching and bounding. In the branching step, the original problem is divided into sub-problems, resulting in a tree structure whose nodes are sub-problems. The bounding step computes the upper and lower bounds for each node in the tree. The intuition behind the B&B optimisation is that if the lower bound for the currently visited node (in the tree) is greater than the upper bounds for some other nodes, we can safely stop exploring (branching) further from that node without affecting the optimality. The branching and bounding continue until all the sub-problems are explored. More technical details about the B&B algorithm and its applications can be found in (Brusco and Stahl, 2005).

In (Thakoor and Gao, 2008), the authors used the B&B optimisation method for solving the two-view multiple structure and motion segmentation problem. Their fitting objective function is defined as

\[ E(\Theta) = N \log\left( \frac{1}{N} \sum_{n=1}^{N} \min_{\theta \in \Theta} (r(x_n, \theta)) \right) + w|\Theta|, \]  

(2.8)

where \( w \) is a scaling parameter. Observe that the first and second terms in Equation (2.8) measures the goodness of fit and the model complexity, respectively.

To find the minimum of (2.8), the authors resort to the B&B optimisation method. Firstly, a rooted binomial tree over the collection of model hypotheses \( \mathcal{H} \) is constructed, where each node in the tree represents a possible solution \( \Theta \); the tree must represent all possible solutions, each of which is listed only once. We refer readers to the original work (Thakoor and Gao, 2008) for the detailed construction. Figure 2.3 shows an example of a rooted binomial tree with 5 hypotheses; at the node A, the current solution is \( \Theta = \{\theta_1, \theta_2, \theta_3\} \). Note that \( \theta_1 \) is a dummy hypothesis which is always included in the solution to explain outliers. Starting from the root (e.g., \( \Theta = \{\theta_1\} \)), their algorithm
Figure 2.3: A binomial solution tree constructed with a hypothesis set \( \mathcal{H} = \{ \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 \} \). \( \{ z_i \}_{i=1}^5 \) represent the height of the tree. At node A, the corresponding solution is \( \Theta = \{ \theta_1, \theta_2, \theta_3 \} \); \( \mathcal{H}_f = \{ \theta_4, \theta_5 \} \) are hypotheses which could be included to the current solution \( \Theta \) in the subsequent branching steps.

performs a series of forward moves, right moves and retraction moves to explore the optimal solution. The forward move simply adds a new hypothesis to the current solution \( \Theta \), and the move is accepted if the lower bound (at the current node) is smaller than current objective value. The right move attempts to explore the right next branch if the forward move is rejected. If both the forward and right moves are impossible, the algorithm goes back to a lower level (e.g., \( z_3 \to z_2 \), see Figure 2.3) by a retraction move and then explores other branches. The algorithm continues until the retraction leads to the root node.

The crucial aspect in their algorithm is to compute the lower bound at each node. Suppose that we are at a node A in the solution tree with an associated solution \( \Theta \); \( \mathcal{H}_f \) is a set of nodes (hypotheses) in the sub-tree counted from the node A. These hypotheses could be added to the current solution \( \Theta \) in the future branching steps. The lower bound \( E_{lb} \) at the node A is computed incrementally as:

\[
E_{lb} = N \log \left( \frac{1}{N} \sum_{n=1}^{N} \min(D(x_n, \Theta), D(x_n, \mathcal{H}_f)) \right) + w(|\Theta| + 1), \tag{2.9}
\]

where

\[
D(x_n, \Theta) = \min_{\theta \in \Theta} r(x_n, \theta), \quad \tag{2.10}
\]

\[
D(x_n, \mathcal{H}_f) = \min_{\theta \in \mathcal{H}_f} r(x_n, \theta). \quad \tag{2.11}
\]

Although the B&B optimisation method guarantees to provide a globally optimal solution, the expensive computational cost is a crucial drawback of this method. The worst case complexity of the B&B method exponentially increases with the size of the hypothesis set. Indeed, in (Thakoor and Gao, 2008), only few dozens of model hypotheses can
be handled. Therefore the method is intractable in practice, where we commonly deal with thousands of hypotheses.

2.2.3 Linear Programming Relaxation

In (Li, 2007), the two-view motion segmentation problem was viewed as an assignment task. In particular, let \( d_{nm} = r(x_n, \theta_m)^2 \) be the squared distance from a point \( x_n \) to a motion hypothesis \( \theta_m \), \( a_{nm} \) are membership variables (\( a_{nm} = 1 \) if \( x_n \) is assigned to model \( \theta_m \)), \( b_m \) are indicating variables (\( b_m = 1 \) if model hypothesis \( \theta \) is selected as a member of \( \Theta \)), the objective function has the following form

\[
\min_{a, b} \sum_{n=1}^{N} \sum_{m=1}^{M} d_{nm} a_{nm} + w \sum_{m=1}^{M} b_m,
\]

s.t., \( \sum_{m=1}^{M} a_{nm} = 1, \forall n, \)

\( a_{nm} \leq b_m, \forall n, m, \)

\( a_{nm} \in \{0, 1\}, b_m \in \{0, 1\}, \forall n, m, \)

where \( w \) is a weighting parameter. In (2.12), the first term accounts for the goodness of fit while the second term computes the model complexity. The constraint (2.13) implies that each point is assigned to exactly one model, while the inequality (2.14) ensures that the model \( \theta_m \) must be counted (i.e., \( b_m = 1 \)) if there exists at least one point assigned to \( \theta_m \). The expression (2.12) is also known as the facility location (FL) problem\(^1\) — a well-known problem in operations research

The FL problem is generally NP-hard (Mahdian et al., 2002), therefore only approximate solutions can be found. In (Li, 2007), the author applied a Linear Programming Relaxation (LPR) strategy. The idea is that the integer constraint (2.15) is relaxed into linear inequalities of real variables. That is

\[
0 \leq a_{nm} \leq 1, \ 0 \leq b_m \leq 1, \forall n \in [1, N], \forall m \in [1, M].
\]

Then, the optimisation (2.12) becomes a Linear Program (LP) problem (i.e., a linear objective function with linear equality and inequality constraints), and can be solved by

\(^1\)The facility location problem consists of a set of potential facilities, and a set of customers. We would like to open a number of facilities to serve all the customers. However we need to pay a specific cost for opening each facility (assume that the opening costs for all the facilities are equal). The goal is to open only a subset of facilities, and assign customers to the opened facilities, such that the total opening costs, plus the sum of distances from each customer to the assigned facility is minimised.
using any standard LP technique. Once the variables \(a\) and \(b\) have been optimised, a rounding step (e.g., \(b_m = 1\) if \(b_m > 0.5\) and \(b_m = 0\) otherwise) is applied for de-relaxation.

Although the linear programming optimisation guarantees a globally optimal solution, it is generally not an optimal solution to the original FL problem. Moreover, similar to the Branch and Bound optimisation, linear programming in the worst case requires the computational effort that exponentially increases with the number of hypotheses.

### 2.2.4 Message Passing

In (Lazic et al., 2009), a subspace segmentation problem is considered. The task is to estimate multiple linear subspaces embedded in a set of data. Particularly, let \(\mathcal{X} = \{x_n\}_{n=1}^N\) be a set of \(N\) data points, assume that a large collection of \(M\) candidate subspaces \(\mathcal{H} = \{\theta_m\}_{m=1}^M\) have been prepared. The task becomes selecting a subset \(\Theta \subset \mathcal{H}\) on which the data \(\mathcal{X}\) is most likely lying. Similar to the previously discussed motion segmentation problem (Li, 2007), here the subspace segmentation is also formulated as a FL problem, denoted as FLoSS (Facility Location for Subspace Segmentation), i.e.,

\[
\min_a \sum_{n=1}^N \sum_{m=1}^M d_{nm}a_{nm} + \sum_{\theta_m \in \Theta} c_m \tag{2.17}
\]

subject to \(\sum_{m=1}^M a_{nm} = 1, \forall n\),

\[
\Theta = \{\theta_m | \sum_{n=1}^N a_{nm} > 0\}, \tag{2.19}
\]

\(a_{nm} \in \{0, 1\}, \forall n, m, \quad (2.20)\)

where \(\{a_{nm}\}\) are binary assignment variables, \(d_{nm}\) measures the distance from \(x_n\) to subspace \(\theta_m\), \(c_m\) is a positive cost associated with subspace \(\theta_m\).

Instead of resorting to a Linear Programming Relaxation approach as in (Li, 2007), here the authors find the approximate solutions to this NP-hard model selection problem (2.17) by performing maximum-a-posteriori (MAP) inference in a probabilistic graphical model. Particularly, minimising (2.17) is equivalent to maximising the following joint probability function:

\[
p(a) \approx \prod_{n,m} \exp(h_{nm}(a_{nm})) \prod_m \exp(j_m(a_{1m}, a_{2m}, \ldots, a_{Nm})) \prod_n \exp(l_n(a_{n1}, a_{n2}, \ldots, a_{nM})) \tag{2.21}
\]
where

\[
h_{nm}(a_{nm}) = -d_{nm}a_{nm},
\]

(2.22)

\[
j_m(a_{1m}, a_{2m}, \ldots, a_{Nm}) = \begin{cases} 
-e_m & \text{if } \sum_n a_{nm} > 0 \\
0 & \text{otherwise}, 
\end{cases}
\]

(2.23)

\[
l_n(a_{n1}, a_{n2}, \ldots, a_{nM}) = \begin{cases} 
0 & \text{if } \sum_m a_{nm} = 1 \\
-\infty & \text{otherwise}. 
\end{cases}
\]

(2.24)

Observe that the \(h_{nm}, j_m\) and \(l_n\) functions encode the first and second terms in (2.17) and the constraint (2.18) respectively.

The probability distribution (2.21) is then represented using a factor graph (Kschischang et al., 1998) whose variable nodes model the variables \(\{a_{nm}\}\), and factor nodes evaluate the potential functions (i.e., \(h_{nm}, j_m\) and \(l_n\)) (see Figure 2.4). Once a probabilistic graphical model encapsulating the distribution (2.21) is constructed, the MAP inference can be performed by using a belief propagation algorithm (Kschischang et al., 2001). The belief propagation inference algorithm is known to be efficient, however only approximate solutions are available.

### 2.2.5 Graph-cuts and Expansion Move Algorithm

In stead of using boolean assignment variables as in (Li, 2007; Lazic et al., 2009), in a recent work (Delong et al., 2010, 2012b; Isack and Boykov, 2012), the model selection problem is solved by optimising multi-valued assignment variables. Specifically, given \(N\) data points and \(M\) hypotheses, define a vector of \(N\) labels \(f = [f_1, f_2, \ldots, f_n, \ldots, f_N]\),
where \( f_n \) is the label of \( x_n \) and takes value in \( \{1, 2, \ldots, m, \ldots, M\} \). For example, \( f_n = m \) means that \( x_n \) is assigned to hypothesis \( \theta_m \). Note that for simplicity we make no distinction between a label and its index (e.g., \( f_n = m \iff f_n = \theta_m \)). Given a particular \( f \), we can obtain the subset of hypotheses \( \Theta \) easily, i.e.,

\[
\Theta = \{\theta_m \subset \mathcal{H} \mid \exists n \in [1, N], f_n = m\}. \tag{2.25}
\]

Therefore, finding an optimal assignment (labeling) \( f \) corresponds to finding an optimal \( \Theta \). In (Delong et al., 2010, 2012b; Isack and Boykov, 2012), the optimal labeling \( f \) can be found by solving the following optimisation problem

\[
f^* = \arg \min_f \left[ \sum_{n=1}^{N} D(x_n, f_n) + w_1 \sum_{(i,j) \in \mathcal{N}} V(f_i, f_j) + w_2 C(f) \right]. \tag{2.26}
\]

\( D(x_n, f_n) \) computes how appropriate the label \( f_n \) is assigned to data point \( x_n \). For example, they used \( D(x_n, f_n) = r(x_n, f_n) \). \( V(\cdot, \cdot) \) is a function which enforces the smoothness constraint — spatially close points should be assigned to the same label. \( \mathcal{N} \) is some neighbourhood structure defined over \( \mathcal{X} \). \( C(f) \) simply counts the number of unique labels in \( f \). \( w_1 \) and \( w_2 \) are two scaling parameters. It is clear that the data cost term measures the goodness of fit while the label cost term measures the model complexity. Different from the previous robust criteria, the objective function (2.26) includes an additional smoothness term, which aims to improve the labelling coherence, and potentially the parameter estimation accuracy.

In general, minimising the multi-label function (2.26) is a NP-hard problem, thus approximate methods must be deployed. Prior to showing how the optimisation (2.26) was solved in (Delong et al., 2010, 2012b; Isack and Boykov, 2012), let us consider the problem without the label cost term, i.e.,

\[
E(f) = \sum_{n=1}^{N} D(x_n, f_n) + w_1 \sum_{(i,j) \in \mathcal{N}} V(f_i, f_j). \tag{2.27}
\]

An energy function such as (2.27) has been widely used in computer vision, for example image restoration (Boykov et al., 2001) and stereo vision (Szeliski and Zabih, 2000). If the smoothness function \( V(\cdot, \cdot) \) is submodular\(^2\) on the label space \( \mathcal{H} \), the energy (2.27) can be efficiently minimised with a guaranteed optimality bound using the expansion move method (Boykov et al., 2001). The idea of the expansion move algorithm is that the multi-label problem (2.27) can be converted to a series of binary-label problems,

\(^2\) \( V \) is submodular if \( V(f_i, f_j) + V(f_k, f_k) \leq V(f_i, f_k) + V(f_j, f_k) \) for any \( f_i, f_j, f_k \).
each of which can be optimised exactly using a graph-cut method. Specifically, given an arbitrary label $f$ and a label $\theta \in \mathcal{H}$, the algorithm finds an optimal labelling $\hat{f}$, which either changes $f_n$ ($n \in [1, N]$) to $\theta$ or keeps $f_n$ unchanged, by optimising a binary (i.e., changed and unchanged) energy. Intuitively, we would like to expand the label $\theta$ (if possible), such that more points are assigned to $\theta$. Readers may refer to (Boykov et al., 2001) for technical details how to construct the binary energy given $f$ and $\theta$. The algorithm iterates through all labels in $\mathcal{H}$ until convergence. The optimality property of the expansion move method is described in Theorem 2.1. Algorithm 2.2 summarises the basic steps of the method.

**Algorithm 2.2: Expansion move algorithm (Boykov et al., 2001)**

1. Start with an arbitrary labelling $f$
2. for each label $\theta \in \mathcal{H}$ do
3. Construct a binary energy that encapsulates the energy (2.27) given $\theta$ and $f$.
4. Optimise the binary energy using graph-cuts.
5. Infer the new labelling $\hat{f}$.
6. if $E(\hat{f}) < E(f)$ then
7. $f = \hat{f}$.
8. end if
9. end for

**Theorem 2.1.** Let $\hat{f}$ be a local minimum obtained by the expansion move algorithm (Boykov et al., 2001) and $f^*$ be the globally optimal solution. Then $E(\hat{f}) \leq 2cE(f^*)$, where

$$c = \max_{(i,j) \in \mathcal{V}} \left( \frac{\max_{f_i \neq f_j} V(f_i, f_j)}{\min_{f_i \neq f_j} V(f_i, f_j)} \right).$$

(2.28)

To tackle the energy with label costs (2.26), the expansion move algorithm was then extended to encode the label cost term using additional auxiliary variables. The expansion move with label costs was reported in (Delong et al., 2010). However the downside of this extension is that more computational cost is required since more variables (for encoding the label costs) are optimised. Moreover, the optimality bound is getting looser with the increase of the number of labels.

### 2.2.6 Quadratic Programming

Rather than selecting a subset $\Theta \subset \mathcal{H}$, the work (Yu et al., 2011) tries to rank the hypotheses in $\mathcal{H}$ such that the top ranked hypotheses represent distinct structures in the data. Specifically, their algorithm associates each model hypothesis $\theta_m$ with a variable weight $t_m \in [0, 1]$. Subsequently, the ranking is achieved by computing the optimal weights $t = [t_1, t_2, \ldots, t_M]$, then sorting them in non-ascending order. The optimal
weights must reflect the requirements that the top ranked hypotheses are of high quality and diversity. The optimal \( t \) is obtained by minimising the following objective function

\[
E(t) = \sum_{m=1}^{M} t_m F(I_m, \theta_m) - w_1 \sum_{m=1}^{M} t_m S(I_m) + w_2 \sum_{\theta_m, \theta_m'} t_m t_{m'} O(\theta_m, \theta_{m'})
\] (2.29)

subject to

\[
t \in [0, 1]^M, \quad \sum_{m=1}^{M} t_m > T,
\] (2.30)

where \( T \in [1, M] \) is a positive integer used to ensure that at least \( T \) model hypotheses will have non-zero weights, \( w_1, w_2 \) are weighting parameters. The following paragraphs explain the terms \( I_m, F(I_m, \theta_m), \theta_m, S(I_m) \) and \( O(\theta_m, \theta_{m'}) \) used in Equation (2.29).

\( I_m \) is a set of inliers associated with the hypothesis \( \theta_m \). \( F(I_m, \theta_m) \) measures the quality of the hypothesis \( \theta_m \). Formally,

\[
F(I_m, \theta_m) = \frac{1}{|I_m|} \sum_{i \in I_m} r(x_i, \theta_m),
\] (2.31)

which is the average residual from the data in \( I_m \) to the model hypothesis \( \theta_m \). Intuitively, a good hypothesis \( \theta_m \) should have low residual \( F(I_m, \theta_m) \). \( S(I_m) \) computes the similarity between data in \( I_m \), i.e.,

\[
S(I_m) = \frac{1}{|I_m|} \sum_{i \in I_m} \text{median}_{j \in I_m} s(x_i, x_j),
\] (2.32)

where \( s(x_i, x_j) \) evaluates the similarity between \( x_i \) and \( x_j \). The higher similarity \( S(I_m) \) suggests that inliers in \( I_m \) are more likely from the same structure, thus \( \theta_m \) is a good hypothesis. It is clear that the model fidelity and inlier similarity together quantify the hypothesis quality. \( O(\theta_m, \theta_{m'}) \) measures the degree of overlap between any pair of model hypotheses \( \theta_m \) and \( \theta_{m'} \), which implicitly encourages the model diversity. Indeed, the model overlapping term will penalise simultaneously assigning high weights to approximately similar model hypotheses, thus the top ranked hypotheses only represent distinct structures. We refer readers to (Yu et al., 2011) for detailed definitions of the inlier set \( I_m \), the similarity function \( s(x_i, x_j) \), as well as the overlapping function \( O(\theta_m, \theta_{m'}) \).

Although the objective function (2.29) is quite different from the previously discussed fitting criteria, (2.29) still encodes the goodness of fit and model complexity. Observe that the model fidelity terms, computed based on the fitting residuals, implicitly measure the goodness of fit. Also the model overlapping terms, which prohibit all good hypotheses from being ranked highly, indirectly penalise the model complexity.
It can be seen that (2.29) is quadratic function of $t$ with linear constraints (2.30). Therefore, the optimisation problem can be solved by using any Quadratic Programming (QP) solvers such as MOSEK\(^3\). Moreover, since the function $O(\theta_m, \theta_{m'})$ used in (Yu et al., 2011) is a positive-definite kernel function, the objective function $E(t)$ is convex. As a result, a globally optimal solution can be found. Nevertheless, the QP solvers (e.g., MOSEK) are generally less efficient than other optimisation methods (e.g., graph-cuts).

2.2.7 Discussion

We have described numerous global optimisation based robust estimation methods in computer vision. The main differences between these methods relate to the objective functions and the corresponding optimisation strategies. Although the objective functions are in different forms, they commonly encode the goodness of fit and the model complexity, which are the two important requirements in multi-structure robust estimation. Subsequently, depending on the characteristics of the objective functions, various optimisation techniques are used. The similarity of these approaches is that they all operate on a pre-established hypothesis set $\mathcal{H}$. In effect, the optimisations simply select a subset of hypotheses in $\mathcal{H}$ that best meet the fitting criteria.

The main advantage of the energy-minimisation based approaches to multi-model fitting is that all the structures embedded in the data are jointly estimated in a unified optimisation framework. We just need to define a robust energy/objective function that can be optimised using standard optimisation algorithms. The number of models can be discovered automatically. These approaches are clearly more advantageous than sequential methods (e.g., sequential RANSAC), which estimate a single model at a time and require the number of models to be specified. As shown in (Isack and Boykov, 2012), sequential RANSAC performs badly when measurement noise is high and the structures partially overlap.

Moreover, the energy-based approaches are able to incorporate prior knowledge (e.g., spatial coherence) into the parameter estimation. The priors are simply encoded into the objective function. Such a capability is quite impossible in the methods (e.g., (Toldo and Fusiello, 2008; Wang et al., 2012)) which estimate the structures independently.

Nevertheless, the crucial issue of the energy-based multi-model fitting is that the overall estimation performance heavily depends on the hypothesis set $\mathcal{H}$. If the sampled hypotheses $\mathcal{H}$ are inaccurate, or worse, if not all valid structures are sampled, the selection or optimisation step will be affected. Thus these approaches are highly vulnerable to sampling inadequacies, even with theoretical assurances on the optimisation step (e.g.,

---

\(^3\)http://www.mosek.com
globally optimal over the sampled hypotheses (Li, 2007; Delong et al., 2010; Yu et al., 2011)). Conversely, sampling an excessively large number of hypotheses (e.g., $10^{10}$) would ensure the adequacies, however, the computational cost (for both sampling and model selection) will significantly increase. Addressing these issues is one of the research objectives conducted in this thesis (see Chapters 3 and 4).

### 2.3 Random Hypothesis Generation

So far in this chapter, we have described various optimisation based robust methods. All these methods rely on a pre-sampled hypothesis set $\mathcal{H}$. In this section, we review techniques for sampling the set $\mathcal{H}$, which are based on elemental subsets.

#### 2.3.1 Elemental Subsets

An elemental subset consists of the minimum number of points required to fully instantiate the parameters of a geometric model. Thus an elemental subset is also called a minimal subset. For example, two points are required to determine a line, three points for a circle and four points for a planar homography. Although the uses of elemental subsets in model fitting have appeared in eighteenth century (see (Farebrother, 1997) for a summary of the history of elemental subset methods), the extension of the idea for robust estimation (handling outliers) was not popular until the works of (Fischler and Bolles, 1981) and (Rousseeuw, 1984). These two works are known as RANSAC (summarised in Algorithm 2.3) and LMS, respectively. Basically, RANSAC and LMS resort to elemental subsets to optimise the robust criteria (i.e., maximum consensus (1.2) and least median squares (1.3)), which do not have closed-form solutions.

Intuitively, minimal subsets are used to generate model hypotheses $\mathcal{H}$ which sample the continuous parameter space. Roughly speaking, there are two types of minimal subsets: uncontaminated and contaminated. An uncontaminated (clean) minimal subset contains only inliers from a genuine structure while other cases are contaminated subsets (e.g., subsets with outliers). Apparently, only clean minimal subsets are of interest because they are expected to produce reasonably good model hypotheses. Therefore, the target becomes generating minimal all-inlier subsets from the data. However it is quite impossible to dichotomise inliers from outliers without knowing the underlying model parameters. Furthermore, in practice it is computationally intractable to generate all possible minimal subsets, thus random sampling (as done in RANSAC) has to be used. The idea is to randomly sample minimal subsets such that at least one clean subset is found on each structure. Nevertheless, it is commonly known that obtaining clean
subsets is not sufficient, the subsets should also be high quality (i.e., subsets with low amount of noise or large spatial spans (Tran et al., 2013)). In the subsequent sections, we discuss techniques for sampling clean and high quality minimal subsets.

**Algorithm 2.3: RANSAC (Fischler and Bolles, 1981).**

**Input:** Input data $X = \{x_n\}_{n=1}^N$, maximum number of iterations $M$, size of minimal subset $p$, and inlier threshold $\sigma$.

**Output:** The best model $\theta^*$.

1: Initialise the current best solution $\theta^*$.
2: for $m = 1, 2, \ldots, M$ do
3: Randomly sample a minimal subset $S$ from $X$.
4: Hypothesise a model $\theta$ from $S$.
5: Find a set of inliers $I_\theta$ of $\theta$, i.e., the data points whose residual is less than $\sigma$.
6: if $|I_\theta| > |I_{\theta^*}|$ then
7: $\theta^* = \theta$.
8: end if
9: end for

### 2.3.2 Random Sampling

Pure random sampling is the simplest strategy for sampling minimal subsets. Suppose that we would like to sample minimal subsets of size $p$, a trivial sampling (for each subset) is to randomly select each point at a time until there are $p$ points (points are sampled independently). Since we are only interested in clean subsets, a natural question is that how many subsets are required such that at least one of them is uncontaminated? To answer the question, without loss of generality assume that the data contains only one structure, there are $N_{\text{inliers}}$ inliers among a total of $N$ points, the number of independently sampled $p$-tuple subsets is $M$, then the probability that at least one all-inlier subset has been drawn is

$$P = 1 - \left[1 - \left(\frac{N_{\text{inliers}}}{N}\right)^p\right]^M.$$  \hspace{1cm} (2.33)

Given a high probability $P$ (e.g., 0.999), we can infer the value $M$ that ensures this probability, i.e.,

$$M = \frac{\log(1 - P)}{\log(1 - w^p)},$$  \hspace{1cm} (2.34)

where $w = \frac{N_{\text{inliers}}}{N}$ is the inlier ratio. It can be seen that $M$ exponentially increases with the outlier ratio $1 - w$ and the size of minimal subsets $p$.

Let’s consider an example of fitting a fundamental matrix (Luong and Faugeras, 1996) onto the data, where at least 8 points ($p = 8$) are required to create a model candidate.
Suppose that the inlier ratio $w = 0.5$, with $P = 0.9999$, one would need to sample at least $M = 2353$ minimal subsets. When $w = 0.2$, $M = 3597785$. Figures 2.5 and 2.6 visualise the exponential increase of $M$ (the number of required minimal subsets) against the outlier ratios and the sizes of the minimal subsets respectively.

Note that when data contains multiple structures, an outlier ratio up to 80% is not unexpected. For example, consider the data with 3 structures, which occupy 20%, 20% and 10% of the data respectively, and 50% gross outliers, then the outlier ratio with respect to the first structure is 80%.
Clearly, pure random sampling is intractable in practice, especially when the data contains many high-order geometric structures. Therefore it is crucial to devise guided sampling techniques to accelerate the hypothesis generation process. In the next section, we review such techniques.

### 2.3.3 Guided Sampling

Guided sampling is a class of methods which aim to rapidly find high quality all-inlier subsets. As opposed to the standard random sampling method which treats all the data points equally and chooses members for minimal subsets randomly and independently, guided sampling methods select members probabilistically and/or conditionally. The idea is to associate each data point with a prior probability of being an inlier such that inliers are more likely to be simultaneously selected into a subset. Such (conditional) probabilities can be induced from the domain- or application-specific knowledge, assumptions on data distribution or the history of sampled subsets. The following paragraphs explain representative methods in detail.

The first group of methods (Tordoff and Murray, 2005; Chum and Matas, 2005) were specifically developed for two-view geometric estimation (see Chapter 5), where the sampling probabilities can be directly calculated from the data themselves. In these works, each datum is pair of SIFT features (Lowe, 1999), matched across two images, and its associated matching score is assumed to reflect its confidence of being an inlier, i.e., the higher score of a datum is, the more likely it is an inlier. Relying on that assumption, guided-MLESAC (Tordoff and Murray, 2005) uses the matching scores as the inlier probabilities, then applies a weighted sampling method to select members for each minimal subset (see Algorithm 2.4). As a consequence, the probability of sampling minimal subsets containing only inliers is considerably improved. Instead of assuming that the matching scores exactly translate into the inlier probabilities, PROSAC (Chum and Matas, 2005) restricts the sampling of minimal subsets to a potential collection of top highest score data. This collection expectedly contains only inliers. Technically, the inlier probability for a datum is high (≈ 1) if it belongs to the potential collection, and low (≈ 0) otherwise. The method incrementally expands the potential collection as the sampling progresses, such that diverse subsets can be generated, and eventually converges to the standard random sampling.

Although both guided-MLESAC and PROSAC have been shown to be computationally efficient, they still encounter several limitations. Firstly, these two techniques break down easily for multi-structure data. The primary reason is that the keypoint matching scores reveal no information about the inlier memberships, i.e., using the matching
Algorithm 2.4: The guided-MLESAC sampling algorithm (Tordoff and Murray, 2005)

**Input:** Input data $X = \{x_n\}_{n=1}^N$, maximum number of hypotheses $M$, size of a

minima subset $p$ and a set of sampling weights induced from the matching scores

$w = \{w_1, w_2, \ldots, w_N\}$.

**Output:** A hypothesis set $\mathcal{H}$.

1. $\mathcal{H} = \emptyset$
2. for $m = 1, 2, \ldots, M$ do
3.     for $k = 1, 2, \ldots, p$ do
4.         Sample a datum $s_k$ from $X$ using the sampling weight $w$.
5.         $S = S \cup \{s_k\}$.
6.     end for
7.     Hypothesise a model $\theta$ from $S$.
8.     $\mathcal{H} = \mathcal{H} \cup \{\theta\}$.
9. end for

Figure 2.7: Different types of “all-inlier” minimal subsets for the line and circle fitting problem. (a) An all-inlier minimal subset leads to a good model hypothesis. (b) A minimal subset contains inliers from different structures, leading to an inaccurate model hypothesis. (c) A minimal subset containing spatially close inliers also results in a bad model hypothesis.

scores only can not differentiate inliers from different structures. Therefore, these methods might select inliers from different structures into a single subset, which results in an incorrect hypothesis (see Figure 2.7(b) for an example). Another issue is that the matching scores do not necessarily translate to the inlier probabilities. This occurs because, for example, in many cases outliers could also have high scores, as observed in (Sattler et al., 2009; Serradell et al., 2010). More importantly, the prior inlier probabilities based on the application-specific knowledge (e.g., keypoint matching scores) are not always available in the general robust estimation.

Rather than directing the sampling onto a collection of potential data derived from the matching scores as in PROSAC, LO-RANSAC (Chum et al., 2003) focuses the sampling onto a set of inliers found thus far. LO-RANSAC requires no prior knowledge
about the inlier probabilities, but an inlier threshold. Specifically, given the best hypothesis sampled thus far (using a standard random sampling method) and an inlier threshold, LO-RANSAC extracts a set of inliers, then executes a local sampling procedure within the inlier set. Algorithm 2.5 shows the implementation of LO-RANSAC. Broadly speaking, LO-RANSAC belongs to a class of methods that simultaneously conducts global and local sampling to rapidly search for high quality hypotheses. This idea was then extended in a number of works such as BEEM (Goshen and Shimshoni, 2008), BLOGS (Brahmachari and Sarkar, 2009), Cov-RANSAC (Raguram et al., 2009), LO^+-RANSAC (Lebeda et al., 2012). The main differences between these methods rely on how the global and local sampling are performed, and when the global sampling is switched to the local sampling. Table 2.1 summarises the differences.

The usefulness of the local sampling step is only visible if the inliers and outliers are correctly classified. However, it might not possible to classify inliers and outlier completely without having a correct model parameter, and vise versa a correct model could not be achieved without knowing inliers (e.g., all-inlier subsets). In most cases, the inlier classification is based on an inlier threshold. Unfortunately, choosing a proper

<table>
<thead>
<tr>
<th>Methods</th>
<th>Global sampling</th>
<th>Switching schedule from global to local</th>
<th>Inlier classification</th>
<th>Local sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>LO-RANSAC</td>
<td>Random sampling as in RANSAC.</td>
<td>When a better hypothesis is found.</td>
<td>Use an inlier threshold.</td>
<td>Randomly sample data subsets from the inlier set.</td>
</tr>
<tr>
<td>BEEM</td>
<td>Random sampling using keypoint matching scores.</td>
<td>Based on the quality of the currently sampled hypothesis.</td>
<td>Use an inlier threshold.</td>
<td>Randomly sample minimal data subsets using the inlier probabilities.</td>
</tr>
<tr>
<td>BLOGS</td>
<td>Random sampling using keypoint matching scores.</td>
<td>With probability 0.5</td>
<td>Compute the conditional inlier probability for any data point, given the currently sampled hypothesis.</td>
<td>Randomly sample minimal data subsets from the inlier set.</td>
</tr>
<tr>
<td>Cov-RANSAC</td>
<td>Random sampling as in RANSAC.</td>
<td>When a better hypothesis is found.</td>
<td>Use model uncertainty for inlier classification.</td>
<td>Randomly select $7 \times p$ points from the inlier set, and apply an iterative least squares on the $7 \times p$ chosen points.</td>
</tr>
<tr>
<td>LO^+-RANSAC</td>
<td>Random sampling as in RANSAC.</td>
<td>When a better hypothesis is found.</td>
<td>Use an inlier threshold.</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Different local-and-global search strategies for hypothesis sampling. $p$ is the size of a minimal subset.
Algorithm 2.5: The LO-RANSAC sampling algorithm (Chum et al., 2003).

**Input:** Input data \( X \), maximum number of hypotheses \( M \), size of a minimal subset \( p \), inlier threshold \( \sigma \), local number of samples \( M_l \), and current best hypothesis \( \theta^* \).

**Output:** A hypothesis set \( \mathcal{H} \).

1. \( \mathcal{H} = \emptyset \).
2. **while** \( m < M \) **do**
3. Randomly sample a minimal subset \( S \subset X \).
4. Hypothesise a model \( \theta \) from \( S \).
5. \( \mathcal{H} = \mathcal{H} \cup \{ \theta \} \).
6. \( m = m + 1 \).
7. Compute the inliers set \( I_\theta \) of \( \theta \), i.e., points whose residual is less than \( \sigma \).
8. **if** \( |I_\theta| > |I_{\theta^*}| \) **then**
9. \( \theta^* = \theta \).
10. **for** \( i = 1, 2, \ldots, M_l \) **do**
11. Randomly sample a data subset \( S \subset I_\theta \). (\(|S| \geq p\))
12. Hypothesise a model \( \theta \) from \( S \).
13. \( \mathcal{H} = \mathcal{H} \cup \{ \theta \} \).
14. \( m = m + 1 \).
15. Compute the inliers set \( I_\theta \) of \( \theta \).
16. **if** \( |I_\theta| > |I_{\theta^*}| \) **then**
17. \( \theta^* = \theta \).
18. **end if**
19. **end for**
20. **end if**
21. **end while**

Inlier threshold is a non-trivial task, especially when data with multiple structures is presented since each structure might have a different level of noise.

Another class of guided sampling methods (Myatt et al., 2002; Kanazawa and Kawakami, 2004; Sattler et al., 2009; Ni et al., 2009) exploit the assumption that inliers of a genuine structure tend to be spatially close while outliers distribute sparsely. That is if a datum is an inlier, its neighbours are more likely to be inliers as well. The earliest attempt in this group is NAPSAC (Myatt et al., 2002), which restricts the selection of members of minimal subsets within a local region. Specifically, in NAPSAC the first member is selected randomly while the subsequent members lying within a hypersphere centred at the first selection, are sampled uniformly (see Algorithm 2.6). Similarly, the proximity sampling method (Kanazawa and Kawakami, 2004) places a Gaussian distribution centred on the first sampled datum such that the next points are selected probabilistically — points spatially closer to the center are more likely to be chosen next. The prior spatial information is also exploited to remove outliers in the data by using a spatial consistency filter proposed in SCRAMSAC (Sattler et al., 2009).

A crucial drawback of these methods is that the performances are heavily sensitive to the choice the spatial bandwidth (e.g., the radius of the hypersphere or the variance of
the Gaussian distribution). The smaller a bandwidth is, the more likely uncontaminated subsets can be found. However, the hypothesis fitted on a minimal subset of spatially close inliers might be arbitrarily biased from the true one due to the effects of inlier noise. Figure 2.7(c) illustrates a typical example. Increasing the bandwidth size might help improve the hypothesis quality, however also decreases the chance of hitting clean subsets. A comprehensive experiment can be found in (Tran et al., 2013).

**Algorithm 2.6:** The NAPSAC sampling algorithm (Myatt et al., 2002)

**Input:** Input data $X = \{x_i\}_{n=1}^N$, maximum number of hypotheses $M$, size of a minimal subset $p$.

**Output:** A hypothesis set $H$.

1: for $m = 1, 2, \ldots, M$ do
2: Randomly sample $s_1$ from $X$, and initialise $S = \{s_1\}$.
3: Extract a set of points $I_{s_1}$ lying within a hypersphere of radius $r$ centred on $s_1$.
4: for $k = 1, 2, \ldots, p - 1$ do
5: Randomly sample $s_{k+1}$ from $I_{s_1}$.
6: $S = S \cup \{s_{k+1}\}$.
7: end for
8: Construct a hypothesis $\theta$ from $S$.
9: $H = H \cup \{\theta\}$.
10: end for

As opposed to the above-mentioned methods, recent advanced guided sampling methods (Chin et al., 2010b, 2012; Wong et al., 2013) made no assumption on the prior inlier probabilities as well as the spatial distribution of the data. Rather the inlier probabilities are purely learnt from the history of the sampled hypotheses. Moreover in these methods the inlier probability of a datum is dependent on other data, such that the selections of subsequent members are conditioned on the previous selections. This distinct capability is very useful for multi-structure data, which helps to avoid inliers from different structures being included into a single minimal subset.

A notable example in this group is the Multi-GS sampling method (Chin et al., 2010b, 2012). Instead of computing the inlier probabilities for individual data, Multi-GS estimates the inlier probabilities for pairs of points — how likely the two points are inliers from the same structure. The intuition behind Multi-GS relies on the observation that if two data $x_i$ and $x_j$ are from the same structure, their preferences toward a collection of hypotheses $H$ are correlated. Specifically, given a hypothesis set $H$, the preference of a datum $x_i$ toward $H$ is obtained by sorting (in an ascending order) the residuals of $x_i$ to all the model hypotheses in $H$. Then the pairwise data similarities (inlier probabilities) are obtained by measuring the distances between pairwise data preferences.

Algorithm 2.7 summarises the basic steps of Multi-GS for sampling minimal subsets. The first member, say $s_1$, is randomly sampled; the second member, $s_2$, is selected
probabilistically based on the similarities (of all the points) with \( s_1 \); the third one, \( s_3 \), is sampled based on the similarities with both \( s_1 \) and \( s_2 \); and so forth. In effect, if the first selected datum is an inlier, there is a high probability that the subsequent members are all inliers from the same structure. However the downside of Multi-GS is that if the initial selection is an outlier, the subsequent selections, which are relatively expensive, are clearly wasteful.

Algorithm 2.7: The Multi-GS guided sampling algorithm (Chin et al., 2010b).

**Input:** Input data \( \mathcal{X} = \{x_n\}_{n=1}^N \), maximum number of hypotheses \( M \), size of a minimal subset \( p \) and an initial set of hypotheses \( \mathcal{H} \).

1: for \( m = 1, 2, \ldots, M \) do
2: Randomly sample \( s_1 \) from \( \mathcal{X} \), and initialise \( S = \{s_1\} \).
3: for \( k = 1, 2, \ldots, p-1 \) do
4: Compute the sampling weights \( \{w_n\}_{n=1}^N \) by multiplying the similarities between each \( x_n \) and all data in \( S \).
5: Sample \( s_{k+1} \) from \( \mathcal{X} \) using the sampling weight \( \{w_n\}_{n=1}^N \).
6: \( S = S \cup \{s_{k+1}\} \).
7: end for
8: Construct a hypothesis \( \theta \) from \( S \).
9: \( \mathcal{H} = \mathcal{H} \cup \{\theta\} \).
10: end for

To resolve the weakness of Multi-GS, the ModeSamp method (Wong et al., 2013) proposes to simultaneously estimate the inlier probabilities for individual data as well as for pairs of data, such that the first datum can also be sampled probabilistically. Particularly, ModeSamp extracts a number of modes in the data using the pairwise data similarities, as computed in Multi-GS. Given that the pairwise data similarities are approximately correct, these modes will represent inliers in the data. As a result, by restricting the sampling of the first data among these modes, the uncontaminated minimal subsets can be rapidly retrieved. Note that these probabilities are inaccurate in the early stages and are adaptively updated as the sampling progresses. Nevertheless, these approaches (Multi-GS and ModeSamp) require a considerable amount of computation effort for calculating the pairwise similarities and mode seeking, thus the efficiency drops quickly with the increase in the data size as well as the number of hypotheses.

Notice that almost all of the above-discussed guided sampling methods only focus on improving the speed of hitting clean minimal subsets, but not on improving (directly) the quality. (High quality subsets will produce accurate hypotheses.) Although increasing the number of samples will improve the chance of finding high-quality minimal subsets, directly sampling high-quality all-inlier subsets is more preferable. In (Tran et al., 2013), the authors proposed a technique, namely Multi-GS-Offset, that aims to rapidly retrieve all-inlier minimal subsets with large spans. Figure 2.7(a) illustrates a typical subset with a large span, which leads to a good line candidate. Multi-GS-Offset simply
combines Multi-GS with distance-based sampling. More specifically, given the first sampled member \( s_1 \) for a subset, the next member \( s_2 \) is selected probabilistically based on the similarity and spatial distance between \( s_1 \) and \( s_2 \). Intuitively, \( s_1 \) and \( s_2 \) should be simultaneously correlated and far apart. Although having proved the superior performance, similar to Multi-GS and ModeSamp, the computation cost of Multi-GS-Offset increases rapidly along with the number of data points and hypotheses.

### 2.3.4 Discussion

We have summarised various sampling techniques to construct the hypothesis set \( \mathcal{H} \) that is representative of the true structures in the data. Traditionally, a hypothesis is generated by fitting a model onto an elemental subset of the data. The task becomes finding high-quality, clean minimal subsets in the data (at least one on each structure). We showed that such a process is computationally expensive. To speed up the sampling, a majority of sampling methods take advantage of meta-information (e.g., keypoint matching scores) or assumptions on data distribution (e.g., inliers tend to form a cluster) to rapidly retrieve clean subsets. Some other methods (e.g., Multi-GS (Chin et al., 2010b)), on the other hand, analyse the history of sampled hypotheses to infer the inlier probabilities of the data such that subsets containing only inliers can be found rapidly.

However, in practice constructing a sufficient hypothesis set \( \mathcal{H} \) (i.e., \( \mathcal{H} \) covers all true structures in the data) takes far more computational expense than theoretically expected. It has been shown in (Tordoff and Murray, 2002) that the actual number of hypotheses \( \tilde{M} \) required by RANSAC is greatly larger (by an order of magnitude) than the predicted value \( M \), given in Equation (2.34). The primary reason is that clean samples may yield hypotheses that are not accurate (i.e., hypotheses that are arbitrarily far from the true models) because of noise in the data. As an example, Figure 2.7(c) shows an biased circle estimated from three noisy inliers. The higher noise levels cause the model hypotheses more biased. Unfortunately, almost all the existing guided sampling methods only try to improve the probability of finding clean minimal subsets. An exception is the method in (Tran et al., 2013) that tries to sample clean minimal subsets with large spans. Such subsets prove to produce higher-quality hypotheses. However it is unclear how to extend the method to multi-structure cases where different structures may have different spatial spans (e.g., objects with imbalanced sizes). We will propose an alternative solution in Chapter 4.
2.4 Summary

In this chapter, we have reviewed state-of-the-art optimisation based robust estimation methods. In general these methods advocate a hypothesise-then-select approach — first sample a large set of model hypotheses, then select the best subset that optimises a fitting objective function. Previously used hypothesis sampling techniques have been clearly explained in Section 2.3. Once a hypothesis set is available, depending on the specific definition of the objective functions encoding the fitting requirements, various optimisation strategies have been devised to select the best subset of hypotheses.

As discussed in Section 2.2, a disjoint hypothesise-then-select approach is clearly sub-optimal. The inaccuracies in the hypothesis sampling step will heavily affect the overall model estimation even though the fitting optimisation (model selection) guarantees optimality. The post-refinement step will not be effective if the selected models are far from the true ones. We argue that the two steps should be integrated into a unified framework. We will address this problem by proposing a continuous hypothesise-and-select approach to be presented in Chapter 3 and Chapter 4.

Secondly the quality of the sampled hypotheses is critical to the efficiency and effectiveness of the model fitting. State-of-the-art hypothesis generation methods rely on sampling minimal subsets of data, where the members of a minimal subset are sampled sequentially. Unfortunately, we showed in Section 2.3 that such a sampling strategy is computationally expensive, and the resulting hypotheses are not accurate. We propose a novel method that samples larger-than-minimal subsets, where all the members of a subset are selected simultaneously (see Chapter 4).

It is also crucial to note that the fitting accuracy also depends on the fitting criterion on top of the quality of sampled hypotheses and fitting optimisation. A criterion with many constraints would potentially lead to a more accurate fitting, but it is hard to optimise. For tractability, most of the fitting criteria contain only the goodness of fit and model complexity. However those criteria are not robust against non-uniform outliers and non-Gaussian inlier noise. This motivates us to introduce a more robust fitting criterion (see Chapter 6) which additionally considers pairwise geometric constraints between structures, and can be efficiently optimised using the recent energy minimisation method (Ladicky et al., 2010).
Chapter 3

Simultaneous Hypothesis Sampling and Model Selection

As presented in Chapter 2, previous optimisation based robust estimation methods advocate a hypothesise-then-select approach. In particular, a set of model hypotheses are randomly sampled, then the best subset that optimises some fitting criterion is considered as the estimate. This disjoint two-stage approach is arguably suboptimal and inefficient. If the random sampling did not retrieve a good set of hypotheses, the optimised outcome will not represent a good fit. Additionally, if the hypothesis set is excessively large, much more computational cost is paid for both the hypothesis generation and the model selection.

The issue above is due to the lack of a stopping criterion for the hypothesis sampling stage. If there is only one structure in the data, we can easily evaluate the sampled hypothesis quality (e.g., consensus size) on-the-fly and stop as soon as the prospect of obtaining a better hypothesis becomes insignificant (Fischler and Bolles, 1981). Under multi-structure data, it is unknown what a suitable stopping criterion is (unless one solves the overall fitting and model selection problem itself). One can consider iterative local refinement of the structures after data assignment (Isack and Boykov, 2012), but the fact remains that if the initial hypotheses are inaccurate, the subsequent fitting and refinement will be not effective.

In this chapter, we present a more appropriate approach that conducts sampling and model selection simultaneously. Specifically, we propose a new method for multi-structure fitting based on Reversible Jump Markov Chain Monte Carlo (RJMCMC) (Green, 1995). By design our MCMC based method directly optimises the fitting criterion via hypothesis sampling, such that the wastage in the sampling and model selection can be minimised.
Also we show how to exploit the reversible jump mechanism to provide a simple and effective framework for multi-structure model selection.

The main content in this chapter is based on the author’s publication in (Pham et al., 2011); the chapter is organised as follows. In Section 3.1, we briefly provide the background on MCMC methods and technical challenges. Section 3.2 formulates the multi-model fitting using an optimisation framework based on a RJMCMC approach. In Section 3.3, we present our main contributions including the adaptive (hypothesis) proposal distribution for RJMCMC, and the proof that it is a valid adaptive MCMC sampler. We illustrate the properties and capabilities of our approach in Section 3.4. We summarise the chapter in Section 3.5.

3.1 MCMC Methods

MCMC methods belong to a class of sampling-based algorithms for solving integration and optimisation problems in large and high-dimensional spaces. The idea is to draw a set of i.i.d samples \( \{\Theta^m\}_{m=1}^M \) from a density or objective function \( \pi(\Theta) \). These samples are then used to compute the approximate integral or maximum of \( \pi(\Theta) \). For example, the maximum of the objective function \( \pi(\Theta) \) can be achieved as

\[
\Theta^* = \arg \max_{\Theta^m: m=1,...,M} \pi(\Theta^m). \tag{3.1}
\]

In practice, MCMC methods are usually combined with simulated annealing for more efficient optimisation (see Section 3.2.2). Note that it is not always possible to draw samples from the distribution \( \pi(\Theta) \) directly unless \( \pi(\Theta) \) has a standard form (e.g., Gaussian). In such cases, MCMC methods allow one to generate a sequence of samples \( \{\Theta^m\}_{m=1}^M \) that mimic the samples drawn from the distribution \( \pi(\Theta) \). In particular, the sequence is constructed such that it is a Markov chain, i.e.,

\[
\pi(\Theta^m | \Theta^{m-1},...,\Theta^1) = \mathcal{K}(\Theta^m | \Theta^{m-1}), \tag{3.2}
\]

where \( \mathcal{K} \) is a transition kernel. Intuitively, the value of the next state depends solely on the current state. To ensure that the chain has \( \pi(\Theta) \) as its stationary distribution, a sufficient condition is the detailed balance condition, i.e.,

\[
\pi(\Theta^m)\mathcal{K}(\Theta^{m-1}|\Theta^m) = \pi(\Theta^{m-1})\mathcal{K}(\Theta^m | \Theta^{m-1}). \tag{3.3}
\]

Moreover, the chain will converge to the invariant distribution \( \pi(\Theta) \) if the kernel \( \mathcal{K} \) is irreducible and aperiodic. Irreducibility suggests that there exists a non-zero probability
of reaching all other states from any given state $\Theta^m$ of the Markov chain. Aperiodicity ensures that the chain will not cycle determinately between states.

To construct a Markov chain with such properties, one can use the Metropolis-Hastings (MH) algorithm (Hastings, 1970). Basically, given a current state $\Theta^m$, the MH algorithm samples a candidate $\Theta^*$ according to a proposal distribution $q(\Theta^*|\Theta^m)$ (that is easy to sample). Then, the next state $\Theta^{m+1}$ is either $\Theta^*$ or $\Theta^m$ based on an acceptance probability $A(\Theta^*, \Theta^m)$. Algorithm 3.1 outlines the idea. We refer readers to (Andrieu et al., 2003) for a proof that the chain constructed by the MH algorithm is irreducible and aperiodic, and satisfies the detailed balance condition.

Algorithm 3.1: Metropolis-Hastings algorithm.

1: Initialise $\Theta^1$.
2: for $m = 1, \ldots, M$ do
3: Sample $u \sim U[0,1]$.
4: Sample $\Theta^* \sim q(\Theta^*|\Theta^m)$.
5: if $u < A(\Theta^*, \Theta^m) = \min \frac{\pi(\Theta^*)q(\Theta^m|\Theta^*)}{\pi(\Theta^m)q(\Theta^*|\Theta^m)}$ then
6: $\Theta^{m+1} = \Theta^*$.
7: else
8: $\Theta^{m+1} = \Theta^m$.
9: end if
10: end for

In spite of simplicity and universality, the bane of MCMC is the difficulty in designing efficient proposal distributions (e.g., $q(\ldots)$). A poor choice of $q(\ldots)$ can make the chain converge to the stationary distribution extremely slowly. Adaptive MCMC techniques (Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009) promise to alleviate this difficulty by learning the proposal distribution on-the-fly. To accelerate the efficiency of our RJMCMC based model fitting approach, we adopt a hypothesis generator proposed in (Chin et al., 2010b) that progressively updates the proposal distribution using previously generated hypotheses. Care must be taken in introducing such adaptive schemes, since a chain propagated based on a non-stationary proposal is non-Markovian, and unless the proposal satisfies certain properties (Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009), this generally means a loss of asymptotic convergence to the target distribution.

Clearing these technical hurdles is one of the major contributions in this chapter. Using emerging theory from adaptive MCMC (Roberts and Rosenthal, 2007; Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009; Giordani and Kohn, 2010), we prove that the adaptive proposal, despite its origins in robust estimation (Chin et al., 2010b), satisfies the properties required for convergence, most notably diminishing adaptation.
Chapter 3. *Simultaneous Hypothesis Sampling and Model Selection*  38

### 3.2 Model Fitting with Reversible Jump MCMC

In this section, we present a sophisticated optimisation framework for multi-structure fitting, which simultaneously conducts hypothesis sampling and model selection. The framework relies on a simulated annealing optimisation (Geman and Geman, 1984) embedded in a Reversible Jump MCMC algorithm.

#### 3.2.1 Objective Function

In Chapter 2, we have discussed various requirements for the optimal parameters \( \Theta \). Here we aim to find \( \Theta \) which balances goodness of fit and model complexity. We derive the following objective function

\[
E(\Theta) = D(\mathcal{X}, \Theta) + \lambda C(\Theta),  
\]

(3.4)

where \( D(\mathcal{X}, \Theta) \) measures the goodness of fit, \( C(\Theta) \) evaluates the model complexity, \( \lambda \) is a scaling parameter. Assuming i.i.d. Gaussian noise with known variance \( \sigma^2 \), the goodness of fit is defined as

\[
D(\mathcal{X}, \Theta) = \sum_{n=1}^{N} \rho \left( \min_{\theta_k \in \Theta} \frac{r(x_n, \theta_k)}{2\sigma} \right),  
\]

(3.5)

where \( r(x_n, \theta_k) \) is the absolute residual of \( x_n \) to the \( k \)-th structure \( \theta_k \) in \( \Theta \). The residuals are subjected to a robust loss function \( \rho(\cdot) \) to limit the influence of outliers; we use the biweight function (Huber, 2009), i.e.,

\[
\rho(u) = \begin{cases} 
\frac{1}{6}[1 - (1 - u^2)^3] & \text{if } |u| < 1 \\
\frac{1}{6} & \text{if } |u| > 1 
\end{cases}  
\]

(3.6)

The model complexity simply counts the number of structures in \( \Theta \), i.e., \( C(\Theta) = |\Theta| \).

Though the desired fitting now can be described by a simple energy function, minimising the function like (3.4) over a vast space of all possible \( \Theta \) is a formidable task. We will be addressing this challenging optimisation problem in subsequent sections.

#### 3.2.2 Simulated Annealing Optimisation

Simulated annealing (SA) is a well-known method for solving global optimisation problems. The SA technique searches for a good approximation to the global optimum of an objective function (with many local optima) over a large search space. Basically, the SA
technique works by iteratively sampling and accepting new solutions probabilistically, i.e., with some probability worse solutions are still accepted. Consequentially, the local optima can be effectively avoided. Simulated annealing has also proven to be effective for difficult model selection problems (Andrieu et al., 2001; Brooks et al., 2003).

As discussed, our aim is to find a global minimum of the objective function (3.4), which encodes a model selection criterion. Here we show that such an objective (3.4) can be optimised effectively using a MCMC based simulated annealing approach. The idea is to propagate a Markov chain for the Boltzmann distribution encapsulating (3.4), i.e.,

$$\pi_T(\Theta) \propto \exp\left(\frac{-E(\Theta)}{T}\right),$$

where temperature $T$ is progressively lowered until the samples from $\pi_T(\Theta)$ converge to
the global minimum of $E(\Theta)$. Under weak regularity assumptions, there exist cooling schedules (Brooks et al., 2003) that will guarantee that as $T$ tends to zero the samples from the chain will concentrate around the global minimum. Figure 3.1 visualises the idea of simulated annealing optimisation in 2D space.

Algorithm 3.2 shows the main body of the fitting algorithm. In the rest of this section, we will discuss a technique to simulate a Markov chain for $\pi_T(\Theta)$ with a fixed temperature $T$, which is the central part of the Algorithm 3.2 (Step 2).

**Algorithm 3.2:** Monte Carlo Markov Chain simulated annealing for geometric fitting.

1: Initialise temperature $T$ and state $\Theta$.
2: Simulate Markov chain for $\pi_T(\Theta)$ until convergence.
3: Lower temperature $T$ and repeat from Step 2 until $T \approx 0$.

### 3.2.3 Reversible Jump MCMC

Since the parameters $\Theta$ contain a variable number of structures $K$ (e.g., $\Theta = \{\theta_k\}_{k=1}^K$), we rewrite $\Theta$ as $\Theta_K \in \Omega^K$ to make explicit the size of $\Theta$; $\Omega^K$ is a space of all possible $K$ structures. To simulate $\pi_T(\Theta_K)$ we adopt a mixture of kernels MCMC approach (Andrieu et al., 2001). This involves in each iteration the execution of a randomly chosen type of move to update $\Theta_K$. Algorithm 3.3 summarises the idea. We make available 3 types of moves: birth, death and local update. Birth and death moves change the number of structures $K$. These moves effectively cause the chain to jump across parameter spaces $\Omega^K$ of different dimensions. It is crucial that these trans-dimensional jumps are reversible to produce correct limiting behaviour of the chain. Next we explain in detail each type of moves and their reversibility.

**Algorithm 3.3:** Reversible jump mixture of kernels MCMC to simulate $\pi_T(\Theta_K)$

**Input:** Last visited state $\Theta_K$ of the previous chain, probability $\beta$.

1: Sample $u \sim U_{[0,1]}$.
2: if $u \leq \beta$ then
3: With probability $r_B(K)$, attempt birth move, else attempt death move.
4: else
5: Attempt local update.
6: end if
7: Repeat from Step 1 until convergence (e.g., last $V$ moves all rejected).
3.2.3.1 Birth and Death Moves

The birth move propagates \( \Theta_K \) to \( \Theta'_{K'} \), with \( K' = K + 1 \). Applying Green’s (Green, 1995; Richardson and Green, 1997) seminal theorems on RJMCMC, the move is reversible if it is accepted with probability \( \min\{1, A_{\text{birth}}\} \), where

\[
A_{\text{birth}} = \frac{\pi_T(\Theta'_{K'})[1 - r_B(K')]/K'}{\pi_T(\Theta_K)r_B(K)q(u)} \left| \frac{\partial \Theta'_{K'}}{\partial (\Theta_K, u)} \right|. \tag{3.8}
\]

The probability of proposing the birth move is \( r_B(K) \), where

\[
r_B(K) = \begin{cases} 
1 & \text{if } K = 1 \\
0.5 & \text{if } K = 2, \ldots, K_{\text{max}} - 1 \\
0 & \text{if } K = K_{\text{max}} 
\end{cases} \tag{3.9}
\]

In other words, any move that attempts to move \( K \) beyond the range \( [1, K_{\text{max}}] \) is disallowed in Step 3 of Algorithm 3.3. \( K_{\text{max}} \) is an integer number representing a prior on the maximum number of structures allowable in the data.

In the birth move, the extra degrees of freedom required to specify the new item in \( \Theta'_{K'} \) are given by auxiliary variables \( u \), which are in turn proposed by \( q(u) \). Following previous work, we estimate parameters of the new item by fitting the geometric model \( \mathcal{M} \) onto a minimal subset of the data. Thus \( u \) is a minimal subset of \( X \). Our approach amounts to minimising (3.4) over collections \( \{\Theta_K\} \) of minimal subsets of \( X \), where now \( \Theta_K \equiv \{u_k\}_{k=1}^K \). Taking this view the mapping between \( \Theta'_{K'} \) and \( \Theta_K \) is identity (e.g., \( \Theta'_{K'} = \{\Theta_K, u\} \)), so the Jacobian \( \partial \Theta'_{K'}/\partial (\Theta_K, u) \) is simply the identity matrix. \( A_{\text{birth}} \) becomes

\[
A_{\text{birth}} = \frac{\pi_T(\Theta'_{K'})[1 - r_B(K')]/K'}{\pi_T(\Theta_K)r_B(K)q(u)}. \tag{3.10}
\]

The death move is much simpler than the birth move, where an existing structure is chosen randomly and deleted from \( \Theta_K \). The death move brings \( \Theta_K \) to \( \Theta'_{K'} \), with \( K' = K - 1 \). The probability of choosing a death move \( 1 - r_B(K) \). The death move is accepted with probability \( \min\{1, A_{\text{death}}\} \), where \( A_{\text{death}} = A_{\text{birth}}^{-1} \). That is

\[
A_{\text{death}} = \frac{\pi_T(\Theta'_{K'})r_B(K')q(u)}{\pi_T(\Theta_K)[1 - r_B(K)]/K}. \tag{3.11}
\]

3.2.3.2 Local Update

A local update does not change the model complexity \( K \). The move involves randomly choosing a structure \( \theta_k \) in \( \Theta_K \) to update, making only local adjustments to its minimal
subset $u_k$. The outcome is a revised minimal subset $u_k'$, and the move is accepted with probability $\min\{1, A_{\text{update}}\}$, where

$$A_{\text{update}} = \frac{\pi_T(\Theta'_K)q(u_k'|\theta'_k)}{\pi_T(\Theta_K)q(u_k'|\theta_k)}$$

(3.12)

Our local update is also accomplished with the same proposal $q(u|\theta)$, but this time conditioned on the selected structure $\theta_k$. Section 3.3.2 describes and analyses $q(u|\theta)$.

It has been known that the speed of convergence of MCMC simulation highly depends on the proposal distributions — good proposals would lead to faster convergences. In our case, the only concern is the proposal $q(u)$ (and hence $q(u|\theta)$) since there are a colossal number of possible minimal subsets $u$. Therefore, obtaining good overall performance thus hinges on the ability of proposal $q(u)$ to propose minimal subsets that are relevant, i.e., those fitted purely on inliers of valid structures in the data. One way is to learn $q(u)$ incrementally using generated hypotheses. We describe such a scheme (Chin et al., 2010b) in Section 3.3.1 and prove that the adaptive proposal preserves ergodicity in Section 3.3.3.

### 3.3 Adaptive MCMC Proposal

In this section, we propose an adaptive MCMC proposal to speed up the convergence of the RJMCMC simulation, which is one of the central contributions in this chapter. Our work capitalises on the guided hypothesis sampling scheme of Chin et al. called Multi-GS (Chin et al., 2010b) originally proposed for accelerating the hypothesis generation process. The algorithm maintains a series of sampling weights which are revised incrementally as new hypotheses are generated. This bears similarity to the pioneering Adaptive Metropolis (AM) method of Haario et al. (Haario et al., 2001). Here, we prove that our adaptive proposals $q(u)$ and $q(u|\theta)$, based on Multi-GS, satisfy the conditions required to preserve ergodicity.

#### 3.3.1 The Multi-GS Algorithm

Let $\{\theta_m\}_{m=1}^M$ aggregate the set of hypotheses fitted on the minimal subsets proposed thus far in all birth and local update moves in Algorithm 3.2. To build the sampling weights, first for each $x_i \in X$ we compute its absolute residuals as measured to the $M$ hypotheses, yielding the residual vector

$$r^{(i)} := [r_1^{(i)} r_2^{(i)} \cdots r_M^{(i)}].$$
We then find the permutation
\[ a^{(i)} := [a^{(i)}_1 \ a^{(i)}_2 \ \cdots \ a^{(i)}_M] \]
that sorts the elements in \( r^{(i)} \) in non-descending order. The permutation \( a^{(i)} \) essentially ranks the \( M \) hypotheses according to the preference of \( x_i \); the higher a hypothesis is ranked the more likely \( x_i \) is an inlier to it. The weight \( w_{i,j} \) between the pair \( x_i \) and \( x_j \) is obtained as
\[
\begin{align*}
  w_{i,j} &= I_h(x_i, x_j) := \frac{1}{h} |a^{(i)}_h \cap a^{(j)}_h|, \\
  |a^{(i)}_h \cap a^{(j)}_h| &= \text{the number of identical elements shared by the first-}h \text{ elements of } a^{(i)} \text{ and } a^{(j)}. 
\end{align*}
\]
where \( |a^{(i)}_h \cap a^{(j)}_h| \) is the number of identical elements shared by the first-\( h \) elements of \( a^{(i)} \) and \( a^{(j)} \). Clearly \( w_{i,j} \) is symmetric with respect to the input pair \( x_i \) and \( x_j \), and \( w_{i,i} = 1 \) for all \( i \). To ensure technical consistency in our later proofs, we add a small positive offset \( \gamma \) to the weight\(^1\), or
\[
  w_{i,j} = \max(I_h(x_i, x_j), \gamma),
\]
hence \( \gamma \leq w_{i,j} \leq 1 \). The weight \( w_{i,j} \) measures the correlation of the top \( h \) preferences of \( x_i \) and \( x_j \), and this value is typically high iff \( x_i \) and \( x_j \) are inliers from the same structure; Figure 3.3 illustrates. Parameter \( h \) controls the discriminative power of \( w_{i,j} \), and is typically set as a fixed ratio \( c \) of \( M \), i.e., \( h = \lceil cM \rceil \). Experiments suggest that \( c = 0.1 \) provides generally good performance (Chin et al., 2012).

Multi-GS exploits the preference correlations to sample the next minimal subset \( u = \{x_{s_t}\}_{t=1}^p \), where \( x_{s_t} \in \mathcal{X} \) and \( s_t \in \{1, \ldots, N\} \) indexes the particular datum from \( \mathcal{X} \); henceforth we regard \( u = \{s_t\}_{t=1}^p \). The first datum \( s_1 \) is chosen purely randomly. Beginning from \( t = 2 \), the selection of the \( t \)-th member \( s_t \) considers the weights related to the data \( s_1, \ldots, s_{t-1} \) already present in \( u \). More specifically, the index \( s_t \) is sampled according to the probabilities
\[
P_t(i) \propto \prod_{z=1}^{t-1} w_{s_z,i}, \quad \text{for } i = 1, \ldots, N,
\]
i.e., if \( P_t(i) > P_t(j) \) then \( i \) is more likely than \( j \) to be chosen as \( s_t \). A new hypothesis \( \theta_{M+1} \) is then fitted on \( u \) and the weights are updated in consideration of \( \theta_{M+1} \). Experiments comparing sampling efficiency (e.g., all-inlier minimal subsets produced per unit time) show that Multi-GS is superior over previous schemes, especially on multi-structure data; see (Chin et al., 2012) for details.

\(^1\)It can be shown if both \( x_i \) and \( x_j \) are uniformly distributed outliers, the expected value of \( w_{i,j} \) is \( h/M \), i.e., a given pair \( x_i \) and \( x_j \) will likely have non-zero preference correlation.
3.3.2 Adaptive Proposal using Multi-GS

Our RJMCMC scheme in Algorithm 3.3 depends on the Multi-GS-inspired adaptive proposals \( q_M(u) \) and \( q_M(u|\theta) \), where we now add the subscript \( M \) to make explicit their dependency on the set of aggregated hypotheses \( \{\theta_m\}_{m=1}^M \) as well as the weights \( \{w_{ij}\}_{i,j=1}^N \) they induce. The probability of proposing a minimal subset \( u = \{s_i\}_{i=1}^p \) from \( q_M(u) \) can be calculated as

\[
q_M(u) = \frac{1}{N} \prod_{a<b} w_{s_a,s_b} \left[ \prod_{d=1}^{p-1} \left( \bigodot_{e=1}^d w_{s_e} \right) \right]^{-1}, \quad (3.16)
\]

where \( w_i \) is the column vector \([w_{i,1} \ldots w_{i,N}]^T\) and \( \bigodot \) is the sequential Hadamard product over the given multiplicands. The term with the inverse in Equation (3.16) relates to the normalising constants for Equation (3.15). As an example, the probability of selecting the minimal subset \( u = \{s_1, s_2, s_3, s_4\} \) is

\[
q_M(u) = \frac{1}{N} 1^T w_{s_1} 1^T (w_{s_1} \odot w_{s_2}) 1^T (w_{s_1} \odot w_{s_2} \odot w_{s_3}).
\]

The local update proposal \( q_M(u|\theta) \) differs only in the manner in which the first datum \( x_{s_1} \) is selected. Instead of chosen purely randomly, the first index \( s_1 \) is sampled according to

\[
P_{s_1}(i) \propto \exp\left(-\frac{O(g(x_i, \theta))}{n}\right), \quad \text{for } i = 1, \ldots, N, \quad (3.17)
\]

where \( O(r(x_i, \theta)) \) is the order statistic of the absolute residual \( r(x_i, \theta) \) as measured to \( \theta \); to define \( q_M(u|\theta) \) the \( 1/N \) term in Equation (3.16) is simply replaced with the appropriate probability from Equation (3.17). For local updates an index \( i \) is more likely to be chosen as \( s_1 \) if \( x_i \) is close to \( \theta \). Parameter \( n \) relates to our prior belief of the minimum number of inliers per structure; we fix this to \( n = 0.1N \).

Since our proposal distributions are updated with the arrival of new hypotheses, the corresponding transition probabilities are inhomogeneous (they change with time) and the chain is non-Markovian (the transition to a future state depends on all previous states). Therefore the convergence of the Markov chain is no longer evident. In the next section, we show that the chain with such adaptive proposals will still lead to the correct target distribution (3.7).

3.3.3 Convergence

First we restate Theorem 1 in (Giordani and Kohn, 2010), which provides sufficient conditions for the ergodicity of an adaptive proposal. The Theorem is distilled from
other work on Adaptive MCMC (Roberts and Rosenthal, 2007; Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009).

**Theorem 1.** Let $Z = \{Z_n : n > 0\}$ be a stochastic process on a compact state space $\Xi$ evolving according to a collection of transition kernels

$$T_n(z, z') = pr(Z_{n+1}|Z_n = z, Z_{n-1} = z_{n-1}, \ldots, Z_0 = z_0),$$

and let $p(z)$ be the distribution of $Z_n$. Suppose for every $n$ and $z_0, \ldots, z_{n-1} \in \Xi$ and for some distribution $\pi(z)$ on $\Xi$,

$$\sum_{z_n} \pi(z_n)T_n(z_n, z_{n+1}) = \pi(z_{n+1}),$$  \hspace{1cm} (3.18)

$$|T_{n+k}(z, z') - T_n(z, z')| \leq a_n c_k, \quad a_n = O(n^{-r_1}), \quad c_k = O(k^{-r_2}), \quad r_1, r_2 > 0,$$ \hspace{1cm} (3.19)

$$T_n(z, z') \geq \epsilon \pi(z'), \quad \epsilon > 0,$$  \hspace{1cm} (3.20)

where $\epsilon$ does not depend on $n, z_0, \ldots, z_{n-1}$. Then, for any initial distribution $p(z_0)$ for $Z_0$,

$$\sup_{z_n} |p(z_n) - \pi(z_n)| \to 0 \quad \text{for} \quad n \to \infty.$$  

Theorem 1 says that if the three conditions (3.18) (invariance), (3.19) (diminishing adaptation), and (3.20) (ergodicity) hold, then the distribution of $Z_n$ converges to $\pi(z)$.

Next we show that our adaptive proposal satisfies these conditions.

3.3.3.1 Invariance

Equation (3.18) requires that transition probabilities based on $q_M(u)$ permit an invariant distribution individually for all $M$. Since we propose and accept states based on the Metropolis-Hastings algorithm, the detailed balance condition (3.3) is satisfied by construction (Andrieu et al., 2003), which means that a Markov chain propagated based on $q_M(u)$ will admit $\pi(\Theta)$ as its stationary distribution.

3.3.3.2 Diminishing Adaptation

Equation (3.19) dictates that the difference between consecutive transition kernels, and thus the proposal distributions in the Metropolis-Hastings updates in Equations (3.8) and (3.12), decreases with the length of the chain, i.e., the adaptation must diminish. To see that this occurs naturally in $q_M(u)$, first we show that the change $|w'_{i,j} - w_{i,j}|$
Chapter 3. Simultaneous Hypothesis Sampling and Model Selection

for all \(i, j\) gets smaller as \(M\) increases. Without loss of generality assume that \(b\) new hypotheses are generated between successive weight updates \(w_{i,j}\) and \(w'_{i,j}\). Then,

\[
\lim_{M \to \infty} |w'_{i,j} - w_{i,j}| = \lim_{M \to \infty} \left| \frac{|a'_{i}(M+b) \cap a_{j}(M+b)|}{c(M + b)} - \frac{|a_{i}(M) \cap a_{j}(M)|}{cM} \right|
\leq \lim_{M \to \infty} \left| \frac{|a_{i}(M) \cap a_{j}(M)|}{c(M + b)} \pm b(c + 1) \right|
= \lim_{M \to \infty} \left| \frac{|a_{i}(M) \cap a_{j}(M)|}{c + cb/M} \right| = 0,
\]

(3.21)

where \(a'^{(i)}\) is the revised preference of \(x_i\) in consideration of the \(b\) new hypotheses. The result is based on the fact that the extension of \(b\) hypotheses will only perturb the overlap between the top-c percentile of any two preference vectors by at most \(b(c + 1)\) items. It should also be noted that the result is not due to \(w'_{i,j}\) and \(w_{i,j}\) simultaneously converging to zero with increasing \(M\); in general

\[
\lim_{M \to \infty} \frac{|a_{i}(M) \cap a_{j}(M)|}{cM} \neq 0
\]

(3.22)

since \(a^{(i)}\) and \(a^{(j)}\) are extended and revised as \(M\) increases and this may increase their mutual overlap. Figure 3.3 illustrates the vanishing adaptation of \(w_{i,j}\) as \(M\) increases.

Using the above result, it can be shown that the adaptation of the product of any two weights also diminishes

\[
\lim_{M \to \infty} |w'_{i,j}w'_{p,q} - w_{i,j}w_{p,q}| = \lim_{M \to \infty} |w'_{i,j}(w'_{p,q} - w_{p,q}) + w_{p,q}(w'_{i,j} - w_{i,j})|
\leq \lim_{M \to \infty} |w'_{i,j}| |w'_{p,q} - w_{p,q}| + |w_{p,q}| |w'_{i,j} - w_{i,j}| = 0.
\]

(3.23)

This result is readily extended to the product of any number of weights. To show the diminishing adaptation of the normalisation terms in (3.16), we first observe that the sum of weights is bounded away from 0

\[
\forall i, \quad 1^T w_i \geq \mathcal{L}, \quad \mathcal{L} > 0,
\]

(3.24)

due to the offsetting (3.14) and the constant element \(w_{i,i} = 1\) in \(w_i\) (although \(w_{i,i}\) will be set to zero to enforce sampling without replacement (Chin et al., 2012)). It can thus
be established that
\[
\lim_{M \to \infty} \left| \frac{1}{1^T w_i'} - \frac{1}{1^T w_i} \right| = \lim_{M \to \infty} \left| \frac{1^T w_i' - 1^T w_i}{(1^T w_i')(1^T w_i)} \right| 
\leq \lim_{M \to \infty} \left| \frac{1^T w_i' - 1^T w_i}{L^2} \right| = \lim_{M \to \infty} \left| \frac{\sum_{j=1}^{N} (w_{i,j}' - w_{i,j})}{L^2} \right| = 0 \quad (3.25)
\]
since \( \lim_{M \to \infty} |w_{i,j}' - w_{i,j}| = 0 \). The result is readily extended to the inverse of the sum of any number of Hadamard products of weights, since we have also previously established that the adaptation of the product of any number of weights diminishes. Finally, since Equation (3.16) involves only multiplications of diminished quantities, the amount of the adaptation of \( q_M(u) \) will diminish as the update progresses.

### 3.3.3.3 Uniform Ergodicity

Equation (3.20) requires that \( q_M(u) \) for all \( M \) be individually ergodic, i.e., the resulting chain using \( q_M(u) \) is aperiodic and irreducible. Again, since we simulate the target using Metropolis-Hastings, every proposal has a chance of being rejected, thus implying aperiodicity (Andrieu et al., 2003). Irreducibility is satisfied by the offsetting in (3.14) and renormalising (Nott and Kohn, 2005), since this implies that there is always a non-zero probability of reaching any state (minimal subset) from the current state.

The above results apply for the local update proposal \( q_M(u|\theta) \) which differs from \( q_M(u) \) only in the (stationary) probability to select the first index \( s_1 \). Hence \( q_M(u|\theta) \) is also a valid adaptive proposal.

### 3.4 Experiments

So far in this chapter, we have presented a complete sampling-based optimisation algorithm for multi-structure model fitting. In this section, we demonstrate the distinct properties and capabilities of our proposed fitting algorithm, denoted as ARJMC, using both synthetic and real dataset.

### 3.4.1 Synthetic data

We first illustrate the behaviours of our proposed algorithm in fitting lines onto a synthetic 2D point cloud (see Figure 3.2). The data are sampled from 5 line segments, where there are 100 inliers on each segment. The inliers are perturbed by Gaussian noise with standard deviation \( \sigma = 0.01 \). The data is also contaminated by 200 uniform
outliers. The goal is to recover the unknown line parameters used to generate the data. A line is modelled as a 3-tuple $\theta = (a, b, c)$, and the distance (residual) from a point $x_n = (x_n, y_n)$ to $\theta$ is

$$r(x_n, \theta) = \frac{|ax_n + by_n + c|}{\sqrt{a^2 + b^2}}. \quad (3.26)$$

In Algorithm 3.2, the temperature $T$ is initialized as 1 and we apply the geometric cooling schedule $T_{next} = 0.99T$. In Algorithm 3.3, probability $\beta$ is set as equal to the current temperature $T$, thus allowing more global explorations in the parameter space initially before concentrating on local refinement subsequently. Such a helpful strategy is not naturally practicable in disjoint two-stage approaches. We set $K_{max} = 10$, $\lambda = 4$.

Figure 3.3 shows the evolution of the pairwise weights (3.13) as the number of iterations $I$ (thus the number of hypotheses $M$) increases until 10,000 for the data in Figure 3.2. The matrices exhibit a five-block pattern, indicating strong mutual preference among inliers from the same structure. This phenomenon allows accurate selection of minimal subsets in Multi-GS (Chin et al., 2012). More pertinently, as we predicted in Section 3.3.3, the weight variances become smaller as $M$ increases, as evidenced by the stabilising block pattern. Note that only a small number of weights are actually computed in Multi-GS; the full matrix of weights are calculated here for illustration only.

Figure 3.4 depicts the evolution of fitting results as $I$ increases. It can be seen that for small $I$, the algorithm wrongly estimates the model parameters as well as the number of structures. However, since the algorithm inherently keeps sampling and fitting, the result is updated continuously and subsequently converges to a good estimate (when

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure32.png}
\caption{A 2D point cloud includes noisy inliers sampled from 5 straight line segments and outliers. The goal of multi-structure model fitting is to recover the underlying line parameters from the data.}
\end{figure}
Figure 3.3: The evolution of the matrix of pairwise weights (3.13) computed from the data in Figure 3.2 as the number of iterations (and hence the number of hypotheses $M$) is increased. For presentation the data are arranged according to their structure membership, which gives rise to a 5-block pattern. Observe that the block pattern, hence weights, “freeze” as $M$ increases.

Figure 3.4: The evolution of the fitting result of ARJMC as the number of iterations increases using the data in Figure 3.2. The colors suggest the inlier memberships. Detected outliers are shown in red. Observe that our method converges to a “good” result after only 200 iterations.

Figure 3.5: The evolution of the objective function value as the algorithm progresses in fitting lines onto the data in Figure 3.2. The algorithm converges after 200 iterations.

$I > 200)$. This property makes our method distinct from the two-stage approaches, where hypothesis sampling and model selection are treated as the two separate problems.

Figure 3.5 plots the evolution of the underlying objective function value (3.4) when the fitting algorithm progresses. Notice the fluctuations of the objective curve during the early stages of the iterations (when temperature $T$ is high) — This shows that states
with higher objective values are still accepted such that local optima can be avoided. When the iteration number gets higher (hence the temperature $T$ gets lower), only states decreasing the objective are accepted. The algorithm starts converging quickly after 200 iterations. Also Figure 3.6 displays the associate evolution of the estimated number of structures. These plots correctly match to the actual fitting results as shown in Figure 3.4.

### 3.4.2 Real data

Now we demonstrate the practical advantages of our method over the two stage hypothesis-then-select approach using a real dataset. We choose the FLOSS method (see Section 2.2.4 for a review) as our representative competitor since the two methods have the same objective function. Note that an extensive benchmarking against other state-of-the-art methods using a larger dataset will be reported in Chapter 5.

We apply ARJMC and FLOSS to a two-view motion segmentation task and compare the performances. A detailed explanation of the motion segmentation problem, model parameters as well as the residual function are presented in Chapter 5. We select the data *dinobooks* (shown in Figure 3.7(a)) from the AdelaideRMF dataset (Wong et al., 2011)\(^2\) for this experiment.

To evaluate the performance, we design the benchmarking as follows. We run the ARJMC algorithm until 1000 iterations, and for each iteration, we record the objective value and the time elapsed. Since FLOSS is a two-stage method, we test FLOSS with different hypothesis sets (of different sizes 100, 300, 500 and 700). For each hypothesis

(a) Input data: each datum is pair of key-points matched across two views.

(b) Segmentation result of ARJMC at iteration 1000. (c) Segmentation result of FLOSS with 700 hypotheses.

(d) Efficiency comparison.

Figure 3.7: Performance comparison between ARJMC and FLOSS on a motion segmentation problem. (a) show the input data, where the lines connect keypoints in one image matched to corresponding keypoints in another image. There are totally 360 correspondences, 155 of them are wrong matches (outliers). (b) and (c) displays the segmentation results of ARJMC and FLOSS respectively (only one view is shown). (d) compares the efficiency between the two methods. For the FLOSS method, we plot the results with different hypothesis set sizes. In practice, we need to prepare a generous hypothesis set to ensure that at least one all-inlier sample is found on each structure. It is clear that ARJMC is significantly more efficient than FLOSS with 700 hypotheses.
set (sampled by using Multi-GS (Chin et al., 2010a)), we record the optimised objective value and time inclusive of the duration of hypothesis generation. The computation time and objective values are then used for the performance evaluation. Note that in practice it is impossible to test FLOSS with different hypothesis sets, instead we should choose the largest possible set such that at least one good hypothesis is expectedly found on each structure.

Each method is repeated 50 times, and the median results are reported in Figure 3.7(d). The advantage of ARJMC over FLOSS is clearly shown. Apparently, although the two methods approximately converge to the same objective values, ARJMC is considerably more efficient than FLOSS in minimising the objective function. Note that FLOSS actually does not have “convergence” since in practice we run FLOSS with a single hypothesis set. With 700 hypotheses, for example, FLOSS is significantly slower than ARJMC. In contrast, if only 100 hypotheses are given, FLOSS “converges” to a higher objective value than ARJMC. Figures 3.7(b) and 3.7(c) visualise the qualitative segmentation results of ARJMC with 1000 iterations and FLOSS with 700 hypotheses respectively.

3.5 Summary

We started the chapter by indicating the weakness of the existing optimisation based model estimation methods, which conduct hypothesis generation and model selection separately. Ideally, the hypothesis set should sufficiently cover all valid structures in the data, e.g., all valid structures are sampled. However, in practice it might be not possible to measure the adequacy of the hypothesis set until the model selection has been optimised. This leads to several issues. For example, if some of the structures are not sampled, the overall fitting performance will be affected regardless of the optimality of the model selection optimisation. This situation might be avoided by sampling a huge number of hypotheses. However, the cost of the hypothesis sampling is computationally expensive. Plus the efficiency and optimality of the model selection optimisation will be decreased.

We then presented a novel algorithm to tackle the aforementioned issues. The idea is that the fitting algorithm should perform hypothesis sampling, geometric fitting and model selection simultaneously. This strategy encourages faster convergence and minimises wastage in the sampling process. We realised the idea by using a Reversible Jump Monte Carlo Markov Chain algorithm. Underpinning our novel Reversible Jump MCMC method is an efficient hypothesis generator whose proposal distribution is learned online. Drawing from emerging theory on Adaptive MCMC, we proved that our efficient
hypothesis generator satisfies the properties crucial to ensure convergence to the correct target distribution. Our work thus links the latest developments from stochastic optimisation and geometric model fitting.

We note that in general the overall fitting efficiency and accuracy are largely affected by the performance of the hypothesis sampler. This is especially true for high-order geometric models (e.g., fundamental matrices) since it could take an enormous effort to sample all-inlier subsets from all the valid structures present in the data, thus the algorithm will converge slower. Furthermore, as presented in Chapter 2, a hypothesis fitted on an all-inlier subset might be biased largely towards the correct model due to the measurement errors. Hence suboptimal solutions might not be escaped when the noise becomes high, even we simulate the Markov chain over a long period of time. We will propose possible solutions in the next chapter.
Chapter 4

Rapid Hypothesis Sampling using Random Cluster Models

As we have stated several times, random hypothesis generation is central to robust geometric model fitting. The goal is to generate model candidates that are representative of the true structures in the data. As reviewed in Chapter 2, most previous fitting methods require a set of accurate model candidates to work properly. Also in Chapter 3, we introduced a new robust fitting algorithm, which simultaneously optimises the fitting and conducts hypothesis sampling. In general, the quality of the sampled hypotheses is critical to the overall fitting efficiency and accuracy.

The predominant techniques generate hypotheses by fitting models onto randomly sampled minimal subsets of the data (see Chapter 2 for a review). The intuition behind sampling minimal subsets is to maximise the chance of successively hitting inliers in a single sample (see Equation (2.34) in Chapter 2); and a hypothesis fitted on only inliers is expected to be a satisfactory model candidate. However, the desired accuracy of sampling minimal subsets might not be achieved when the measurement noise is high. Figure 4.1(a) illustrates an example. Therefore, to ensure the fitting accuracy much more sampling effort has to be spent so that better minimal subsets (e.g., subsets of noise-free inliers) and thus better hypotheses might be found.

A better strategy is to directly alleviate the effect of noise by generating hypotheses fitted on large subsets of inliers, as shown in Figure 4.1(b). This is statistically true since using more data will result in a better estimate. However, such a strategy is not recommended for the conventional sampling techniques since the probability of obtaining clean subsets decreases exponentially with the increase in the subset size, which entails a greater sampling effort to generate good hypotheses.
In this chapter, we propose an unconventional sampling method, which generates hypotheses based on larger-than-minimal data subsets without the exponential growth in sampling effort required to retrieve large subsets of inliers. Specifically, we produce model hypotheses fitted on clusters of data sampled using Random Cluster Models (RCM), which follows from the Swendsen-Wang method (Swendsen and Wang, 1987) widely used in statistical mechanics to simulate coupled spin systems. The RCM, leveraging on the inherent spatial smoothness of the data, partitions the data into a number of arbitrary-sized subsets (clusters). In effect, all the members of a subset are sampled jointly, hence the probability of gathering all inliers in a subset is not decreasing exponentially with the subset sizes. Figures 4.2(b) and 4.3(b) show examples of random clusters on fitting multiple lines and homographies (Vincent and Laganière, 2001).

To capitalise on our efficient hypothesis generator, we propose a simple annealing method based on graph cuts to optimise the fitting. We follow the work of (Delong et al., 2010; Isack and Boykov, 2012) to formulate geometric fitting and model selection as an optimal labelling problem. We show how our RCM sampler can be integrated closely with graph cuts, such that hypothesis sampling and fitting optimisation are conducted alternately and continuously. Consequentially, the model parameters are continuously revised using the new arrived hypotheses, and in turn the RCM relies on the current labelling to generate new hypotheses. Figures 4.2 and 4.3 demonstrate our ideas.

To ease the discussion, we shall describe the overall fitting optimisation algorithm in
Chapter 4. Rapid Hypothesis Sampling using Random Cluster Models

Figure 4.2: A conceptual illustration of the proposed hypothesis generation method on a multiple line fitting problem. First, an adjacency graph from Delaunay triangulation is defined over the points. (a) An intermediate fitting $\Theta$ (red lines) and labelling $f$ (point colours) result. Points currently labeled as outliers are shown as black crosses. The lines were fitted inaccurately, and another line was not fitted, resulting in a wrongful $f$. (b) Clusters are randomly sampled from the RCM which is conditioned on the current $f$. Note that clusters that contain points with (currently) different labels are forbidden (see Section 4.2). (c) Line hypotheses are fitted on the sampled clusters. Many of the hypotheses lie close to the correct lines, in contrast to the current fitted lines in (a). The hypotheses are then used to improve the current $\Theta$ and $f$ (see Section 4.1).

Figure 4.3: Demonstrating our RCM hypothesis generator on a multi-homography fitting problem (Vincent and Laganère, 2001) (the images show only one of the two views). (a) and (b) are analogous to (a) and (b) in Figure 4.2. Here, to improve legibility data currently labelled as outliers are shown as red crosses. In (a), both $\Theta$ and $f$ are inaccurate (the fitted structures straddle more than one true structure). In (b) many of the generated clusters will give better hypotheses. To encourage global exploration (see Section 4.2.2), we bias the selection of the hypotheses to favour clusters from relatively unexplained regions (the darker regions in (c)).

Section 4.1 before giving the technical details of the RCM hypothesis generator in Section 4.2. The advantages of the RCM over the traditional hypothesis sampling methods are demonstrated in Section 4.3. We summarise the chapter in Section 4.4. The main content in this chapter is based on the author’s publication in (Pham et al., 2012, 2014).

4.1 A Graph Cuts based Simulated Annealing Approach

In Chapter 3, we shown that combining hypothesis generation and model selection into a unified framework is generally more beneficial for robust multi-structure model fitting than performing the two steps disjointly. We developed a fitting method, namely ARJMC (see Section 3.2), which is based on a Reversible Jump Monte Carlo Markov Chain
simulated annealing algorithm. In this chapter, we propose another fitting algorithm, which also conducts hypothesis sampling and model selection jointly. The two algorithms inherently optimise the fitting criteria by performing a series of births, deaths and updates. Another common property is that they are both monitored by a tempering schedule. Nonetheless, the new algorithm, developed in this chapter, has a number of distinctions.

- The new algorithm simultaneously optimises the model parameters as well as the point assignment. The later is not considered in ARJMC. The model parameters are optimised using birth, death and update moves (similar to ARJMC) while the point assignment is optimised using a graph cuts method.

- Different from ARJMC, the new algorithm can work with any hypothesis samplers, although it performs best with the RCM sampler. Note that the RCM sampler cannot be deployed in ARJMC since the probability of proposing a specific random cluster of data cannot be computed precisely.

- The new algorithm is not restricted to the choices of hypothesis samplers. However, the downside is that its simulated chain is not guaranteed to be aperiodic and irreducible, thus the theoretical convergence property as in ARJMC is not available. Nevertheless, the new states are still accepted probabilistically, and the acceptance probabilities are controlled by a simulated tempering such that local optima can be avoided.

In the subsequent sections, we explain the details of the cost function and optimisation algorithm.

### 4.1.1 Energy Function

To jointly optimise the model parameter and point assignment, we define the energy function for multi-structure fitting as

$$E(\Theta, f) = \sum_{n=1}^{N} D(x_n, f_n) + w_1 \sum_{(i,j) \in N} V(f_i, f_j) + w_2 C(\Theta),$$  \hspace{1cm} (4.1)$$

where \( f = [f_n]_{n=1}^{N} \) is a vector of labels that assign each \( x_n \) to one of the structures in \( \Theta \), or designate \( x_n \) as an outlier; weights \( w_1 \) and \( w_2 \) scale the strengths of smoothness constraints and model complexity.
The data cost \( D(x_n, f_n) \) is constructed as
\[
D(x_n, f_n) = \begin{cases}
  r(x_n, \theta_{f_n})^2 & \text{if } f_n \in \{1, \ldots, K\}, \\
  \sigma^2 & \text{if } f_n = 0,
\end{cases}
\]
where \( r(x_n, \theta_{f_n}) \) is the residual of \( x_n \) to structure \( \theta_{f_n} \), while \( \sigma \) is the penalty for labeling \( x_n \) as an outlier.

The function \( V(f_i, f_j) \) imposes the spatial smoothness coherence, which derives from the Potts Model
\[
V(f_i, f_j) = \begin{cases}
  0 & \text{if } f_i = f_j, \\
  1 & \text{if } f_i \neq f_j.
\end{cases}
\]

In (4.1) a neighbourhood graph \( G = (V, N) \) is required. The vertices are simply \( V = \mathcal{X} \), and we follow (Delong et al., 2010; Isack and Boykov, 2012) by constructing \( N \) from the Delaunay Triangulation of \( \mathcal{X} \), hence drawing edges \((i, j) \in N\) between spatially close data. Figure 4.3(a) shows an example from homography detection where each data point \( x_i = (x_1^i, x_2^i) \) is a point in \( \mathbb{R}^4 \), thus the adjacency graph is also constructed in \( \mathbb{R}^4 \). To ease visualisation, the adjacency graph is shown on the feature positions \( \{x_n^1\}_{n=1}^N \) in the first image only.

The model complexity cost \( C(\Theta) \), which penalizes the solutions with many structures, simply counts the number of parameters in \( \Theta \).

### 4.1.2 Energy Optimisation

Algorithm 4.1 details our simulated annealing approach to minimising the energy function (4.1). At each iteration, a new state with model parameters \( \Theta \) and labels \( f \) are proposed, then the proposal is accepted probabilistically such that local minima may be avoided. Note that the acceptance probability depends on the temperature \( T \), which is continuously annealed. By design, our approach conducts hypothesis sampling, model selection and point assignment alternately and continuously. This allows more effective and global exploration of the solution space. The subsequent subsections explain how to generate the potential proposals.

#### 4.1.2.1 Model Selection Optimisation

To optimise the parameters \( \Theta \) our algorithm performs a series of births (sampling and adding a new structure to \( \Theta \)) and deaths (deleting a structure from \( \Theta \)) to explore
Algorithm 4.1: Graph cuts based annealing optimisation for geometric fitting.

Input: Initialize temperature $T$ and structures $\Theta$, obtain labels $f$ via $\alpha$-expansion.

1: Sample $u \sim U_{[0,1]}$.
2: if $u < 0.5$ then
3: Sample a new hypothesis $\theta$ (e.g., using the RCM sampler see Section 4.2).
4: $\Theta' := \Theta \cup \theta$.
5: else
6: Randomly select one $\theta$ in $\Theta$.
7: If $|\Theta| > 1$ then $\Theta' := \Theta \setminus \theta$.
8: end if
9: Obtain labels $f'$ of $\Theta'$ via $\alpha$-expansion (graph cuts).
10: if $E(\Theta', f') < E(\Theta, f)$ then
11: $\Theta := \Theta'$, $f := f'$.
12: else
13: With probability $e^{\frac{E(\Theta, f) - E(\Theta', f')}{T}}$, $\Theta := \Theta'$, $f := f'$.
14: end if
15: If $\Theta$ changed then refine parameters in $\Theta$.
16: $T := 0.99T$ and repeat from Step 1 until $T \approx 0$.

solutions of different complexity. While the death moves (Line 6 and 7) are effortless, the sampling of new hypotheses in the birth step (Line 3) is expensive and decides the performance of Algorithm 4.1. The quicker good hypotheses are sampled, the faster Algorithm 4.1 reaches the optimal $\Theta$. Since Algorithm 4.1 is agnostic to the choice of hypothesis samplers, previous techniques, as reviewed in Chapter 2, can be applied here. In Section 4.2 we propose a novel sampling method based on RCM that allows rapid minimisation of the energy $E(\Theta, f)$.

4.1.2.2 Label Optimisation

In the context of geometric fitting we are mainly interested in optimising the parameters $\Theta$ and not the labels $f$. The estimation of $f$ given $\Theta$ merely assigns the data to the structures in $\Theta$ — the labels $f$ are implicit variables needed to compute the smoothness cost $V(f_i, f_j)$. However for consistency of our annealing procedure $f$ must be deterministic given the parameters $\Theta$ so that the energy $E(\Theta, f)$ is uniquely and consistently defined. Also ideally $f$ should globally minimise the energy (4.1) given $\Theta$.

Given $\Theta$, computing the $f$ which globally minimises (4.1) is NP-hard (Boykov et al., 2001). Similar to (Zabih and Kolmogorov, 2004), we resort to $\alpha$-expansion (Boykov et al., 2001) which can very efficiently reach a local minimum. To ensure that the $\alpha$-expansion
algorithm uniquely returns a single solution $f$ given $\Theta = \{\theta_k\}_{k=1}^{K}$, we initialise
\[
 f_n^{(0)} = \begin{cases} 
 0 & \text{if } \min_k r(x_n, \theta_k) > \sigma, \\
 \arg\min_k r(x_n, \theta_k) & \text{otherwise},
\end{cases}
\]
and use $f^{(0)} = \{f_n^{(0)}\}_{n=1}^{N}$ as the initialiser for $\alpha$-expansion. We then take the outcome of $\alpha$-expansion in minimising (4.1) as the $f$ for fixed $\Theta$.

### 4.1.2.3 Local Refinement of Model Parameters

Line 15 in Algorithm 4.1 refines the parameters in $\Theta = \{\theta_k\}_{k=1}^{K}$ if the birth or death step was accepted. Given the labelling $f$ corresponding to $\Theta$, the set of data assigned to $\theta_k$ is $p_k = \{x_n | x_n \in \mathcal{X}, f_n = k\}$. Structure $\theta_k$ is then refined by re-estimating it as
\[
 \hat{\theta}_k = \arg\min_{\theta} \sum_{x \in p_k} r(x, \theta)^2,
\]
i.e., the least squares fit of the model using the data $p_c$. This local refinement step is partially analogous to (Chum et al., 2003). With $f$ fixed, the spatial coherence terms $V(f_i, f_j)$ are unchanged, and since the complexity of $\Theta$ is also unchanged, refining the parameters in this manner always reduces the data costs $D(x_n, f_n)$ and hence the objective value (4.1).

### 4.2 The Random Cluster Model

The key contribution in this chapter is an efficient hypothesis generator based on RCM (MacKay, 2003). The model follows from the Swendsen-Wang (SW) method for simulating coupled spin systems (Swendsen and Wang, 1987). Given an adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{N})$, the Potts Model is
\[
 P(f) = \frac{1}{Z} \prod_{(i,j) \in \mathcal{N}} g(f_i, f_j),
\]
where $Z$ is the partition function, and the factor $g(f_i, f_j) = \exp(\beta_{ij}1(f_i = f_j))$; $1(\cdot)$ returns 1 if its argument is true and 0 otherwise. Positive coefficients $\{\beta_{ij}\}$ favor large clusters of the same labels. The SW method introduces binary “bond” variables $d = \{d_{ij}\}$ such that
\[
 P(f, d) = \frac{1}{Z'} \prod_{(i,j) \in \mathcal{N}} g'(f_i, f_j, d_{ij}),
\]
where the factor $g'$ is defined as

$$g'(f_i, f_j, d_{ij}) = \begin{cases} 
1 - w_{ij} & \text{if } d_{ij} = 0, \\
w_{ij} & \text{if } d_{ij} = 1 \text{ and } f_i = f_j, \\
0 & \text{if } d_{ij} = 1 \text{ and } f_i \neq f_j.
\end{cases}$$

$w_{ij} = 1 - \exp(-\beta_{ij})$ is the edge probability which indicates how likely $x_i$ and $x_j$ are from the same structure. Marginalising $d$ in (4.6) returns the Potts Model $P(f)$, while marginalising $f$ yields $P(d)$ or the Random Cluster Model.

The following section describes hypothesis generation using the RCM, while Section 4.2.4 shows how the edge probabilities $\{w_{ij}\}$ can be learned online.

### 4.2.1 Sampling Larger-than-minimal Subsets

A realisation of $d$ divides the graph $G$ into a set of connected components or clusters. Vertices in the same component are interconnected by edges with bond value 1. A component is a cluster of spatially close data likely to be from the same structure. Here we explain how $d$ can be sampled from the RCM.

In the SW method, to simulate $f$, $P(d|f)$ and $P(f|d)$ are sampled alternately. This allows large segments to change label simultaneously and is behind the success of SW (MacKay, 2003). In a similar vein, we also obtain $d$ and $f$ alternately; however, while $d$ is sampled from $P(d|f)$, $f$ is obtained deterministically from the current structures $\Theta$ (Steps 3 and 9 in Algorithm 4.1, respectively). Conditioned on $f$, the bond variables are mutually independent and are sampled according to

$$P(d_{ij} = 1|f_i = f_j) = w_{ij},$$

$$P(d_{ij} = 1|f_i \neq f_j) = 0,$$

i.e., a bond straddling two vertices is turned on probabilistically if the vertices have the same label, otherwise the bond is switched off deterministically. For a sparse adjacency graph this process scales linearly with the number of vertices $N$.

Note that the introduction of a neighbourhood structure $G$ and the sampling of bond variables $d$ allows the proposed subset sampler to choose all the members in a large data subset simultaneously. This is contrary to the traditional strategy which samples the members in the subsets individually. Consequently our RCM method can avoid exponentially decreasing probabilities of sampling clean subsets. In Section 4.2.4, we
show that the probabilities of sampling clean subsets by our method is progressively increased by continuously learning the RCM.

Figures 4.3(a) and 4.3(b) depict one iteration of this procedure, where Figure 4.3(a) shows the $\mathbf{f}$ corresponding to an intermediate $\Theta$, and Figure 4.3(b) illustrates the clusters due to a particular sample of $d$ conditioned on the $\mathbf{f}$. Observe that many large random clusters useful for hypothesis generation are obtained in Figure 4.3(b).

### 4.2.2 Hypothesis Generation using the RCM

To sample a hypothesis, we first discard the clusters induced by $d$ with insufficient number of data for fitting the geometric model.

Among the remaining clusters, one can be randomly chosen to produce a hypothesis. While random selection is simple, it may not be the most efficient, since some of the hypotheses may correspond to the same structure. Here, we propose to bias the selection towards unexplained regions in the data domain. Specifically, a cluster $S \subset \mathcal{X}$ is chosen based on

$$w_S := \frac{1}{|S|} \sum_{x_n \in S} D(x_n, f_n),$$

i.e., the average fitting error, where $|S|$ is the size of the cluster, and $f_n$ is the label of the $n$-th datum given the current $\Theta$. Figure 4.3(c) illustrates an example of an exploration map corresponding to an intermediate $\Theta$, where darker regions, which are not well fitted by $\Theta$, are more likely to be sampled next.

Figure 4.4 qualitatively compares the hypotheses generated progressively using RCM, and the hypotheses generated from minimal subset sampling.

### 4.2.3 Validity of Spatial Smoothness Assumption

Underlying the imposed neighbourhood structure $\mathcal{G}$ is the assumption of spatial smoothness, i.e., inliers from the same structure tend to be spatially close. This may not be generally true, since the inliers “lying” on the same geometric structure can be arbitrarily far apart (e.g., a line with two widely separated line segments). Nonetheless it can be argued that for many practical applications, spatial smoothness is an inherent characteristic, e.g., features on the same wall cluster in the same region, features on a moving object are concentrated within the object boundary. Moreover, on such applications the inclusion of spatial smoothness constraints improves the fitting accuracy tremendously, as observed elsewhere (Delong et al., 2010; Isack and Boykov, 2012; Yu et al., 2011).
Chapter 4. Rapid Hypothesis Sampling using Random Cluster Models

Figure 4.4: Qualitative comparison of hypotheses progressively generated using RANSAC, Multi-GS and RCM on a multiple line fitting problem. It is evident that RCM is more capable of producing more relevant hypotheses.

4.2.4 Online Learning of The RCM

Learning the RCM amounts to learning the edge probabilities \( \{w_{ij}\} \). If two neighbouring points \( x_i \) and \( x_j \) are from the same structure, then \( w_{ij} \) should be high to encourage them to be selected into the same cluster for hypothesis generation. In effect the cluster distribution encoded in the RCM serves as a proxy for the parameter distribution \( P(\theta) \) over \( \Omega \), and we incrementally refine it as the fitting progresses.

To learn \( \{w_{ij}\} \) we adopt the idea that the preferences of inliers from the same structure towards a set of putative hypotheses are correlated, as mentioned in Chapter 3 (see Section 3.3.1). Specifically, let \( \mathcal{H} = \{\theta_m\}_{m=1}^M \) contain the history of all hypotheses generated thus far in Algorithm 4.1 (including rejected ones). Let \( \mathbf{r}^{(i)} = [r_1^{(i)} \ r_2^{(i)} \ \cdots \ r_M^{(i)}] \)
be a vector containing the absolute residuals of a datum $x_i$ to the hypotheses in $H$, the “preference” of $x_i$ towards $H$ is the permutation $a^{(i)} = [a_1^{(i)} \ a_2^{(i)} \ \cdots \ a_M^{(i)}]$ which sorts $r^{(i)}$ in increasing order. The edge probability $\{w_{ij}\}$ is computed as

$$w_{ij} = \frac{1}{h} \left| a^{(i)}_h \cap a^{(j)}_h \right|,$$

where $\left| a^{(i)}_h \cap a^{(j)}_h \right|$ is the number of identical elements shared by the first-$h$ elements of $a^{(i)}$ and $a^{(j)}$; $h = [0.1M]$. It is clear that $0 \leq w_{ij} \leq 1$ and $\{w_{ij}\}$ are adaptively updated as the sampling progresses.

To compute $\{w_{ij}\}$ in an online manner, first the preferences $a^{(i)}$ for all $x_i$ need to be adjusted as a new hypothesis $\theta$ is sampled; this can be done efficiently by maintaining a sorted list $\tilde{r}^{(i)}$ of the residuals $r^{(i)}$, and updating the list with the new residual to $\theta$ via insertion sort. However computing the intersection (4.8) naively may cause deterioration in performance, since $h$ grows as the hypothesis set $H$ expands, i.e., updating the RCM becomes progressively expensive. To alleviate this issue, we approximate the intersection $\left| a^{(i)}_h \cap a^{(j)}_h \right|$ by a “mutual inclusion” counter $I_{ij}$, which is initialised to 0. Let $H$ contain the current $M$ hypotheses. If a newly added hypothesis $\theta$ is inserted into the top $[0.1(M+1)]$ preferences of both $x_i$ and $x_j$, we increment $I_{ij}$ by 1, and obtain $w_{ij} = I_{ij}/[0.1(M+1)]$. The process is performed for other neighbours, before $\theta$ is inserted into $H$. Algorithm 4.2 outlines the procedure.

**Algorithm 4.2: Updating Edge Probabilities in RCM**

**Input:** Hypotheses $H$, sorted residual list $\{\tilde{r}^{(i)}\}$, edge probabilities $\{w_{ij}\}$, mutual inclusion counters $\{I_{ij}\}$, new hypothesis $\theta$.

1: for $i = 1, \ldots, N$ do
2: Compute residual $r(x_i, \theta)$.
3: Insert $r(x_i, \theta)$ into $\tilde{r}^{(i)}$.
4: if position of $r(x_i, \theta) \leq [0.1(|H| + 1)]$ then
5: $I_i := 1$.
6: else
7: $I_i := 0$.
8: end if
9: end for
10: for all $(i, j)$ in $N$ do
11: if $I_{ij} > [0.1(|H| + 1)]$ then
12: $I_{ij} := [0.1(|H| + 1)]$
13: end if
14: $w_{ij} := I_{ij}/[0.1(|H| + 1)]$
15: end for
16: $H := H \cup \theta$. 

Figure 4.5: Matrix of edge probabilities for the multiple line fitting task using the data in Figure 4.4 (containing 5 genuine structures) updated using the intersection function (4.8) naively (top row) and Algorithm 4.2 (bottom row).

Note that the counter $I_{ij}$ monotonically increases with $|\mathcal{H}|$, thus it is possible for $I_{ij} > 0.1(|\mathcal{H}| + 1)$ (in practice, however, the growth of $0.1(|\mathcal{H}| + 1)$ usually far outstrips the increase of $I_{ij}$). To ensure that $w_{ij}$ is within $[0, 1]$, in Line 13 the counter $I_{ij}$ is truncated to $0.1(|\mathcal{H}| + 1)$ (if necessary) before computing $w_{ij}$.

Figure 4.5 compares the $\{w_{ij}\}$ updated using (4.8) naively and Algorithm 4.2. Observe that as more hypotheses are sampled, both methods yield similar patterns. In Figure 4.5, the $\{w_{ij}\}$ are computed for all pairs of data for illustration only; in actual operations we only need to compute $\{w_{ij}\}$ between adjacent data. Also observe that the changes in the weights $\{w_{ij}\}$ become less and less significant when the number of hypotheses is excessively large. This property has been observed previously in Chapter 3. Technically, $\lim_{|\mathcal{H}| \to \infty} |w'_{ij} - w_{ij}| = 0$, where $w'_{ij}$ is the updated weight after a new $\theta$ is added. (See Section 3.3.3.2 for the proof.) Intuitively, the gain in updating $w_{ij}$ diminishes as the sampling progresses. Hence we need not sample a large number of hypotheses to enlarge the hypothesis set $\mathcal{H}$.

4.2.5 The Swendsen-Wang Method for Label Optimisation

The application of the SW method is preceded by Barbu et al. (Barbu and Zhu, 2003, 2005), who proposed “SW cuts” to simulate arbitrary Markov Random Field (MRF) distributions under MCMC. The method was proven to be effective on various data labelling problems (e.g., superpixel labelling). Theoretically, SW cuts can be used to optimise our energy function (4.1). However, our empirical study shows that the label inference using MCMC with SW is generally slower than $\alpha$-expansion (see Section 4.1.2.2). The
primary reason is that the MCMC-SW method requires accurate data-driven edge probabilities (Barbu and Zhu, 2003, 2005) to obtain good label proposals. Unfortunately, in our multi-structure fitting problem the edge probabilities are learnt online using Algorithm 4.2, which is not accurate at the early stages (see Figure 4.5). Thus using SW cuts for optimising the geometric fitting energy (4.1) is less efficient than using Algorithm 4.1.

4.3 Experiments

4.3.1 Larger-than-minimal Subsets vs. Minimal Subsets

In this section we demonstrate the practical advantages of the RCM over the previous hypothesis sampling methods in fitting optimisation. Those are pure random sampling, pure spatial sampling (i.e., sampling data that are spatially close, without regard to intermediate labelling results $f$; see Algorithm 2.6), as well as the Multi-GS algorithm (Chin et al., 2012) which is the state-of-the-art guided sampling method. Note that except the RCM, other methods are for sampling minimal subsets.

We design the benchmarking as follow. In Step 3 of Algorithm 4.1, we resort to the above-mentioned methods for sampling new hypotheses and compare their performances on the same data\footnote{“Raglan Castle” from http://www.robots.ox.ac.uk/~vgg/data/} in Figure 4.6(a) for the task of multiple homography detection. More details about the multiple homography estimation problem can be found in Chapter 5. Also for correctly assessing the actual performances of the different hypothesis generators we omit local refinement (Line 15, Algorithm 4.1) and global exploration (see Equation (4.7)) for all the samplers.

Figure 4.7 depicts the evolution of the objective value (4.1) as Algorithm 4.1 was executed using the 4 hypothesis generators. It is clear that RCM provides the fastest minimisation. The differences in objective function value achieved at termination (after 30s) by the 4 samplers are also significant and meaningful. This suggests that given a fixed time budget, our hypothesis sampler RCM produces more accurate hypotheses than other samplers. Observe that only RCM in Figure 4.6(b) correctly partitioned the underlying planar structures (there are 11 visually detectable planes).

Figure 4.8 displays the cumulative probability of hitting all-inlier subsets as Algorithm 4.1 progresses. The cumulative probability is defined as the ratio of accumulated number of all-inlier subsets over the total number of subsets (hypotheses) sampled thus far. It is clear that the RCM provides the highest probability even though the RCM method samples larger-than-minimal subsets. This result explains the superior performance of
Rapid Hypothesis Sampling using Random Cluster Models

Figure 4.6: Multi-homography detection on Raglan Castle. (a) The ground truth labelling with 11 unique structures. Final labelling using (b) RCM, (c) random sampling, (d) pure spatial sampling and (e) Multi-GS. Outliers are marked as red crosses. It can be seen that expectedly using the purely random sampling results in the worst segmentation, i.e., some true planes are not detected while some estimated segments wrongly explain the true flat surfaces (e.g., see the purple segment in (c)). RCM, Multi-GS and the spatial sampling are equally good in segmenting the dominant planes, however Multi-GS and the spatial sampling fail to pick small ones. Moreover, some segments discovered by Multi-GS and the spatial sampling are not spatially consistent (e.g., see the orange segment in (d) and the purple and green segments in (e).

4.3.2 The RCM vs. Local Refinement and Global Exploration

We have shown the superior performance of the RCM over the other hypothesis sampling methods using minimal subsets. Here we further demonstrate the exceptional strength of the RCM by comparing the performance against the other methods equipped with local refinement (Section 4.1.2.3) and global exploration (Section 4.2.2).

The experiment settings are the same as the previous one, except that local refinement (Line 15, Algorithm 4.1) and global exploration are turned on for all the samplers, not including the RCM. To implement global exploration for random sampling, pure spatial
Chapter 4. Rapid Hypothesis Sampling using Random Cluster Models

Figure 4.7: This figure is best viewed in color. Evolution of objective function value by Algorithm 4.1 using different sampling schemes without using local parameter refinement (Section 4.1.2.3) and exploration of unexplained regions (Section 4.2.2).

Figure 4.8: This figure is best viewed in color. The probability of hitting all-inlier subsets for different sampling methods as Algorithm 4.1 progresses.

Sampling and Multi-GS, given the current \( \Theta \) and \( f \), the weight of each \( x_n \) is obtained as \( D(x_n, f_n) \); see Equation (4.2). Hence an \( x_n \) which is not well explained will have high weight and vice versa. The seed datum of the minimal subset is chosen based on the weights. The subsequent data are also sampled based on the weights (4.2) (for random sampling), the spatial distances to the seed datum (for spatial sampling) or using the preference correlations (4.8) with the seed datum (for Multi-GS, refer to Algorithm 2.7 and Section 3.3.1).

Figure 4.9 illustrates the results, where it is clear that local refinement and global exploration are generally beneficial regardless of the hypothesis generators used. More importantly, observe that the raw RCM without using local refinement and global exploration is still faster than other minimal subset samplers equipped with the two functionalities.
The results demonstrate that hypotheses fitted on larger-than-minimal subsets is generally more accurate than using local refinement to improve the bad hypotheses fitted on minimal subsets.

Finally we investigate the maximal advantages of full Algorithm 4.1 (local refinement and global exploration are on) with the RCM sampler together. The evolution of the objective function values is plotted in the same Figure 4.9 for comparisons. It can be seen that the fitting efficiency significantly improves. Note that iterative and online refinement and exploration are not feasible if hypothesis sampling and fitting optimisation are executed disjointly.

A complete implementation of Algorithm 4.1 using the RCM hypothesis samplers contributes a novel robust fitting method namely RCMSA. In Chapter 5, we shall demonstrate the performance of RCMSA on a number of geometric vision applications, and also benchmark against the state-of-the-art fitting methods.

4.4 Summary

In this chapter, we have revisited the known issue related to hypothesis generation using minimal subsets in robust parameter estimation. The issue comes from the incorrect assumption that a hypothesis fitted on an outlier-free minimal subset of data approximates well the genuine structure. However in practice, due to the effects of measurement noise, hypotheses fitted on “clean” minimal subsets may be arbitrary far from the true structures. Increasing the size of data subsets could resolve the problem, however, this is not
recommended by the traditional sampling techniques since the probability of choosing all inliers in a sample will exponentially decreases (Meer, 2004).

The work presented in this chapter showed a possibility to sample larger-than-minimal subsets without decreasing the probability of hitting all-inlier subsets. We proposed an approach based on RCM to sample clean clusters of the data directly. Particularly, instead of sampling the members in a subset individually, we probabilistically partition a neighbourhood graph (defined over the data points) into disjoint clusters. The graph (i.e., the edge weights) is learnt incrementally so that more relevant clusters (i.e., the ones contain only inliers) can be efficiently harvested.

This chapter also presented a novel geometric fitting algorithm which effectively combines hypothesis generation and label optimisation. The algorithm is another realisation of the continuous hypothesise-and-select approach discussed in Chapter 1. Note that the first realisation based on RJMCMC was presented in Chapter 3. We showed that the algorithm maximises its optimisation speed when implemented with the RCM sampler (denoted as RCMSA) compared to other minimal subset samplers.

In the next chapter, we will benchmark the performance of the RCMSA estimator against state-of-the-art estimation methods via various two-view geometry applications. We will also test the capability of RCMSA by applying to a high-level automatic augmented reality application.
Chapter 5

Robust Two-view Geometry Estimation and Applications

Two-view geometry is an important area in computer vision which is essential towards understanding of the 3D world. This chapter presents a number of interesting applications such as: two-view motion segmentation, multiple homography detection and augmented reality, which are rooted in two-view geometry. These applications essentially require a robust technique to estimate geometric relations between two views. Using such applications we aim to demonstrate the capabilities of our two novel robust parameter estimation methods: namely ARJMC and RCMSA (previously proposed in Chapters 3 and 4 separately). We also benchmark our proposed approaches against state-of-the-art fitting methods.

The outline of this chapter is as follows. Section 5.1 reviews a number of important concepts in two-view geometry, which are central to the applications considered in this chapter. Sections 5.2 and 5.3 present comprehensive experimental studies on motion segmentation and homography detection, respectively. We then demonstrate an automatic augmented reality application using a multi-structure robust estimation approach in Section 5.4. Section 5.5 summarises the chapter.

5.1 Two-view Geometry

Two-view geometry studies geometric relations between two different views of the same scene, from which we can extract useful information about the 3D structure of the scene, camera motion and/or scene objects’ motions. Such relations are embedded in the differences between the two images, which are caused by the movements of the camera.
or the changes in the positions of any foreground objects in the scene. Depending on the characteristics of the scene and camera’s motion, the geometric relations between these images can be captured by different types of transformations, such as fundamental matrix or homography matrix. These transformations impose the constraints on the positions of every pair of “matched” points — points (one from each image) that are the images of the same 3D point. Each pair of matched points is called a correspondence.

In the rest of this section, we briefly review two frequently encountered image-to-image transformations, those described by a fundamental matrix and a homograph matrix; which are essential to the applications considered in Sections 5.2 and 5.3. Detailed definitions and derivations of these transformations can be found in (Hartley and Zisserman, 2004).

5.1.1 Fundamental Matrix

Consider two different images $I_1$ and $I_2$ of the same static scene, taken from two different view points. Let $\mathbf{x}^1 = (x^1, y^1)$ and $\mathbf{x}^2 = (x^2, y^2)$ be the positions of a 3D point $\mathbf{X}$ after being projected on $I_1$ and $I_2$ respectively, there exists a fundamental matrix of rank 2 $\mathbf{F} \in \mathbb{R}^{3 \times 3}$ that imposes a (bi)linear constraint on $\mathbf{x}^1$ and $\mathbf{x}^2$. That is

$$(\tilde{\mathbf{x}}^2)^\top \mathbf{F} \tilde{\mathbf{x}}^1 = 0,$$  \hspace{1cm} (5.1)$$

where $\tilde{\mathbf{x}}^1 = (x^1, y^1, 1)$ and $\tilde{\mathbf{x}}^2 = (x^2, y^2, 1)$ are the homogeneous coordinates of $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively. Figure 5.1 illustrates this geometric constraint.

Geometrically, the fundamental matrix $\mathbf{F}$ will map a point $\mathbf{x}^1$ in the first image $I_1$ to a line $l_2 = \mathbf{F} \tilde{\mathbf{x}}^1$ in the second image $I_2$, which passes through the corresponding point $\mathbf{x}^2$. Similarly, $l_1 = \mathbf{F}^\top \tilde{\mathbf{x}}^2$ defines a line in image $I_1$ which goes through the image point $\mathbf{x}^1$. 

![Figure 5.1: Two-view geometry and fundamental matrix. Two cameras locating at $V_1$ and $V_2$ capture a 3D point $\mathbf{X}$. The images of $\mathbf{X}$ projected on the first and second image are $\mathbf{x}^1$ and $\mathbf{x}^2$ respectively. The positions of $\mathbf{x}^1$ and $\mathbf{x}^2$ are constrained by a fundamental matrix $\mathbf{F}$, in that $\mathbf{x}^2$ must lie on the epipolar line $l_2$ (red line) which is mapped from $\mathbf{x}^1$ via the transformation $\mathbf{F}$.](image-url)
5.1.2 Multiple Fundamental Matrices for Dynamic Scenes

If the scene contains \( K \) dynamic foreground objects, such as moving cars, the geometric constraints between the two images are a bit more complicated. Assume that the objects are moving independently in different directions and the camera is also moving between the two views. Firstly, let’s consider the correspondences belonging to the static background. Apparently, every correspondence on the background is constrained via a fundamental matrix \( F_b \), which is induced by the movement of the camera similar to the standard static scenes. Likewise, the movement of the camera relative to the static background can be viewed as the “movement of the background” relative to a fixed-position camera. Viewing the problem by this way, any dynamic object has its own motion relative to the camera. As a result, we could associate each moving object with a fundamental matrix \( F_o \). Therefore, the two images are now constrained by \( K + 1 \) fundamental matrices, one for each dynamic foreground objects and one for background. This scenario has been widely studied in the field of multi-body structure from motion (Vidal et al., 2006; Schindler and Suter, 2006; Ozden et al., 2010).

5.1.3 Homography Matrix

Now suppose that the 3D point \( X \) lies on a flat surface in the scene, there exists a stronger geometric constraint on the positions of \( x^1 \) and \( x^2 \). Specifically, there is a bijective transformation that uniquely maps \( x^1 \) in the first image \( I_1 \) to \( x^2 \) in the second image \( I_2 \). The transformation, also known as the plane projective transformation, can be described by a homography matrix \( H \in \mathbb{R}^{3 \times 3} \). That is

\[
\tilde{x}^2 \simeq H \tilde{x}^1,
\]
where \( \simeq \) denotes equality up to scale. To obtain the location \( x^2 \) in \( I_2 \), we simply transform \( \tilde{x}^2 \) from homogeneous to Cartesian coordinates. Similarly, \( H^{-1} \tilde{x}^2 \) results in the position \( x^1 \) in the first image \( I_1 \). Figure 5.2 illustrates this special geometric constraint.

### 5.1.4 Multiple Homography Matrices

If the scene contains multiple planar surfaces such as building walls, the two images can be related by multiple homography matrices, each of which is assigned to a unique plane in the scene. For instance, let \( X_1 \) and \( X_2 \) be two 3D points lying on two separate planes \( \pi_1 \) and \( \pi_2 \), \( X_1 \) is projected onto the image \( I_1 \) at \( x^1_1 \) and image \( I_2 \) at \( x^2_1 \), while \( x^1_2 \) and \( x^2_2 \) is the projection of \( X_2 \) onto \( I_1 \) and \( I_2 \) respectively. Then, there exist two distinct homography matrices \( H_1 \) and \( H_2 \) such that

\[
\tilde{x}^2_1 \simeq H_1 \tilde{x}^1_1, \tag{5.3}
\]

and

\[
\tilde{x}^2_2 \simeq H_2 \tilde{x}^1_2. \tag{5.4}
\]

### 5.1.5 Estimating Image-to-image Transformations

The task of two-view geometry estimation becomes that of estimating the image-to-image transformations. Since the transformations constrain the positions of the correspondences between the images, we can estimate these transformation parameters based on a set of established correspondences.

In practice, the correspondences between two images can be obtained by first extracting the features (e.g., corners) from each image, then matching them across the two images. Intuitively, similar features in the two images are expected to correspond to the same 3D point. Unfortunately, the matching step will unavoidably result in a number of wrong correspondences (i.e., pairs of matched points that are not corresponding to the same 3D points). Figure 5.3 shows an example of matching between two views. Consequently, it is necessary to resort to robust techniques for such estimation tasks.

In the next Sections 5.2 and 5.3, we shall discuss two practical applications: motion segmentation and multiple homography detection, which essentially require estimating multiple fundamental matrices and homography matrices from correspondences. We apply our proposed estimation methods, discussed in Chapter 3 and 4, and state-of-the-art robust methods to these two applications for performance evaluation.
5.2 Two-view Motion Segmentation

5.2.1 Problem Statement

Consider two views of a dynamic scene where a number of objects have moved between the views. Let there be $N$ correspondences $\{x_n\}_{n=1}^N$, where each $x_n = (x_1^n, x_2^n)$ contains matched features. The problem of motion segmentation is to segment the matches into a number of distinct motions, as well as to identify the false matches not belonging to any motion. The number of motions must also be estimated.

As discussed in Section 5.1.2, each motion (structure) can be modelled by a fundamental matrix $F \in \mathbb{R}^{3 \times 3}$ which describes the epipolar constraint $(\tilde{x}_2^2)^\top F \tilde{x}_1^n = 0$ on the matches arising from that motion ($\tilde{x}^1$ and $\tilde{x}^2$ are $x^1$ and $x^2$ in homogeneous coordinates). Therefore, the task becomes fitting a set parameters $\Theta = \{F_k\}_{k=1}^K$ ($K$ is unknown) onto $N$ observations $\{x_n\}_{n=1}^N$, then using the estimated $\{F_k\}_{k=1}^K$ to classify the matches.
The distance from a match $x_n$ to a fundamental matrix $F$, denoted as $r(x_n, F)$, is computed as the Sampson distance (Hartley and Zisserman, 2004), i.e.,

$$r(x, F) = \frac{(\tilde{\mathbf{x}}^2)^{\top} F \tilde{\mathbf{x}}^1}{(F \tilde{\mathbf{x}}^1)^2 + (F \tilde{\mathbf{x}}^1)^2 + (F^{\top} \tilde{\mathbf{x}}^2)^2 + (F^{\top} \tilde{\mathbf{x}}^2)^2},$$  \hspace{1cm} (5.5)$$

where $(F \tilde{\mathbf{x}}^1)^2$ is the squared of the $i$-th entry of the vector $F \tilde{\mathbf{x}}^1$. We use DLT method (Hartley and Zisserman, 2004) to instantiate $F$ from a minimum of 8 matches (the rank of each estimated $F$ is also enforced to be 2).

### 5.2.2 Estimation Methods

To tackle the above motion segmentation problem, we consider the following optimisation based robust fitting methods:

- **ARJMC** is our first proposed method presented in Chapter 3.
- **RCMSA** is our second proposed method presented in Chapter 4.
- **PEARL** (Isack and Boykov, 2012) casts the robust estimation as an optimal labeling task, then uses the $\alpha$-expansion (Boykov et al., 2001) algorithm to optimise the labeling. (See Section 2.2.5 for a review.)
- **FLOSS** (Lazic et al., 2009) formulates the robust estimation as a facility location problem and solves via a message passing algorithm. (See Section 2.2.4 for a review.)
- **QP-MF** (Yu et al., 2011) solves the robust estimation using a quadratic programming. (See Section 2.2.6 for a review.)

We do not include older optimisation based fitting methods with known problems such as lower computational efficiency (Li, 2007), lower accuracy due to greedy search (Schindler and Suter, 2006), and an inability to naturally handle outliers (Jian and Chen, 2007).

### 5.2.3 Experimental Design

#### 5.2.3.1 Datasets

In this experiment, we use image pairs from the AdelaideRMF dataset (Wong et al., 2011)\(^1\) where keypoints were detected and matched using SIFT (Lowe, 1999). Ground

\(^1\)http://cs.adelaide.edu.au/~hwong/doku.php?id=data
truth segmentation is also provided. We use only image pairs with 3 or more unique motions to better differentiate the methods.

5.2.3.2 Benchmarking

Before designing an objective benchmarking, we firstly revisit the distinct properties of all the considered methods. As characterized in Chapter 3 and 4, our proposed methods ARJMC and RCMSA conduct hypothesis sampling and fitting optimisation alternatingly and continuously. In contrast, the methods PEARL, FLOSS and QP-MF take a hypothesize-then-select approach — firstly a large set of hypotheses are generated, then the best subset of hypotheses are selected by optimising a fitting criterion. Clearly, the size of the hypothesis set affects the fitting efficiency and accuracy. To enable an objective comparison the two-stage methods are conducted as follows: Sampling is conducted to produce hypothesis sets $\mathcal{H}$ of sizes $M = 100, 200, \ldots, 1000$. We use Multi-GS (Chin et al., 2010b), which is one of the most efficient algorithm available, to generate hypotheses. For each hypothesis set the fitting/optimization is performed until termination, and we record the fitting error (the precise metric will be defined below) and time elapsed inclusive of the sampling duration (note: for each $M$ we tune the competitors to return the correct number of structures). For ARJMC and RCMSA, the fitting error and duration are simply recorded at iterations 10, 20, $\ldots$, 1000.

5.2.3.3 Error Metric

Since the methods optimise different fitting criteria, we compare them using segmentation error which penalises wrongful assignments as outliers/inliers and incorrect groupings of inliers:

$$SE(f) = \min_{\Gamma} \frac{1}{N} \sum_{n=1}^{N} \delta(f_{n}^{\Gamma} \neq f_{n}^{*}),$$

(5.6)

where $f^{*}$ are the true labels, $\delta(\cdot)$ is the delta function, and $\Gamma$ is a permutation on $f$ which leaves the zero elements unchanged, while the non-zero elements have their values permuted. Function (5.6) also penalises $f$ which contains the wrong number of structures (which may happen at the early stages of ARJMC and RCMSA).

Since by design the RCMSA and PEARL methods simultaneously optimise the model parameters $\Theta$ and labelling $f$, to be objective we perform $\alpha$-expansion (see Section 4.1.2.2, Chapter 4) on the optimised $\Theta$ of all other methods to obtain the best labelling $f$ for the error metric before evaluating (5.6).
To examine the repeatability of the results, we perform 50 runs for all methods on each dataset. To evaluate a particular run, we record the lowest segmentation error achieved within the run, and the time at which that error was achieved. We then take the median of these values over the 50 runs.

5.2.4 Results

Table 5.1 shows the comparison results. It can be seen clearly that RCMSA gives the best results on almost all the cases. Indeed, in terms of accuracy (segmentation error), RCMSA is comparable with QP-MF, and considerably better than others. In terms of computational efficiency, RCMSA is the fastest in all the cases, and the improvements are very significant. RCMSA is approximately 12 times faster than PEARL, 22 times faster than QP-MF and FLOSS, and 3 times faster than ARJMC. It is also important to notice the considerable improvement of ARJMC in saving computational cost over the other methods such as PEAL, FLOSS and QP-MF. ARJMC is about 7 times faster than FLOSS and QP-MF.

The efficiency improvements of RCMSA and ARJMC over the other methods (i.e., PEARL, QP-MF, FLOSS) can be explained by the differences in the optimisation strategy. While RCMSA and ARJMC inherently optimise the fitting criteria by continuously sampling hypotheses, the others conduct hypothesis sampling and fitting optimisation disjointly. As a result, RCMSA and ARJMC can minimise the wastage in the hypotheses generation and optimisation steps. Moreover, the fact that RCMSA generates hypothesis by using larger-than-minimal subsets further improves the efficiency because accurate hypotheses can be produced efficiently. Consequently, RCMSA is more efficient than ARJMC because ARJMC still uses minimal subsets to produce hypotheses.

Though the differences in fitting accuracy (segmentation error) between the methods is not significant, the methods imposing additional constraints (e.g., smoothness constraints) generally perform better. The result shows that PEARL, RCMSA and QP-MF are more accurate than FLOSS and ARJMC. In fact, whereas FLOSS and ARJMC only consider the goodness of fit and model complexity in their fitting criteria, PEARL and RCMSA further impose the pairwise smoothness constraints; QP-MF alternatively enforces the similarities between inliers of a structure.

Figure 5.4 depicts the evolution in minimizing SE within a 10s window for a subset of the image pairs. Note that some of the methods do not terminate within 10s. The graphs show that RCMSA and ARJMC converge to its lowest error in less than 1s while the others take much longer. Figures 5.9, 5.10, 5.11 and 5.12 provide qualitative comparisons on a subset of the datasets.
Table 5.1: Summary statistics (median) over 50 repetitions on fundamental matrices detection. For each run, we record the lowest SE achieved within 1000 iterations/hypotheses and the time when the lowest SE was achieved (SE in percent, time in seconds). On each dataset, the lowest and second lowest median SE among all methods are highlighted in blue and light blue respectively; the lowest and second lowest computing time among all methods are highlighted in yellow and light yellow respectively. S = number of structures; I = number of inliers; and O = number of outliers.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>PEARL</th>
<th>QP-MF</th>
<th>FLOSS</th>
<th>ARJMC</th>
<th>RCMSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biscuitbookbox</td>
<td>Seg. Error</td>
<td>8.11</td>
<td>7.72</td>
<td>11.58</td>
<td>11.58</td>
</tr>
<tr>
<td>S=3, I=162, O=97</td>
<td>Run Time</td>
<td>2.51</td>
<td>5.59</td>
<td>3.76</td>
<td>0.4</td>
</tr>
<tr>
<td>Boardgame</td>
<td>Seg. Error</td>
<td>16.85</td>
<td>17.2</td>
<td>17.92</td>
<td>19.35</td>
</tr>
<tr>
<td>S=3, I=166, O=113</td>
<td>Run Time</td>
<td>3.16</td>
<td>5.74</td>
<td>5.91</td>
<td>1</td>
</tr>
<tr>
<td>S=4, I=155, O=82</td>
<td>Run Time</td>
<td>2.84</td>
<td>6.51</td>
<td>5.29</td>
<td>1.35</td>
</tr>
<tr>
<td>Breadcubeschips</td>
<td>Seg. Error</td>
<td>9.57</td>
<td>6.96</td>
<td>11.74</td>
<td>10.43</td>
</tr>
<tr>
<td>S=3, I=149, O=81</td>
<td>Run Time</td>
<td>2.76</td>
<td>4.37</td>
<td>4.16</td>
<td>0.55</td>
</tr>
<tr>
<td>Breadtoyecar</td>
<td>Seg. Error</td>
<td>10.24</td>
<td>8.73</td>
<td>11.75</td>
<td>10.84</td>
</tr>
<tr>
<td>S=3, I=110, O=56</td>
<td>Run Time</td>
<td>1.94</td>
<td>4.93</td>
<td>3.54</td>
<td>0.85</td>
</tr>
<tr>
<td>Carchipscube</td>
<td>Seg. Error</td>
<td>10.3</td>
<td>9.09</td>
<td>16.97</td>
<td>15.76</td>
</tr>
<tr>
<td>S=3, I=105, O=60</td>
<td>Run Time</td>
<td>2.28</td>
<td>3.78</td>
<td>3.66</td>
<td>0.43</td>
</tr>
<tr>
<td>Cubebreadtoyschips</td>
<td>Seg. Error</td>
<td>9.02</td>
<td>7.34</td>
<td>11.31</td>
<td>9.94</td>
</tr>
<tr>
<td>S=4, I=239, O=88</td>
<td>Run Time</td>
<td>3.55</td>
<td>6.37</td>
<td>6.15</td>
<td>1.36</td>
</tr>
<tr>
<td>Dinabooks</td>
<td>Seg. Error</td>
<td>19.17</td>
<td>17.78</td>
<td>20.28</td>
<td>20.56</td>
</tr>
<tr>
<td>S=3, I=205, O=155</td>
<td>Run Time</td>
<td>3.77</td>
<td>4.77</td>
<td>7.19</td>
<td>1.32</td>
</tr>
<tr>
<td>Toycube2car</td>
<td>Seg. Error</td>
<td>12</td>
<td>10.5</td>
<td>13.75</td>
<td>13.5</td>
</tr>
<tr>
<td>S=3, I=128, O=72</td>
<td>Run Time</td>
<td>2.6</td>
<td>4.47</td>
<td>5.04</td>
<td>0.38</td>
</tr>
</tbody>
</table>

5.3 Two-view Multi-homography Detection

5.3.1 Problem Statement

Consider two views of a static scene which contains a number of planes. Given a set of point matches \( \{x_n\}^N_{n=1} \) across the two views, where \( x_n = (x^1_n, x^2_n) \). As discussed in Section 5.1.4, the positions of these point matches are constrained by a number of homography matrices \( \{H_k\}^K_{k=1} \). The goal of multi-homography detection is to recover the unknown \( \Theta = \{H_k\}^K_{k=1} \) (the quantity is also unknown), segment the points matches according to the homographies and identify the false matches.

\( H \) can be instantiated from at least 4 point matches using DLT method (Hartley and Zisserman, 2004). The residual \( r(x_n, H) \) is computed as the Sampson distance (Hartley and Zisserman, 2004).

The experimental settings including estimation methods, benchmarking and error metric are exactly the same as the previous motion segmentation problem.
5.3.2 Datasets

We test on image pairs from AdelaideRMF (Wong et al., 2011) and Oxford Visual Geometry Group\(^2\) (we chose Raglan Castle and Merton College 1 and 3 since these have 3 or more distinct homographies). On the Oxford images SIFT features were detected and matched. Ground truth segmentation was established manually.

5.3.3 Results

Table 5.2 summarizes the quantitative results. Expectedly, our methods, ARJMC and RCMSA, are again visibly the fastest ones, similar to the previous motion segmentation problem. Also it is clear that the gap in performance between our methods and the others is larger in homography fitting (cf. Figures 5.4 and 5.5). A primary factor is that the homography data contains a higher number of true structures (cf. Tables 5.1 and 5.2). Secondly, the number of matches \( \{x_n\}_{n=1}^N \) on the homography data also tends

\(^2\)http://www.robots.ox.ac.uk/~vgg/data/data-mview.html
Table 5.2: Summary statistics (median) over 50 repetitions on multi-homography detection. For each run, we record the lowest SE achieved within 1000 iterations/hypotheses and the time when the lowest SE was achieved (SE in percent, time in seconds). On each dataset, the lowest and second lowest median SE among all methods are highlighted in blue and light blue respectively; the lowest and second lowest computing time among all methods are highlighted in yellow and light yellow respectively. \( S \) = number of structures; \( I \) = number of inliers; and \( O \) = number of outliers.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>PEARL</th>
<th>QP-MF</th>
<th>FLOSS</th>
<th>ARJMC</th>
<th>RCMSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonython Hall</td>
<td>Seg. Error</td>
<td>18.63</td>
<td>14.42</td>
<td>15.68</td>
<td>13.34</td>
</tr>
<tr>
<td>S=6, I=1002, O=66</td>
<td>Run Time</td>
<td>18.3</td>
<td>42.52</td>
<td>31.81</td>
<td>5.6</td>
</tr>
<tr>
<td>Johnsona</td>
<td>Seg. Error</td>
<td>5.63</td>
<td>23.19</td>
<td>6.17</td>
<td>4.02</td>
</tr>
<tr>
<td>S=4, I=295, O=78</td>
<td>Run Time</td>
<td>7.61</td>
<td>19.6</td>
<td>13.71</td>
<td>2.38</td>
</tr>
<tr>
<td>Johnsonb</td>
<td>Seg. Error</td>
<td>18.64</td>
<td>23.73</td>
<td>15.49</td>
<td>15.18</td>
</tr>
<tr>
<td>S=7, I=571, O=78</td>
<td>Run Time</td>
<td>9.86</td>
<td>20.76</td>
<td>29.79</td>
<td>2.94</td>
</tr>
<tr>
<td>Ladysymon</td>
<td>Seg. Error</td>
<td>7.59</td>
<td>18.78</td>
<td>7.59</td>
<td>2.53</td>
</tr>
<tr>
<td>S=3, I=160, O=77</td>
<td>Run Time</td>
<td>3.92</td>
<td>19.41</td>
<td>6.55</td>
<td>1.54</td>
</tr>
<tr>
<td>Merton Colleges 1</td>
<td>Seg. Error</td>
<td>8.12</td>
<td>7.09</td>
<td>7.96</td>
<td>5.85</td>
</tr>
<tr>
<td>S=5, I=1503, O=437</td>
<td>Run Time</td>
<td>32.12</td>
<td>43.48</td>
<td>59.55</td>
<td>7.27</td>
</tr>
<tr>
<td>Merton Colleges 3</td>
<td>Seg. Error</td>
<td>6.53</td>
<td>8.65</td>
<td>5.95</td>
<td>4.39</td>
</tr>
<tr>
<td>S=6, I=1707, O=275</td>
<td>Run Time</td>
<td>27.59</td>
<td>80.93</td>
<td>72.07</td>
<td>13.63</td>
</tr>
<tr>
<td>Neem</td>
<td>Seg. Error</td>
<td>13.49</td>
<td>28.01</td>
<td>13.28</td>
<td>1.66</td>
</tr>
<tr>
<td>S=3, I=153, O=88</td>
<td>Run Time</td>
<td>4.02</td>
<td>7.89</td>
<td>8.07</td>
<td>2.2</td>
</tr>
<tr>
<td>Oldclassicswing</td>
<td>Seg. Error</td>
<td>3.96</td>
<td>5.01</td>
<td>3.96</td>
<td>1.32</td>
</tr>
<tr>
<td>S=3, I=256, O=123</td>
<td>Run Time</td>
<td>5.06</td>
<td>15.18</td>
<td>6.53</td>
<td>1.2</td>
</tr>
<tr>
<td>Raglan Castle</td>
<td>Seg. Error</td>
<td>22.89</td>
<td>29.82</td>
<td>19.17</td>
<td>22.63</td>
</tr>
<tr>
<td>S=11, I=2420, O=214</td>
<td>Run Time</td>
<td>29.3</td>
<td>49.65</td>
<td>112.19</td>
<td>8.17</td>
</tr>
<tr>
<td>Sene</td>
<td>Seg. Error</td>
<td>6.8</td>
<td>7.6</td>
<td>6.8</td>
<td>0.4</td>
</tr>
<tr>
<td>S=3, I=132, O=118</td>
<td>Run Time</td>
<td>2.43</td>
<td>20.29</td>
<td>4.73</td>
<td>0.41</td>
</tr>
<tr>
<td>Unionhouse</td>
<td>Seg. Error</td>
<td>5.18</td>
<td>4.75</td>
<td>4.56</td>
<td>7.2</td>
</tr>
<tr>
<td>S=5, I=1739, O=345</td>
<td>Run Time</td>
<td>22.47</td>
<td>62.89</td>
<td>83.01</td>
<td>15.73</td>
</tr>
</tbody>
</table>

to be bigger. This suggests that our approaches, using simultaneous sampling and fitting, are able to provide superior efficiency under more difficult data.

More interestingly, RCMSA tends to be considerably more accurate than others, especially for the complex data (e.g., data that contains many and unbalanced structures). Qualitatively, notice in Figures 5.14 and 5.15 that only RCMSA correctly detects all the evident planes in *Merton College I* and *Unionhouse* while the others are just able to detect the dominant ones. This again shows the exceptional advantages of the RCM hypothesis sampler as well as the fitting algorithm RCMSA (see Sections 4.2.2, and 4.1). Note that the RCM sampler can not be employed under the two-stage framework, as ahead discussed in Chapter 4.

Nevertheless, Figures 5.16(b) and 5.16(d) show failures of RCMSA on *Bonython Hall* and *Merton College 3*. The undetected structures have extremely small populations relative to the other structures — very few SIFT features are extracted from these structures;
this may happen due to small physical size or lack of textures on the surface. Moreover, the inliers of the undetected structures are far apart, and this challenges the spatial smoothness assumption. Note that this weakness also affects the other methods, and may be exacerbated by minimal subset sampling.

![Graphs showing evolution of SE versus run time within a 30s window.](image)

**Figure 5.5**: Comparing efficiency on multiple homography detection on a subset of the data. The graphs show the evolution of SE versus run time within a 30s window. Not all methods complete 1000 iterations in 30s. Best viewed in color.

### 5.4 Augmented Reality using Homography Mappings

So far in this chapter, we have demonstrated the performance of our proposed robust estimation methods, namely ARJMC and RCMSA, (as well as other state-of-the-art methods) on two popular intermediate-level applications in two-view geometry, which extract transformations (i.e., homographies, fundamental matrices) from two images. In this section, we apply robust estimation methods to a high-level application, which deals with the usage or interpretation of the extracted transformations. Particularly, we apply our proposed RCMSA method to an augmented reality application using homography mappings.
Figure 5.6: Using RCMSA in an augmented reality application. The left column shows the original frame overlaid with segmented keypoints (keypoints are matched with a reference frame not shown here). The right column shows the result of inserting synthetic posters. Red crosses are keypoints labelled as outliers.
Augmented reality is an area with significant commercial potential. A simple example is photo-realistically inserting synthetic objects into a video sequence (e.g., real time camera recordings, motion pictures). In many cases, the objects are planar (e.g., posters, notices, advertisement), and inserting them into a video can be achieved by first detecting the planar surfaces in a scene from key-point matches (that include outliers) between the current frame and the reference frame. Once a plane of interest is discovered, the synthetic object can be anchored to the surface by warping the object from the reference frame to the current frame.

We argue that the task of inserting synthetic flat objects into a video sequence should be accomplished by a multi-structure geometric fitting approach, mainly because most scenes of interest (e.g., indoor, urban) will contain multiple planar surfaces (structures). An approach which assumes only one structure (e.g., RANSAC) is bound to fail (which homography gets retrieved is unpredictable) or be too inefficient (due to the multiple instances of executions required). Moreover, there is the issue of model selection, i.e., determining the number of genuine planar structures embedded in the scene. Figure 5.6 demonstrates our RCMSA method on an indoor scene, where we detect multiple homographies and insert multiple synthetic posters into the scene. Note that the number of homographies are not constant and must be estimated on-the-fly. The subsequent sections describe a complete implementation.

### 5.4.1 Offline Preprocessing

First a reference frame is acquired and synthetic posters are anchored to the reference frame; see Figure 5.7. The scene contains a number of planes of interest \( \{ P_{j}^{\text{ref}} \}_{j=1}^{K} \). To anchor a poster on a non-fronto-parallel plane in the reference frame, we manually mark the 4 corners of the poster in the reference image. This yields 4 point matches which allow us to estimate a homography to warp the poster onto the plane. We also annotate every plane \( P_{j}^{\text{ref}} \) with a centroid \( m_{j} \) whose position is the center of the region.
5.4.2 Online Multiple Homography Detection

In order to insert the synthetic posters into the current frame at time $t$, we need to detect the planes in the scene viewed from the current frame. The task can be obtained by estimating the homographies between the reference frame and the current frame. SIFT keypoints are extracted and matched between the reference and the current frame. We apply RCMSA on the point matches to recover a set of homographies $\{H_i\}_{i=1}^{K'}$. Note that the number of detected homographies $K'$ may not be equal to the number of planes of interest $K$ in the reference frame. To increase efficiency, the estimated homographies $\{H_i\}_{i=1}^{K'}$ are also used as the initializer for RCMSA in the next frame $t+1$.

5.4.3 Plane Matching and Image Augmentation

An important step in inserting posters into the current frame is plane matching (see Figure 5.8). First, we need to associate a plane in the reference frame to each detected homography $H_i$. A plane $P_{j}^{ref}$ is considered to be associated with the homography $H_i$ if the inliers of $H_i$ in the reference frame are close to the centroid $m_j$ of $P_{j}^{ref}$. Specifically, if the average distance from $m_j$ to $n$ nearest inliers of $H_i$ in the reference frame is less than a threshold $th$ (we use $n = 10$ and $th = 50$ for a $350 \times 700$ pixel image), we associate $P_{j}^{ref}$ with $H_i$. $H_i$ is then used to warp the poster anchored at $P_{j}^{ref}$ to the current frame.

Figure 5.6 shows the image augmentation result on a number of selected frames. The full video can be viewed from the link\(^3\). The corresponding results in video show that our method detects homographies correctly most of the time.

Note we are not claiming that our augmented reality system is sophisticated and optimal. Our aim is to demonstrate the practical usefulness of our robust model fitting algorithm to an interesting application. We show that our method applied to multiple homography detection provides sufficient robustness and accuracy to achieve such applications.

\(^3\)http://www.youtube.com/watch?v=37gYIIEc_KA
5.5 Summary

In this chapter, we have demonstrated the superior performances of our proposed robust estimation methods, namely ARJMC and RCMSA over the state-of-the-art methods using two-view geometry estimation datasets. Here we summarize a number of important results.

- The experimental results show that our proposed methods (ARJMC and RCMSA) are more efficient than the previous approaches which treat hypothesis generation and fitting optimisation as two independent problems. This affirms the validity of our belief discussed in Chapter 2 that it is generally better to conduct hypothesis sampling and fitting optimisation simultaneously.

- The advantage of using larger-than-minimal subsets for hypothesis generation (i.e., RCM hypothesis sampler), analysed in Chapter 4, was further confirmed. Our method RCMSA, which relies on the RCM hypothesis sampler, was shown to be not only faster but also more accurate than the others. This is because the ability of the RCM sampler to efficiently produce accurate hypotheses.

- We also observe that our methods, especially RCMSA, perform particularly better than the disjoint two-stage methods when dealing with difficult data (e.g., data that contains a large number of valid structures and data points). This is because of the rapid degradations in performances of the two-stage methods, where much more effort is spent on sampling good hypotheses and optimising the fitting criteria.
Figure 5.9: Qualitative comparisons on motion segmentation on breadtoychips dataset (only one of the two views are shown). The markers indicate the structure segmentation. Red crosses are outliers.

Figure 5.10: Qualitative comparisons on motion segmentation on cubebreadtoychips dataset (only one of the two views are shown). The markers indicate the structure segmentation. Red crosses are outliers.
Figure 5.11: Qualitative comparisons on motion segmentation on breadtoycar dataset (only one of the two views are shown). The markers indicate the structure segmentation. Red crosses are outliers.

Figure 5.12: Qualitative comparisons on motion segmentation using biscuitbookbox dataset (only one of the two views are shown). The markers indicate the structure segmentation. Red crosses are outliers.
Figure 5.13: Qualitative comparisons on multi-homography detection using Johnsona dataset (only one of the two views are shown). Color indicates the point membership. Red crosses are detected outliers. All the methods perform equally well on this dataset, except QP-MF (c). Notice in (c) that one true plane is missed, and another plane is broken into 2 segments.

Figure 5.14: Qualitative comparisons on multi-homography detection using Merton College 1 dataset (only one of the two views are shown). Colors indicate the point membership. Red crosses are detected outliers. It can be seen that the segmentation result of RCMSA is qualitatively better than others. Except RCMSA, the results by others are not spatially consistent.
Chapter 5. Robust Two-view Geometry Estimation and Applications

Figure 5.15: Qualitative comparisons on multi-homography detection using Union-house dataset (only one of the two views are shown). Colors indicate the point membership. Red crosses are detected outliers. Note that only RCMSA is able to correctly detect 5 planes while others just detect 4 dominant ones.

Figure 5.16: Two examples where RCMSA fails to recover all the structures in multi-homography detection. (a)(c) show the true labels. (b)(d) are the final labelling results of RCMSA. The undetected homographies correspond to structures with extremely small populations relative to the other structures (i.e., very few SIFT features are detected in these planes).
Chapter 6

Fitting with Geometric Priors

In the previous chapters, we have seen how global optimisation methods are used to solve robust fitting problems in computer vision. Typically, these approaches find the optimal fitting by optimising robust objective functions over the parameter space. Because vision data is inherently ambiguous and noisy, the objective functions should include regularisation terms such that reasonable fitting can be achieved. These regularisations usually encode prior knowledge of the data and the problem. However, for tractability, almost all the existing fitting methods only impose two types of regularisations: model complexity and low-level spatial smoothness. The model complexity penalises the number of structures in the data, whereas the spatial smoothness enforces spatially close points to lie on the same structure. However, in many cases, such regularisations are not powerful enough to produce desired estimates, especially when the data is largely contaminated by non-normal noise and non-uniform outliers.

In this chapter, we show that incorporating higher level priors, which consider interactions between geometric models, can further improve fitting accuracy and yield more plausible results. Such high-level priors capture knowledge (e.g., geometric shape) of the scenes or objects. For example, in detecting planes from indoor images (see Fig. 6.1(a)), the room walls are usually either orthogonal or parallel to each other. In a vanishing point detection problem (see Fig. 6.1(b)), the three dominant vanishing directions are mutually orthogonal under the Manhattan world assumption (Coughlan and Yuille, 1999).

Fundamentally, using high-level geometric priors helps combat against undesirable effects from assuming an unrealistic noise and outlier models, e.g., i.i.d. normal noise for the inliers, uniform distribution for outliers. Such assumptions, typically encoded in the goodness-of-fit term, may break down unpredictably. These issues often happen in practice, and could be minimised with geometric priors.
To this end, we propose an energy function that encapsulates goodness-of-fit, spatial smoothness, model complexity, and adherence to the high-level geometric priors. (The spatial regularisation can be omitted when the smoothness assumption is not valid.) Although such an energy function formulation has been acknowledged in vision (e.g., in image segmentation (Ladicky et al., 2010)), it has not been proposed previously in geometric model fitting problems in computer vision.

To optimise the energy, we first sample the continuous model parameter space, then resort to the graph-cuts based inference technique proposed in (Ladicky et al., 2010) to find the optimal fitting.

![Image](image.png)

(a) Planar surface detection.  
(b) Vanishing point detection.

**Figure 6.1:** Two examples of computer vision applications where geometric priors should be effectively exploited to produce plausible results. (a) Detecting planar homographies using two views of an indoor scene (the first view is shown). In indoor environments, the planar surfaces (e.g., ground floors, walls) are usually either parallel or orthogonal to each other. (b) Estimating vanishing points in a single image. In a Manhattan world (Coughlan and Yuille, 1999), the three vanishing directions (coloured in red, green and blue) are mutually orthogonal.

The main content in this chapter is based on the author’s work in (Pham et al., 2013). The remainder of the chapter is organised as follows. Section 6.1 reviews existing application-specific algorithms which have considered high-level interactions between objects/models to enhance the performance. In Section 6.2 we describe our general fitting framework including the energy formulation, energy normalisation, as well as the optimisation method. Section 6.3 presents various applications to support our approach. We summarise the chapter in Section 6.4.

### 6.1 High-Level Priors in Vision

There have been various works that consider high-level constraints between models for multi-model fitting. The majority of these fall under the stochastic geometry framework (Baddeley and van Lieshout, 1993; Ortner et al., 2008; Lafarge et al., 2010), where...
the distribution of objects in a spatial coordinate space is modelled. Given a set of measurements, the most likely object configuration under the posterior is inferred. A prior is introduced to encourage favoured object configurations. This framework is, however, unable to handle more general “objects” not lying in a spatial space such as motions and affine transformations. Moreover, the framework has not considered the low-level pairwise potentials over the data points. Therefore, the methods cannot simultaneously consider (low-level) spatial smoothness and (high-level) geometric priors, unlike our method to be presented in this chapter. Other application-specific multi-model fitting (using geometric priors) include RGB-D image segmentation (Jia et al., 2013) and vanishing point detection (Schindler and Dellaert, 2004; Tardif, 2009; Mirzaei and Roumeliotis, 2011).

Recently, Olsson and Boykov (Olsson and Boykov, 2012) have introduced a curvature based regularisation for fitting surfaces onto 3D point clouds. Their method assigns a tangent plane to each noisy data point such that the estimated tangent planes conform a smooth surface. To impose a smoothness constraint, the method models interactions between tangent planes at pairwise neighbouring points. Note that by imposing a low-level smoothness constraint on the pairwise tangent planes, it is also a high-level constraint on the surfaces embedded in the original data. Though their energy function can be approximately minimised using combinatorial optimisation techniques (e.g., TRW-S (Kolmogorov, 2006)), making the inter-model relations depend on the spatial neighbourhood system limits the potential applications in the general geometric fitting. In contrast, the interplays between models in our formulation are independent of the underlying neighbourhood structure. Moreover, their method does not consider model selection (i.e., estimate the quantity of models) as each point is assigned a unique model, whereas our fitting method only returns a small number of models conforming with the geometric priors.

Though with a different application and emphasis, the work most related to our fitting algorithm presented in this chapter is (Ladicky et al., 2010). In (Ladicky et al., 2010), the problem of semantic image segmentation is modelled using Conditional Random Field (CRF) with co-occurrence statistics. Typically, the co-occurrence terms can help prevent implausible combinations of semantic labels (e.g, cow, lion) appearing in a single image. We extend the co-occurrence idea (and their inference algorithm) to the general geometric multi-model fitting problem, where consistent geometric models should be recovered from the data. Unlike the image segmentation task where the co-occurrence potentials possibly help overcome the weaknesses of the object detectors/classifiers, our geometric consistency is specifically used to alleviate the influences of noise and outliers in the data. (Note that noise and outliers are the main roots of biased parameter estimation.) Moreover, the inference method proposed in (Ladicky et al., 2010) is for
a discrete set of labels, to deal with continuous geometric model parameter spaces, we discretize the parameter spaces by randomly sampling model candidates. Therefore, the geometric consistency between models are computed directly from the model parameters rather than learning from the training data.

### 6.2 Robust Fitting with Geometric Priors

In this section, we describe our robust fitting framework which seamlessly imposes geometric constraints during the parameter estimation. Similar to the estimation methods proposed in Chapter 3 and 4, here we formulate the multi-model fitting as an energy minimisation task, i.e.,

$$\Theta^* = \arg\min_{\Theta \subset \Omega} E(\Theta), \quad (6.1)$$

where $E$ is some energy function, and $\Omega$ is a continuous space of parameters (e.g., circle parameters). Next we define the formulation of the energy function $E(\Theta)$.

### 6.2.1 Energy Function

We extend the energy function (4.1), used in Chapter 4, to further encode the high-level geometric priors, i.e.,

$$E(\Theta, f) = \sum_{n=1}^{N} D(x_n, f_n) + w_1 \sum_{(p,q) \in \mathcal{N}} V(f_p, f_q) + w_2 G(\Theta), \quad (6.2)$$

where $D$ and $V$ functions respectively encode the fitting error and spatial (ir)regularity; $G(\Theta)$ encodes the high-level geometric constraints on $\Theta$; $w_1$ and $w_2$ are weighting parameters; $\mathcal{N}$ represents a sparse neighbourhood structure defined on $\mathcal{X}$; $f = [f_n]_{n=1}^{N}$ is a vector of labels, where each $f_n \in \Theta$ assigns $x_n$ to either one of the structures in $\Theta$ or an outlier label.

The data term $D(x_n, f_n)$ evaluates the fitting error on $x_n$

$$D(x_n, f_n) = \begin{cases} \frac{r(x_n, f_n)^2}{\sigma^2} & \text{if } f_n \in \Theta \\ 1 & \text{if } f_n = \text{null} \end{cases} \quad (6.3)$$

where $r(x_n, f_n)$ gives the absolute residual of $x_n$ from the model/label $f_n$, $\sigma$ refers to the noise level.
The pairwise term \( V(f_p, f_q) \) is defined over the labels of neighbouring points \( p, q \in X \), where \((p, q) \in N\). We have

\[
V(f_p, f_q) = \begin{cases} 
1 & \text{if } f_p \neq f_q \\
0 & \text{if } f_p = f_q,
\end{cases}
\]

which implements the spatial smoothness priors, since it encourages neighbouring points to be assigned to the same model.

The function \( G(\Theta) \), in many practical cases, can be defined as the sum over the subsets of models, i.e.,

\[
G(\Theta) = \sum_{S \subseteq \Theta} C(S),
\]

where \( S \) is any subset of \( \Theta \), and \( C(S) \geq 0 \) measures the “compatibility” between the models in \( S \). When \(|S| = 1\), \( C(S) \) merely computes the model complexity.

The size of \( S \) and the compatibility measure \( C(S) \) vary with applications. For example, in detecting planar surfaces (from images) of the man-made environments (e.g., buildings), it is natural to enforce every pair of planes (\(|S| = 2\)) to be either orthogonal or parallel. It is also possible to impose more complicated constraints, e.g., the three planes (\(|S| = 3\)) that intersect at the same corner should be mutually orthogonal. Another application is estimating vanishing points in images. Under the Manhattan world, the three vanishing directions are mutually orthogonal (see Figure 6.1(b)). Therefore we could place an orthogonality constraint on every triplet (\(|S| = 3\)) of vanishing points. We also could enforce every pair (\(|S| = 2\)) of vanishing points to be orthogonal.

For simplicity and reducing the energy approximation error (see Section 6.2.3), in this work, we consider (at most) pairwise interactions between the geometric models. That is

\[
G(\Theta) = \sum_{\theta \in \Theta} C(\theta) + \sum_{\beta, \gamma \in \Theta} C(\beta, \gamma).
\]

\( C(\theta) > 0 \) is a per-model cost function. \( C(\beta, \gamma) \geq 0 \) is a pair-model cost function imposing the geometric consistency. For example, \( C(\beta, \gamma) \) can be defined as:

\[
C(\beta, \gamma) = \begin{cases} 
0 & \text{if } \beta \sim \gamma \\
l & \text{if } \beta \not\sim \gamma
\end{cases}
\]
where \( \sim (\approx) \) implies the consistency (inconsistency) between two models, \( l \) is a positive penalty. The actual definition of the geometric consistency depends on the models and problems. (See Section 6.3 for real examples.)

### 6.2.2 Energy Normalisation

In order for the minimum of (6.2) to produce accurate fitting results, the terms in (6.2) should be balanced — none of the terms in (6.2) dominates the other ones. This can be achieved by simply tuning the parameters \( w_1 \) and \( w_2 \). However manually selecting the best values for \( w_1 \) and \( w_2 \) for every single dataset is impractical and tedious.

Here we suggest a strategy to choose \( w_1 \) and \( w_2 \) such that the energy scales well with different datasets. Notice that in (6.2) the first (fitting error) term linearly increases with the number of data points \( N \), and the second (pairwise smoothness) cost linearly grows with the number of edges in \( \mathcal{N} \), which is only a few times bigger than \( N \) (assuming sparsely connected graphs). Thus, they are approximately in the same scale. Our empirical study shows that a value of \( w_1 \) within a range \([0.1, 0.5]\) would work well.

Selecting a good value of \( w_2 \) is, however, more challenging. In part, it is because the model consistency term \( G(\Theta) \) depends on the size of \( \Theta \), which is unknown and has to be estimated. In this work, we propose \( w_2 = \lambda d \log(N)/K_{\text{max}} \), where \( \lambda > 0 \) is a free parameter, \( d \) is the number of parameters to define the geometric model (e.g., \( d = 3 \) for line, \( d = 4 \) for plane), \( K_{\text{max}} \) is the maximum expected number of structures in the data. The amount \( d \times \log(N) \) fundamentally has the same functionality as the classical model selection criteria such as AIC (Akaike, 1974), BIC (Schwarz, 1978) or MDL (Rissanen, 1989) widely used in the statistical estimation. \( K_{\text{max}} \) works as a normalising constant, which can be chosen easily depending on the specific problems. And \( \lambda \) is an actual tuning parameter. Empirically, we found that small values of \( \lambda \) (e.g., 0.1) would work reasonably well. We will show the exact settings of these parameters for each particular application in Section 6.3.

### 6.2.3 Energy Minimisation

Minimising the energy (6.2) is generally intractable. For tractability, as discussed in Chapter 2, a common strategy is to randomly sample a set of model hypotheses \( \mathcal{H} = \{\theta_m\}_{m=1}^M \), which sufficiently covers all true structures in the data, then search for the optimal subset \( \Theta \subset \mathcal{H} \) and labels \( \mathbf{f} \) that minimise the energy \( E(\Theta, \mathbf{f}) \) (6.2). As a result, by extending the domain of the variables \( \mathbf{f} \), i.e., \( f_n \in \mathcal{H} \) (\( f_n \) assigns \( x_n \) to one of
Chapter 6. Fitting with Geometric Priors

97

the structure in \( \mathcal{H} \), minimising the energy (6.2) over \( \mathcal{H} \) corresponds to minimising the following function

\[
E(f) = \sum_{n=1}^{N} D(x_n, f_n) + w_1 \sum_{\langle p, q \rangle \in \mathcal{N}} V(f_p, f_q) + w_2 \left[ \sum_{\theta \in \mathcal{H}} C_{\theta} \delta_{\theta}(f) + \sum_{\beta, \gamma \in \mathcal{H}} C_{\beta, \gamma} \delta_{\beta}(f) \delta_{\gamma}(f) \right],
\]

(6.8)

where \( C_{\theta} = C(\theta), C_{\beta, \gamma} = C(\beta, \gamma) \), \( \delta(.) \) is an indicator function, i.e.,

\[
\delta_{\theta}(f) = \begin{cases} 
1 & \text{if } \exists x_n : f_n = \theta \\
0 & \text{otherwise.}
\end{cases}
\]

(6.9)

To optimise the energy (6.8), we follow the extended expansion move algorithm proposed in (Ladicky et al., 2010). The idea of the expansion move method is to project a multi-label problem (e.g., (6.8)) into a sequence of binary-label subproblems, where each binary energy can be solved very efficiently using graph-cuts (assuming that the binary energy is pairwise and submodular).

More specifically, in the expansion move framework, starting from a current labelling \( f \), for some label \( \alpha \in \mathcal{H} \), the method estimates a new labelling \( f^{\alpha} \) such that a label variable either changes to \( \alpha \) or retains its old value from \( f \). This “expansion” step is iterated for all other labels until convergence. The new labelling \( f^{\alpha} \) can be obtained via a transformation function \( T \) over binary variables \( t = [t_1, t_2, \ldots, t_N] \), i.e.,

\[
f^{\alpha}_n = T_{\alpha}(f_n, t_n) = \begin{cases} 
\alpha & \text{if } t_n = 0 \\
\ f_n & \text{if } t_n = 1.
\end{cases}
\]

(6.10)

The task reduces to an inference over the binary variables \( t \). Consequently we need to derive an expansion move energy \( E^{\alpha}(t) \) so that \( E^{\alpha}(t) \) can be solved using graph-cuts. We rewrite the energy (6.8) in terms of the binary variables \( t \)

\[
E^{\alpha}(t) = E^{\alpha}_1(t) + w_2 E^{\alpha}_2(t),
\]

(6.11)

where \( E^{\alpha}_1(t) \) encodes the unary and local pairwise potentials, and \( E^{\alpha}_2(t) \) encodes the model complexity and model pairwise interactions in (6.8). The expansion move algorithm was originally proposed by Boykov et al. (Boykov et al., 2001) to tackle the standard unary and pairwise energy such as \( E^{\alpha}_1(t) \). We refer readers to (Boykov et al., 2001) for detailed derivation of \( E^{\alpha}_1(t) \) and the associate graph construction. Here we focus on the second term \( E^{\alpha}_2(t) \).
Let $\Theta$ be a set of models corresponding to the current labelling $f$, and assume that $\alpha \notin \Theta$. Let us define the following terms.

$$X_\theta = \{ x_n \in X | f_n = \theta \} \quad \forall \theta \in \Theta. \quad (6.12)$$

$$\delta_\alpha(t) = \begin{cases} 
1 & \text{if } \exists x_n \in X \text{ s.t. } t_n = 0 \\
0 & \text{otherwise.} 
\end{cases} \quad (6.13)$$

$$\forall \theta \in \Theta, \delta_\theta(t) = \begin{cases} 
1 & \text{if } \exists x_n \in X_\theta \text{ s.t. } t_n = 1 \\
0 & \text{otherwise.} 
\end{cases} \quad (6.14)$$

Intuitively, $X_\theta$ is a set of points currently assigned to label $\theta$; $\delta_\alpha(t)$ (or $\delta_\theta(t)$) is set to 1 if model $\alpha$ (or $\theta$) is present in the solution after the expansion move and 0 otherwise.

Now the binary energy $E_2^\alpha(t)$ can be derived as:

$$E_2^\alpha(t) = \sum_{\theta \in \Theta} C_\theta \delta_\theta(t) + \sum_{\beta, \gamma \in \Theta} C_{\beta, \gamma} \delta_\beta(t) \delta_\gamma(t) + C_\alpha \delta_\alpha(t) + \sum_{\theta \in \Theta} C_{\theta, \alpha} \delta_\theta(t) \delta_\alpha(t). \quad (6.15)$$

Since either $\alpha$ or $\theta$ ($\forall \theta \in \Theta$) must be present in $\Theta$ after any expansion move, the following statement holds

$$\delta_\theta(t) \delta_\alpha(t) = \delta_\theta(t) + \delta_\alpha(t) - 1. \quad (6.16)$$

After replacing the term $\delta_\theta(t) \delta_\alpha(t)$ and disregarding the constants\(^1\), the energy (6.15) becomes

$$E_2^\alpha(t) = \sum_{\beta, \gamma \in \Theta} C_{\beta, \gamma} \delta_\beta(t) \delta_\gamma(t) + [C_\alpha + \sum_{\theta \in \Theta} C_{\theta, \alpha}] \delta_\alpha(t) + \sum_{\theta \in \Theta} [C_{\theta, \alpha} + C_\theta] \delta_\theta(t). \quad (6.17)$$

The energy (6.17) still cannot be readily minimised using graph-cuts since (6.17) contains a higher-order component, i.e., the first term. Therefore an approximation technique must be used.

Because the $\delta(.)$ function is binary, it is clear that

$$\delta_\beta(t) \delta_\gamma(t) \leq 0.5 [\delta_\beta(t) + \delta_\gamma(t)]. \quad (6.18)$$

And since $C_{\beta, \gamma} \geq 0 \forall \beta, \gamma,$

$$C_{\beta, \gamma} \delta_\beta(t) \delta_\gamma(t) \leq 0.5 C_{\beta, \gamma} [\delta_\beta(t) + \delta_\gamma(t)]. \quad (6.19)$$

\(^1\)Adding an arbitrary constant to $E_2^\alpha(t)$ does not affect the optimal of $E_2^\alpha(t)$. 
Chapter 6. Fitting with Geometric Priors

Summing up all possible pairs $\beta$ and $\gamma$ in inequality (6.19), we obtain the following inequality

$$\sum_{\beta, \gamma \in \Theta} C_{\beta, \gamma} \delta_\beta(t) \delta_\gamma(t) \leq \sum_{\beta, \gamma \in \Theta} 0.5 C_{\beta, \gamma} \left[ \delta_\beta(t) + \delta_\gamma(t) \right]. \quad (6.20)$$

Because $C_{\beta, \gamma} = C_{\gamma, \beta}$ $\forall \beta, \gamma$, we have

$$\sum_{\beta, \gamma \in \Theta} C_{\beta, \gamma} \delta_\beta(t) \delta_\gamma(t) = \sum_{\beta, \gamma \in \Theta} \left[ C_{\beta, \gamma} \delta_\beta(t) + C_{\gamma, \beta} \delta_\gamma(t) \right] = \sum_{\beta \in \Theta} \delta_\beta(t) \sum_{\gamma \in \Theta \setminus \beta} C_{\beta, \gamma}. \quad (6.21)$$

As a result, the first term in (6.17) can be over-estimated as:

$$\sum_{\beta, \gamma \in \Theta} C_{\beta, \gamma} \delta_\beta(t) \delta_\gamma(t) \leq \sum_{\beta \in \Theta} 0.5 \delta_\beta(t) \sum_{\gamma \in \Theta \setminus \beta} C_{\beta, \gamma}. \quad (6.22)$$

After simplification, the over-estimated energy of (6.17) becomes:

$$\hat{E}_2^\alpha(t) = c_\alpha \delta_\alpha(t) + \sum_{\beta \in \Theta} c_\beta \delta_\beta(t), \quad (6.23)$$

where

$$c_\alpha = C_\alpha + \sum_{\theta \in \Theta} C_{\theta, \alpha} \geq 0, \quad (6.24)$$

$$c_\beta = C_\beta + C_{\beta, \alpha} + \sum_{\gamma \in \Theta \setminus \beta} 0.5 C_{\beta, \gamma} \geq 0. \quad (6.25)$$

Note that the over-estimation strategy as in Equation (6.22) is not recommended for larger subsets $S$. In the general form, the over-estimation is

$$C(S) \prod_{\beta \in S} \delta_\beta(t) \leq C(S) \sum_{\beta \in S} \frac{1}{|S|} \delta_\beta(t). \quad (6.26)$$

Therefore the energy of the moves which remove one of the labels from $S$ will be excessively over-estimated.

**Lemma 6.1.** The over-estimated expansion move energy $\hat{E}_2^\alpha(t)$ in (6.23) can be represented as a binary pairwise energy

$$\hat{E}_2^\alpha(t, z) = c_\alpha (1 - z_\alpha) + \sum_{\beta \in \Theta} c_\beta z_\beta + \sum_{x_i \in X'} c_\alpha (1 - t_i) z_\alpha + \sum_{\beta \in \Theta} \sum_{x_i \in X_{\beta'}} c_\beta t_i (1 - z_\beta), \quad (6.27)$$

where $z$ is a vector of $|\Theta| + 1$ binary auxiliary variables.
Chapter 6. Fitting with Geometric Priors

A proof of the Lemma 6.1 can be found in (Ladicky et al., 2010). The binary energy (6.27) is pairwise and submodular, therefore (6.27) can be minimised using graph-cuts. Figure 6.2 shows the graph construction for the binary energy (6.27). In the case $\alpha \in \Theta$, we can simply discard the term $c_\alpha \delta_\alpha(t)$ from (6.23), and re-compute the energy $E^\alpha(t^*)$ with the optimised $t^*$. If the new energy decreases, the move is accepted, otherwise rejected.

Note that the above-discussed minimisation strategy assumes that the hypothesis set $\mathcal{H}$ is adequate, i.e., all valid structures in the data are sampled. However, as shown in the previous chapters, if $\mathcal{H}$ is constructed by fitting models onto randomly sampled minimal subsets of data (as in RANSAC), the sufficiency of $\mathcal{H}$ might not be ensured before the optimisation, thus possibly resulting in inaccurate estimates. The issue can be tackled by conducting hypothesis sampling and fitting optimisation simultaneously, as done in ARJMC and RCMSA (see Chapters 3 and 4 for details).

6.2.4 Interacting Geometric Costs vs. Category Costs

Delong et al. (Delong et al., 2010, 2012a) have introduced an energy function with the “category costs”, which is somewhat related to our geometric consistency cost $G(\Theta)$. Their category costs have a form:

$$\sum_{H \subset \mathcal{H}} L(H) \delta_H(f),$$

where

$$\delta_H(f) = \begin{cases} 
1 & \text{if } \exists x_n \in \mathcal{X} : f_n \in H \\
0 & \text{otherwise}.
\end{cases}$$
Notice that the energy (6.28) pays a category cost $L(H)$ as soon as a single label from the subset $H$ appears in $f$ (see Equation (6.29)). Clearly this kind of category costs is different from our geometric consistency costs, where a penalty $C(\beta, \gamma)$ is paid only if both labels $\beta$ and $\gamma$ are used.

6.3 Experiments and Applications

In this section, we aim to test the capability of the proposed fitting approach on various vision applications including two-view multi-homography estimation, vanishing point detection, circular object detection and 3D planar surface reconstruction. We compare our approach, denoted as MFIGP (Model-Fitting-with-Interacting-Geometric-Priors), with the state-of-the-art fitting methods such as: Jlinkage\textsuperscript{2} (Toldo and Fusiello, 2008), PEARL (reviewed in Section 2.2.5) as well as other application-specific algorithms. We exclude other optimisation based fitting methods which perform, at most, similarly to PEARL.

6.3.1 Two-view Multi-homography Estimation

We first apply MFIGP to estimate multiple homographies between two views of planar scenes. This problem has been discussed in Chapter 5. We use the Michigan Indoor Corridor 2011 Video Dataset (Tsai et al., 2011). The videos were obtained by a camera placed on a wheeled vehicle. The camera was moving with zero tilt and roll angle with respect to the ground floor. We apply the KLT method (Shi and Tomasi, 1994) to track the features across the video frames, then choose the matches from the first and 30th frames for estimating the homographies. We only select the sequences which contain sufficient features on the scene planes, and satisfy the “box” assumption (explained next), namely Library, Library2, Locker Room, Object and EECS Building.

In the indoor environments, the planes (e.g., walls) usually conform the “box” assumption, i.e., the planes are either mutually orthogonal or parallel. The “box” constraint can be imposed by the orthogonality and parallelism between pairs of homographies. In particular, given two homographies $H_i$ and $H_j$, MFIGP defines the pair-model cost function as

$$C(H_i, H_j) = \min \left( e^{\frac{90 - A(n_i, n_j)}{2}}, e^{\frac{A(n_i, n_j)}{2}} \right) - 1, \quad (6.30)$$

\textsuperscript{2}Matlab source code: http://www.diegm.uniud.it/fusiello/demo/jlk/JLinkage.rar
where $n_i$ and $n_j$ are the normal vectors extracted from the homographies $H_i$ and $H_j$; $A(\cdot, \cdot)$ measures the angle between two vectors. Although for each homography matrix $H_i$ or $H_j$, there are at most two physically possible sets of parameters (i.e., translation, rotation and normal) decomposed from $H_i$ or $H_j$ (Ma et al., 2004), we can select the true normals $n_i$ and $n_j$ based on the fact that $H_i$ and $H_j$ have the same (true) rotation and translation parameters.

Moreover, since the camera is aligned parallel to the ground floor, the normal vector of the ground plane is known, i.e., $n_g = [0, 1, 0]$. To take advantage of this prior knowledge, we define the per-model cost function as

$$C(H_i) = \min \left( e^{90 - A(n_i, n_g) / 2}, e^{A(n_i, n_g) / 2} \right)$$

(6.31)

to penalise the planes/homographies inconsistent with the ground. Note that the label cost function (6.31) is also applied for PEARL method.

**Parameter settings.** Given a set of point matches $\mathcal{X}$, the neighbourhood structure $\mathcal{N}$ is constructed by a Delaunay triangulation, which is used by both MFIGP and PEARL. For MFIGP, we fix $\sigma = 0.008$, $\lambda = 0.1$ and $K_{max} = 10$ for all the image pairs. For PEARL and Jlinkage, the same inlier noise scale $\sigma$ is used; we tune the remaining required parameters so that exactly 4 homographies are returned. These homographies are expected to correspond to the ground floor, left wall, right wall and the ceiling.

**Hypothesis sampling.** As discussed in Section 6.2.3, the geometric model fitting can be achieved by optimising the energy (6.2) over a set of model hypotheses $\mathcal{H}$, under the assumption that $\mathcal{H}$ sufficiently covers all the true models. Traditionally, $\mathcal{H}$ is constructed by sampling minimal subsets as in RANSAC. However, since the model is high-order (homography has 8 free parameters) and the data is noisy, as shown in Chapter 4 using minimal subsets might not produce a sufficient and accurate set of hypotheses $\mathcal{H}$. To tackle the issue, we propose an iterative algorithm which alternately samples hypotheses (using non-minimal subsets) and optimises the energy. Algorithm 6.1 outlines the idea. First, our algorithm generates 1000 hypotheses using minimal subsets, and optimises the energy (6.2) to select the best subset of models $\Theta$ and labelling $f$. Based on the labelling $f$, we partition the data into a number of clusters using the RCM method proposed in Chapter 4 (see Section 4.2), then generate a hypothesis set $\mathcal{H}$ by fitting the model onto each cluster of data. (Only clusters whose sizes are larger than minimal are used.) These hypotheses are concatenated with the previously estimated models $\Theta$, and input to the optimisation to select the best $\Theta$ and labelling $f$. The process is repeated
until convergence\(^3\). In our experiments, the convergence is typically reached after dozens of iterations. To make a fair comparison, PEARL is also adapted to this algorithm.

**Algorithm 6.1:** Iterative hypothesis sampling and model selection for homography estimation.

1: Sample a set of hypothesis \( \mathcal{H} \) as in RANSAC.
2: Optimise (6.2) over \( \mathcal{H} \) to obtain labeling \( f \) and \( \Theta \).
3: Partition the data into clusters using \( f \) (see Section 4.2 for details).
4: Generate new hypotheses \( \mathcal{H} \) using the sampled clusters.
5: \( \mathcal{H} = \mathcal{H} \cup \Theta \).
6: Optimise (6.2) over \( \mathcal{H} \) to obtain new labeling \( f \) and \( \Theta \).
7: Goto Step 3.

**Qualitative comparisons.** Figure 6.3 depicts the results on the Library, Object and EECS Building datasets. It is evident that Jlinkage and PEARL are unable to satisfactorily segment the data into the groups corresponding to the actual scene planes. The flaws of Jlinkage and PEARL could be because the changes (the camera motions) between two views are relatively small. In such cases, all the valid homographies are not much different. On the contrary, MFIGP better classifies the data into the ground, left wall, right wall and the ceiling.

**Quantitative comparisons.** To quantify the estimation accuracy, we use the point transfer error. Given a true match \( x = (x^1, x^2) \) and a homography \( H \), the point transfer error is defined as \( d(x^2, Hx^1) \), where \( d(.,.) \) is a Euclidean distance measure. More specifically, for each image pair, we extract three sets of true matches from the ground floor, left wall and right wall separately by running RANSAC on each plane individually. These sets of true matches are denoted as \( \mathcal{X}_g \), \( \mathcal{X}_l \) and \( \mathcal{X}_r \). The ceiling and front wall are not considered because they do not contain sufficient tracked features and they are not always detected by all the methods (see Figure 6.3). The total error of an estimate \( \Theta \) is computed as

\[
\min_{H \in \Theta} \sum_{x \in \mathcal{X}_g} d(x^2, Hx^1) + \min_{H \in \Theta} \sum_{x \in \mathcal{X}_l} d(x^2, Hx^1) + \min_{H \in \Theta} \sum_{x \in \mathcal{X}_r} d(x^2, Hx^1). \quad (6.32)
\]

Table 6.1 reports the mean and median errors over 20 repetitions on the 5 tested image pairs. It is clear that in most of the cases MFIGP outperforms its competitors. Table 6.1 also depicts the average computation time, where MFIGP is comparable with PEARL and faster than Jlinkage.

\(^3\)By setting the initial labelling for the expansion move algorithm (see Section 6.2.3) to the previously estimated labelling, the energy (6.2) will be non-increasing. Therefore the algorithm 6.1 is guaranteed to converge.
Chapter 6. Fitting with Geometric Priors

6.3.2 Vanishing Point Detection in a Manhattan World.

The detection of vanishing points (VP) from a single image has many interesting applications including single-view reconstruction (Hedau et al., 2009), rectangular structure detection (Zhang and Kosecka, 2003; Micusik et al., 2008) and pose estimation (Caprile and Torre, 1990). The task can be accomplished by first extracting edges from the image, then searching for the most likely vanishing points that the extracted edges pass through. This problem is clearly a multi-model fitting problem where the models are vanishing points and the data are line segments.

In the Manhattan world, the 3D lines are considered to belong to the three mutually orthogonal directions. Therefore the orthogonality constraints must be imposed to ensure the detection accuracy.
Datasets | Jlinkage  | PEARL  | MFIGP  |
---|---|---|---|
Library | Mean error | 629.55 | 550.64 | 442.55 |
      | Median error | 539.21 | 579.40 | 430.82 |
      | Run time (seconds) | 14.71 | 8.85 | 9.39 |
Library2 | Mean error | 571.44 | 513.43 | 385.13 |
        | Median error | 556.01 | 541.54 | 366.04 |
        | Run time (seconds) | 13.59 | 8.35 | 9.19 |
Locker Room | Mean error | 201.23 | **120.28** | 122.46 |
         | Median error | 173.86 | 120.43 | **120.37** |
         | Run time (seconds) | 12.82 | **8.64** | 9.13 |
Object | Mean error | 1028.19 | 715.02 | **624.99** |
       | Median error | 1001.94 | 783.73 | **548.72** |
       | Run time (seconds) | 11.16 | **8.46** | 9.31 |
EECS Building | Mean error | 1095.43 | 840.83 | **663.39** |
            | Median error | 1142.83 | 748.67 | **574.44** |
            | Run time (seconds) | 11.29 | **8.19** | 8.77 |

Table 6.1: Quantitative comparison results on two-view multiple homography estimation. The lowest error and computation time are boldfaced.

We apply MFIGP and its main competitor, PEARL, to solve the VP detection task; then compare the performances. We also compare with the recent vanishing point detectors such as Tardif’s method (Tardif, 2009) and RNS (Mirzaei and Roumeliotis, 2011). Tardif’s method is based on the multi-model robust fitting method Jlinkage (Toldo and Fusiello, 2008). The method first clusters the edges into groups (i.e., pencils of lines), then estimates the vanishing points separately on each group. Finally, a triplet of most orthogonal vanishing points is heuristically selected. RNS assumes that the three dominant directions in the image correspond to the three Manhattan directions, then formulates the problem as a single-model fitting where a model is a triplet of orthogonal vanishing points. A RANSAC-like algorithm is used to search for the best model.

We use the York Urban Dataset (Denis et al., 2008), which consists of 102 calibrated images of man-made environments. For each image, the database provides the ground truth orthogonal triplet of vanishing points, and groups of manually extracted line segments corresponding to the vanishing points. The camera internal parameters are also given.

Assume that we have extracted a set of edges $\mathcal{X} = \{x_n\}_{n=1}^N$ from the image. Each edge is represented by two end points and one middle point, i.e., $x_n = [e_{1n}, e_{2n}, e_{3n}]$. The task is to estimate a set of mutually orthogonal vanishing points $\Theta = \{\theta_k\}_{k=1}^K$. Points are in the homogeneous coordinate, i.e., $e = [e_1, e_2, 1]$, $\theta = [\theta_1, \theta_2, 1]$. Following the work (Tardif,
the residual from an edge $x_n$ to a vanishing point $\theta_k$ is computed as
\begin{equation}
 r(x_n, \theta_k) = \frac{|l^T e_1|}{\sqrt{l_1^2 + l_2^2}} \quad \text{with} \quad l = [l_1, l_2, l_3] = e_n \times \theta_k. \tag{6.33}
\end{equation}

$l$ is the line passing through the middle point $\bar{e}_n$ and the vanishing point $\theta_k$. (See Figure 6.4.)

In MFIGP, the geometric consistency between two vanishing points is defined as:
\begin{equation}
 C(\beta, \gamma) = \begin{cases} 
 0 & \text{if } A(\beta, \gamma) \in [89, 91] \\
 10^2 & \text{otherwise}
\end{cases} \tag{6.34}
\end{equation}

where $A(\beta, \gamma)$ measures the angle between two directions corresponding to two vanishing points $\beta$ and $\gamma$. Specifically, $A(\beta, \gamma)$ is defined as (Hartley and Zisserman, 2004)
\begin{equation}
 A(\beta, \gamma) = \arccos \left( \frac{\beta^T w \gamma}{\sqrt{\beta^T w \beta} \sqrt{\gamma^T w \gamma}} \right) \frac{180}{\pi}, \tag{6.35}
\end{equation}

where $w$ is the image of absolute conic given by the camera internal matrix $K$ (i.e., $w = K^{-T} K^{-1}$ (Hartley and Zisserman, 2004)). A constant per-model cost function is used, i.e., $C(\theta) = 1 \forall \theta$.

In this application, the spatial smoothness assumption is not sensible because the vertical and horizontal edges within a genuine plane belong to two structures (vanishing directions) although they are spatially close. Thus we remove the spatial smoothness costs from both MFIGP and PEARL (e.g., by setting $w_1$ in (6.2) to 0). MFIGP uses the following parameters $\sigma = 1, \lambda = 0.1, K_{max} = 10$. PEARL also uses the same noise level of 1 pixel, and is manually tuned to return the best results. Moreover, we randomly sample 1000 VP hypotheses for MFIGP and PEARL using minimal subsets of 2 edges.

To evaluate the performances, we use two different measures. The first one is the accuracy of the detected vanishing points. Let $\Theta$ and $\bar{\Theta}$ be the estimated and ground truth vanishing points respectively, the estimation accuracy is computed as the angular
deviation between $\Theta$ and $\hat{\Theta}$. That is

$$AD(\Theta) = \min_{\Gamma} \frac{1}{3} \sum_{i=1}^{3} A(\theta_i, \bar{\theta}_i)$$  (6.36)

where $\Gamma$ is a permutation on $\Theta$ and the function $A$ is defined as in (6.35). Since some images contain only one or two vanishing directions, a post-processing step is carried out before evaluating (6.36). If $\Theta$ contains only two vanishing points (i.e., $\Theta = \{\theta_1, \theta_2\}$), the third one can be computed as:

$$\theta_3 = \text{Null}(w(\theta_1, \theta_2)),$$  (6.37)

where $\text{Null}(M)$ returns the left nullspace of $M$. $\theta_3$ will be orthogonal with $\theta_1$ and $\theta_2$. If $\Theta$ consists of only one vanishing point (i.e., $\Theta = \{\theta_1\}$), $\theta_2$ and $\theta_3$ are set to $[0, 0, 1]$. If $|\Theta| > 3$, the top three largest (in terms of the number of supporting inliers) vanishing points are selected.

The second error measure is the consistency between the estimated vanishing points and the ground truth edges. That is

$$CE(\Theta) = \frac{1}{|\bar{X}|} \sum_{\bar{x} \in \bar{X}} \min_{\theta \in \Theta} r(\bar{x}, \theta),$$  (6.38)

where $\bar{X}$ is a set of ground truth line segments. The median angular deviation and consistency errors over 50 repetitions are recorded for evaluation.

![Figure 6.5: Quantitative comparisons of the methods for detecting vanishing points in the images using the manually extracted lines. (a) and (b) show the cumulative histograms of the angular deviations and consistency errors respectively. The graphs show that all the methods are relatively comparable. The performance of all the methods is not much different because the data is not contaminated by outliers.](image-url)
Chapter 6. Fitting with Geometric Priors

Figure 6.6: Quantitative comparisons of the methods for detecting vanishing points in the images using automatically extracted lines. (a) and (b) show the cumulative histograms of the angular deviations and consistency errors respectively. Observe that MFIGP performs significantly better than its competitor (PEARL). The low performance of PEARL is mainly due to its incapability to handle data with strong noise and outliers. In contrast, MFIGP with the use of geometric constraints is highly robust. Also MFIGP is still competitive with RNS (Mirzaei and Roumeliotis, 2011) and considerably better than Tardif’s method (Tardif, 2009).

Using the supplied edges. We first test the performance of all the methods using the supplied edges given by YorkUrban Database. Figure 6.5 shows the quantitative results. Expectedly, all the methods are comparable since the data contains no outliers.

Using extracted edges. We perform the same experiment again, but using the edges automatically extracted by the method in Tardif (2009). The quantitative comparison results are shown in Figure 6.6. It is evident that MFIGP significantly outperforms PEARL in both error measures, which proves the robustness of our fitting approach. Compared with the specialised VP detectors, MFIGP is considerably better than Tardif’s VP method and comparable with RNS. Our improvements over Tardif’s VP method suggest that it is generally better to impose the geometric constraints during the model parameter estimation rather than after the estimation as in Tardif’s method.

Figure 6.7 qualitatively demonstrates the edge clustering results of MFIGP and PEARL on 4 difficult images. The results of Tardif and RNS are visually similar to MFIGP’s, so we do not show those here. It can be seen that PEARL incorrectly estimates the three Manhattan directions when there exist non-Manhattan directions in the images (see the first two images in Figure 6.7). Moreover, the last two images show another weakness of PEARL, where the large structures (e.g., the vertical directions) are split into smaller similar structures. On the contrary, these issues can be tackled effectively by MFIGP (see the last column of Figure 6.7). Indeed, as opposed to PEARL, MFIGP would unlikely return two approximately identical vanishing points since they are geometrically inconsistent.
Figure 6.7: Qualitative comparisons of MFIGP and PEARL for detecting vanishing points in the images using automatically extracted lines (Best viewed on screen). The left column shows the images with the extracted lines. The middle and right columns show the line segmentation results according to the vanishing points detected by PEARL and MFIGP separately, where the red, green and blue lines show the three estimated directions, the black lines are detected outliers. Notice the first two images (the top two rows), PEARL fails to recover the three Manhattan directions due to the presences of non-Manhattan directions. In contrast, MFIGP exactly returns three orthogonal directions. The last two images demonstrate another advantage of MFIGP over PEARL. It can be seen that the images contain uneven-structures (i.e., planes with different number of edges). In such cases, PEARL tends to split the big structures into smaller ones, resulting in wrong detections. However MFIGP can tackle this problem easily since inconsistent vanishing points (e.g., two similar vanishing points) are discouraged to jointly appear in the solution.
6.3.3 Circular Object Detection in Images

Circular object detection is an important problem in computer vision, in which we are interested in determining size, location and the quantity of objects in the images. The task can be achieved by either fitting circles or disks to the images, depending on the properties of the objects. For example, if the objects of interest have approximately homogeneous intensities (colours), one can fit disks to the image pixels using colour models. When the objects have high contrast boundaries with respect to the background and inhomogeneous colours, it is more computationally efficient to fit circles to the extracted edge pixels. We will consider these two models in this experiment.

A circle/disk is parameterised as \( \theta = (x_\theta, r_\theta) \) where \( x_\theta = (x_\theta, y_\theta) \) and \( r_\theta \) are the centre and radius, respectively. If \( \theta \) is a circle, the distance (residual) between an edge pixel \( x_n = (x_n, y_n) \) and \( \theta \) is computed as

\[
r(x_n, \theta) = |r_\theta - \sqrt{(x_n - x_\theta)^2 + (y_n - y_\theta)^2}|
\]

(6.39)

If \( \theta \) is a disk, the residual is defined as

\[
r(x_n, \theta) = \begin{cases} D(\theta) & \text{if } x_n \in R_\theta \\ \infty & \text{otherwise,} \end{cases}
\]

(6.40)

where \( R_\theta \) is the region occupied by the object \( \theta \) in the image space, \( D(\theta) \) measures the quality of \( \theta \) using some colour model. We use the colour model proposed in (Gamal-Eldin et al., 2012; Verdié and Lafarge, 2014), which is based on the contrast between foreground objects and background.

Here we consider the problem of counting blobs (e.g., cells, flamingoes) in the microscope or overhead satellite images. These objects inherently have circular shapes and are not mutually overlapping (see Figure 6.8(a) for an example). Therefore the non-overlapping constraint must be imposed on the object configurations. In MFIGP, this constraint is enforced via the interactions between any pair objects \( \beta \) and \( \gamma \), i.e.,

\[
C(\beta, \gamma) = \begin{cases} 0 & \text{if } \frac{R_\beta \cap R_\gamma}{\min(R_\beta, R_\gamma)} < 0.1 \\ l & \text{otherwise,} \end{cases}
\]

(6.41)

where \( l \) is a positive cost penalising overlapping objects. We set \( l \) to \( 10^2 \). Larger values for \( l \) (e.g., \( > 10^3 \)) is not recommenced since it will increase the energy approximation error (see Section 6.2.3). The per-model cost function is defined as \( C(\theta) = r_\theta^2 \) to penalize overly large objects.
6.3.3.1 Fitting Circles

Preprocessing. Given an image, we extract a set of edge pixels using Canny detector. These edge pixels are taken as the input data $X$. Note that for simplicity we only use the raw edge pixels though the oriented pixels could be used, e.g., for evaluating the spatial smoothness consistency.

Hypothesis sampling. We propose to sample circle candidates $H$ using the Hough transform. Specifically, every edge pixel $x = (x, y)$ votes for all possible bins of a 3D accumulator (the Hough space), in which each bin represents the centre $(x_c, y_c)$ and radius $r_c$ of a circle. Subsequently we detect 1000 peaks in the Hough space, which are taken as the circle hypotheses. The resolution of 1 pixel for the three axes (i.e., $x_c, y_c, r_c$) has been used. We limit the centres to be within the images and radii to be in a range $[20, 100]$ (pruning out too small or too large objects). The peaks are identified sequentially. In particular, after the peak with highest value has been detected, the next highest peak not within a radius of 1 from the first peak is found.

Parameter settings. We minimise the fitting energy (6.8) over $H$ with the following parameters: $\sigma = 1$, $\lambda = 0.1$, $K_{max} = 10$. The neighbourhood structure $N$ is constructed by Delaunay triangulation on $X$. We remove edges that are longer than 5 pixels.

![Image](image-url)

Figure 6.8: Circular object detection using an image of diatom. The image (a) contains various types of objects, however we are only interested in finding circular ones. Therefore the other objects become outliers as shown in the edge map (b). It can be seen that MFIGP with the non-overlapping constraint gives the best result. Note that there are some objects that none of the methods can detect. This is because of the insufficiency of the hypothesis set.
Chapter 6. **Fitting with Geometric Priors**

112

(a) Input image  
(b) Edge map of (a)  
(c) Hough transform  
(d) Jlinkage  
(e) AKSWH  
(f) PEARL  
(g) MFIGP  
(h) MBC

**Figure 6.9:** Circular object detection using an image of cells. Note that the cells are not well separated from the background. Also each cell tends to make two concentric circles. Consequently, the edge map (b) extracted from (a) is very ambiguous. Notice the duplicated circles produced by HT, Jlinkage, AKSWH and PEARL as shown in (c), (d), (e), (f), respectively. In contrast, MFIGP with non-overlapping constraints returns a better result. Not surprisingly, MBC performs badly for this dataset (see (h)) since the method heavily relies on the object intensity.

**Qualitative comparisons.** Figure 6.8 depicts the detection results of MFIGP and the other existing multi-model fitting methods such as the Hough transform, AKSWH (Wang et al., 2012), JLinkage and PEARL using an image of diatom. The result of MBC (Gamal-Eldin et al., 2012) is also added for comparisons. Note that MBC is a representative of the stochastic geometric framework (see Section 6.1), which fits disk models to the image and imposes the non-overlapping constraint as MFIGP. It can be seen that MFIGP mostly recovers all fully visible objects in the image (see Figure 6.8(g)). In contrast, the other fitting methods either return redundant objects or miss-detect some true ones. The issues are mainly due to the effects of noise and cluttered outliers. Figure 6.8(h) shows the result of MBC, where there are also some false- and miss-detections. This is because the colour model used by MBC (i.e., the contrast between the objects and the background) is not sufficiently effective to model the complex background and objects with non-uniform intensities. Figure 6.9 demonstrates another similar result using an image of cells.

6.3.3.2 Fitting Disks

Next we test MFIGP for detecting objects which have uniform colors, but are not well contrasted. Also the objects are often highly concentrated, see Figure 6.10(a) for a typical image of flamingoes (Gamal-Eldin et al., 2012). In such cases, fitting disk models to the image pixels is more appropriate than fitting circles to the edge pixels. However, the downside of this model is the high computational expense. To reduce the computational
cost, we roughly pre-segment the image into foreground and background based on the pixel intensity (see Figure 6.10(e) for an example). The foreground pixels are then used as the input data \( \mathcal{X} \).

**Experimental settings.** To generate the disk hypothesis set \( \mathcal{H} \), for each foreground pixel \( \mathbf{x} \), we sample all possible disks whose centres are at \( \mathbf{x} \) and radii fall within a range \([r_{\text{min}}, r_{\text{max}}]\). In this application, MFIGP does not enforce the local smoothness constraints since the disk model and the non-overlapping constraints implicitly account for the spatial smoothness. Other parameters are \( \sigma = 0.5, \lambda = 0.1, K_{\text{max}} = 500 \).

We compare MFIGP with PEARL and MBC. For PEARL method, the set of hypotheses \( \mathcal{H} \), the residual function, the label cost function and the noise scale \( \sigma \) are same as MFIGP. The neighbourhood structure \( \mathcal{N} \) is obtained by 4-connectivity. Theoretically, by fitting disks to the image, the strongly overlapping objects are unlikely to be returned since they simultaneously explain the same image region. Also, the spatial smoothness constraint would discourage the overlapping objects since the discontinuity is high when the objects overlap.

![Figure 6.10](image.png)

**Figure 6.10:** Counting flamingoes via disk model fitting. (a) is a satellite image of flamingoes (Gamal-Eldin et al., 2012). Based on the prior knowledge of the object intensity (i.e., the objects are brighter than the background), we roughly segment the image into the object foreground and background regions (shown in (e)). We then fit disk models to the foreground pixels. Since the image is overhead, the objects should not overlap each others. Thus, MBC and MFIG with the non-overlapping constraint give reasonably results, as shown in (c) and (d). Although PEARL with the spatial smoothness constraint could penalize the overlapping objects, it is unable to detect small and low-contrasted objects due to the over-smoothing (see (f)). Decreasing the smoothness penalty or increasing the model complexity cost still can not solve the issue completely (see (g) and (h)).
Ground truth  PEARL  MFIGP  MBC

Figure 6.11: Counting cells. Left to right are the ground truth and the detection results of MBC, MFIGP and PEARL. Observe that PEARL tends to merge nearby small cells into a bigger one because of the spatial smoothness constraint. Also overlapping objects cannot be completely avoided by PEARL. On the contrary, MFIGP and MBC result in better detections.

Qualitative comparisons. Figure 6.10 shows the performances of all the tested methods for detecting flamingoes from a satellite image. Our result is clearly comparable with MBC’s result. Figures 6.10(f), 6.10(g) and 6.10(h) show detection results by PEARL with different values of the weighting parameters $w_1$ and $w_2$ (see Equation (2.26) in Chapter 2). As expected, PEARL is able to produce the configuration with non-overlapping objects thanks to the spatial smoothness constraint, however the downside of the spatial smoothness is that some objects are miss-detected due to over-smoothing (the objects are relatively small with respect to the image size) (see Figure 6.10(f)). Reducing the smoothness penalty $w_1$ can detect more objects, but redundant objects start appearing (see Figure 6.10(g)). Increasing the model complexity weight $w_2$ still cannot completely solve the issue (see Figure 6.10(h)). Another result using a microscope image of cells (Lempitsky and Zisserman, 2010) is shown in Figure 6.11.

6.3.4 Planar Surface Reconstruction from 3D Point Clouds

Fitting multiple primitives to 3D point cloud has received much interest recently because of the maturity of the 3D reconstruction methods and the increasing popularity of advanced 3D laser scanners. The estimated primitives (e.g., planes, cylinders) can serve as a proxy for 3D scene modelling, or generally as meaningful abstractions of the data.

In this section, we aim to fit multiple planes to 3D data which has been reconstructed from a set of images of buildings. We select the publicly available point cloud Merton College 3 from the Oxford Colleges building reconstruction dataset\(^4\), and the point clouds used in (Farenzena et al., 2009) (specifically Pozzoveggiani and Piazza Dante). These point clouds were obtained using Structure from Motion (SfM).

\(^4\)http://www.robots.ox.ac.uk/~vgg/data/data-mview.html
Chapter 6. Fitting with Geometric Priors

Given a set of 3D points $\mathcal{X} \subset \mathbb{R}^3$, our goal is to compactly model the data by a small number of planes, where each plane is represented by parameters $\theta = (a, b, c, d) \in \mathbb{R}^4$. The distance (residual) from a point $x_n = (x_n, y_n, z_n)$ to plane $\theta$ is measured as:

$$r(x_n, \theta) = \frac{|ax_n + by_n + cz_n|}{\sqrt{a^2 + b^2 + c^2}}.$$  \hspace{1cm} (6.42)

In this application, we exploit the orthogonality and parallelism characteristics of the buildings to enhance the fitting. Accordingly, the geometric consistency between two planes $\beta$ and $\gamma$ is computed as:

$$C(\beta, \gamma) = \begin{cases} 0 & \text{if } A(\beta, \gamma) \in [0, 0.5] \cup [89.5, 90] \\ 10^2 & \text{otherwise} \end{cases}$$  \hspace{1cm} (6.43)

where $A(\beta, \gamma)$ measures the angle between two planes $\beta$ and $\gamma$. A constant per-model cost function is used, i.e., $C(\theta) = 1$, $\forall \theta \in \mathcal{H}$.

Parameter settings and hypothesis sampling. The neighbourhood structure $\mathcal{N}$ is obtained by a Delaunay triangulation on $\mathcal{X}$. We remove edges that are too long. We set $\sigma = 0.5$, $\lambda = 0.1$, $K_{max} = 10$. We generate a set $\mathcal{H}$ of 1000 plane hypotheses using the guided sampling method (Chin et al., 2012). Note that both Jlinkage and PEARL use the same inlier noise scale $\sigma$ and hypothesis set $\mathcal{H}$ as MFIGP. Other parameters are manually tuned to return the best fitting.

Qualitative comparisons. Figures 6.12, 6.13, 6.14 provide qualitative comparison results. We choose to plot the top views to easily percept the angles between the estimated planes. It is clear that the planes fitted by MFIGP more accurately reflect the actual design of the buildings. Observe that although the data segmentations of Jlinkage and PEARL seem reasonably correct, their fitted planes are considerably biased. This could be explained by the fact that the data errors do not follow a Gaussian distribution. The Merton College 3 dataset (see Figure 6.14) clearly demonstrates the case. Under the non-Gaussian noise, the models (planes) with low fitting residuals could be arbitrarily far from the true ones. On the contrary, in MFIGP, the fitted models balance between the fitting error and the geometric consistency, thus resulting in the better estimations.

6.3.5 Tuning Parameter Sensitivity

It is worth noting that our framework (MFIGP) has used a single set of tuning parameters for different datasets in each specific application, and there has been a relatively small adjustment for different applications. The results suggest that the proposed
Figure 6.12: Qualitative comparisons of Jlinkage, PEARL and MFIGP for fitting planes to Pozzoveggiani 3D point cloud. By observing the angles between the estimated planes, one can appreciate the improvements due to our high-level geometric priors. It can be seen that other methods produce a lot of misalignments of the estimated planes. Remark: arrows point to wrongly aligned planes with respect to the actual building design; detected outliers have been removed from the fitting results for clarity.

framework is not very sensitive to the choice of the tuning parameters. We suggest the \( \lambda \) parameter should not be much smaller than 0.1 since it would cause over-fitting. If we are working with images, an amount of a few pixels for \( \sigma \) is sufficiently good.

### 6.4 Summary

The importance of geometric priors has been acknowledged in various specific settings in computer vision. In this chapter, we proposed a general geometric fitting framework (named as MFIGP), which allows the integration of high-level geometric priors into the model hypothesis evaluation. Besides the essential hypotheses evaluation using an energy minimisation, our framework can work with various customisable routines, such as hypothesis generation, geometric constraint specifications for different types of models.

Empirically, MFIGP outperforms the existing model fitting methods on a variety of computer vision applications such as homography estimation, vanishing point detection, planar surface reconstruction and circular object detection. We showed that geometric constraints are useful to eliminate the influences of (non-normal) noise and (non-uniform)
Chapter 6. Fitting with Geometric Priors

Figure 6.13: Qualitative comparisons of Jlinkage, PEARL and MFIGP for fitting planes to Piazza Dante point cloud. The caption is the same as in Figure 6.12.

Figure 6.14: Qualitative comparisons of Jlinkage, PEARL and MFIGP for fitting planes to Merton College 3 3D point cloud. The caption is the same as in Figure 6.12.
outliers. We believe that our fitting framework will prove to be useful to many other multi-model fitting applications, where prior knowledge of the scenes/objects is available and can be translated into interactions between structures.
Chapter 7

Conclusions and Future Work

Recovering the underlying model parameters from vision data is a challenging problem. Since the available data is often contaminated by noise and outliers, almost all of the robust fitting criteria do not have closed-form solutions. Therefore, for computational tractability, sampling and evaluating model hypotheses is a technique of choice to obtain good estimates. Nevertheless, generating hypotheses (usually constructed by fitting models onto randomly sampled data subsets) is an expensive process, thus critically affects the overall estimation efficacy. Moreover, in many cases, the number of structures (model instances) embedded in the data is unknown, thus testing the sampled model hypotheses to select the best models is also a difficult problem. The research conducted in this thesis partially addressed those challenges.

In this last chapter, we conclude the thesis with a short summary of key contributions, and possible research directions in the robust estimation field.

7.1 Summary of Contributions

The main contributions presented in this thesis are:

1. A novel robust estimator (ARJMC, Algorithm 3.3) based on a Reversible Jump Monte Carlo Markov Chain algorithm. Different from traditional fitting methods which conduct hypothesis sampling and testing in two independent steps, ARJMC effectively combines hypothesis sampling and evaluation (fitting optimisation) in a unified framework. ARJMC relies on an adaptive hypothesis sampler to simulate a Markov chain, which is proved to asymptotically converge to the target distribution. (The target distribution encapsulates the fitting criterion.) ARJMC
demonstrates competitive accuracy and a significant gain in efficiency when benchmarked against the state-of-the-art estimators.

2. A fitting algorithm (RCMSA, Algorithm 4.1) which is based on a simulated annealing with graph-cuts. Similar to ARJMC, RCMSA also jointly conducts hypothesis sampling and fitting optimisation. However, as opposed to ARJMC, RCMSA simultaneously optimises model parameters (using hypothesis sampling) and point segmentation (using graph-cuts), which is enforced to be spatially smooth. RCMSA is shown not only more efficient but also more accurate than many existing robust fitting methods, including ARJMC.

3. An efficient hypothesis sampler (RCM, see Section 4.2) based on Random Cluster Model. Different from conventional sampling techniques which generate hypotheses from minimal subsets of data, our RCM sampler uses larger-than-minimal subsets. We showed that using larger-than-minimal subsets (of inliers) can result in more accurate hypotheses. To quickly generate such subsets of inliers, RCM adaptively partitions the data into clusters, which are then considered as data subsets. In effect, members of the subsets are sampled jointly, thus RCM can avoid the exponential growth of sampling effort in finding all-inlier subsets. Apparently, this sampling strategy is different from the standard methods, where members of the subsets are chosen in succession. The RCM sampler (when integrated into the fitting Algorithm 4.1) shows its superior efficacy over the existing sampling methods.

4. A new multi-structure fitting algorithm (MFIGP, Section 6.2) which seamlessly integrates geometric constraints into the hypothesis evaluation. MFIGP encodes the geometric constraints into a single energy function via interactions between every pair of geometric models. The function is then used to evaluate model hypotheses via an energy function optimisation. The experimental studies show that MFIGP is very robust against noise and outliers without making assumptions on Gaussian noise and uniform outliers.

5. Extensive experiments using various interesting computer vision applications such as two-view motion segmentation, multiple homographies detection, single-view vanish point detection, circular object detection and 3D planar surface reconstruction from 3D point clouds. Moreover, we developed an automatic augmented reality system for videos, which is based on the RCMSA estimator.


7.2 Future Research Directions

Even though the work in this thesis has made considerable progress in the field of robust parameter estimation in computer vision, several important issues remain unresolved. In the following, we suggest a number of possible future directions.

1. **Large-scale geometric model fitting.** Though the robust techniques proposed in this thesis have been comprehensively tested on many multi-structure vision datasets, the techniques do not scale well to large-scale datasets — those contain up to millions of data points and hundreds to thousands of structures (e.g., 3D point clouds of urban environments with many planar surfaces). In such cases, randomly sampling hypotheses and computing residuals from all the data points to the sampled hypotheses (for hypothesis evaluation) apparently become computationally infeasible. It would be interesting and useful to investigate how to extend the frameworks presented in this thesis, or innovate completely new robust techniques for large-scale problems in the future.

2. **Exploiting geometric priors.** In many cases, it is quite impossible to estimate exactly the quantity (and parameters) of structures in the data without a precise definition of what is a structure. Implicitly or explicitly, a structure is traditionally defined using a noise scale and population size, e.g., given a noise threshold, a valid structure should occupy at least 10% of the data points. However, in practice different structures may have very different noise scales and population sizes, preventing robust methods from estimating the structures accurately. One avenue for improving estimation accuracy is maximally exploiting geometric prior knowledge of the data. By utilising such geometric constraints, structures can be described more precisely; a valid structure not only stands alone by itself, but also is consistent with others, as preliminarily demonstrated in Chapter 6. We believe that this research direction is promising and needs further investigations. For example, how to effectively impose the geometric constraints without leading to intractable or hard-to-optimise objective functions?
Bibliography


