Exploiting Low-dimensional Structures in Motion Problems

Zhuwen Li

A THESIS SUBMITTED FOR THE DEGREE OF

Doctor of Philosophy

Department of Electrical & Computer Engineering

National University of Singapore

2014

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Zhuwen Li

October 2, 2014

Acknowledgements

I would like to thank my research supervisors, Prof. Loong-Fah Cheong and Prof. Steven Zhiying Zhou, for their guidance during my candidate years. Prof. Cheong, in his wisdom and patience, has taught me everything I need to know about research, including genuine research motive, academic writing, professional presentations etc. I will never forget the time we spent in preparing my oral presentation in ICCV 2013. Prof. Zhou, with his passion and drive, has taught me that diligence is the key ingredient for success. He has also widened my research view and kept me interested in different research topics, which help to inspire new ideas.

I owe deep appreciation to my friend Choon Meng Lee, who is very supportive and patient whenever I have questions about mathematical derivations or other problems. He has also led me to learn about convex optimization, which has proven very useful in my research pursuit.

Special thanks to my friends and lab mates, Jiaming Guo, Tran Lam An, Wen Kou, Yuxiang Wang, Ye Luo, Zhi Gao, Shahzor Ahmad, Shuaicheng Liu, Zhe Wu, Zhaopeng Cui, Kaimo Lin, Rui Huang, Zhenlong Zhou, Yinda Zhang, Danping Zou, Chengyao Shen and Qiang Zhou for the discussions, seminars, lunches, dinners, every movie and hiking.

Finally, I would like to thank my parents' selfless love. I still remember the last time I went home, I was preoccupied with

this thesis. They understood me and never complained about it. I know I owe them everything and my gratitude to them is beyond what I can express in words.

Contents

Contents **v**

Summary **ix**

List of Tables **xi**

List of Figures **xiii**

List of Abbreviations **xvii**

1 Introduction **1**

1.1 Motion Segmentation 3

1.2 Model Selection 4

1.3 Visual Tracking 5

1.4 Structure of the Thesis 6

2 Collaborative Clustering for Perspective Motion Segmentation **9**

2.1 Introduction 10

2.1.1 Related work 18

2.2 The TPV Motion Subspace 22

2.3 Clustering Motion Subspaces 24

2.3.1 Sparse subspace clustering 24

2.3.1.1 Single image pair 25

2.3.1.2 Multiple image pairs 26

2.3.1.3 Handling ambiguous matches 28
2.3.2 Merging via coefficient analysis 28
2.4 Experiments 32
2.4.1 Results on single image pairs 32
2.4.2 Results on multiple image pairs 33
2.5 Conclusions 38

3 Simultaneous Clustering and Model Selection 41
3.1 Introduction 42
3.2 Related works 45
3.3 Clustering with Model Selection 47
3.3.1 Problem formulation 47
3.3.2 Solver 49
3.4 Constrained Boolean Matrix Factorization 52
3.5 Experiments 55
3.5.1 Synthetic data 56
3.5.1.1 Different noise levels 58
3.5.1.2 Varying group numbers 58
3.5.2 Motion segmentation 60
3.5.3 Face clustering 61
3.6 Discussion and Conclusion 63

4 Visual Tracking via Sparsity Patten Learning and An Alternative Sparsity Model 65
4.1 Introduction 66
4.2 Related Works 71
4.3 Background 73
4.3.1 Particle filter for visual tracking 73
4.3.2 The ℓ_1 tracker 74
4.4 Visual Tracking via Sparsity Pattern Learning 75

4.4.1 Sparsity Pattern Learning 75
4.4.2 Fast update of the sparsity patterns 77
4.5 An Alternative Sparse Representation Approach 79
4.6 Experiments 80
4.7 Conclusions 84

5 Conclusions and Future Works **89**
5.1 Summary of Contributions 89
5.2 Future Works 91
5.2.1 Motion segmentation 91
5.2.2 Clustering and model selection 91
5.2.3 Visual tracking 92

Bibliography **93**

Appendix A **107**
A.1 Proof of Theorem 1 107
A.2 Proof of Theorem 2 107

Appendix B **109**
B.1 Blockwise Inversion 109
B.2 Sherman - Morrison Formula 109

Appendix C **111**
C.1 More Visual Tracking Results based on Attributes 111

Summary

Video-related problems often involve high-dimensional data analysis. In this thesis, we explore the theoretical and algorithmic aspects of their low-dimensional structures, including sparsity in vectors and low rank matrices, among others. Specifically, we address various motion-related problems such as motion segmentation and object tracking.

In the first part of the thesis, we re-formulate the 3-D motion segmentation from two perspective views as a subspace clustering problem, utilizing the classic epipolar constraint of an image pair. We then combine the point correspondence information across multiple image frames via a collaborative clustering step, in which tight integration is achieved via a mixed norm optimization scheme. Our method effectively addresses several longstanding real-world challenges in the motion segmentation problem, including perspective effects, model selection and missing data, obtaining state-of-the-art performance in handling the aforementioned challenges.

In the preceding, the model selection methods to estimate the number of motion groups is based on an over-segment and merge approach, where the merging step is based on the property of the ℓ_1-norm of the mutual sparse representation of two over-segmented groups. In the next part of the thesis, we propose a more general model selection approach, which only needs an affinity matrix as input. This approach solves clustering and

model selection in a joint manner with an indicator matrix formulation, in which the clustering cost is penalized by a Frobenius inner product term and the group number estimation is achieved by a rank minimization. We further add a sparsity term to discover structures in the data. Rather than adopting the conventional convex relaxation approach wholesale, we represent the original problem more faithfully by taking full advantage of the particular structure present in the optimization problem and solving it efficiently using the Alternating Direction Method of Multipliers. The highly constrained nature of the optimization provides our algorithm with the robustness to deal with the varying and often imperfect input affinity matrices arising from different applications and different group numbers.

Lastly, we exploit the low-dimensional structures present in the object tracking problem to speed up the ℓ_1 Tracker. We learn the coefficient patterns of the sparsity model and solve small scale ℓ_2 norm minimization problems instead of the high cost ℓ_1 norm minimization problems, resulting in a very fast tracking algorithm. We also propose a novel sparsity model by considering the problem from a different angle, leading to an algorithm with better tracking robustness.

List of Tables

2.1 Classification errors (%) for sequences with 2 motions 33

2.2 Classification errors (%) for sequences with 3 motions 34

2.3 Classification errors (%) on *Hopkins155* 34

2.4 Classification results on 62-clip dataset 36

2.5 Classification results on 62-clip dataset (only considering sequences where the number of motions is correctly estimated) 36

3.1 RI on *Hopkins155* 61

List of Figures

2.1 Motion segmentation results of two sequences with strong perspective effects using SSC. The ground truths are shown in (a) and (c), and the SSC results in (b) and (d) respectively. In (b), part of the green object is classified as belonging to the background, and in (d) the green object captures some of the background points. 13

2.2 (a) 60 trajectories obtained with the full-length requirement, and (b) 524 trajectories without the full-length requirement. (c) The data matrix, with black area indicating missing entries. 14

2.3 Illustration of the $\ell_{1,1,2}$ norm minimization. The entries (i, j) of $\mathbf{C}^{(l)}$ should be sparse and its support set should be consistent across different $\mathbf{C}^{(l)}$. 26

2.4 Illustration of the magnitudes of the ℓ_1 norm when point p is represented by points from the "red" subspace (left) or from the "bule" and "green" subspace (right). The ℓ_1 norm of the later case is larger. 29

2.5 Qualitative results of the real data with missing entries. The segmentation results of the 50-th frames of the sequences are presented. From left to right are the "Van", "Girl" and "Swing" clips. 39

3.1 Left: A contaminated affinity matrix $\mathbf{A}$ with 5 clusters. Right: The recovered $\mathbf{G}$ contains 5 almost perfect blocks. Further processing by the proposed Boolean matrix factorization algorithm will obtain perfect blocks from this $\mathbf{G}$. . . 44

3.2 Comparison on the Synthetic Data when the noise level changes. 57

3.3 Comparison on the Synthetic Data when the number of subspaces changes. 59

3.4 Examples of face images. Images of 6 subjects (32 images for each subject) are shown here, where each row corresponds to a subject. 61

3.5 Comparison on the Extended YaleB dataset with increasing number of subjects. 62

4.1 The coefficient vectors of the best patches in different frames are shown in sequence. The supports of these coefficient vectors contain only a few combinations. And coefficient vectors in neighboring frames has shown similar patterns. . . 68

4.2 Overall visual tracking performance plots 83

4.3 Visual tracking performance plots - Occlusion 85

4.4 Visual tracking performance plots - Deformation 86

4.5 Visual tracking performance plots - Illumination Variation . 87

C.1 Visual tracking performance plots - Fast Motion 111

C.2 Visual tracking performance plots - Background Clutter . . . 112

C.3 Visual tracking performance plots - Motion Blur 113

C.4 Visual tracking performance plots - In-Plane Rotation 114

C.5 Visual tracking performance plots - Low Resolution 115

C.6 Visual tracking performance plots - Out-of-Plane Rotation . 116

C.7 Visual tracking performance plots - Out-of-View 117

C.8 Visual tracking performance plots - Scale Variation 118

List of Abbreviations

ADMM Alternating Direction Method of Multipliers

AIC Akaike Information Criterion

ALC Agglomerative Lossy Compression

APG Accelerated Proximal Gradient

BMF Boolean Matrix Factorization

BPR Bounded Particle Resampling

CBMF Constrained Boolean Matrix Factorization

CC Correlation Clustering

CT Compressive tracking

CXT Context Tracker

EM Expectation Maximization

GH Gap Heuristic

GPCA Generalized Principal Component Analysis

GRIC Geometrically Robust Information Criterion

IVT Incremental Visual Tracking

KKT Karush-Kuhn-Tucker

LBF Local Best-fit Flats

LRR Low Rank Representation

LSA Local Subspace Affinity

MCMC Markov Chain Monte Carlo

MDL Minimum Description Length

MIL Multiple Instance Learning

MRF Markov Random Field

MSMC Multi-Scale Motion Clustering

NNMF Non-Negative Matrix Factorization

NRSfM Non Rigid Structure from Motion

ORK Ordered Residual Kernel

PCA Principal Component Analysis

PSD Positive Semi-Definite

QSAP Quadratic Semi-Assignment Problem

RAS Robust Algebraic Segmentation

RIP Restricted Isometry Property

RI Rand Index

SCAMS Simultaneous Clustering and Model Selection

SCM Sparsity based Collaborative Model

SOD Second Order Difference

SPL Sparsity Pattern Learning

SSC Sparse Subspace Clustering

ST Soft Thresholds

SVD Singular Value Decomposition

TLD Tracking-Learning-Detection

TPV Two-Perspective-View

VTD Visual Tracking Decomposition

Chapter 1

Introduction

Our society has invested massively in the collection and processing of data of all kinds, resulting in an overwhelming amount of data generated and collected every day. Among these data, many of them are high-dimensional. For example, a single digital image with modest quality contains more than a million pixels. Such high dimensionality is usually considered impossible to analyze using classic techniques in statistics because the number of data points required to successfully fit a general Lipschitz function increases exponentially with the dimension of the data. This is often described metaphorically as the "curse of dimensionality" [35]. Fortunately, it is often valid that real data have some certain low-dimensional structures, such as sparsity and low-rank. In this case, the high dimensionality can result in desirable data redundancy which makes it possible to provably and exactly recover the correct parameters of the structure. This is often referred to as the "blessing of dimensionality" [35].

This phenomenon appears in many computer vision problems. For example, face recognition community has observed that the images of faces under varying illumination and expression lie on low-dimensional subspaces [15]. This observation motivates many dimension reduction approaches to exploiting the low-dimensional structures in the raw image data. The ear-

liest work is the famous Eigenface [104], which essentially adopt principal component analysis (PCA) [53] to select an optimal low-rank approximation in the ℓ_2 sense. Later works include Fisherfaces [15], Laplacianfaces [49] and some variants [57, 62]. More recently, the theories from compressive sensing [23, 36] offer better representation of the data, leading to a breakthrough in face recognition [113]. Other similar examples from computer vision community include foreground detection [40], non rigid structure from motion (NRSfM) [32], photometric stereo [114], motion segmentation [39, 68], etc. In all these examples, compressive sensing plays a key role in recent developments of the algorithms and make significant advancements in performance.

The advent of the compressive sensing builds upon the fundamental fact that we can reconstruct sparse or compressible signals accurately via ℓ_1 minimization (convex relaxation of ℓ_0) from a very limited number of measurements if the sensing matrix obeys the restricted isometry property (RIP) property [20, 23, 36]. This result equivalently shows its ability to correct sparse errors/outliers when recovering signals, leading to success in handling occlusions in face recognition [113] and visual tracking [73]. In the spectral domain, since cardinality corresponds to the rank of a matrix, sparsity corresponds to low-rank. Thus, nuclear norm (defined as the sum of singular values) is a convex relaxation of the rank function. Notably, nuclear norm minimization are shown effective in completing a partially observed low-rank matrix (low-rank matrix completion) [22] and in recovering a low-rank matrix with sparse corruptions (Robust PCA) [21]. This result leads to success in foreground detection in [40] and shadow/highlight removal in photometric stereo [114].

With these main results of compressive sensing at our disposal, the challenge now is to identify and model the low-dimensional structures in the various specific research problems. For example, the low-dimensional

subspace might contain structures such as a union of subspaces or other sparsity patterns. The class of "sparse" models could also be extended beyond sparse vectors and low rank matrices to include other low-dimensional structures such as a sum of few permutation matrices, which are ubiquitous in many problem domains. We envisage the use of these new class of sparse models will offer more generality and better performance than many conventional approaches to problems in video analysis, thereby overcoming the challenges that limit the use and growth of video analytic software. In this thesis, we focus on motion-related problems and aim to exploit the underlying structures of the data inherent in these problems. More specifically, we consider the motion segmentation problem and the attendant model selection, as well as the object tracking problem.

1.1 Motion Segmentation

Motion segmentation is a challenging problem in visual motion analysis. The idea is to segment the scene into multiple rigid-body motions, based on the point trajectories or optical flow observed in multiple camera views. It is a challenging problem because it requires simultaneous estimation of an unknown number of motion models, without knowing which measurements correspond to which model. This problem can be cast as a subspace clustering problem in which point trajectories associated with each motion are to be clustered together. Recent works [39, 68, 83] introduced compressed sensing techniques to subspace segmentation. We seek to extend these sparsity-based techniques as there are many difficulties with the current motion segmentation techniques. For instance, most current techniques cannot handle perspective effects because of the subspace assumption. Moreover, they cannot automatically estimate the number of motion clusters and can only tolerate a small amount of missing entries

and outliers. Our novel contributions include:

- better capturing the global structures of data than current sparse techniques via the mixed norm approach.
- better estimating the number of motion clusters via a over-segment and merge approach, where the merging step is based on the property of the ℓ_1-norm of the mutual sparse representation of two over-segmented groups.
- better handling the conditions where missing data, noise, and outliers are prevalent.

1.2 Model Selection

As have been mentioned above, estimating the number of motion groups remains very much an open problem in motion segmentation. In the wider context, this is also known as the model selection problem and it appears in many clustering or segmentation tasks, such as image segmentation [91], protein clustering [70] and so on. Just like in the case of motion segmentation, model selection is a common and essential problem in all such tasks.

A common way to estimate the group number follows from the spectral clustering framework [71]; it counts the number of zero eigenvalues of the Laplacian matrix of the affinity graph. However, this method does not perform very well in practice when the data contain structures at different scales of size and density, and when data are contaminated by noise. In these cases, these eigenvalues deviate from zero in a complex manner, and it is non-trivial to determine the number of eigenvalues close to zero in a robust manner. While this method belongs to the spectral graph method [3, 68, 87, 93, 97], the other kind of method is the information-theoretic method [2, 55, 72, 85, 94, 100], which aims to balance the goodness of fit

against the complexity of the model. The major drawback of this kind of method is that they are usually model-dependent. To overcome these problems, we propose in this thesis a general and robust algorithm to perform simultaneous clustering and model selection (SCAMS) with only an affinity matrix as input. To explore the low-dimensional structures hidden in the affinity matrix, we apply the low rank and sparsity constraints and solve the original non-convex problems, yet by taking advantage of the particular structure present in the optimization problem, we are able to put forth a tractable solution. Note that in many cases, the convex proxy to the original NP-hard problems may not be a good approach - an approximate solution to the right problem can be better than the exact solution to the wrong problem. This problem is especially severe when there are outliers very large in magnitudes, a situation that could very well arise in many real problems. In this case, there might be a need to represent the original problem more faithfully rather than just adopting the conventional convex relaxation approach.

1.3 Visual Tracking

It has been shown that promising tracking accuracy can be achieved by modeling the target appearance by a sparse representation of the template set, resulting in the so-called ℓ_1 Tracker [73]. However, the ℓ_1 Tracker is limited by its high computational cost which is dominated by that of the ℓ_1-norm minimization. Though significant acceleration is achieved by the Minimum Error Bound [74] and a fast solver to the ℓ_1-norm minimization using Accelerated Proximal Gradient (APG) [13], it is still not fast enough for a normal PC. To further accelerate the ℓ_1 Tracker, we propose to learn the sparsity patterns for the template set, performing a quick update on these patterns when the templates are changed. With the learnt sparsity

patterns, we are able to recover the sparse coefficients by ℓ_2 norm minimization. Note that the ℓ_1-norm minimization only needs to be carried out when the template set is updated, thus leading to significant saving of computational cost.

Other than the preceding proposed acceleration, we also propose a novel sparsity model to describe the visual tracking problem. Unlike previous methods, we model a template appearance using a sparse representation of the candidate set, instead of the other way round (i.e. modeling a candidate using the template set). As a result, a large number of candidates can be filtered out, followed by some simple manipulations to determine the best candidate from the remaining small set.

1.4 Structure of the Thesis

The organization of this thesis is as follows.

In Chapter 2, we better exploit the sparsity patterns in the 3D motion segmentation problem, such that several well-known real-world challenges in this problem are effectively addressed; these challenges include perspective effects, missing data, and unknown number of motions. We first formulate the 3-D motion segmentation from two perspective views as a subspace clustering problem, utilizing the epipolar constraint of an image pair. We then combine the point correspondence information across multiple image frames via a collaborative clustering step, in which tight integration is achieved via a mixed norm optimization scheme. For model selection, we propose an over-segment and merge approach, where the merging step is based on the property of the ℓ_1-norm of the mutual sparse representation of two over-segmented groups. The resulting algorithm can deal with incomplete trajectories and perspective effects substantially better than state-of-the-art two-frame and multi-frame methods. Part of the results in

this chapter appeared in [65].

In Chapter 3, we jointly address the clustering and model selection problems in a more general setting with an indicator matrix formulation, in which the clustering cost is penalized by a Frobenius inner product term and the group number estimation is achieved by a rank minimization. As affinity graphs generally contain positive edge values, a sparsity term is further added to avoid the trivial solution and exploit the structures. Rather than adopting the conventional convex relaxation approach wholesale, we represent the original problem more faithfully by taking full advantage of the particular structure present in the optimization problem and solving it efficiently using the Alternating Direction Method of Multipliers. The highly constrained nature of the optimization provides our algorithm with the robustness to deal with the varying and often imperfect input affinity matrices arising from different applications and different group numbers. Part of the results in this chapter appeared in [64]

In Chapter 4, we accelerate the ℓ_1 Tracker by learning the sparsity patterns of the template set. With the learnt sparsity patterns, we are able to recover the sparse coefficients of candidate samples by some small-scale ℓ_2 norm minimizations, resulting in a very fast algorithm. In addition to that, we propose an alternative sparsity model, which models the template appearance by a sparse approximation over the candidate set. In this case, a large number of candidates are immediately filtered out according to whether they are chosen to represent the templates or not. Then the optimal candidate is chosen as the one with the largest observation likelihood from the retained candidate set. This sparsity model exploits the tracking problem from a novel perspective and achieves better performance even with the simplest setting.

In Chapter 5, we conclude the thesis with some discussions and list some open questions and potential future developments related to this thesis.

Chapter 2

Collaborative Clustering for Perspective Motion Segmentation

This chapter addresses real-world challenges in the motion segmentation problem, including perspective effects, missing data, and unknown number of motions. It first formulates the 3-D motion segmentation from two perspective views as a subspace clustering problem, by utilizing the epipolar constraint of an image pair. It then combines the point correspondence information across multiple image frames via a collaborative clustering step, in which tight integration is achieved via a mixed norm optimization scheme. For model selection, we propose an over-segment and merge approach, where the merging step is based on the property of the ℓ_1-norm of the mutual sparse representation of two over-segmented groups. The resulting algorithm can deal with incomplete trajectories and perspective effects substantially better than state-of-the-art two-frame and multi-frame methods. Experiments on a 62-clip dataset show the significant superiority of the proposed idea in both segmentation accuracy and model selection.

2.1 Introduction

Scenes with multiple motions are very common in reality, which leads to an increasing interest in dynamic scene analysis. Among all issues in dynamic scene understanding, 3-D motion segmentation is an essential problem. It refers to the problem of clustering trajectories according to n motions. These trajectories correspond to several objects undergoing n different rigid-body motions relative to a static or moving camera. The success of motion segmentation helps to further develop applications in dynamic scenes, such as tracking, recognition, reconstruction, etc. The challenge in this problem is to segment the trajectories only considering motion cues in the scene. Previous approaches to this problem can be roughly separated into the multi-frame and the two-frame methods.

Multi-frame methods. Multi-frame methods have been studied mostly under the affine assumption. This kind of methods can be traced back to the early work of [17, 98], and the ensuing multi-frame methods [31, 39, 42, 45, 56, 68, 83, 96, 105, 118] are based on this assumption and one can solve the problem using either a factorization or a subspace separation framework. Under the affine assumption, the trajectories of a rigid motion across multiple frames lie in an affine subspace with a dimension of no more than 3, or a linear subspace with a dimension of at most 4. That is, let $\mathbf{x}_{fp} \in \mathbb{R}^2$ be the image coordinate of 3-D points $\widetilde{\mathbf{X}}_p \in \mathbb{P}^3$ in frame f, where "$\sim$" denote the homogeneous representation, then

$$\mathbf{x}_{fp} = \mathbf{A}_f \widetilde{\mathbf{X}}_p \tag{2.1}$$

where $\mathbf{A}_f = \mathbf{K}_f \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{2\times 4}$ is the affine camera matrix for frame f, which depends on the camera intrinsic parameters $\mathbf{K}_f \in$

$\mathbb{R}^{2\times3}$, the camera relative rotation matrix $\mathbf{R}_f \in \mathbb{R}^{3\times3}$ and the translation vector $\mathbf{t}_f$.

Assume there are F frames and P 3-D points, from (2.1), the measurement matrix $\mathbf{W} \in \mathbb{R}^{2F\times P}$, whose columns are the image point trajectories, can be presented as

$$\mathbf{W} = \overbrace{\begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix}}^{2F\times P} = \overbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_F \end{bmatrix}}^{2F\times 4} \overbrace{\begin{bmatrix} \widetilde{\mathbf{X}}_1 & \cdots & \widetilde{\mathbf{X}}_P \end{bmatrix}}^{4\times P} \tag{2.2}$$

It is immediate that $rank(\mathbf{W}) \leq 4$. Since the last entry of $\widetilde{\mathbf{X}}_p$ is always 1, the trajectories lie in an affine subspace of dimension at most 3. However, most works consider the trajectories lie in a linear subspace of dimension at most 4. Therefore, motion segmentation problem can be formulated based on a factorization or subspace separation framework. For independent rigid-body motions, trajectories undergoing different motions live in an independent linear subspace and have no intersection [31]. For articulated motions, trajectories undergoing different motions live in an different linear subspace but have one or two dimensional intersection [101, 117, 118].

As an extension to perspective camera model, the projection equation (2.1) becomes

$$\lambda_{fp}\widetilde{\mathbf{x}}_{fp} = \mathbf{P}_f\widetilde{\mathbf{X}}_p \tag{2.3}$$

where λ_{fp} is the projective depth of point p relative to frame f, $\widetilde{\mathbf{x}}_{fp} \in \mathbb{P}^2$ denote the homogeneous representation of the image coordinate and $\mathbf{P}_f \in \mathbb{R}^{3\times4}$ is the general projective matrix for frame f. Because λ_{fp} is unknown for all trajectories, the measurement matrix is now a function of λ, which

can also be factorized as before

$$\mathbf{W}(\lambda) = \overbrace{\begin{bmatrix} \lambda_{11}\widetilde{\mathbf{x}}_{11} & \cdots & \lambda_{1P}\widetilde{\mathbf{x}}_{1P} \\ \vdots & \ddots & \vdots \\ \lambda_{F1}\widetilde{\mathbf{x}}_{F1} & \cdots & \lambda_{FP}\widetilde{\mathbf{x}}_{FP} \end{bmatrix}}^{3F\times P} = \overbrace{\begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_F \end{bmatrix}}^{3F\times 4} \overbrace{\begin{bmatrix} \widetilde{\mathbf{X}}_1 \cdots \widetilde{\mathbf{X}}_P \end{bmatrix}}^{4\times P} \tag{2.4}$$

It is clear from (2.4) that if λ_{fp} is known for all trajectories, the rank of $\mathbf{W}(\lambda)$ is less than 4, thus subspace clustering can be applied to segment the scene into multiple motions. The Sturm/Triggs(ST) algorithm [95] analyzed the case of static scene to recovery the structure and camera pose of the scene, while Li *et al.* [63] extended the iterative ST algorithm [48, 102] to the case of multiple rigid-body motions by simultaneously estimating the depth information and separating the motion groups iteratively.

Two-frame methods. Two-view methods are usually based on the epipolar geometry, and are thus capable of handling perspective effects. The motion model fitting and selection are carried out by either statistical methods [52, 61, 88, 100] or algebraic methods [84, 106, 112]. The statistical methods start with a random or guided sampling, followed by estimating the likelihoods of the generated motion model hypotheses, at the end of which the models with high quality (likelihood) are selected. The algebraic methods fit a mixture of fundamental matrices by linearizing the multilinear relationship between correspondences in a high-dimensional space. Then correspondences are assigned to the motion models with the smallest fitting errors.

The multi-frame methods have been better developed, partly due to the elegance of its formulation and partly due to the release of the *Hopkins155* database [103], which contains largely clips with little perspective effects. Recently, the class of multi-frame affine methods has been further enlarged by the powerful subspace clustering algorithms [39, 68]. However, we argue

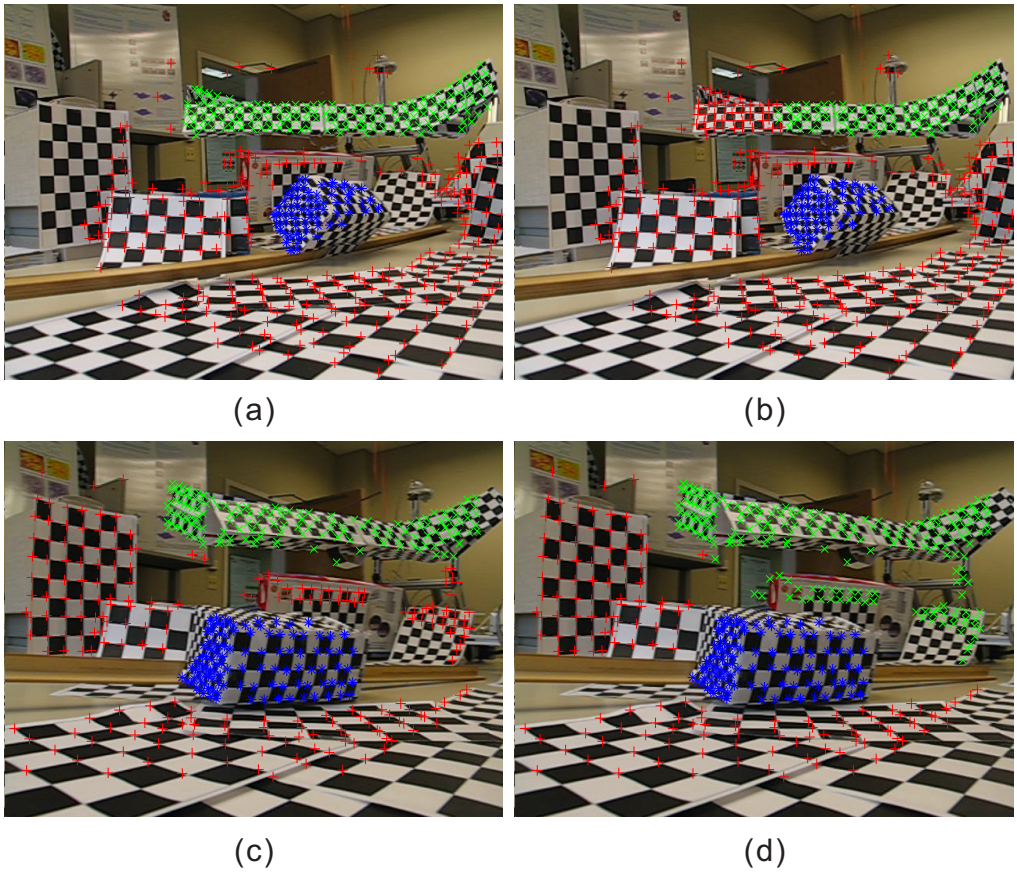

Figure 2.1: Motion segmentation results of two sequences with strong perspective effects using SSC. The ground truths are shown in (a) and (c), and the SSC results in (b) and (d) respectively. In (b), part of the green object is classified as belonging to the background, and in (d) the green object captures some of the background points.

that the current crop of multi-frame affine methods does not confront several real world issues, despite ever-decreasing and near perfect classification rate on *Hopkins155*. There are three major drawbacks of the multi-frame affine methods when compared to the two-frame methods.

Firstly, multi-frame affine methods suffer from their inability to deal with perspective effects, while this presents no problem in the two-frame method; it becomes a significant consideration when using shorter lenses for shooting outdoor sequences. Figure 2.1 shows the results of two sequences with perspective effects from *Hopkins155*; these results are produced by the state-of-the-art clustering algorithm – sparse subspace clustering (SSC) [39]. Compared to the near zero errors achieved by SSC for the other

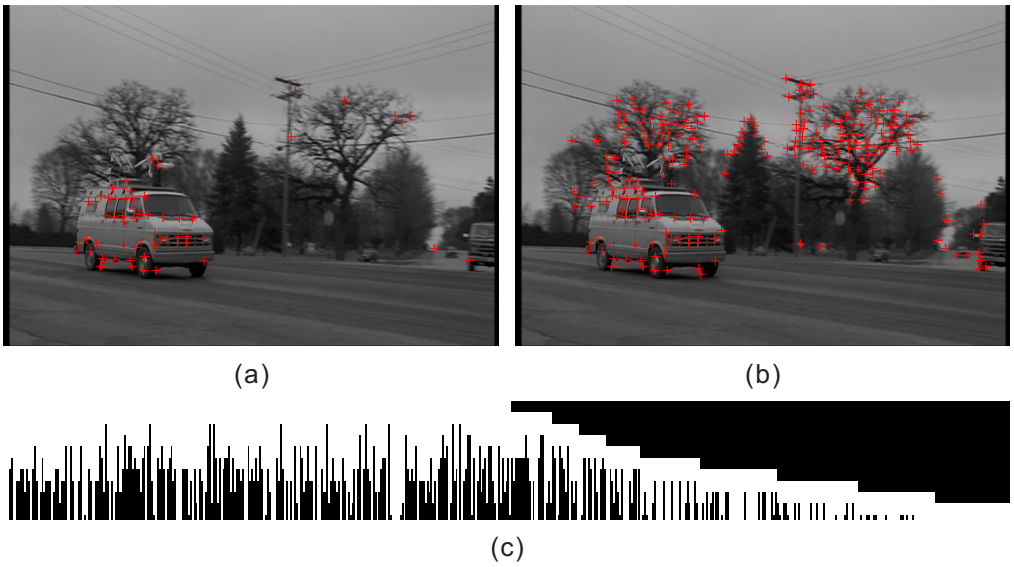

Figure 2.2: (a) 60 trajectories obtained with the full-length requirement, and (b) 524 trajectories without the full-length requirement. (c) The data matrix, with black area indicating missing entries.

sequences in *Hopkins155* without strong perspective effects, the erroneous segmentation results in these clips are especially notable: in Figure 2.1(b), part of the green object is classified as belonging to the background, and in Figure 2.1(d) the green object captures some of the background points.

Secondly, multi-frame affine methods generally require the trajectories to have full-length. If one simply filters out the trajectories which are absent in some frames, the density of the trajectories is likely to be significantly decreased, resulting in lack of coverage of many parts of the sequence. The full-length requirement also makes it difficult to deal with objects entering into or departing from the scene and suffering from temporary occlusion. Figure 2.2(a) shows the feature points of the "delivery van" data with the full-length requirement on the trajectories. It is observed that they are much sparser than the density of those in Figure 2.2(b), which only requires the trajectories to appear in at least two frames. Clearly, two-frame methods suffer to a much lesser extent from the missing entry issue. One may argue that matrix completion techniques can help to fill in the missing entries [26]. However, Candès and Tao [24] have proven a lower bound on

the necessary number of uniformly distributed samples, below which no algorithm can guarantee correct recovering of the missing entries. Unfortunately, motion segmentation data often violate this condition. Figure 2.2(c) shows the data matrix of the "delivery van" data, which has about 50% missing entries and is non-uniformly distributed. Even it is by no means the most challenging data, it is difficult to recover the missing entries.

Thirdly, the number of motion groups is usually assumed to be known *a priori* for multi-frame affine methods. It is indeed a strong indication that model selection is actually difficult for motion segmentation. Related to this issue is the fact that the number of motion groups in each clip of the *Hopkins155* dataset remains unchanged throughout the frames, which makes it easy to indulge in the aforementioned assumption. In real videos, the number of motion groups may change throughout a clip as moving objects enter or leave the scene. Without coming to grips with this fundamental issue, the application of these works to real life problems will be severely hampered. By comparison, the two-frame methods are much better-placed to estimate exactly when moving objects enter or leave the scene.

Despite the relative merits of the two-frame methods over the multi-frame affine methods, less effort is devoted to the two-frame approach in recent years. On the one hand, it is partly due to the belief that multiple frames contain much more information that should be exploited. Contrary to such belief, we will show in Section 2.4 that the performance of the two-frame method is generally quite adequate; we may indeed question the wisdom of abandoning the two-frame method too hastily, especially in view of the information we lost through these feature points discarded because of the full-length requirement. On the other hand, there are clearly scenes where an observation period as short as two frames may confound the two-frame approach. For example, two objects may be moving with the same

motion for a short while but diverge thereafter.

In this chapter, we propose a multi-frame approach that is rooted in two-frame analysis, with a mixed norm formulation that couples the multi-frame information in an integrated manner. Beginning with a single image pair, we revisit the epipolar constraint of two-perspective-view (TPV) , leading to a subspace segmentation problem formulation that segments the null spaces of the appropriate equations. Thus, the idea of subspace separation applies and one can follow the SSC approach in converting the motion segmentation problem into a graph partitioning problem based on an affinity matrix. We prefer the sparse self-expression affinity of SSC, because of its good performance and some degree of tolerance to dependent subspaces [93]. A more powerful formulation that integrates multiple frames then follows, in which we derive an aggregated affinity matrix from multiple image pairs and seek a joint sparse coefficient recovery across multiple image pairs, *i.e.*, the sparse affinity coefficients of a particular trajectory should be consistently distributed across multiple image pairs in the sense that this trajectory should use the same set of other trajectories to express itself across all image pairs. This is formulated as a constrained mixed norm minimization problem, whose relaxed version is convex and can be solved efficiently with Alternating Direction Method of Multipliers (ADMM) method [18, 66].

Another important contribution of our work in this chapter lies in its robust model selection scheme. We first make a rough model estimation by analyzing the Laplacian matrix of the affinity matrix and over-segment the data into groups. Then we perform merging by a scheme that takes advantage of the loose grouping already available. Specifically, we use the data points in one group to sparsely represent each data point in another group. Based on Soltanolkotabi and Candès' scheme of outlier rejection [93], which declares a data point to be an outlier if the ℓ_1-norm of its

sparse coding vector is above a fixed threshold, we can decide which data points in the second group are inliers w.r.t. the first group and which are outliers. Based on the statistics of the ℓ_1-norm, they can then be proceed to merge the groups or leave them as they are.

To summarize, our major contributions are as follows.

- We return motion segmentation to its two-frame perspective root and then tightly integrate the information from correspondences across multiple frames into a unified mixed-norm optimization scheme. This results in a collaborative clustering algorithm that deals with perspective effects naturally and yet can leverage fully on the information present in multiple frames. It also handles incomplete trajectories much more reliably than those generic matrix completion schemes or motion segmentation methods with built-in completion schemes such as [83, 105]. Ambiguous feature matches can also be handled naturally.

- Inspired by the efficient outlier rejection scheme [93], we propose a simple yet coherent model selection algorithm, which also solves a series of mixed norm optimization problems; it follows an over-segment and merge scheme where the merging is based upon the mutual sparse representation of two groups.

- We then carry out extensive evaluation over a dataset containing 64 video sequences, with a balanced mix of clips with two and three motions, ranging from small to wide field of view, and with different amount of missing data. The results show that our joint inference scheme can produce significantly more accurate and reliable results than those methods individually estimating two-view motion models, followed by a loosely-coupled fusion step, or those state-of-the-art multi-frame methods such as SSC and LRR (low rank representation

[68]). More importantly, it offers scope for hope in realizing a motion segmentation scheme that is more adequate to the purpose of dealing with real world sequences with challenges such as missing data, unknown number of motions, and perspective effects.

2.1.1 Related work

There have been a plethora of multi-frame approaches. In the literature. Costeira and Kanade [31] propose to segment the motion of multiple independently moving objects according to the shape interaction matrix which is built from the singular value decomposition (SVD) of the trajectory matrix. However, this method fails when motion groups are partially dependent and it is very sensitive to noise [56]. Multi-Stage Learning [96] is a probabilistic approach which learns the parameters of a mixture model using the Expectation Maximization (EM) algorithm. Gruber [45] also presents an EM based algorithm which handles noise as well as missing data and can easily incorporate prior knowledge. Generalized principal component analysis (GPCA) [105] is an algebraic method, which equates subspace clustering with polynomial fitting and differentiation. The local subspace affinity method (LSA) [118] unifies the mixture of dependent and independent motions by estimating a local linear manifold. Then, an affinity matrix is established from the principal angles between these manifolds, after which spectral clustering is applied. While LSA uses a fixed size of neighboring points, local best-fit flats (LBF) [122] finds the optimal local neighborhoods, which is proven to improve the performance significantly. Agglomerative lossy compression (ALC) [83] find the segmentations by minimizing the coding length of the segmented data. Most recently, Elhamifar and Vidal [39] bring sparse representation into subspace clustering and apply them to motion segmentation. The key idea is to sparsely represent a feature point trajectory by other trajectories from the same subspace.

LRR [68] is another compressive sensing technique brought into subspace clustering. It finds the lowest rank representation of all data jointly, upon which an affinity graph is defined for subsequent clustering. Another thread of research is the projective factorization method [63] which extends the camera model to perspective, but it needs an iterative process that alternates between the estimation of the depths and the segmentation of the point trajectories. Furthermore, it still has the full-length requirement on the trajectories, and the depth estimation is highly dependent on the initial segmentation. Unlike the previous trajectory based multi-view methods, Cheriyadat and Radke [27] decompose the velocity profiles of point tracks into different motion components and corresponding non-negative weights using non-negative matrix factorization (NNMF) . Then the motions are segmented based on the derived weights. Our method revisits the two-view epipolar constraint equation in the language of subspace clustering and thus there is much similarity in terms of how subspace separation is performed. However, it does not have to make concession in terms of the camera modeling, and its multi-frame extension does not suffer from the strictness of requiring features to be present in all frames.

While many of these methods perform very well with *Hopkins155*, significant problems remain, as reviewed in the preceding paragraphs. Our key concern here is to tackle these challenges not well represented in *Hopkins155*. In contrast to the aforementioned approaches, our modeling of the problem is based on the epipolar constraint and does not make concession in terms of the camera projection, and its multi-frame extension does not suffer from the restriction of requiring features to be present in all frames. While projective factorization [63] extends the camera model to perspective, it needs an iterative process that alternates between the estimation of the depths and the segmentation of the trajectories. Furthermore, it still requires full-length trajectories, and the depth estimation is

highly dependent on the initial segmentation.

There have also been a lot of two-frame methods in the literature. There are many early works that deal with the two-frame case, early examples being [111, 79, 92, 99] and the more recent work of Wolf and Shashua's two-body fundamental matrix [112]. Then Vidal *et al.* [106] extend the later to general multi-body fundamental matrix and linearize it in a high dimensional space. The clustering of correspondences is then achieved by choosing the minimum Sampson distance from the correspondence to the estimated fundamental matrix. Li [61] proposes another mixture-of-fundamental-matrices model and formulates it as a linear programming problem. The Robust Algebraic Segmentation (RAS) algorithm[84] uses a hybrid perspective constraint to unify the representation of epipolar and homography constraints; its algebraic process is similar to GPCA. Jian *et al.* [52] simultaneously obtain the number of motions and group the motion trajectories based on the mixture of Dirichlet process models. Schindler and Suter [88] randomly sample sufficient motion models and choose the models by maximizing the geometrically robust information criterion (GRIC) . Our two-view method is similar in that it uses the two-view epipolar constraints, though we do not explicitly estimate the fundamental matrices but directly cluster the correspondences. More importantly, our formulation allows multi-frame extension in an integrated manner and can handle incomplete and ambiguous features in a natural way.

Model selection remains very much an open problem in motion segmentation. While the number of zero eigenvalues of the Laplacian matrix can be related to the number of connected components of the affinity matrix, the challenge lies in determining the number of eigenvalues close to zero in a robust manner [68, 93]. Some other methods [29, 38, 52, 61, 89, 88, 100] explicitly generate motion hypotheses and balance the goodness of fit against the complexity of the model. In general, the hypothesis generation step

is crucial in determining its success. Models with a high number of parameters face the predicament of generating a sufficiently large number of hypotheses while coping with the prohibitive computational cost. Bad samplings often result in failure for these methods, with the results varying each time due to the sampling procedure. Moreover, it is difficult, probabilistically speaking, to sample an all-inlier minimal set when estimating a high order model, because the number of samples required by the minimal set is relatively larger. Thus, [52, 89] uses calibrated cameras and [38] uses homography, both to reduce the number of points necessary to estimate a motion model. For the same purpose, [61, 88] design guided sampling steps. A unconventional method [28] uses multiple kernel learning to conduct a series of kernel optimizations. Then the model selection criterion stems from the idea that if two structures are indeed separate instances of the generic model, the optimized kernel will have a high alignment with the target kernel. However, it also suffers from the sampling step. Our method eschews this costly hypothesis generation step but instead takes advantage of the over-segmented grouping provided by the spectral clustering. We then leverage on the recent theoretical result [93] which provides a principled way to detect outlier points based on the ℓ_1 norm of the sparse representation of the point. This in turn allows us to perform merging of two over-segmented groups in a very robust way.

Two closely related works [89, 38] use two-view constraint to segment trajectories of a video sequence. However, our work differs from theirs in several key ways. First, the two related works both sample many model candidates for each image pair, followed by a model estimation. It is worthwhile noting that poor sampling often results in failure for these methods, and the results may vary every time due to the sampling procedure. Moreover, it is difficult to sample an all-inlier minimal set when estimating a high order model, because the number of the minimal set is relatively larger.

Thus, [89] uses calibrated cameras and [38] uses homography, where both can reduce the number of points necessary to estimate a motion. Second, the two-view motion segmentation and the linking of the frame-to-frame correspondences are somehow separated in their cases. In our scenario, we put all frame-to-frame correspondences into a global optimization scheme, and the linking information is also considered to construct a unified affinity matrix, which makes the linking more natural and is expected to achieve more optimal solutions.

Lastly, some recent research addresses the need to obtain a denser set of trajectories [19, 60]. These works aim to cover the image domain without too many large gaps. However, they only carry out the segmentation in the 2D domain, mainly due to computational consideration. Thus, motions that deviate from the simple 2D model may lead to a wrong segmentation. Our work pays the price of a lower trajectory density for a more accurate motion model and a higher quality data input.

The rest of this chapter is organized as follows. Section 2 discusses the TPV subspace in detail. Section 3 describes the joint clustering algorithm and the ℓ_1-norm based merging scheme. Then, our experimental results are illustrated in Section 4. Finally, we draw the conclusion in Section 5.

2.2 The TPV Motion Subspace

Assume $\mathbf{x}_p = (x_p, y_p, 1)^T$ and $\mathbf{x}'_p = (x'_p, y'_p, 1)^T$ are the homogeneous coordinates of two corresponding points of a 3-D point p in two frames. Their relationship is governed by the epipolar constraint [48] expressed as follows:

$$\mathbf{x}_p'^T \mathbf{F} \mathbf{x}_p = 0, \tag{2.5}$$

where $\mathbf{F} \doteq \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \in \mathbb{R}^{3\times3}$ is the fundamental matrix, which connects correspondences under the same rigid motion in two views. A classic algorithm to compute $\mathbf{F}$ is the 8-point algorithm [48], in which each correspondence gives rise to one linear equation in the unknown entries of $\mathbf{F}$ as follows:

$$(\; x'_p x_p \;\; x'_p y_p \;\; x'_p \;\; y'_p x_p \;\; y'_p y_p \;\; y'_p \;\; x_p \;\; y_p \;\; 1 \;)\mathbf{f} = 0, \tag{2.6}$$

where $\mathbf{f} = (\; f_{11} \;\; f_{12} \;\; f_{13} \;\; f_{21} \;\; f_{22} \;\; f_{23} \;\; f_{31} \;\; f_{32} \;\; f_{33} \;)^T$ is the 9×1 vector made up of the entries of $\mathbf{F}$ in row-major order. The coefficients of this equation are arranged in a column vector, denoted as $\mathbf{w}_p$. Clearly, those $\mathbf{w}_p$ under the same rigid motion k form a hyperplane perpendicular to $\mathbf{f}_k$, which we refer to as the TPV motion subspace. Since $\mathbf{f}_k$ is a 9×1 vector, the dimension of this subspace is at most 8.

A fundamental matrix determines the relationship of a camera pair uniquely [48]. Thus, in general the set of $\mathbf{w}_p$ for points undergoing the same rigid motion k forms a unique hyperplane perpendicular to $\mathbf{f}_k$. However, for points in special configuration, they fail to uniquely determine the fundamental matrix. These include correspondences lying on a plane in space or those only related by a pure rotation about the camera center. In both cases, point correspondences are related by a homography matrix $\mathbf{H} \doteq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \in \mathbb{R}^{3\times3}$, *i.e.*,

$$[\mathbf{x}'_p]_\times \mathbf{H}\mathbf{x}_p = 0, \tag{2.7}$$

where $[\mathbf{x}]_\times \in \mathbb{R}^{3\times3}$ denotes the skew-symmetric matrix associated with $\mathbf{x}$.

If we rewrite equation (2.7) in a linear form, $\mathbf{w}_p$ is related to a 9×3 matrix $\mathbf{H}'$:

$$\mathbf{w}_p^T \mathbf{H}' = \mathbf{0}, \tag{2.8}$$

where

$\mathbf{H}' = [\ \mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3\ ]$

$\mathbf{h}_1 = (\ 0\ \ 0\ \ 0\ \ h_{31}\ \ h_{32}\ \ h_{33}\ \ -h_{21}\ \ -h_{22}\ \ -h_{23}\ )^T,$

$\mathbf{h}_2 = (\ -h_{31}\ \ -h_{32}\ \ -h_{33}\ \ 0\ \ 0\ \ 0\ \ h_{11}\ \ h_{12}\ \ h_{13}\ )^T,$

$\mathbf{h}_3 = (\ h_{21}\ \ h_{22}\ \ h_{23}\ \ -h_{11}\ \ -h_{12}\ \ -h_{13}\ \ 0\ \ 0\ \ 0\ )^T.$

It can be observed from (2.8) that those $\mathbf{w}_p$ under the aforementioned degenerate configurations fall on the intersection of three hyperplanes, each of which is perpendicular to one column of $\mathbf{H}'$. Here, each column of $\mathbf{H}'$ is independent of one another in general and thus the rank of $\mathbf{H}'$ is 3. Thus, $\mathbf{w}_p$ under these degenerate configurations lie in a lower dimensional subspace with dimension no more than 6. Fortunately, there are various subspace separation algorithms [39, 68] that can handle subspaces with different dimensions and the above situation should pose no special problem.

2.3 Clustering Motion Subspaces

2.3.1 Sparse subspace clustering

The preceding section has reduced the motion segmentation task to that of clustering subspaces of dimension at most 8 in $\mathbb{R}^9$ in general. The data are now collected in a data matrix $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_P]$. The SSC algorithm can be used directly to perform subspace clustering for the case of single image pair; the case of multiple image pairs requires joint sparsity and will be discussed in Section 2.3.1.2.

2.3.1.1 Single image pair

We briefly review the SSC algorithm in the context of the TPV motion subspace: each column $\mathbf{w}_p$ can be represented as a linear combination of the other columns $\mathbf{w}_q$

$$\mathbf{w}_p = \sum_{q=1, q\neq p}^{P} c_q \mathbf{w}_q = \mathbf{W}_{\widehat{p}} \mathbf{c}_p, \tag{2.9}$$

where P is the number of correspondences, $\mathbf{W}_{\widehat{p}} = [\mathbf{w}_1 \cdots \mathbf{w}_{p-1}\, \mathbf{w}_{p+1} \cdots \mathbf{w}_P] \in \mathbb{R}^{D \times P-1}$ is the matrix obtained from $\mathbf{W}$ by removing its p-th column and $\mathbf{c}_p \in \mathbb{R}^{P-1}$ is the vector made up of the coefficients c_q. Generally, the solution for (2.9) is not unique and the key idea of SSC is to obtain a sparsest solution for $\mathbf{c}_p$ via solving the following relaxed ℓ_1 optimization problem

$$\min \|\mathbf{c}_p\|_1 \quad \text{s.t.} \quad \mathbf{w}_p = \mathbf{W}_{\widehat{p}} \mathbf{c}_p. \tag{2.10}$$

The nonzero entries in the optimal solution $\mathbf{c}_p$ indicate that the corresponding trajectories in $\mathbf{W}_{\widehat{p}}$ belong to the same subspace as $\mathbf{w}_p$. The optimization problem for every trajectory is collected and written succinctly in matrix form as

$$\min \|\mathbf{C}\|_1 \quad \text{s.t.} \quad \mathbf{W} = \mathbf{W}\mathbf{C}, diag(\mathbf{C}) = 0. \tag{2.11}$$

where $diag(\mathbf{C})$ are the diagonal entries of the matrix $\mathbf{C}$, and $diag(\mathbf{C}) = 0$ is introduced to avoid the trivial solution.

According to [39], since the optimal solution $\mathbf{C}^*$ to problem (2.11) measures the pairwise linear correlations among trajectories, it can be naturally used to construct an affinity matrix $\mathbf{A}$ with $\mathbf{A}_{ij} = |\mathbf{C}^*_{ij}| + |\mathbf{C}^*_{ji}|$, after which spectral clustering algorithms can be applied to obtain the desired segmentation into the respective subspaces.

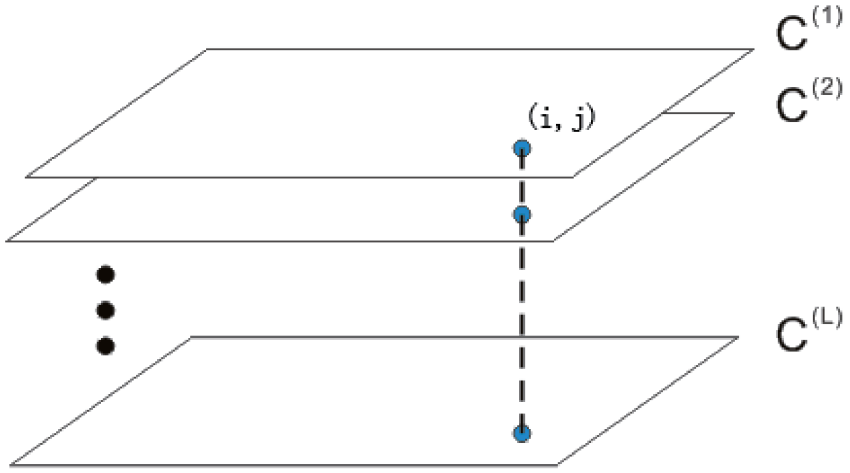

Figure 2.3: Illustration of the $\ell_{1,1,2}$ norm minimization. The entries (i, j) of $\mathbf{C}^{(l)}$ should be sparse and its support set should be consistent across different $\mathbf{C}^{(l)}$.

2.3.1.2 Multiple image pairs

A naive way to extend the SSC algorithm to multi-view case is to compute results from many image pairs individually and design a voting scheme to determine to which group the data points should belong. An alternative way is to accumulate the individual affinity matrices or adopt the multi-view spectral clustering method [124]. However, these methods operate on each image pair separately, and have not exploited the linkage between the multiple image pairs in a more integral manner. Here, we seek to incorporate all image pairs into a unified optimization process.

Assuming we have L image pairs, and since each image pair yields a correspondence matrix $\mathbf{W}^{(l)}$, L corresponding coefficient matrices $\mathbf{C}^{(l)}$ will be constructed by SSC. The key here is to solve for all $\mathbf{C}^{(l)}$ together and require them to share a common sparsity profile. In other words, the non-zero entries of $\mathbf{C}^{(l)}$ should be sparse and those columns corresponding to the same trajectory across the different $\mathbf{C}^{(l)}$ should share the same support set. This amounts to solving a joint sparse optimization problem [81], which

can be relaxed into the following mixed norm minimization problem:

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{P} \sum_{j=1}^{P} \sqrt{\sum_{l=1}^{L} (c_{ij}^{(l)})^2} \\ \text{s.t.} \quad &\mathbf{W}^{(l)} = \mathbf{W}^{(l)} \mathbf{C}^{(l)}, \quad diag(\mathbf{C}^{(l)}) = 0, \\ &l = 1, \ldots, L, \end{aligned} \tag{2.12}$$

where $c_{ij}^{(l)}$ is the (i, j)-th element of $\mathbf{C}^{(l)}$ for the l-th image pair. Referring to Figure 2.3, this operation can be visualized as stacking all $\mathbf{C}^{(l)}$ into a tensor $\mathcal{C} \in \mathbb{R}^{P \times P \times L}$, and then minimizing the number of non-zero entries in the aggregate matrix formed by summing all $c_{ij}^{(l)}$ along the third dimension l. In analogy to the $\ell_{1,2}$ norm being the norm that approximately measures the number of non-zero columns, we can call our norm the $\ell_{1,1,2}$ norm. Denote $\mathcal{C}^*$ as the optimal solution. We similarly construct an affinity matrix $\mathbf{A}$ with its element $\mathbf{A}_{ij} = \sqrt{\sum_{l=1}^{L} (c_{ij}^{*(l)})^2} + \sqrt{\sum_{l=1}^{L} (c_{ji}^{*(l)})^2}$. Then spectral clustering is applied as in the two-frame case.

Notice that the correspondences can be missing in some image pairs, here "missing" means a trajectory is invisible in either one or both of the image pair. In this case, we fill in with a $\mathbf{0}_{9\times1}$ column vector for the missing data so as to ensure that all $\mathbf{W}^{(l)}$ have the same dimension. More specifically, if a trajectory p is missing in the image pair l, then in the l-th correspondence matrix $\mathbf{W}^{(l)}$, the p-th column $\mathbf{w}_p^{(l)} = \mathbf{0}_{9\times1}$. Our rationales for filling in with $\mathbf{0}_{9\times1}$ are twofold: 1) when we want to obtain the sparse coding for the p-th point, the optimal solution for the missing data in the l-th image pair is $\mathbf{0}_{(P-1)\times1}$, not incurring any cost in equation (2.12), nor biasing the solution for other $\mathbf{C}^{(l)}$ in any way. 2) Conversely when we want to recover the sparse coding for other points, e.g. q, the missing data will not be chosen to represent the point q in the l-th image pair since it contributes nothing to the representation of q. This allows us to treat a trajectory with missing data in a uniform manner, without affecting the joint optimization scheme.

2.3.1.3 Handling ambiguous matches

In real applications, feature trackers often bring in noisy or even heavily corrupted trajectories, especially if we want to seek a denser coverage of features over the entire image. In order to recover the sparse coefficients from the corrupted observations, it is straightforward to consider the following regularized minimization problem:

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{P} \sum_{j=1}^{P} \sqrt{\sum_{l=1}^{L} (c_{ij}^{(l)})^2} + \lambda \sum_{l=1}^{L} \|\mathbf{E}^{(l)}\|_{\ell} \\ &\text{s.t.} \quad \mathbf{E}^{(l)} = \mathbf{W}^{(l)} - \mathbf{W}^{(l)}\mathbf{C}^{(l)}, \quad diag(\mathbf{C}^{(l)}) = 0, \\ &\qquad\qquad l = 1, \ldots, L, \end{aligned} \tag{2.13}$$

where λ is a weight used to adjust the effect of the two parts and $\|\cdot\|_{\ell}$ indicates a particular choice of regularization strategy. Here we choose $\ell_{1,2}$ norm to model sample-specific corruptions and outliers [68], whose minimization forces $\mathbf{E}^{(l)}$ to be column sparse.

After obtaining an optimal solution $(\mathcal{C}^*, \mathcal{E}^*)$ (where $\mathcal{E}^* \in \mathbb{R}^{D \times P \times L}$ is a tensor stacked from $\mathbf{E}^{*(l)}$), we could detect erroneous matches by looking for those columns with large ℓ_2 norms in any of the $\mathbf{E}^{*(l)}$. If a corrupted match is detected in $\mathbf{E}^{*(l)}$, we will delete it from image pair l but preserve the correct matches of that trajectory in other image pairs unless all matches of that trajectory are corrupted.

2.3.2 Merging via coefficient analysis

As the number of motion groups is usually not known *a priori* in reality, we have to come to grips with the model selection problem. In view of the difficulty of cluster detection, we propose to first over-segment the data based on the number of zero eigenvalues of the Laplacian matrix of the affinity matrix, and then attempt to merge the clusters later via the following model selection scheme. Based on the work of Soltanolkotabi and

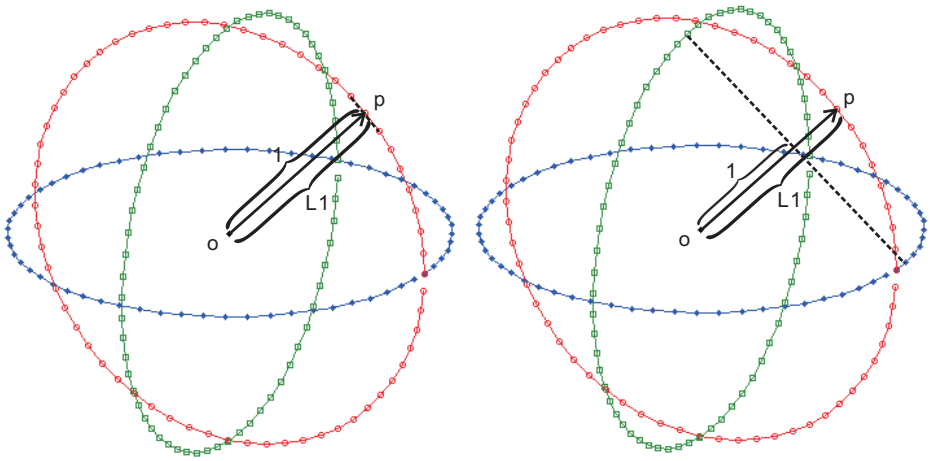

Figure 2.4: Illustration of the magnitudes of the ℓ_1 norm when point p is represented by points from the "red" subspace (left) or from the "bule" and "green" subspace (right). The ℓ_1 norm of the later case is larger.

Candès' outlier detection scheme [93], we propose a simple yet coherent model selection algorithm based on analysis of the ℓ_1-norm of the mutual representation of two over-segmented groups, which is able to correctly merge groups from the same subspace by a simple threshold.

For multiple independent subspaces, a point can be treated as an outlier *w.r.t* a group to which it does not belong. On the one hand, the coefficient vector of an outlier is expected to be less sparse due to the optimization scheme of the sparse affinity pursuit. On the other hand, even if the expansion of a data point is also sparse enough, its ℓ_1 norm is more likely to be large if the points chosen to represent this point are from different subspaces. The former situation is well explained in [93], and Figure 2.4 illustrates the second situation. We assume data points are located uniformly at random on the unit hypersphere (data vectors are normalized) as in [93]. As can be seen in Figure 2.4, if the points chosen to represent the point p in the "red" subspace are from the "blue" and the "green" subspace, the ℓ_1 norm is larger than the norm obtained in the case that the points are chosen from the "red" subspace. In summary, if the ℓ_1 norm of the coefficients vector is large, it is more likely to connect points from

different subspaces.

Given a data point $\mathbf{q} \in \mathbb{R}^D$ and a group of points $\{\mathbf{p}_i\}_{i=1}^M$ stacked as the columns of the matrix $\mathbf{P} \in \mathbb{R}^{D\times M}$ and spanning the subspace $\mathcal{S}$, if we use $\mathbf{P}$ to represent $\mathbf{q}$, *i.e.* $\mathbf{q} = \mathbf{P}\mathbf{c}$, we can obtain a coefficient vector $\mathbf{c} \in \mathbb{R}^M$. According to Theorem 1.3 of [93], the data point $\mathbf{q}$ has a high probability of being an outlier w.r.t $\mathcal{S}$ if the ℓ_1-norm of the sparsest solution $\mathbf{c}$ is larger than a threshold $\epsilon = \lambda(\frac{M-1}{D})\sqrt{D}$ (λ is a threshold ratio function; for details, see [93]). Based on this theorem, we can determine the relationship between two groups.

Now consider two groups of points obtained from the over-segmentation step, $\mathbf{P} \in \mathbb{R}^{D\times M}$ and $\mathbf{Q} \in \mathbb{R}^{D\times N}$, whose columns $\{\mathbf{p}_i\}_{i=1}^M$ and $\{\mathbf{q}_i\}_{i=1}^N$ are extracted from subspaces $\mathcal{S}_u$ and $\mathcal{S}_v$ respectively. If we sparsely represent the points in $\mathbf{P}$ using the points in $\mathbf{Q}$:

$$\begin{aligned} &\min \|\mathbf{C}\|_1 \\ &\text{s.t.} \quad \mathbf{P} = \mathbf{Q}\mathbf{C}, \end{aligned} \tag{2.14}$$

the columns of $\mathbf{C} \in \mathbb{R}^{N\times M}$ are the coefficient vectors corresponding to the data points in $\mathbf{P}$. Based on the aforementioned outlier determination scheme, if $u = v$ and $\mathbf{Q}$ adequately represents $\mathcal{S}_v$, the points in $\mathbf{P}$ should be inliers w.r.t $\mathbf{Q}$, and thus the ℓ_1-norms of columns $\{\mathbf{c}_i\}_{i=1}^M$ in $\mathbf{C}$ are expected to be small. For robustness, we compare the median value of all ℓ_1-norms of $\{\mathbf{c}_i\}_{i=1}^M$ against the threshold ϵ to decide if $\mathbf{P}$ should be merged into $\mathbf{Q}$. For notational convenience, we denote the above using a relationship matrix $\mathbf{R}$ with its elements defined as

$$\mathbf{R}_{pq} = median_{i=1}^M(\|\mathbf{c}_i\|_1). \tag{2.15}$$

Similarly, we can obtain $\mathbf{C}' \in \mathbb{R}^{M\times N}$ by representing $\mathbf{Q}$ using $\mathbf{P}$ and compute the relationship $\mathbf{R}_{qp}$. Note that this relationship is oriented, and in

general, $\mathbf{R}_{pq} \neq \mathbf{R}_{qp}$.

The above analysis can be extended to the case for the multiple image pairs in a manner analogous to the collaborative clustering algorithm in (2.12). Assuming L image pairs, we rewrite (2.14) as

$$\begin{aligned} &\min \textstyle\sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{\sum_{l=1}^{L} (c_{ij}^{(l)})^2} \\ \text{s.t.} \quad &\mathbf{P}^{(l)} = \mathbf{Q}^{(l)} \mathbf{C}^{(l)}, \quad diag(\mathbf{C}^{(l)}) = 0, \\ &l = 1, \ldots, L, \end{aligned} \tag{2.16}$$

where $\mathbf{P}^{(l)}$ and $\mathbf{Q}^{(l)}$ are the data matrices of the two groups in the l-th image pair, $\mathbf{C}^{(l)}$ is the corresponding coefficient matrix, and $c_{ij}^{(l)}$ is the (i,j)-th element of $\mathbf{C}^{(l)}$. Notice that this formulation is similar to the collaborative clustering algorithm in (2.12), but the data matrices in the $l.h.s.$ and $r.h.s.$ are different. And here we solve for this optimization problem to calculate the relationship of two groups. The relationship $\mathbf{R}_{pq}$ (2.15) is also changed accordingly:

$$\mathbf{R}_{pq} = median_{i=1}^{M}(median_{l=1}^{L}(\|\mathbf{c}_i^{(l)}\|_1)). \tag{2.17}$$

After obtaining the oriented relationships of all over-segmented groups, we claim the two groups belong to each other if both descriptions are smaller than a threshold ϵ, thus merging is applied between these two groups. If one description is larger than ϵ but the other one is smaller than ϵ, we claim one group belongs to the other group, thus we will also merge these two groups. If both descriptions are larger than ϵ, we claim these two groups exclude each other, so we will not merge them. We iteratively merge two groups according to the aforesaid threshold ϵ until there is no more merging possible. The details of the merging step are summarized in Algorithm 1.

One might question what if some of the groups are too small or degenerate such that they do not adequately represent the underlying subspace $\mathcal{S}$. Clearly, such groups are common occurrences, but it is also true that there

Algorithm 1 ℓ_1-norm based merging

Input: Set of motion groups $\{\mathbf{P}_k\}_{k=1...K}$, ϵ
  $\mathcal{P}_0 \leftarrow$ Current set of groups
  for $k = 1 \rightarrow (K-1)$ **do**
    for each group pair **do**
      Compute relationship matrix $\mathbf{R}$ according to (2.17).
    end for
    if $\min(\mathbf{R}) < \epsilon$ **then**
      1.$(i, j) = \text{find}(\min(\mathbf{R}))$
      2.Merge the groups i and j
      3.$\mathcal{P}_k \leftarrow$ Current set of groups
    else
      return $\mathcal{P}_k$
    end if
  end for
  return $\mathcal{P}_k$

invariably exist some other groups whose points fully span the subspace $\mathcal{S}$. In such cases, the former will be judged to belong to and merged into the latter. [1]

2.4 Experiments

2.4.1 Results on single image pairs

In this subsection, we evaluate the performance of the two-frame version of our algorithm on the *Hopkins155* database (denoted as TPV in Tables 2.1, 2.2 and 2.3) to gauge the effectiveness of our two-frame method. We compute the classification error as the percentage of misclassified points w.r.t the ground truth and list the average classification errors. We choose the first and the last frames of all sequences as the image pair for the testing, which avoids cases with short observation periods and ensures that all correspondences in the scene have sufficient displacements in the image plane. For the sake of comparison, we assume the number of motion groups

[1] Even if a motion group consists of say, just two walls, the degenerate case of the over-segmentation yielding two walls cleanly (and thus not mergeable) seldom arises; instead, the points of the two walls are usually segmented non-exactly by our over-segmentation step.

Table 2.1: Classification errors (%) for sequences with 2 motions

Method	ALC	GPCA	LSA	SSC	LRR	TPV
Checkerboard: 78 sequences						
Mean	1.49	6.09	2.57	1.12	1.50	1.81
Median	0.27	1.03	0.27	0.00	0.00	0.00
Traffic: 31 sequences						
Mean	1.75	1.41	5.43	0.02	0.52	1.10
Median	1.51	0.00	1.48	0.00	0.00	0.00
Other: 11 sequences						
Mean	10.70	2.88	4.10	0.62	2.41	1.26
Median	0.95	0.00	1.22	0.00	0.00	0.00
All: 120 sequences						
Mean	2.40	4.59	3.45	0.82	1.33	1.57
Median	0.43	0.38	0.59	0.00	0.00	0.00

is known in this experiment, like what many algorithms did. We also list the classification errors when applying ALC[83], GPCA[105], LSA[118], SSC[39] and LRR[68] to the affine motion subspace for comparison.

It can be seen from Table 2.3 that TPV yielded average classification errors of less than 5% for the two and three motions, which is only slightly worse off than those of SSC and LRR applied to multiple views assuming affine model. The results indicate that segmentation from two properly chosen views is almost as good as segmentation from the multiple views. What is noteworthy is that the 2-frame TPV algorithm outperforms the multi-frame GPCA and LSA algorithms on all categories. We believe that this is due to a combination of factors such as the better modeling of perspective effect and the choice of better clustering methods.

2.4.2 Results on multiple image pairs

We now evaluate the complete algorithm using multiple image pairs without knowing the number of motion groups and with challenges like missing data and perspective effects. The data used in this evaluation comprise 62 video sequences, of which 50 are from *Hopkins155*. Since *Hopkins155*

Table 2.2: Classification errors (%) for sequences with 3 motions

Method	ALC	GPCA	LSA	SSC	LRR	TPV
Checkerboard: 26 sequences						
Mean	5.00	31.95	5.80	2.97	2.56	5.12
Median	0.66	32.93	1.77	0.27	0.10	1.57
Traffic: 7 sequences						
Mean	8.86	19.83	25.07	0.58	1.80	1.82
Median	0.51	19.55	23.79	0.00	0.00	0.18
Other: 2 sequences						
Mean	21.08	16.85	7.25	1.42	4.25	1.93
Median	21.08	16.85	7.25	1.42	4.25	1.93
All: 35 sequences						
Mean	6.69	28.66	9.73	2.45	2.51	4.98
Median	0.67	28.26	2.33	0.20	0.00	0.79

Table 2.3: Classification errors (%) on *Hopkins155*

Method	ALC	GPCA	LSA	SSC	LRR	TPV
All: 155 sequences						
Mean	3.36	10.02	4.86	1.18	1.59	2.34

has a very unbalanced number of 2-motion and 3-motion clips (120 and 35 respectively), we retain only the 50 original seed videos (the other 105 2-motion clips are created by splitting off from the 3-motion clips). More importantly, to evaluate the performance under missing data and perspective effects, we added 12 clips with incomplete trajectories, of which 4 are from [89] and the other 8 are captured by us using a handheld camera with a wide angle lens. The newly captured sequences contain about 100 frames each, some of which experience heavy occlusions, posing significant challenge to the matrix completion task, as we shall see later. Of the resultant 62 motion clips, 26 contain two motions, 36 contain three motions, 12 suffer from missing data, and 9 have strong perspective effects (some of these categories are not mutually exclusive). We refer to this combined dataset as the 62-clip dataset.

We denote our complete algorithm as M-TPV for multiple-TPV. We

compare the performance of M-TPV to seven state-of-the-art approaches: ALC [83], GPCA [106], LBF [122], LRR [68], Multi-Scale Motion Clustering (MSMC) [38], Ordered Residual Kernel (ORK) [29] and SSC [39]. For ALC, we use the provided rather simple matrix completion method and test 101 different values from 10^{-5} to 10^{3} for the noisy level as in [83], and then we record the best segmentations with the smallest average error rate. For MSMC, since the default scales (the number of interval frames between an image pair, with the default scales being h_1, h_5 and h_{25}) did not perform well in these sequences, we tried several combinations and report the error rates corresponding to the following scales: h_5, h_{10} and h_{25}. For SSC, since the model selection method based on spectral gap[93] performed poorly in these real data, we choose the second order difference (SOD) method as in LBF. Note that the SOD method is also used in a similar manner to support SSC in [122]. For those algorithms which do not explicitly handle missing data, such as LBF, LRR, ORK and SSC, we recover the data matrix using Chen's matrix completion approach [26], which in our experience has the best performance among various competing algorithms (such as OptSpace [78], GROUSE [10] and etc.). For those algorithms which have a random element in their results, such as ORK and MSMC, we repeat 100 times and record the best results.

Table 2.4 shows the performance of these methods on the 62-clip dataset. Since the estimated number of motion groups may not be the same as the ground truth number, we exhaustively test all the cluster pairings to obtain the best error rates. Furthermore, to investigate if good model selection results in good segmentation, the error rates obtained by only considering sequences where the number of motions is correctly estimated are shown in Table 2.5. We also show some qualitative results obtained with the newly captured clips in Figure 2.5.

The evaluation in Table 2.4 can be divided into three parts.

Table 2.4: Classification results on 62-clip dataset

Method	ALC	GPCA	LBF	LRR	MSMC	ORK	SSC	M-TPV
Classification error (%) - clips with perspective effect: 9 clips								
Mean	16.18	43.66	20.00	16.31	19.17	22.94	25.68	**8.20**
Classification error (%) - clips with missing data: 12 clips								
Mean	25.38	39.64	20.17	26.03	14.64	24.11	27.41	**7.71**
Classification error (%) - clips without missing data: 50 clips								
Mean	22.03	16.89	15.66	9.82	14.19	12.98	13.09	**7.56**
Classification error (%) - all: 62 clips								
Mean	22.67	21.29	16.53	12.98	14.27	15.13	15.86	**7.59**
Group number estimation - all 62 clips								
# correct	21	33	29	35	25	37	33	**46**

Table 2.5: Classification results on 62-clip dataset (only considering sequences where the number of motions is correctly estimated)

Method	ALC	GPCA	LBF	LRR	MSMC	ORK	SSC	M-TPV
Classification error (%) - clips with perspective effect: 9 clips								
Mean	0.35	40.83	12.14	14.83	0.58	20.24	9.68	0.46
Classification error (%) - clips with missing data: 12 clips								
Mean	0.43	28.77	18.47	29.46	1.06	22.33	17.22	0.91
Classification error (%) - clips without missing data: 50 clips								
Mean	18.28	16.20	11.90	5.26	2.59	4.15	2.01	2.78
Classification error (%) - all: 62 clips								
Mean	14.88	16.58	5.90	5.95	2.34	8.08	5.17	2.37

In the first part, the classification error rates of the 9 clips with strong perspective effects are presented. Our method is the only one with an error rate of less than 10%, which shows the superiority of the proposed approach. Although ALC and MSMC also reported good results when the number of motion groups is correctly estimated, perspective effects have a significant detrimental impact on their model selection steps, resulting in substantially higher error rates of ALC and MSMC.

In the second part of Table 2.4, the impact of missing data is investigated. Our approach again outperformed the other methods with a less than 10% error rate. GPCA broke down mainly due to the instability of

the Power Factorization method used for filling in missing data. Those methods based on the matrix completion of [26] for filling in, such as LBF, ORK and SSC, performed well in some sequences, but the overall deleterious impact is evident, attesting to the difficulty faced by a general-purpose matrix completion algorithm in dealing with the structured pattern of the missing data. Among these methods, it is also remarkable that the so-far top-performing LRR failed in the model selection of 11 sequences, which implies that the model selection step in LRR is very sensitive to how the spectral values have been changed in the recovered matrix. Of the only sequence whose motion number is correctly estimated (the "Van" clip, first column of Figure 2.5), LRR has a very poor classification error rate. MSMC failed in those sequences with complicated objects and backgrounds due to its simple motion model based on homography. Even if this method uses a higher-order motion model, the significant increase in model complexity will pose a lot of difficulties for the sampling procedure, rendering its performance very much suspect.

The last comparison is based on the 50 seed videos from the *Hopkins155*'s dataset. These clips are relatively easy, because they have complete trajectories. The average classification error of our method on all 50 clips is 7.56%, while that considering only cases having correct motion number estimation is 2.78%. The more meaningful figure of 7.56% is clearly the best compared to other state-of-the-art motion segmentation algorithms. These figures also demonstrate that model selection remains a recalcitrant problem, and to achieve real progress in motion segmentation, we must meet this challenge heads-on.

If we only consider the segmentation results of the sequences with correct number of motion estimation in Table 2.5, all approaches except ALC and GPCA yielded near zero error rates. This fact demonstrates that these methods can almost give perfect segmentation if the model selection part

can be well solved. Thus, we believe model selection is the urgent problem we need to improve next.

The last two rows of metrics in Table 2.4 measure the overall performance, from which it can be seen that our method outperformed the rest in all significant aspects. It has 46 correct motion number estimation out of 62 clips (next best is 37), and the average classification error of all clips is 7.59% (next best is 12.98%). These overall performances demonstrate that our method is capable of handling the various real challenges in the motion segmentation problem.

2.5 Conclusions

We solve the 3D motion segmentation problem of multiple frames rooted in the epipolar geometry of two perspective views via a collaborative clustering algorithm. This approach highly integrates multiple frame information with a mixed norm optimization, which is able to avoid the disadvantages of multi-frame methods and enjoy the rich information provided by multiple frames. We also propose a method to evaluate the relationship of two groups based on a similar optimization scheme. Leveraging on this, we first over-segment the motion groups, and then merge them according to the relationships. The experiments on the *Hopkins155* database and the new sequences showed that the proposed algorithm outperforms the state-of-the-art methods in meeting the various challenges.

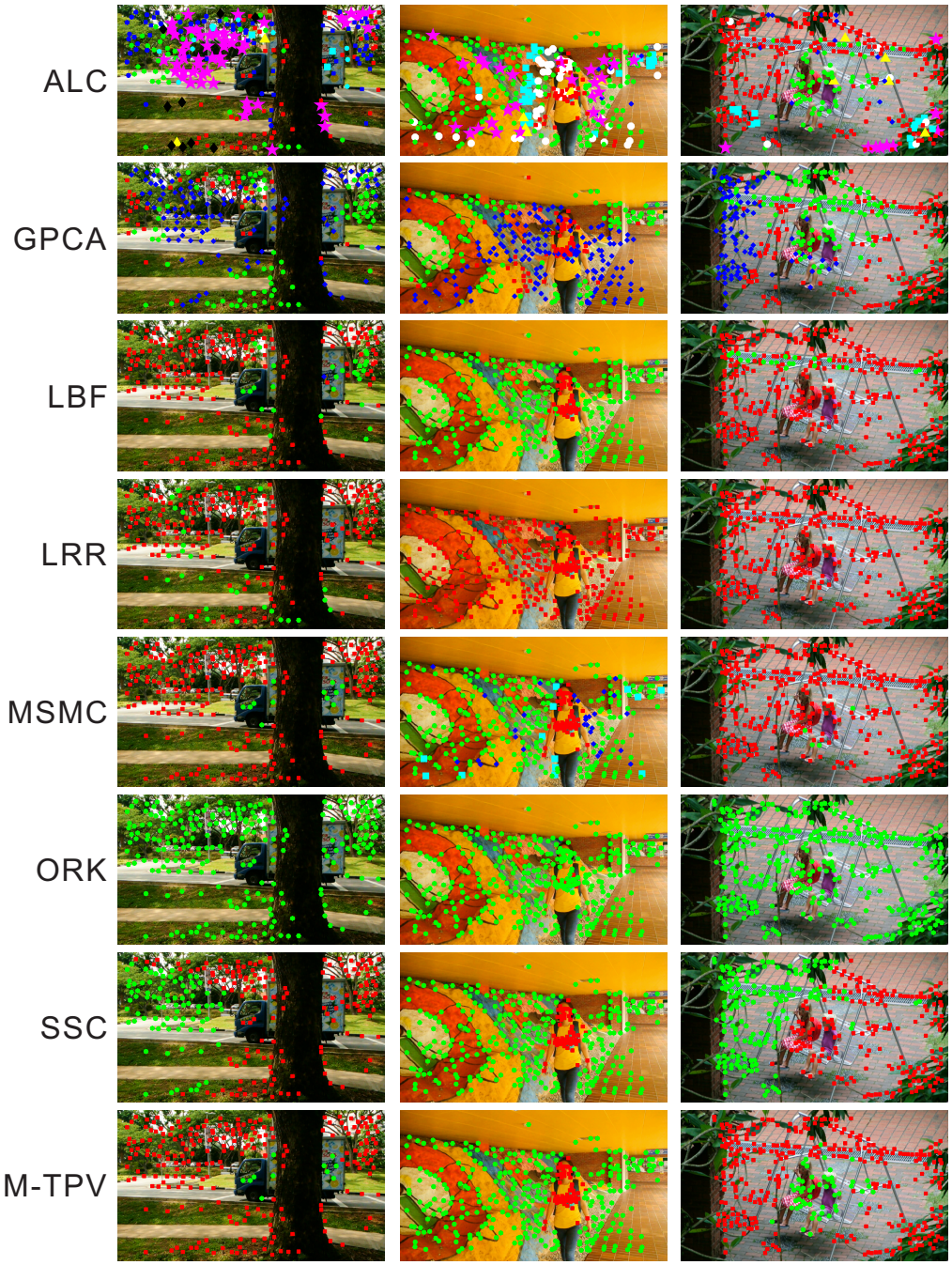

Figure 2.5: Qualitative results of the real data with missing entries. The segmentation results of the 50-th frames of the sequences are presented. From left to right are the "Van", "Girl" and "Swing" clips.

Chapter 3

Simultaneous Clustering and Model Selection

In the preceding chapter, the model selection method to estimate the number of motion groups is based on an over-segment and merge approach, where the merging step is based on the property of the ℓ_1-norm of the mutual sparse representation of two over-segmented groups. In this chapter, we propose a more general model selection approach

While clustering has been well studied in the past decade, model selection has drawn less attention. In this chapter, we address both problems in a joint manner with an indicator matrix formulation, in which the clustering cost is penalized by a Frobenius inner product term and the group number estimation is achieved by a rank minimization. As affinity graphs generally contain positive edge values, a sparsity term is further added to avoid the trivial solution. Rather than adopting the conventional convex relaxation approach wholesale, we represent the original problem more faithfully by taking full advantage of the particular structure present in the optimization problem and solving it efficiently using the ADMM. The highly constrained nature of the optimization provides our algorithm with the robustness to deal with the varying and often imperfect input affinity

matrices arising from different applications and different group numbers. Evaluations on the synthetic data as well as two real world problems show the superiority of the method across a large variety of settings.

3.1 Introduction

Many computer vision problems, such as image segmentation, multi-structure recovery and so on, involve solving the clustering problem at some point. Often, an affinity graph is set up and then fed into a spectral clustering framework [71] to infer the clustering of the data into groups. Such spectral graph methods include Ratio Cut [46], Normalized Cut [91], *etc.* However, deciding on the number of clusters remains an open problem for all such algorithms.

The simplest way to estimate the group number is to count the number of zero eigenvalues of the Laplacian matrix of the affinity graph. However, it does not perform very well in practice when data contain structures at different scales of size and density, and when data are contaminated by noise. In these cases, these eigenvalues deviate from zero in a complex manner, and it is non-trivial to determine the number of eigenvalues close to zero in a robust manner.

In this chapter, we propose a novel algorithm to perform simultaneous clustering and model selection (SCAMS). Given an affinity matrix $\mathbf{A}$ with non-negative entries, our task can be conceptually viewed as discovering which $\mathbf{A}(i, j)$ are small enough; this is essentially saying that elements i and j are dissimilar and should be placed in different clusters. Just as importantly, we should also ensure that elements i and j are not linked indirectly through other elements in the graph. This is realized by adopting an indicator matrix formulation explained as follows. We take the Frobenius inner product of the affinity matrix $\mathbf{A}$ and $\mathbf{G} = \mathbf{Z}\mathbf{Z}^T$, where $\mathbf{Z}$ is an

indicator matrix whose rows indicate to which group a point belongs. We maximize this Frobenius inner product term $\langle \mathbf{A}, \mathbf{G} \rangle$ so as to keep $\mathbf{G}$ as close to the data term $\mathbf{A}$ as possible, while at the same time, we impose several constraints so as to ensure meaningful solutions for $\mathbf{G}$. Firstly, there should be a trade-off between the complexity of the model and goodness of fit. The model complexity is indicated by the rank of $\mathbf{G}$ (see Section 3.3); thus, we seek to minimize the rank of $\mathbf{G}$ to discriminate against a more complex model. Secondly, we should also limit the cardinality of $\mathbf{G}$ — the number of nonzero entries in $\mathbf{G}$ — so as to discover structure in the data (indicated by the sparsity pattern of $\mathbf{G}$). In fact, without this penalty term on cardinality, we will end up with the trivial solution of $\mathbf{G}$ being the all-one matrix (all data belong to one cluster). Together with the $\{0, 1\}$ constraint on $\mathbf{G}$, this formulation in effect examines the connectivity of the entire graph and tends to set $\mathbf{G}(i, j)$ to one if elements i and j are linked indirectly through other elements. This highly constrained formulation also provides our algorithm with the robustness to deal with the varying and often imperfect input affinity matrices generated from different applications and different group numbers (despite the best efforts of works to generate these matrices [39, 68, 108]). Figure 3.1 shows a recovery result of our algorithm. Notice that our algorithm is able to recover a nearly perfect 0-1 block diagonal $\mathbf{G}$ from the contaminated affinity matrix.

Our problem now involves solving for a low-rank and sparse matrix $\mathbf{G}$, subject to a number of constraints over the integer variables, all of which lead to an NP-hard problem. In many problem instances, the convex proxy to an NP-hard problem may not be a good approach. Instead, there might be a need to represent the original problem more faithfully — an approximate solution to the right problem can be better than the exact solution to the wrong problem. In our case, we take full advantage of the particular structure present in the optimization problem, optimizing over

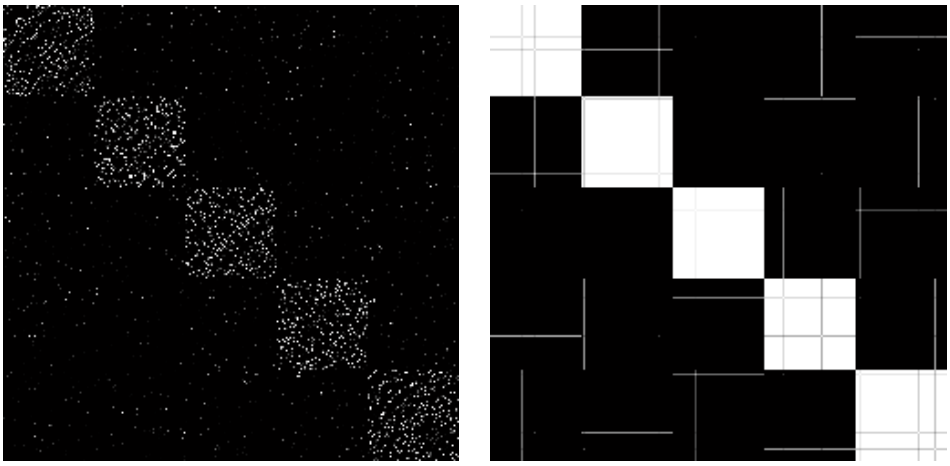

Figure 3.1: Left: A contaminated affinity matrix $\mathbf{A}$ with 5 clusters. Right: The recovered $\mathbf{G}$ contains 5 almost perfect blocks. Further processing by the proposed Boolean matrix factorization algorithm will obtain perfect blocks from this $\mathbf{G}$.

the rank and ℓ_0-norm directly and yet solving the problem efficiently using the ADMM method [18, 66].

A common heuristic to obtain the final clustering is to factorize $\mathbf{G}$ back to $\mathbf{Z}\mathbf{Z}^T$ using Cholesky decomposition [44], and assign each data point to the index with the maximum value in each row of $\mathbf{Z}$. However, Cholesky decomposition occasionally produces bad results even if $\mathbf{G}$ contains nearly perfect blocks because it does not impose any Boolean constraint on the factor matrices. Thus, we propose a variant of an existing Boolean matrix factorization (BMF) algorithm [76] to finesse a better decomposition.

The contribution of this chapter is summarized as follows.

- We formulate the model selection as a rank minimization problem, leading to a joint optimization of clustering and model selection. Trivial solution is avoided by adding a sparsity penalty term. The low rank penalty, together with other constraints that enforce the indicator matrix formulation, highly constrains the solution space and provide our algorithm with the ability to repair imperfections in the affinity matrix, *e.g.* filling in the connectivity gap or ignoring dubious connections.

- The inner optimization subproblems in each iteration are designed to take full advantage of the particular structure present in our problem. This results in an effective and efficient algorithm that represents the original problem more faithfully and works well under a wider range of changing conditions such as increasing group number and noise level. Our extensive experiments shed light on how the different attributes of the affinity matrices constructed by different methods impact on model selection, further highlighting the strength of our algorithm.

- We propose a novel Boolean matrix factorization algorithm to obtain a better decomposition which lends itself to more accurate clustering.

3.2 Related works

There have been many algorithms devised for the clustering problem; we will briefly review some major approaches here. In the spectral graph approach, one needs to determine the number of zero eigenvalues of the Laplacian matrix of the affinity graph in a robust manner. Heuristics particularly designed for this purpose include the eigengap heuristic, the elbow criterion, the gap statistic [97], the silhouette index [87], and several recent measures [3, 68, 93]. In the information-theoretic approach, one aims to balance the goodness of fit against the complexity of the model. A classical measure is the Akaike Information Criterion (AIC) [2], which is followed by many variants [55, 100]. Another measure is based on compression efficiency, such as the Minimum Description Length (MDL) [72, 85, 94]. The major drawback of this kind of methods is that they are usually model-dependent. Among the many clustering methods, one can also distinguish another category which is based on the stability of the solutions [16, 59]. The stability is measured by the pairwise similarities between clustering results with respect to perturbations such as sub-sampling or the addition

of noise, and the optimal number of clusters is then given by the most stable solution. Many of the above methods involve particular choices to be made at the outset, for example the value of a particular thresholding parameter. Many of them also require that the number of clusters to be found by another criterion. That is, a two-step procedure is performed: a clustering criterion determines the optimal assignments for a given number of clusters and a separate criterion measures the goodness of the classification to determine the number of clusters. Our method involves very little domain-specific assumptions, and it performs a joint optimization of clustering and model selection in one single step. While our algorithm also involves choice of weights, the experimental results show that these chosen values works well across a wide range of different settings, which is not what can be said about other compared methods.

Our method is also related to the probabilistic mixture model approach in the sense that both combine clustering and model selection in a single step. However, in the probabilistic mixture approach, one needs to assume that the data can be described by a mixture of multivariate distributions with some parameters that determine their shape with known distribution of the data. Our method involves no such assumption. Another similarity between such probabilistic mixture model approach and our method lies in the objective function. In fact, if we view our affinity matrix $\mathbf{A}$ as a covariance matrix, the objective functions are identical except for the integer constraint(*e.g.* see [11, 25, 77]).

Lastly, we have in the preceding section likened the optimization as one of discovering which affinity values are small enough to be set as zero. This can be regarded as a thresholding operation on the affinity values. In fact, if we know the threshold, we can convert our problem into a correlation clustering (CC) problem [12]. We can either use the original unweighted form of CC, in which the affinity matrix $\mathbf{A}$ defines a graph with all edges

assigned weights of either $+1$ or -1 (representing "similar" and "dissimilar" respectively), or one can use the general form of CC with real edge weights [9, 33]. In either case, CC maximizes the Frobenius inner product term $\langle \mathbf{A}, \mathbf{Z}\mathbf{Z}^T \rangle$ which is identical to our problem. The difficulty of this line of approach is in determining a proper threshold to distinguish between "similar" and "dissimilar". Our method eschews such direct thresholding and instead utilizes the generic low rank and sparsity assumption to perform the operation. Furthermore, the CC problem is an instance of the quadratic semi-assignment problem (QSAP) [107], which is NP-complete when the cluster number is unknown. Our method provides a tractable solution via carefully exploiting the structure of the problem and appropriate relaxations, and we show in our experiments that the results are of good quality and stable across a range of noise level and cluster number.

3.3 Clustering with Model Selection

3.3.1 Problem formulation

Suppose we are given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{v_i\}_{i=1}^N$ is the set of the N nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of the edges between the nodes, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is an affinity matrix constructed by some method, with each element $\mathbf{A}(i,j) \geq 0$ being the affinity between sample v_i and v_j. $\mathbf{A}(i,j) = 0$ suggests that v_i and v_j are completely dissimilar, and thus likely to be disconnected, while $\mathbf{A}(i,j) > 0$ means there is the possibility for the two nodes to be clustered into the same group. The larger the value, the more likely these two nodes should be in the same group. Now the task is to cluster these N nodes into K groups, where the group number K is unknown a priori and needs to be estimated.

For ease of problem formulation, let us assume for now that K is known. Denote $\mathbf{Z} \in \mathbb{R}^{N \times K}$ as the indicator matrix, whose row entries indicate to

which group the points belong, *i.e.*., if point i belongs to group k, $\mathbf{Z}(i,k) = 1$ and the remaining entries of the i-th row are all 0's. Thus, if point i and j belong to the same group, $\langle \mathbf{Z}(i,:), \mathbf{Z}(j,:) \rangle = 1$; otherwise, $\langle \mathbf{Z}(i,:), \mathbf{Z}(j,:) \rangle = 0$, where $\langle \cdot, \cdot \rangle$ denote the inner product of two vectors, or the Frobenius inner product of two matrices, as the case may be. As discussed before, we want to maximize the following objective function:

$$f(\mathbf{Z}) = \langle \mathbf{A}, \mathbf{Z}\mathbf{Z}^T \rangle = tr(\mathbf{A}^T \mathbf{Z}\mathbf{Z}^T), \tag{3.1}$$

where $tr(\cdot)$ indicates the trace operator of the given matrix.

From the preceding, we have $\mathbf{G} = \mathbf{Z}\mathbf{Z}^T$; therefore, $\mathbf{G}$ is positive semi-definite (PSD) and the rank of $\mathbf{G}$ is exactly K. We can convert the above problem into the following minimization problem over $\mathbf{G}$ by adding a negative sign in front of the affinity matrix and denoting $\mathbf{W} = -\mathbf{A}$:

$$\begin{aligned} \min. \quad & tr(\mathbf{W}^T \mathbf{G}), \\ s.t. \quad & \mathbf{G} \in \mathbf{S}_+, \\ & diag(\mathbf{G}) = 1, \\ & rank(\mathbf{G}) = K, \\ & \mathbf{G} \in \{0,1\}^{N \times N}, \end{aligned} \tag{3.2}$$

where $\mathbf{S}_+$ is the PSD cone and $diag(\cdot)$ are the diagonal entries of the matrix, this constraint merely reflecting the fact that the same point cannot be split into different groups.

Since K is unknown a priori and usually $K \ll N$, we estimate it by minimizing the rank of $\mathbf{G}$. However, this will result in a trivial solution for $\mathbf{G}$, *i.e.*, the all one matrix, which is rank-one and "covers" all the entries of the affinity matrix by 1. To avoid the trivial solution, we further add an ℓ_0 penalty on $\mathbf{G}$ to enforce sparsity on its entries. This would force the optimization to only insert ones at those $\mathbf{G}(i,j)$ locations where the

magnitude of the corresponding $\mathbf{A}(i,j)$ is large. Accordingly, we now have

$$\begin{aligned} \min. \quad & tr(\mathbf{W}^T\mathbf{G}) + \lambda rank(\mathbf{G}) + \gamma||\mathbf{G}||_0, \\ s.t. \quad & \mathbf{G} \in \mathbf{S}_+, \\ & diag(\mathbf{G}) = 1, \\ & \mathbf{G} \in \{0,1\}^{N\times N}, \end{aligned} \tag{3.3}$$

where $||\cdot||_0$ is the ℓ_0 norm, which counts the number of nonzero elements, and λ and γ are the parameters to weigh the respective penalty terms. To make the problem tractable, we first relax the constraint $\mathbf{G} \in \{0,1\}^{N\times N}$ to obtain real-valued entries $\mathbf{G} \in [0,1]^{N\times N}$. Next, instead of replacing the rank and the ℓ_0 norm with their convex proxies, we optimize them directly by taking full advantage of the particular structure present in the problem. In particular, as we will show later, the resulting inner optimization problems can be solved analytically by eigen-decomposition and soft-thresholding operations. By now, the problem to be solved has the following form

$$\begin{aligned} \min. \quad & tr(\mathbf{W}^T\mathbf{G}) + \lambda rank(G) + \gamma||\mathbf{G}||_0, \\ s.t. \quad & \mathbf{G} \in \mathbf{S}_+, \\ & diag(\mathbf{G}) = 1, \\ & \mathbf{G} \in [0,1]^{N\times N}. \end{aligned} \tag{3.4}$$

3.3.2 Solver

For efficiency, we adopt the ADMM method [18, 66] to solve this problem. We first convert (3.4) to the following equivalent problem:

$$\begin{aligned} \min. \quad & tr(\mathbf{W}^T\mathbf{G}) + \lambda rank(\mathbf{G}) + \gamma||\mathbf{H}||_0 + g(\mathbf{H}), \\ s.t. \quad & \mathbf{G} \in \mathbf{S}_+, \\ & \mathbf{G} = \mathbf{H} - diag(\mathbf{H}) + \mathbf{I}, \end{aligned} \tag{3.5}$$

where g is the indicator function of the convex set $[0,1]^{N\times N}$, which returns 0 if it is in the set, ∞ otherwise, and $\mathbf{H}$ is an intermediate variable introduced to make the problem tractable. The augmented Lagrange function is

$$\begin{aligned}\mathcal{L} = \quad & tr(\mathbf{W}^T\mathbf{G}) + \lambda rank(\mathbf{G}) + \gamma||\mathbf{H}||_0 + g(\mathbf{H}) + \\ & tr(\mathbf{Y}^T(\mathbf{G}-\mathbf{H}+diag(\mathbf{H})-\mathbf{I})) + \\ & \tfrac{1}{2\mu}\|\mathbf{G}-\mathbf{H}+diag(\mathbf{H})-\mathbf{I}\|_F^2, \\ s.t. \quad & \mathbf{G}\in\mathbf{S}_+,\end{aligned} \tag{3.6}$$

where $\mathbf{Y}$ is the Lagrange parameter, and $\mu > 0$ is a penalty parameter. The function can be minimized with respect to $\mathbf{G}$ and $\mathbf{H}$ alternatingly, by fixing the other variable, and then updating the Lagrange multipliers $\mathbf{Y}$. The overall framework of the alternating direction method is shown in Algorithm 2, with the detailed solver for each subproblem to be described later.

Algorithm 2 Solving (3.4) by ADMM

Input: Negative affinity matrix $\mathbf{W}$, parameters λ and γ.
Initialize: $\mathbf{G}=\mathbf{H}=\mathbf{Y}=\mathbf{0}_{N\times N}$, $\mu=10^6$, $\rho=1.1$, $\mu_{\min}=10^{-10}$ and $\epsilon=10^{-8}$.
while not converged **do**
Step 1 Fix the others and update $\mathbf{G}$ as
$\mathbf{G} = arg\min_{\mathbf{G}} \|\mathbf{G}-\mathbf{H}+\mu(\mathbf{W}+\mathbf{Y})\|_F^2 + 2\mu\lambda rank(\mathbf{G})$,
$s.t.\ \mathbf{G}\in\mathbf{S}_+$.
Step 2 Fix the others and update $\mathbf{H}$ as
$\mathbf{H}' = arg\min_{\mathbf{H}} \|\mathbf{H}-\mathbf{G}-\mu\mathbf{Y}\|_F^2 + 2\mu\gamma\|\mathbf{H}\|_0 + g(\mathbf{H})$,
$\mathbf{H} = \mathbf{H}' - diag(\mathbf{H}') + \mathbf{I}$.
Step 3 Update the multipliers
$\mathbf{Y} = \mathbf{Y} + \frac{1}{\mu}(\mathbf{G}-\mathbf{H})$.
Step 4 Update the parameter μ by $\mu = \max(\frac{\mu}{\rho}, \mu_{\min})$.
Step 5 Check the convergence conditions:
$\|\mathbf{G}-\mathbf{H}\|_\infty \le \epsilon$.
end while

Solving G. In step 1 of Algorithm 2, the solution of $\mathbf{G}$ involves minimizing the rank plus a convex quadratic function in the PSD cone. It can be efficiently solved using the following theorem. The proof is analogous

to that of Theorem 16 in [80], with the nuclear norm replaced by the rank.

Theorem 1. *For any square matrix* $\mathbf{S} \in \mathbb{R}^{N \times N}$*, the unique closed form solution to the optimization problem*

$$\begin{aligned} \mathbf{G}^* = \quad & arg\min_{\mathbf{G}} \|\mathbf{G} - \mathbf{S}\|_F^2 + \lambda rank(\mathbf{G}), \\ s.t. \quad & \mathbf{G} \in \mathbf{S}_+. \end{aligned} \tag{3.7}$$

takes the form

$$\mathbf{G}^* = \mathbf{Q}\mathcal{H}_\lambda(\mathbf{\Lambda})\mathbf{Q}^T, \tag{3.8}$$

where $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ *is the spectrum(eigen-) decomposition of* $\widehat{\mathbf{S}} = (\mathbf{S} + \mathbf{S}^T)/2$ *and* $\mathcal{H}_\lambda(\cdot)$ *is the thresholding operator acting on each element of the matrix, and defined as*

$$\mathcal{H}_\lambda(v) = \begin{cases} 0 & if \quad v < 0 \quad or \quad v^2 \leq \lambda, \\ v & otherwise\ . \end{cases} \tag{3.9}$$

Proof. See Appendix A.1. □

Solving H. In step 2 of Algorithm 2, the update of $\mathbf{H}'$ involves minimizing the ℓ_0 norm plus a convex quadratic function in the convex set $[0, 1]^{N \times N}$. Since this problem is obviously separable, each element can be optimized individually and simple manipulation suggests the following theorem.

Theorem 2. *For any matrix* $\mathbf{M} \in \mathbb{R}^{M \times N}$*, the unique closed form solution to the optimization problem*

$$\mathbf{H}^* = arg\min_{\mathbf{H}} \|\mathbf{H} - \mathbf{M}\|_F^2 + \gamma\|\mathbf{H}\|_0 + g(\mathbf{H}), \tag{3.10}$$

takes the form

$$\mathbf{H}^* = \mathcal{T}_\gamma(\mathbf{M}). \tag{3.11}$$

where $\mathcal{T}_\gamma(\cdot)$ *is the thresholding operator acting on each element of the ma-*

trix, and is defined as

$$\mathcal{T}_\gamma(v) = \begin{cases} 1 & \textit{if} \quad v > 1 \quad \textit{and} \quad \min(v^2, 2v-1) > \gamma \\ 0 & \textit{else if} \quad v < 0 \quad \textit{or} \quad v^2 \leq \gamma \quad \textit{or} \quad v > 1 \\ v & \textit{otherwise} \ . \end{cases} \tag{3.12}$$

Proof. See Appendix A.2. □

With the closed-form solutions, global minimums are assured for both sub-problems. Nevertheless, the algorithm as a whole does not have guarantee to convergence as the two sub-problems are non-convex. As far as we know, there is no general convergence theory for ADMM applied to non-convex problems, but numerical results in [90] on low-rank matrix factorization show that ADMM performed well for solving certain non-convex models. Indeed, our algorithm also has strong convergence behavior empirically.

3.4 Constrained Boolean Matrix Factorization

As the Cholesky decomposition occasionally yields poor binary result of $\mathbf{Z}$ even if $\mathbf{G}$ is nearly a 0-1 block diagonal matrix, we adapt the idea of BMF to achieve a better decomposition. Our proposed BMF method is similar to the Asso algorithm [75, 76] but takes into account the additional PSD constraint, and that each row of $\mathbf{Z}$ contains only one 1 (this latter constraint can be interpreted as an orthonormal constraint under Boolean algebra).

For the sake of completeness, we first give a brief introduction of BMF; for more details, see [75, 76]. We then formally define our BMF problem with its PSD and orthonormal constraints.

BMF aims to (approximately) represent a Boolean matrix as the Boolean product of two Boolean matrices. Here "Boolean" matrix means that the matrix contains only 0's and 1's. Using the superscript b to stand for Boolean matrix, let $\mathbf{B}^b \in \{0,1\}^{N\times K}$ and $\mathbf{C}^b \in \{0,1\}^{K\times M}$ be the two Boolean matrices, whose *Boolean matrix product*, $\mathbf{B}^b \circ \mathbf{C}^b$ yields $\mathbf{A}^b$, with $\mathbf{A}^b(i,j) = \vee_{k=1}^{K} \mathbf{B}^b(i,k)\mathbf{C}^b(k,j)$, and the *OR* operation $\vee$ is the normal sum but with addition defined as $1+1=1$. Our problem can now be formally defined as

Problem 1. ***Constrained Boolean Matrix Factorization (CBMF)*** *with the PSD and Boolean orthonormal constraints. Given a Boolean matrix* $\mathbf{G}^b \in \{0,1\}^{N\times N}$ *and an upper bound* K_0*, find Boolean matrix* $\mathbf{Z}^b \in \{0,1\}^{N\times K}$*,* $K \leq K_0$*, such that* $\mathbf{Z}^b$ *satisfies*

$$\begin{aligned} \min. \quad & |\mathbf{G}^b \oplus (\mathbf{Z}^b \circ \mathbf{Z}^{bT})|, \\ s.t. \quad & \mathbf{Z}^{bT} \circ \mathbf{Z}^b = \mathbf{I}_{K\times K}, \end{aligned} \tag{3.13}$$

where $|\cdot|$ is the norm of a Boolean matrix and defined as the number of 1's in it, *i.e.*, $|\mathbf{A}^b| = \sum_{i,j} \mathbf{A}^b(i,j)$, and $\oplus$ is the *Exclusive-OR* operation applied element-wise, and defined as the normal addition but with $1+1=0$.

The original Asso algorithm solves the BMF problem via the heuristic approach of generating the candidate columns using pairwise association accuracies. More specifically, it generates a matrix $\mathbf{D}$ with $\mathbf{D}(i,j) = \langle \mathbf{G}^b(i,:), \mathbf{G}^b(j,:)\rangle / \langle \mathbf{G}^b(j,:), \mathbf{G}^b(j,:)\rangle$, *i.e.*, $\mathbf{D}(i,j)$ is the association accuracy as defined in association rule mining [1] for rule $\mathbf{G}^b(j,:) \Rightarrow \mathbf{G}^b(i,:)$. After $\mathbf{D}$ is binarized to a Boolean matrix $\mathbf{D}^b$ (see Algorithm 3), the columns of the factor matrices are selected from the columns of $\mathbf{D}^b$ in a greedy fashion. In the context of our problem with the two additional constraints, the algorithm is modified as follows. Firstly, each candidate column of $\mathbf{D}^b$ is concatenated to the current $\mathbf{Z}^b$, and the next best $\mathbf{Z}^b$ is the one that minimizes (3.13). Note that by virtue of the formulation, the PSD constraint is

Algorithm 3 The AssoCBMF algorithm

Input: $\mathbf{G}$, K_0

Initialize: Construct the Boolean matrix $\mathbf{G}^b$ from $\mathbf{G}$ with rounding threshold $t_b = 0.5$, $\mathbf{Z}^b \leftarrow [\,]$, $e = \infty, t_d = 0.1$.
for $\tau = 0.1, 0.2, \ldots, 1$ **do**
  Construct $\mathbf{D}^b$ with $\mathbf{D}^b(i,j) = \frac{\langle \mathbf{G}^b(i,:), \mathbf{G}^b(j,:) \rangle}{\langle \mathbf{G}^b(j,:), \mathbf{G}^b(j,:) \rangle} > \tau$.
  for $k = 1, 2, ..., K_0$ **do**
    $i = arg\min_i |\mathbf{G}^b \oplus ([\mathbf{Z}^b\ \mathbf{D}^b(:,i)] \circ [\mathbf{Z}^b\ \mathbf{D}^b(:,i)]^T)|$.
    $\mathbf{Z}^b \leftarrow [\mathbf{Z}^b\ \mathbf{D}^b(:,i)]$.
    Delete all j-th columns with $\frac{\langle \mathbf{D}^b(:,i), \mathbf{D}^b(:,j) \rangle}{\|\mathbf{D}^b(:,i)\| \|\mathbf{D}^b(:,j)\|} > t_d$ from $\mathbf{D}^b$.
    if $\mathbf{D}^b$ is empty **or** (3.13) is not reduced in this loop
      break
    end if
    if $\|\mathbf{G} - \mathbf{Z}^b \mathbf{Z}^{bT}\|_F^2 < e$
      $\mathbf{Z}^{b*} = \mathbf{Z}^b$.
      $e = \|\mathbf{G} - \mathbf{Z}^b \mathbf{Z}^{bT}\|_F^2$.
    end if
  end for
end for
return $\mathbf{Z}^{b*}$

automatically satisfied. This step is repeated $K \leq K_0$ times until there is no candidate column in $\mathbf{D}^b$ left or (3.13) cannot be reduced anymore. Secondly, to reduce the probability that a row of $\mathbf{Z}^b$ contains multiple 1's and violates the Boolean orthonormal constraint, we only retain as candidate those columns which are sufficiently different from the selected columns (based on some threshold t_d) for the next iteration. The full details are presented in Algorithm 3, in which the input K_0 is usually selected as the rank of $\mathbf{G}$.

Since we only approximately enforce the orthonormal constraint, it is possible for a row of $\mathbf{Z}^b$ to contain multiple 1's. Usually, these constitute a very small proportion of the rows. Thus, most points can be uniquely assigned to clusters and the clusters are adequately populated. As a result, we can resolve the assignment conflict by a simple post-processing step as follows. We postpone the cluster assignment of all those points with conflicts. Assuming the resultant clustering is $\mathbf{X} = \{X_1, \ldots, X_K\}$ and that there is an unassigned data point i, we assign the point i to the

group $X_{K'}$ with whose members it has the largest affinity; that is, $K' = arg\max_k \sum_{j \in X_k} \mathbf{A}(i,j)$, where $\mathbf{A}$ is the affinity matrix as defined in Section 3.1.

3.5 Experiments

In this section, we compare our method with various model selection methods. In the spectral graph approach, the key to performance lies in how well one is able to determine the number of eigenvalues close to zero in the Laplacian matrix. We choose as representatives of these spectral graph methods both the basic gap heuristic (GH) method [71] as baseline, as well as one of the most robust ones—the soft thresholds (ST) method [68] which produces the best result reported in the motion segmentation problem so far. In addition to these two methods, we also compare with a model specific method—the second order difference (SOD) method [122], which reports state-of-the-art results in several datasets. A potential disadvantage of SOD is that it requires knowledge of the model; in particular, the subspace dimension is assumed known and constant. Note also that its model selection does not depend solely on the affinity matrix, hence requiring the original data as input. Since the performance of the model selection step also depends on the type of affinity matrix passed in, we also experiment with different ways of constructing the affinity matrix. We choose the two state-of-the-art algorithms in subspace clustering, SSC [39] and LRR [68] [1], to construct affinity matrices. For ST and SOD, we use the same parameter settings as in the original papers; for SCAMS, we use the fixed values of $\lambda = 2$ and $\gamma = 0.005$ in all the experiments.

To evaluate algorithm performance, we adopt the Rand index (RI) [82] as a measure of similarity between two data clusterings. This metric counts

[1] Here, by SSC and LRR, we refer only to those part of the respective algorithms that produce the affinity matrix, *i.e.*, not including the original model selection step proposed by the authors.

the pairs of points on which two clusterings agree or disagree. It is a better metric compared to the classification error rate when the number of groups is unknown. It is defined as follows.

Definition 1. *Given a set of N elements $\mathcal{V} = \{v_i\}_{i=1}^N$ and two clusterings of $\mathcal{V}$, namely $\mathbf{X} = \{X_1, \ldots, X_r\}$ with r clusters and $\mathbf{Y} = \{Y_1, \ldots, Y_s\}$ with s clusters. We define*

- *a: the number of pairs that are in the same cluster in both $\mathbf{X}$ and $\mathbf{Y}$.*
- *b: the number of pairs that are in the different clusters in both $\mathbf{X}$ and $\mathbf{Y}$.*
- *c: the number of pairs that are in the same cluster in $\mathbf{X}$ but in the different clusters in $\mathbf{Y}$.*
- *d: the number of pairs that are in the different clusters in $\mathbf{X}$ but in the same cluster in $\mathbf{Y}$.*

The Rand index, RI, is

$$RI = \frac{a+b}{a+b+c+d}. \tag{3.14}$$

Note that RI has a value between 0 and 1, with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clusters are exactly the same.

3.5.1 Synthetic data

We first investigate the performance of the various methods using synthetic data with different noise levels and varying number of groups. Similar to [93], we sample K subspaces chosen uniformly at random from d-dimensional subspaces in $\mathbb{R}^{50}$. We then sample 50 points on each subspace and normalize them to unit-norm vectors for the experiments. When sampling a subspace, we randomly sample d orthogonal basis vectors in $\mathbb{R}^{50}$.

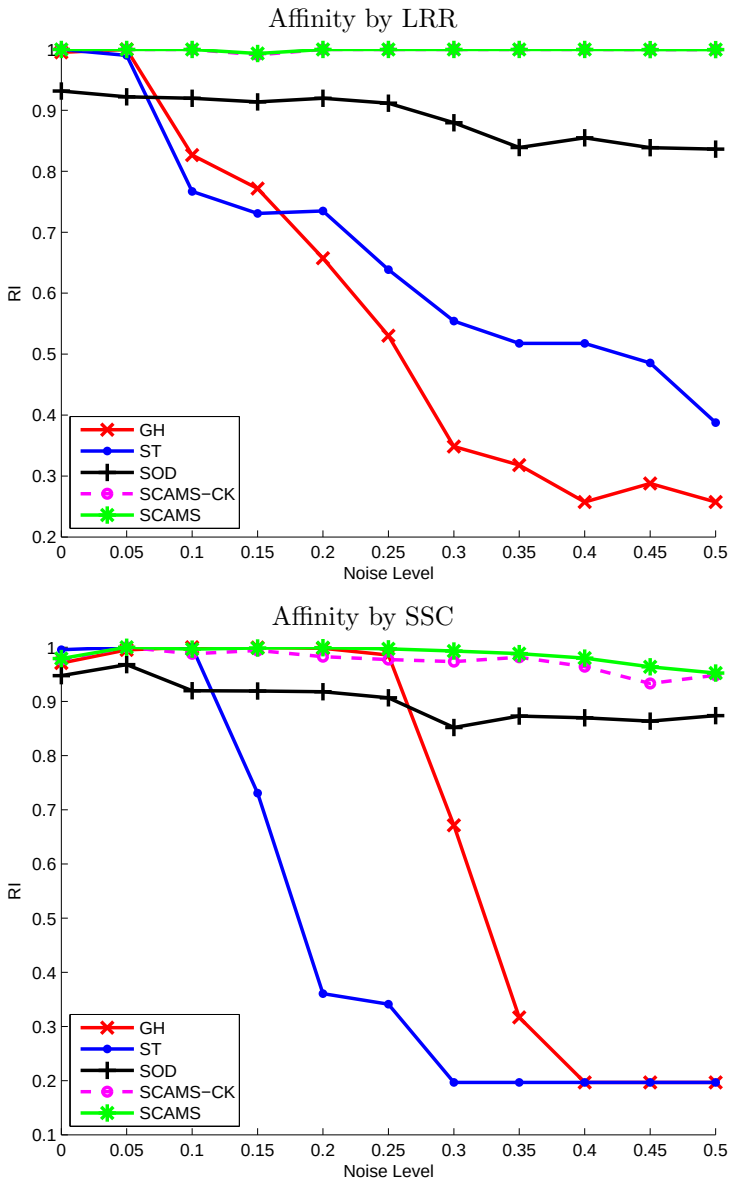

Figure 3.2: Comparison on the Synthetic Data when the noise level changes.

For each subspace, we sample 50 points from it by randomly combining the d basis vectors, and then we normalize the sampled point vectors to unit-norm vectors. To add noise, we perturb each unit-norm point vector by a noisy vector chosen independently and uniformly at random on the sphere of radius ρ (the larger this radius is, the bigger the noise is, so it reflects the noise level). And then, it is normalized to have unit norm again. More specifically, if $\mathbf{x}$ is the point vector, $\mathbf{z}$ is the noise, the noisy sample $\tilde{\mathbf{x}} = \frac{\mathbf{x}+\mathbf{z}}{\|\mathbf{x}+\mathbf{z}\|_2^2}$, where $\|\mathbf{z}\|_2^2 = \rho$.

3.5.1.1 Different noise levels

In the noise level test, we fix $K = 5$, with each group having different dimensions of $d = [2, 4, 6, 8, 10]$ respectively. The latter is to reflect model degeneracy, quite a common occurrence in real-world applications. As per [93], we perturb each unit-norm data point by adding a noisy vector chosen independently and uniformly at random on the sphere of radius ρ (noise level) in $\mathbb{R}^{50}$. We consider 11 different noise levels: $\rho = 0, 0.05, \ldots, 0.5$. The test runs 20 times and the average results are reported in the top of Figure 3.2.

As can be seen, despite the increasing noise, SCAMS performs consistently well (above 0.9) using either SSC or LRR to construct the affinity matrix. SOD performs less well although its performance also does not degrade much with increasing noise level. In contrast, the performances of GH and ST degrade significantly when the noise level increases. This experiment shows that SCAMS is more robust to noise. One may also notice that when the affinity matrix is provided by SSC, the RIs of all methods are somewhat off the perfect score of 1 even with the noise level at 0. This is probably because the LASSO version of SSC that we use is designed for noisy data at all levels. Unfortunately, this results in a slight loss of accuracy in the affinity matrix when the noise level is 0.

3.5.1.2 Varying group numbers

In the group number test, we fix the noise level $\rho = 0.05$ and gradually increase the group number K from 1 to 12. For a given K, each of the K groups has a different dimension d ranging from $[2, 4, \ldots, 2K]$ respectively. Note that the sum of the dimension of the subspaces is greater than the ambient dimension of 50 when $K > 6$. As K increases still more, the various subspaces become increasingly dependent, posing difficulties for the construction of affinity matrix by SSC and LRR. This raises the spectre of

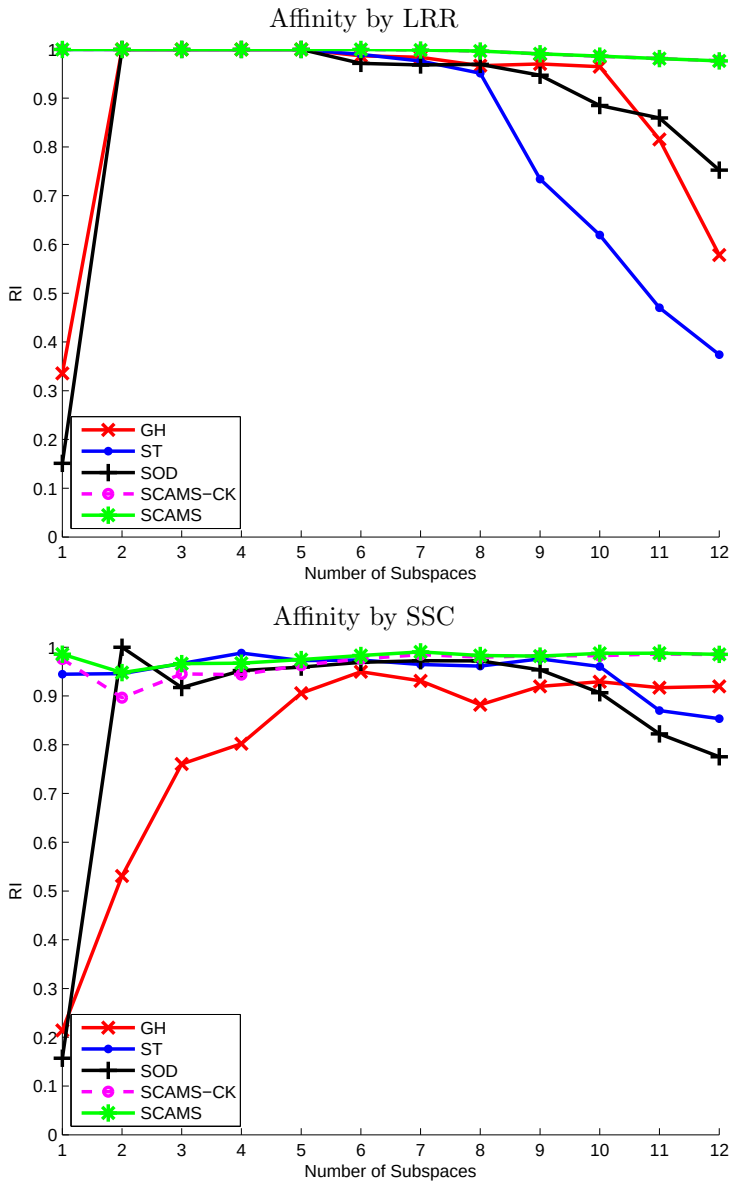

Figure 3.3: Comparison on the Synthetic Data when the number of subspaces changes.

poor-quality affinity matrix as the number of groups increases. We again repeat the experiment 20 times and report the average results in the bottom of Figure 3.3.

As is evident again, SCAMS performs consistently well (above 0.9) with both versions of affinity matrix. SOD is a second order method, and its mechanism can only handle those cases when group number is greater than one. Other than this drawback, SOD again produces fairly competitive results, its performance not degrading significantly until group number

exceeds 8 or 9. ST is also fairly competitive but degrades earlier when the affinity matrix is constructed by LRR. GH and SOD perform badly when the group number is 1. In general, one can say that the performances of most methods are affected by the declining quality of affinity matrices when the subspaces or groups increasingly overlap, with the effect being more pronounced in the case of LRR-constructed affinity matrix. On the other hand, some methods (notably GH) are seemingly affected by the sparser connectivity of the SSC-constructed affinity matrix, especially when the group number is small. Only our method is adequate to the handling of the varied attributes of the affinity matrices produced by different methods and under changing conditions.

To show the improvement brought about by the CBMF algorithm in Section 3.4, we also report the result of SCAMS using just Cholesky decomposition (SCAMS-CK) to perform the $\mathbf{G} = \mathbf{Z}\mathbf{Z}^T$ factorization. While the improvement is not significant in the case of the affinity matrix produced by LRR, it is significant when the affinity matrix is constructed by SSC and the group number is small. This performance boost is further corroborated in the later motion segmentation experiment in which CBMF improves the RI score by about 0.02.

3.5.2 Motion segmentation

We further evaluate the performance of SCAMS in dealing with real world problems. In this subsection, we tackle the motion segmentation problem using the *Hopkins155* [103] as dataset. This dataset comprises 155 sequences containing either two or three motions. This problem can be formulated as a subspace clustering problem, because the trajectories of a rigid motion across multiple frames lie in an affine subspace with a dimension of no more than 3, or a linear subspace with a dimension of at most 4 under the affine camera assumption [103]. In our experiments, we use

the original $2F$-dimensional feature trajectories without any compression, where F is the number of frames in each sequence. The results in Table 3.1 report the RI scores averaged over the 155 sequences.

Table 3.1: RI on *Hopkins155*

	Affinity by LRR		Affinity by SSC	
Method	Mean	Median	Mean	Median
GH	0.6584	0.6490	0.7699	0.7418
ST	0.9154	0.9815	0.9095	0.9972
SOD	0.9026	0.9923	0.8834	0.9944
SCAMS	0.9202	0.9827	0.9068	0.9740

Since this dataset is almost noise-free and contains a small number of subspaces in each sequence, all the methods except GH perform well and there is no significant difference among these methods. GH's poor performance can be correlated with the corresponding simulation results in the preceding section. Firstly, when the affinity matrix is produced by LRR, slight noise can be detrimental to the GH method. Secondly, when the affinity matrix is produced by SSC, GH performs badly with a small group number.

3.5.3 Face clustering

The other real world problem that we address is the face clustering problem. In this subsection, we test the algorithms on the Extended YaleB dataset [43], which contains cropped frontal human face images of 38 subjects. Each subject has 64 images taken under different light illuminations. Figure 3.4

Figure 3.4: Examples of face images. Images of 6 subjects (32 images for each subject) are shown here, where each row corresponds to a subject.

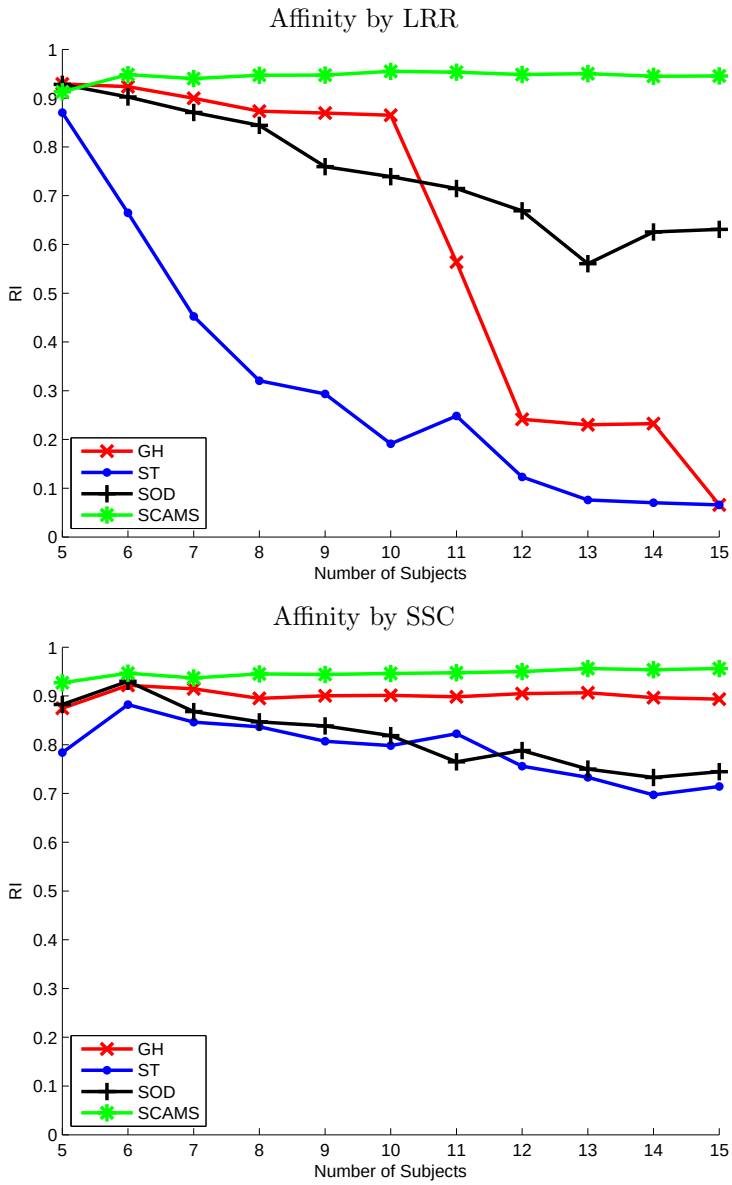

Figure 3.5: Comparison on the Extended YaleB dataset with increasing number of subjects.

shows some image examples. This problem can also be cast as a subspace clustering problem, because images of a subject with a fixed pose and varying illumination lie close to a linear subspace of dimension 9 [14]. To evaluate the performance of our algorithm, we randomly pick K subjects (K ranging from 5 to 15) and cluster the features associated with these subject images. As a preprocessing step, we resize the images to 42×48, and then use PCA to reduce the dimensionality of the vectorized raw pixel features to 30. We repeat the experiment 20 times and show the average

results in Figure 3.5.

As can be seen from Figure 3.5 and has been observed earlier, the affinity matrix constructed by LRR still poses problems for most methods (though to varying degrees) when the number of groups increases. In contrast, SCAMS performs consistently well (above 0.9) even when the LRR-constructed affinity matrix is not in an obliging form for most other methods. With SSC-constructed affinity matrix, all methods yield promising and more stable results, at least with respect to the number of subjects tested in this experiment. SCAMS performs consistently better than most other algorithms, with GH also turning in a stable performance. This latter phenomenon is again consistent with the results of the synthetic experiment.

3.6 Discussion and Conclusion

We simultaneously solve the model selection and clustering problems in a unified optimization scheme. The original structure of the affinity matrix is preserved by the Frobenius inner product (the data term) and the sparsity penalty, both terms acting locally. The rank minimization enforces global smoothness and tends to reduce the complexity of the model. These global and local considerations reveal the underlying structure of the clusters, resulting in a near-perfect 0-1 block diagonal matrix. Our highly-constrained indicator matrix formulation also has the effect of rectifying imperfections in the affinity matrix, such as filling in connectivity gap in the SSC-constructed affinity matrix. We then propose a constrained BMF to obtain a better decomposition and this in turn yields better assignments of data points. The experiments on the synthetic data as well as two real world problems show that our method performs significantly better with noisy data and large number of groups. Our experiments with both the

LRR- and SSC-constructed affinity matrix reveal their different characters, and further showcase the strength of our proposed SCAMS method in handling different types of affinity matrices.

Chapter 4

Visual Tracking via Sparsity Patten Learning and An Alternative Sparsity Model

Recently sparse representation has been successfully applied to visual tracking by modeling the target appearance using a sparse approximation over the template set. However, this approach is limited by its high computational cost, which is dominated by that of the ℓ_1-norm minimization. In the first part of this chapter, we speed up the method by learning the sparsity patterns of the template set. With the learnt sparsity patterns, we are able to recover the "sparse coefficients" of the candidate samples by some small-scale ℓ_2-norm minimizations; this results in a very fast tracking algorithm. In the second part of this chapter, we propose an alternative sparsity model, which, reversing the role of the template and candidate, models the template appearance using a sparse approximation over the candidate set. In this case, a large number of candidates can be immediately filtered out according to whether they are chosen to represent the templates or not. Then the optimal candidate is chosen as the one with the largest observation likelihood from the retained candidate set. This

sparsity model exploits the tracking problem from a novel perspective and achieves better performance even with the simplest setting. Experiments on a recently released benchmark with 50 challenging video sequences show significant runtime efficiency and tracking accuracy achieved by the two proposed algorithms.

4.1 Introduction

Visual tracking (or object tracking) plays an important role in numerous vision applications such as security surveillance, vehicle navigation, activity recognition and human computer interface. Given an initial state (such as position and size) of a target either manually annotated or automatically detected in the first frame of a video sequence, the goal of visual tracking is to estimate the states of the target in the subsequent frames. Although many tracking methods have been proposed [8, 30, 47, 50, 73, 86] in recent decades, it remains a challenging problem due to various factors such as partial occlusions, illumination changes, pose changes, background clutter and viewpoint variation.

Among these methods, the ℓ_1 tracker [73] proposed by Mei and Ling is especially notable as it is the first work that brings the sparse representation and compressed sensing techniques [23, 36] to the visual tracking problem. Similar to the sparsity based method for the face recognition problem [113] mentioned in Chapter 2, the ℓ_1 tracker represents candidate samples by a sparse linear combination of templates; these templates include true templates from the tracked object and trivial templates used to handle noise or occlusion. The optimal candidate should use as few trivial templates as possible and keep a low reconstruction error as well.

For the sparsity based methods, the candidate states are usually estimated in a particle filter framework, which approximates the posterior

distribution by importance sampling. The accuracy of the approximation generally increases with the number of samples used. To achieve a reasonable accuracy, these methods usually need to solve hundreds of ℓ_1-norm related minimization problems for each frame during the tracking process. As a consequence of the large computational cost of the ℓ_1-norm minimization, the tracker is prevented from being used in a real time system such as real time security surveillance.

In this chapter, we propose two sparsity based ideas to improve the computational speed of visual tracking: 1) the first idea speeds up the conventional ℓ_1 tracker by sparsity pattern learning (SPL) ; 2) the other idea considers the visual tracking problem from a different perspective, reversing the roles played by the template and candidate.

Visual tracking via sparsity pattern learning. We propose to learn the sparsity patterns for the template set. Rather than solving hundreds of ℓ_1-norm minimization problems, we solve the small-scale ℓ_2-norm minimization problems with the learnt sparsity patterns. More specifically, we express each object template in the template set using the other templates (including the trivial templates) and record the positions of the nonzero coefficients as a sparsity pattern of that template. When we test a candidate, we choose the basis vectors (i.e the templates) according to each learnt sparsity pattern, reconstruct the candidate by solving the ℓ_2-norm minimization problems with the chosen basis vectors, and represent the observation likelihood with the minimum reconstruction errors from different sparsity patterns. Since the patterns are sparse, the scales of the ℓ_2-norm minimization problems are small and thus can be solved rapidly. Subsequently the sparsity patterns would need to be recomputed with the update of the templates. However, this only happens occasionally in the tracking process. Moreover, we design fast methods to update the sparsity patterns w.r.t. the previously learnt results.

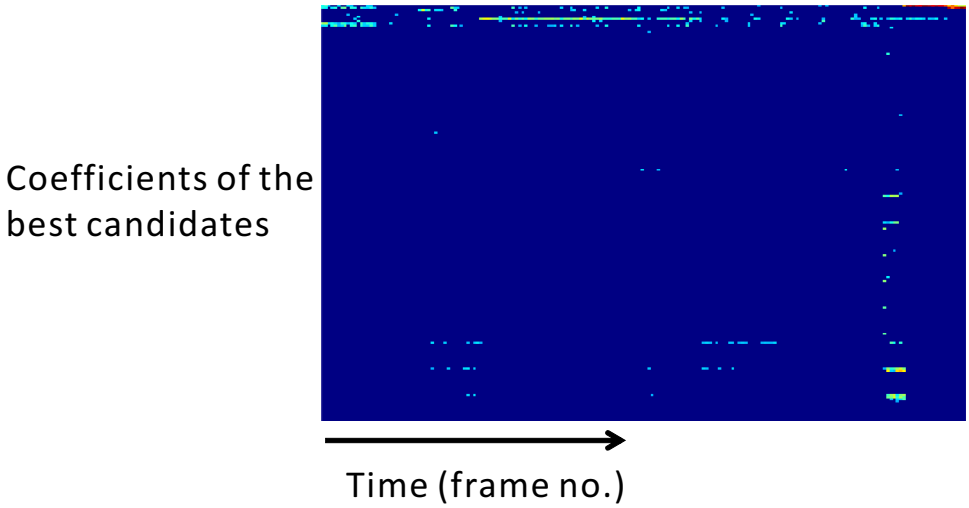

Figure 4.1: The coefficient vectors of the best patches in different frames are shown in sequence. The supports of these coefficient vectors contain only a few combinations. And coefficient vectors in neighboring frames has shown similar patterns.

The reason that we express the candidate in this way is based on the consideration that the appearance of the target object will not usually change significantly or only change gradually, which has been verified in figure 4.1. Therefore, it is likely to be able to use the previously used basis vectors to represent the current target appearance. Since the sparsity patterns are updated with the changes of the templates, it means that any occlusion that introduces appearance change gradually will be tracked and the sparsity pattern updated in a timely fashion. Thus our approach can also handle occlusions like previous ℓ_1 trackers. One critical issue for our approach might be rapid changes of the object appearance resulting from abrupt illumination changes or fast motions. In this case, the learnt sparsity pattern may not be able to represent the target appearance and thus causes failure in tracking.

An alternative sparsity model for visual tracking. We also propose an alternative sparsity model that relooks at how the tracking problem can be formulated. In this model, we swap the roles of the candidate and object template sets in the ℓ_1-norm minimization problem. In other words, we

express the template appearance using a sparse linear combination of the candidates. If a candidate sample is not chosen to represent any template, it is unlikely to be the correct candidate for further tracking and hence directly rejected.

Since this new model pursues a sparsest representation from the candidates, it is likely to choose the true target as a basis vector if it exists in the candidate set. That is, in the ideal case, if the appearance of the object does not change and the true target exists in the candidate set, the sparsest solution would be a coefficient vector with all zeros except one entry corresponding to the true target. In the real world with noise and outliers, it is likely to choose the true target as the most important vector (large coefficient value), together with a few other candidates to model the noise or outliers (e.g. occlusions). Since the true target furnishes the most important basis vector, the reconstruction error would be very large if we remove it from the retained candidate set. Therefore, the observation likelihood can be computed from the reconstruction error by removing each candidate from the retained candidate set. The larger the error is, the more important the candidate is and the higher the likelihood of the candidate.

Given the fact that the number of object templates is usually very small (10 in most ℓ_1 trackers), the new sparsity model only needs to solve several ℓ_1 minimization problems. Though the dictionary will be larger, we do not need to use trivial templates. Therefore, there are $n \times N$ unknowns in our case and $(N + 2D) \times n$ unknowns for the conventional ℓ_1 tracker, where D, N and n are the data dimension, number of templates and candidates respectively. And if we implement the algorithms in matrix form using ADMM, the scale of our problem is smaller, given the fact that ADMM is dominated by matrix multiplications and elementwise operations. As can be seen from the experimental results later, the speed of our simple novel model is close to the ℓ_1 tracker that has been accelerated by using

both minimum error bound [74] and APG [13] together. It should be noted that in this case, the SPL approach mentioned above cannot be applied, because now the dictionary is always changing for each frame. In terms of tracking accuracy, a very basic version of this novel idea with the alternative holistic sparse representation shows better performance comparing to the previous ℓ_1 trackers [73, 13, 74, 121] using the conventional holistic sparse representation. Note that it has been pointed out in the latest benchmark paper [115] that besides holistic representation, other components such as local sparse representation and background information are critical for effective tracking. We did not implement these other components in this thesis as we are only looking at how adopting a different perspective to the sparse representation might improve performance. As a consequence, compared to those trackers with more information (e.g. [51] and [123]), the performance of our simple tracker with the holistic sparse representation alone is somewhat inferior. However, it is not difficult to integrate these information into our tracker, just as [51] and [123] did.

To summarize, our major contributions are as follows.

- We accelerate the ℓ_1 tracker by first learning the sparsity patterns of the template set and then reconstructing the candidates with the learnt patterns, resulting in a very fast algorithm. We also propose fast methods to update the sparsity patterns using the previously learnt results.

- We formulate the visual tracking problem from a different perspective and propose an alternative sparsity model for visual tracking. A simple setting of the novel sparsity model shows better performance than conventional ℓ_1 trackers on a recently released benchmark.

4.2 Related Works

Previous approaches to the visual tracking problem can be roughly separated into the generative and discriminative methods.

Generative methods learn the appearance model of the target object and track the object by a state searching scheme w.r.t. the matching score. Incremental visual tracking (IVT) [86] is a subspace based tracking method, which learns appearance changes by incremental PCA. Visual tracking decomposition (VTD) [58] decomposes the observation and motion model into multiple basic models by sparse PCA, resulting in multiple basic trackers. Then all basic trackers communicate with one another through an interactive Markov Chain Monte Carlo (MCMC) framework to achieve the tracking result.

The aforementioned ℓ_1 tracker also belongs to this category and there are many extensions [13, 51, 67, 74, 109, 110, 116, 121, 123]. Among them, bounded particle resampling (BPR) [74] and L1APG [13] are two direct extensions proposed to improve the tracking speed. L1BPR [74] calculates the minimum error bound of candidates by solving the ℓ_2-norm minimization problems and discards the candidates with large errors in a resampling stage. Thus the number of ℓ_1-norm minimizations is reduced. Essentially, it rejects some candidates by ℓ_2-norm minimization, thus avoiding having to perform ℓ_1-norm minimization for these candidates. Our SPL algorithm, however, converts all ℓ_1-norm minimization problems into ℓ_2-norm minimization problems and is thus more efficient. L1APG [13] accelerates the ℓ_1 tracker via a fast numerical solver (i.e. APG), while our algorithm adopts ADMM to solve the ℓ_1-norm minimization problem during the SPL stage. Note that ADMM has been proven to be faster than APG and also shows higher precision in the RPCA problem [66].

Other representative sparsity based tracking methods are reviewed below. For a comprehensive survey, please refer to [120]. ASLA [51] exploits

the partial and spatial information of the target by sparse presentation of local patches. LSK [67] also uses a local sparse appearance model. Sparsity based collaborative model (SCM) [123] exploits both holistic templates and local representations and designs a collaborative model with both generative and discriminative abilities. MTT [121] utilizes joint sparsity to respect the underlying relationships between sampled particles. [110] updates the object templates via an online robust non-negative dictionary learning algorithm and establishes its equivalence to the ℓ_1 tracker.

Discriminative methods learn binary classifiers and find the best decision boundary for separation of the target object and the background. Multiple instance learning (MIL) [8] proposes an online multiple instance learning approach, which considers the samples within positive or negative bags. Tracking-Learning-Detection (TLD) [54] decomposes the tracking task into tracking, learning and detection. It uses an online semi-supervised learning algorithm and is able to recover from failure because of its detection phase. Context tracker (CXT) [34] exploits the context information; the similar regions are also tracked to avoid drifting and the local keypoints around the truth target with consistent co-occurrence and motion correlation are used to support the tracker. Compressive tracking (CT) [119] reduces the dimensionality of foreground and background samples by a random sparse projection matrix, resulting in effective features, which can be separated using a naive Bayes classifier. Struck [47] formulates the tracking problem as one of structured output prediction, which directly predicts the change in object location between frames.

Besides visual tracking, another closely related research problem is dynamic compressive sensing [41, 5]. Dynamic compressive sensing considers dynamic systems when recovering sparse signals. These dynamics may arise from time varying signals, streaming measurements or adaptive signal transforms. One focus of the dynamic compressive sensing problem is

to quickly update the solution of the ℓ_1-norm minimization problem for a varied system from an already solved ℓ_1 problem of the original system, which is closely related to the update of sparsity patterns in our algorithm. A popular solution to this problem is the homotopy methods [41, 6, 7], which solve an optimization problem by gradually transforming it into a related problem for which the solution is either available or easy to compute, following a so called homotopy path.

Another closely related work is the fast abnormal event detector [69], which also learns the sparsity patterns but uses it to detect outliers (abnormal events). In this work, there is no step for sparsity pattern update.

4.3 Background

To facilitate the presentation of our approaches, we first give a brief review of the particle filter framework for visual tracking [4] and the ℓ_1 tracker [73].

4.3.1 Particle filter for visual tracking

Particle filter [37] is also known as the sequential Monte Carlo method for importance sampling. It has been widely used in visual tracking due to its simplicity and effectiveness. In the context of visual tracking, let $\mathbf{z}_t$ be the observation at frame t and $\mathbf{s}_t$ be the state variable describing the location and shape of a target. The tracking problem estimates the posterior state distribution $p(\mathbf{s}_t|\mathbf{z}_{1:t})$, where $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\}$. Mathematically, it can be calculated using a two-stage Bayesian sequential estimation, which updates the filtering distribution recursively:

$$p(\mathbf{s}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{s}_{t-1}, \tag{4.1}$$

$$p(\mathbf{s}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})}. \tag{4.2}$$

Since direct calculation of the above distribution is practically intractable, the particle filter approach approximates $p(\mathbf{s}_t|\mathbf{z}_{1:t})$ by a set of samples $\{\mathbf{s}_t^i\}_{i=1}^n$ (a.k.a. particles) with importance weights $\{w_t^i\}_{i=1}^n$. These samples are generated by sequential importance distribution $q(\mathbf{s}_t|\mathbf{s}_{1:t-1}, \mathbf{z}_{1:t})$ and the weights are updated as

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{z}_t|\mathbf{s}_t^i)p(\mathbf{s}_t^i|\mathbf{s}_{t-1}^i)}{q(\mathbf{s}_t|\mathbf{s}_{1:t-1}, \mathbf{z}_{1:t})}. \tag{4.3}$$

Following the assumption of the first order Markov process, $q(\mathbf{s}_t|\mathbf{s}_{1:t-1}, \mathbf{z}_{1:t}) = p(\mathbf{s}_t|\mathbf{s}_{t-1})$, the weights are then updated as $w_t^i = w_{t-1}^i p(\mathbf{z}_t|\mathbf{s}_t^i)$. In this case, the weights of some particles may keep increasing, falling into a degenerate case. To avoid such a case, in each step, samples are re-sampled to generate a new sample set with equal weights according to their weights distribution.

4.3.2 The ℓ_1 tracker

For the ℓ_1 tracker, the state variable $\mathbf{s}_t$ is the affine transformation with six parameters. The state transition distribution $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is modeled independently by a Gaussian distribution, and the observation model $p(\mathbf{z}_t|\mathbf{s}_t)$ reflects the similarity between the candidate and the target templates. Under the particle filter framework, it is important to model this similarity, and the ℓ_1 tracker formulates it from the error approximated by the target templates using ℓ_1-norm minimization.

Given a template set $\mathbf{T}_t \in \mathbb{R}^{D\times N_t}$, whose columns are the vectorized and normalized templates at frame t, let $\mathbf{Y}_t \in \mathbb{R}^{D\times n_t}$ be the corresponding candidate set, whose columns represent the vectorized target patches. For

each candidate patch $\mathbf{y}_t^i$, the ℓ_1 tracker [73] solves the following problem:

$$\mathbf{c}_t^i = \min_{\mathbf{c}} \frac{1}{2}\|\mathbf{y}_t^i - \mathbf{D}_t\mathbf{c}\|_F^2 + \lambda\|\mathbf{c}\|_1, \quad s.t. \quad \mathbf{c} \geq 0, \tag{4.4}$$

where $\mathbf{D}_t = [\mathbf{T}_t, \mathbf{I}, -\mathbf{I}]$ and $\mathbf{I}$ is the identity matrix, which presents the trivial template set. Then $\mathbf{c}_t^i$ can be divided into two parts $[\mathbf{c}_t^i(1:N_t); \mathbf{c}_t^i(N_t+1:N_t+2D)]$, which correspond to the coefficients for the template set and trivial template set respectively. And the observation likelihood is derived from the reconstruction error

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{1}{\Gamma} exp(-\alpha\|\mathbf{y}_t^i - \mathbf{T}_t\mathbf{c}_t^i(1:N_t)\|_F^2), \tag{4.5}$$

where α is a constant controlling the shape of the Gaussian kernel and Γ is a normalization factor. The optimal patch is chosen as the candidate with the maximum observation likelihood. Then the template set is updated accordingly. For more details about the ℓ_1 tracker and template update, please refer to [73, 74].

4.4 Visual Tracking via Sparsity Pattern Learning

4.4.1 Sparsity Pattern Learning

With a slight abuse of notation, we ignore the subscript t from now on, and let $\mathbf{T} \in \mathbb{R}^{D\times N}$ be the template set as in the previous section. For each template $\mathbf{t}^i$ in $\mathbf{T}$, we solve the following minimization problem:

$$\begin{aligned} \min_{\mathbf{c}} \quad & \tfrac{1}{2}\|\mathbf{t}^i - \mathbf{D}\mathbf{c}\|_F^2 + \lambda\|\mathbf{c}\|_1, \\ s.t. \quad & \mathbf{c} \geq 0, \quad \mathbf{c}(i) = 0, \end{aligned} \tag{4.6}$$

which essentially uses the other template and trivial patches to sparsely represent the template patch $\mathbf{t}^i$. We rewrite it in a matrix form:

$$\begin{aligned} \min_{\mathbf{C}} \quad & \tfrac{1}{2}\|\mathbf{T}-\mathbf{D}\mathbf{C}\|_F^2+\lambda\|\mathbf{C}\|_1, \\ s.t. \quad & \mathbf{C}\geq 0, \quad \Omega(\mathbf{C})=0, \end{aligned} \tag{4.7}$$

where $\Omega(\cdot)$ are the entries of the matrix with the same row and column number, i.e. $\{\mathbf{C}(i,i)\}_{i=1}^N$. This minimization is very similar to the outlier version of the sparse subspace clustering algorithm [39], but the purpose of this minimization in our algorithm is completely different.

(4.7) can be efficiently solved using the ADMM method [18, 66]. We first convert (4.7) to the following equivalent problem:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\mathbf{T}-\mathbf{D}\mathbf{C}\|_F^2+\lambda\|\mathbf{J}\|_1, \\ s.t. \quad & \mathbf{C}=\mathbf{J}-\Omega(\mathbf{J}), \\ & \mathbf{J}\geq 0, \end{aligned} \tag{4.8}$$

where $\mathbf{J}$ is the intermediate variables introduced to make the problem tractable. The augmented Lagrange function is

$$\begin{aligned} \mathcal{L} =& \tfrac{1}{2}\|\mathbf{T}-\mathbf{D}\mathbf{C}\|_F^2+\lambda\|\mathbf{J}\|_1+ \\ & tr(\mathbf{L}^T(\mathbf{C}-\mathbf{J}+\Omega(\mathbf{J}))+\tfrac{1}{2\mu}\|\mathbf{C}-\mathbf{J}+\Omega(\mathbf{J})\|_F^2, \end{aligned} \tag{4.9}$$

where $\mathbf{L}$ is the Lagrange multiplier, and $\mu>0$ is a penalty parameter, which affects the convergence of the algorithm. The function can be minimized with respect to $\mathbf{C}$ and $\mathbf{J}$ alternatingly, by fixing the other variable, and then updating the Lagrange multiplier $\mathbf{L}$. The overall framework of the alternating direction method is shown in Algorithm 4,

After solving $\mathbf{C}$, we record the sparsity pattern of each column $\mathbf{c}^i$. More specifically, for each column $\mathbf{c}^i$ in $\mathbf{C}$, assume there are k^i non-zero entries. We construct a selection matrix $\mathbf{S}^i\in\mathbb{R}^{(N+2D)\times k^i}$ such that $\mathbf{D}\mathbf{S}^i$ comprises

Algorithm 4 Solving (4.8) by ADMM

Input: template set $\mathbf{T} \in \mathbb{R}^{D\times N}$, parameters λ and μ.
Initialize: $\mathbf{C} = \mathbf{J} = \mathbf{L} = \mathbf{0}_{(N+2D)\times N}$, $\mathbf{D} = [\mathbf{T}, \mathbf{I}, -\mathbf{I}] \in \mathbb{R}^{D\times(N+2D)}$, $\epsilon = 10^{-8}$.
while not converged **do**
Step 1 Fix the others and update $\mathbf{C}$ as
$\mathbf{C} = (\mathbf{D}^T\mathbf{D} + \frac{1}{\mu}\mathbf{I})^{-1}(\mathbf{D}^T\mathbf{T} - \mathbf{L} + \frac{1}{\mu}\mathbf{J})$.
Step 2 Fix the others and update $\mathbf{J}$ as
$\mathbf{J}' = \mathcal{T}_{\lambda\mu}(\mu\mathbf{L} + \mathbf{C})$,
$\mathbf{J} = \mathbf{J}' - \Omega(\mathbf{J}')$,
$\mathbf{J} = max(0, \mathbf{J})$,
where $\mathcal{T}_\tau(\cdot)$ is the shrinkage-thresholding operator acting on each element of the matrix, and is defined as $\mathcal{T}_\tau(v) = \Pi_g(|v| - \tau)sgn(v)$, and Π_g is the projection operator acting on each element of the matrix, and is defined as $\Pi_g(v) = \begin{cases} 0 & \text{if} \quad v < 0, \\ v & \text{otherwise} . \end{cases}$
Step 3 Update the multipliers
$\mathbf{L} = \mathbf{L} + \frac{1}{\mu}(\mathbf{C} - \mathbf{J})$.
Step 4 Check the convergence conditions:
$\|\mathbf{C} - \mathbf{J}\|_\infty \leq \epsilon$.
end while

only those columns in $\mathbf{D}$ corresponding to the non-zeros entries in $\mathbf{c}^i$. Thus, when we test the candidate patches $\mathbf{Y}$, we only solve the following least squares problem:

$$\begin{aligned} \min_{\hat{\mathbf{c}}} \quad & \tfrac{1}{2}\|\mathbf{y}^i - \mathbf{D}\mathbf{S}^j\hat{\mathbf{c}}\|_F^2, \\ s.t. \quad & \hat{\mathbf{c}} \geq 0, \\ & i = 1, \ldots, n, \\ & j = 1, \ldots, N. \end{aligned} \tag{4.10}$$

The optimal candidate patch is then obtained according to the usual measure of observation likelihood. Now the remaining problem is how to quickly update $\mathbf{D}$ and $\mathbf{C}$ if we know the solution at the previous time t.

4.4.2 Fast update of the sparsity patterns

Since the most costly part of Algorithm 4 is the inverse operation $(\mathbf{D}^T\mathbf{D} + \frac{1}{\mu}\mathbf{I})^{-1}$, we perform a fast update of the inverse using the results from the previous computation.

Since the template update of the ℓ_1 tracker replaces only one column of $\mathbf{D}$ during the update [73], we assume one column $\mathbf{p}$ of $\mathbf{D}$ is replaced by a column $\mathbf{q}$ at time $t+1$, so $\mathbf{D}_{t+1} = \mathbf{D} - [\mathbf{0} \ \cdots \ \mathbf{p} \ \cdots \ \mathbf{0}] + [\mathbf{0} \ \cdots \ \mathbf{q} \ \cdots \ \mathbf{0}] = \mathbf{D} + [\mathbf{0} \ \cdots (\mathbf{q}-\mathbf{p}) \ \cdots \ \mathbf{0}]$. Let $\mathbf{Q} = [\mathbf{0} \ \cdots (\mathbf{q}-\mathbf{p}) \ \cdots \ \mathbf{0}]$. Then

$$\begin{aligned} \mathbf{D}_{t+1}^T\mathbf{D}_{t+1} + \tfrac{1}{\mu}\mathbf{I} &= \mathbf{D}^T\mathbf{D} + \tfrac{1}{\mu}\mathbf{I} + \mathbf{Q}^T\mathbf{D} + \mathbf{D}^T\mathbf{Q} + \mathbf{Q}^T\mathbf{Q} \\ &= \mathbf{A} + \mathbf{v}_1\mathbf{u}_1^T + \mathbf{u}_1\mathbf{v}_1^T + \mathbf{u}_2\mathbf{u}_2^T, \end{aligned} \tag{4.11}$$

where $\mathbf{u}_1 = \mathbf{D}^T(\mathbf{q}-\mathbf{p})$, $\mathbf{v}_1 = [0 \ \cdots \ 1 \ \cdots \ 0]^T$, $\mathbf{u}_2 = [0 \ \cdots \ q \ \cdots \ 0]^T$, and $q = \sqrt{(\mathbf{q}-\mathbf{p})^T(\mathbf{q}-\mathbf{p})}$. Consequently, the inverse of $(\mathbf{D}_{t+1}^T\mathbf{D}_{t+1} + \frac{1}{\mu}\mathbf{I})$ can be updated by applying the Sherman - Morrison formula (see Appendix B.2) three times. Note that we use the same ADMM algorithm as in Algorithm 4, but $\mathbf{C}_{t+1}$ is initialized as is the optimal solution of (4.7) at time t, so usually $\mathbf{C}_{t+1}$ is already very close to the optimal solution.

The fast update can also be achieved if $\mathbf{D}$ is incrementally updated. We assume $\mathbf{D}$ is updated at time $t+1$ and $\mathbf{D}_{t+1} = [\mathbf{D} \ \mathbf{d}]$. We now update $\mathbf{C}_{t+1}$. We again use the same ADMM algorithm as in Algorithm 4, but $\mathbf{C}_{t+1}$ is initialized as $\mathbf{C}_{t+1} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$, where $\mathbf{C}$ is the optimal solution of (4.7) at time t. Meanwhile, the most time consuming part – the inverse of $(\mathbf{D}_{t+1}^T\mathbf{D}_{t+1} + \frac{1}{\mu}\mathbf{I})$ – is updated by a blockwise inversion (see Appendix B.1). Since

$$\begin{aligned} \mathbf{D}_{t+1}^T\mathbf{D}_{t+1} + \tfrac{1}{\mu}\mathbf{I} &= \begin{bmatrix} \mathbf{D}^T \\ \mathbf{d}^T \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{d} \end{bmatrix} + \tfrac{1}{\mu}\mathbf{I} \\ &= \begin{bmatrix} \mathbf{D}^T\mathbf{D} + \frac{1}{\mu}\mathbf{I} & \mathbf{D}^T\mathbf{d} \\ \mathbf{d}^T\mathbf{D} & \mathbf{d}^T\mathbf{d} + \frac{1}{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & d \end{bmatrix}, \end{aligned} \tag{4.12}$$

where $\mathbf{A} = \mathbf{D}^T\mathbf{D} + \frac{1}{\mu}\mathbf{I}$, $\mathbf{b} = \mathbf{D}^T\mathbf{d}$ and $d = \mathbf{d}^T\mathbf{d} + \frac{1}{\mu}$. Thus,

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & d \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \frac{1}{m}\mathbf{A}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{A}^{-1} & -\frac{1}{m}\mathbf{A}^{-1}\mathbf{b} \\ -\frac{1}{m}\mathbf{b}^T\mathbf{A}^{-1} & \frac{1}{m} \end{bmatrix}, \tag{4.13}$$

where $m = d - \mathbf{b}^T\mathbf{A}^{-1}\mathbf{b}$.

With the fast update procedures, an optimal strategy is to start one thread for updating the dictionary and sparsity patterns and another thread for solving (4.10) for tracking.

4.5 An Alternative Sparse Representation Approach

In this section, we propose a novel sparsity model for visual tracking, in which we reverse the roles of the template and candidate set in the sparse representation.

With the candidate samples $\mathbf{Y} \in \mathbb{R}^{D\times n}$, the conventional ℓ_1 tracker models the candidate appearance using a sparse representation of the template set. Unlike the conventional ℓ_1 tracker, we use the candidate set as a dictionary and model the template appearance using a sparse linear combination of the candidate set. Formally, for each template $\mathbf{t}^i$ in $\mathbf{T}$ we solve the following problem:

$$\begin{aligned} \min_{\mathbf{c}} \quad & \tfrac{1}{2}\|\mathbf{t}^i - \mathbf{Y}\mathbf{c}\| + \lambda\|\mathbf{c}\|_1, \\ s.t. \quad & \mathbf{c} \geq 0. \end{aligned} \tag{4.14}$$

If a candidate sample turns in zero scores in $\mathbf{c}$ for all templates, it can be directly filtered out from the candidate set. In other words, if this candidate sample is not selected to represent any template, it is unlikely to be the tracked target. Since the template appearance is sparsely reconstructed,

there are only a few candidates retained for each template. For the retained candidate set of each template, we remove a candidate each time and calculate the reconstruction error to derive the observation likelihood. More specifically, let $\mathbf{Y}_i \in \mathbb{R}^{D \times n_i}$ be the retained candidate set for the template $\mathbf{t}^i$. For the j-th candidate $\mathbf{y}_i^j$, we remove it from $\mathbf{Y}_i$ and denote the new matrix as $\mathbf{Y}_i^{\hat{j}}$. We then solve

$$\mathbf{c}_j = \min_{\mathbf{c}} \|\mathbf{t}^i - \mathbf{Y}_i^{\hat{j}}\mathbf{c}\|_F^2, \tag{4.15}$$

and compute the observation likelihood from the reconstruction error

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{1}{\Gamma} exp(\alpha \|\mathbf{t}^i - \mathbf{Y}_i^{\hat{j}}\mathbf{c}_j\|_F^2), \tag{4.16}$$

where α is a constant controlling the shape of the Gaussian kernel and Γ is a normalization factor as before. Note that in (4.16), the larger the reconstruction error is, the more important this candidate is, which is in contrast to (4.5). The tracking result is then chosen as the candidate with the maximum observation likelihood. For candidates appearing in multiple sets for different templates, we choose the maximum value as this candidate's observation likelihood.

4.6 Experiments

In this section, we evaluate the performance of the two proposed trackers on a recent online object tracking benchmark [115]. This dataset consists of 50 commonly used video sequences with fully annotated bounding boxes. It categorizes the sequences by annotating them with different attributes, such as illumination variation, occlusion, fast motions, background clutters and so on. Note that one sequence may have several attributes, so these categories are not mutually exclusive. With the categories, we can better

analyze the performance of the trackers under different conditions.

We denote our tracker with sparsity pattern learning as L1SPL and the alternative sparsity model as L1ASM in the evaluation. Our trackers are implemented with MATLAB and run on a PC with Intel i7 CPU 2.9 GHz. L1SPL runs at about 15 frames per second (fps), which is much faster than the original ℓ_1 tracker [73] and 7-8 times faster than L1APG [13] [1] (2 fps on the same PC). L1ASM also runs at about 2 fps without any accelerating technique, which is comparable to L1APG. We note that the cost of sparse reconstruction using L1SPL is very low, being several orders faster than L1APG. However, the real speed bottleneck is now the sampling procedure with affine transformation.

Following the evaluation methodology in the benchmark paper [115], we use the precision plot measuring the center location error and success plot measuring the bounding box overlap for quantitative analysis.

The center location error is defined as the average Euclidean distance between the center locations of the tracked objects and the groundtruth. Then the average error over all the frames of a sequence is used to rate the overall performance on that sequence. However, when the tracker loses the target, the output location can be arbitrary and the average error value may not show the tracking performance authentically. Thus precision plot has been adopted to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth.

The bounding box overlap is defined as $\frac{area(r_t \bigcap r_g)}{area(r_t \bigcup r_g)}$, where $area(\cdot)$ returns the area of the region, r_t and r_g are the tracked bounding box and groundtruth bounding box respectively, and $\bigcap$ and $\bigcup$ represent the intersection and union of two regions respectively. Following [115], we count

[1] The tested L1APG has been accelerated by both the minimum error bound [74] and APG [13] together, whereby the minimum error bound significantly reduces the number of ℓ_1-norm minimizations

the number of successful frames whose overlap area is larger than a given threshold. The success plot shows the ratios of successful frames at the thresholds varying from 0 to 1. Using one success rate value at a specific threshold for evaluation may not be fair or representative. Therefore, we use the area under curve (AUC) of each success plot to rank the tracking algorithms as in [115].

We test our trackers using the one-pass evaluation (OPE). In other words, we run them throughout a test sequence with initialization from the ground truth position in the first frame and draw the average precision or success rate. For the compared 29 trackers which have been evaluated in [115], we directly use the reported results in that paper. Most of these trackers with high performance have been reviewed in section 4.2. For the works which have not been reviewed, more details can be found in [115]. Figure 4.2 shows the overall performance of the trackers. Due to the limited space, those approaches ranked very low are dropped in the figure.

It can be observed from the precision plots in Figure 4.2 that L1SPL outperforms L1APG by 2.4%, and its performance is very close to that of L1APG considering the success rate in Figure 4.2. The performance of L1APG is worse off probably because it only iterates a maximum of 5 times when solving the ℓ_1-norm minimization, which means the APG ℓ_1 solver may not have converged in practice. This amounts to sacrificing performance for efficiency, but even then, it is still not fast enough. Given that the L1SPL is 7-8 times faster than L1APG, the accuracy performance of L1SPL is very satisfactory; we believe that L1SPL achieves better balance between performance and efficiency.

In figure 4.2, the performance of L1ASM is shown to be better than all those algorithms using only holistic sparse representation i.e. L1SPL, L1APG and MTT. It is ranked 9 in the precision plots and 10 in the success plots. Of those methods whose performances are superior to L1ASM, there

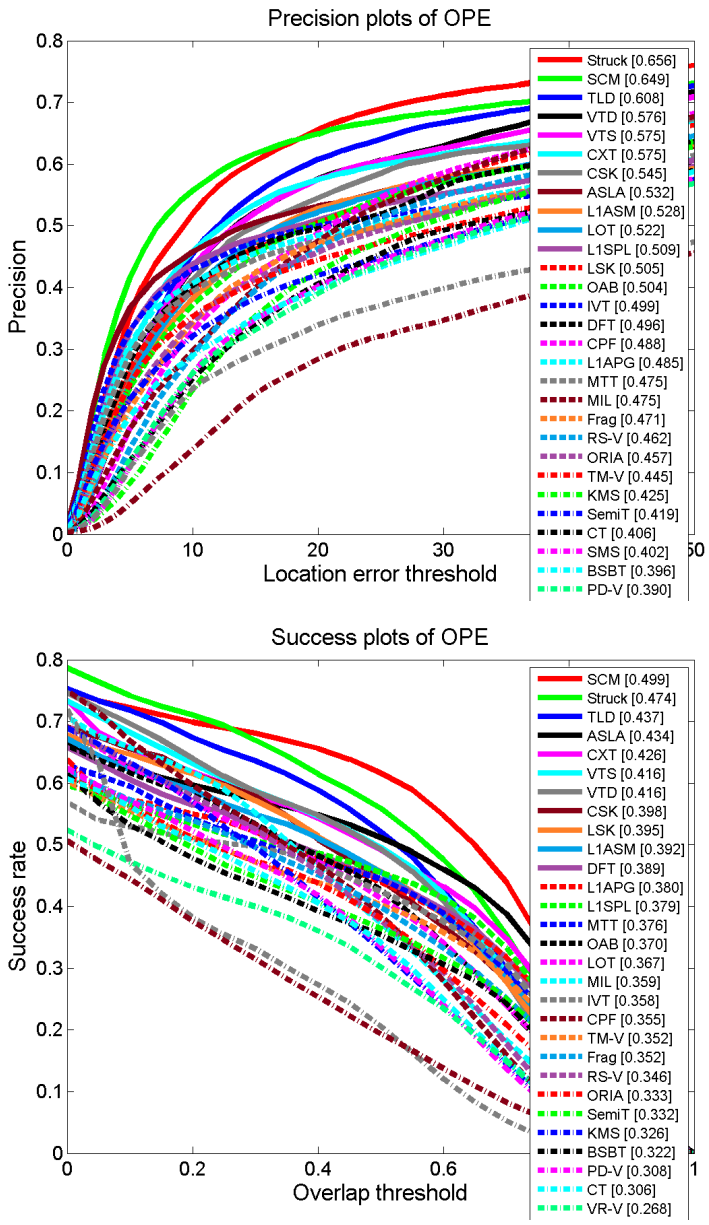

Figure 4.2: Overall visual tracking performance plots

are two other sparsity based methods, i.e. ASLA [51] and SCM [123]. ALSA uses local sparse representation and SCM further adds discriminative ability to the tracker. As mentioned in [115], local models are useful when the appearance of target is partially changed, such as partial occlusion or deformation; background information is critical for effective tracking because discriminative ability makes the tracker more robust to the drifting problem. Thus, a combination of all (i.e. SCM) is ranked top in the evaluation. As a method with only holistic sparse representation, L1ASM

is ranked a creditable top 10 overall. We believe that it is a better sparsity model and there is room to further improve it by combining the local and background information.

We also depict the performance plots of various categories with several different attributes and analyze the performance of our trackers under different factors. Here, we only depict those categories with occlusion, deformation and illumination variation in Figure 4.3, Figure 4.4, and Figure 4.5 respectively. The complete attribute-based performance plots are shown in Appendix C.1

In the occlusion category, both L1PSL and L1ASM perform well. It shows that the acceleration using SPL does not degrade the ability of handling occlusion. Indeed, L1ASM shows better results, which are even close to that of those methods with structured learning and local sparse representations (e.g. Struck and TLD).

In the deformation category, L1SPL works well. Again it shows that gradual changes of the object appearance won't affect the accelerated ℓ_1 tracker using SPL. L1ASM works less well, but it still outperforms L1APG, MTT and many other methods.

In the next category, illumination variation poses significant challenge to L1SPL. As illumination variation may cause rapid appearance changes, the learnt sparsity patterns may not be suitable for the reconstruction in the next frame. Thus, the ranking of L1SPL in this category is very low. Fortunately, L1ASM still performs well in this category.

4.7 Conclusions

We propose two visual tracking approaches. L1SPL is a fast ℓ_1 tracker, which learns sparsity patterns of the templates and thus only needs to solve small-scale ℓ_2-norm minimization problems. The reconstruction step

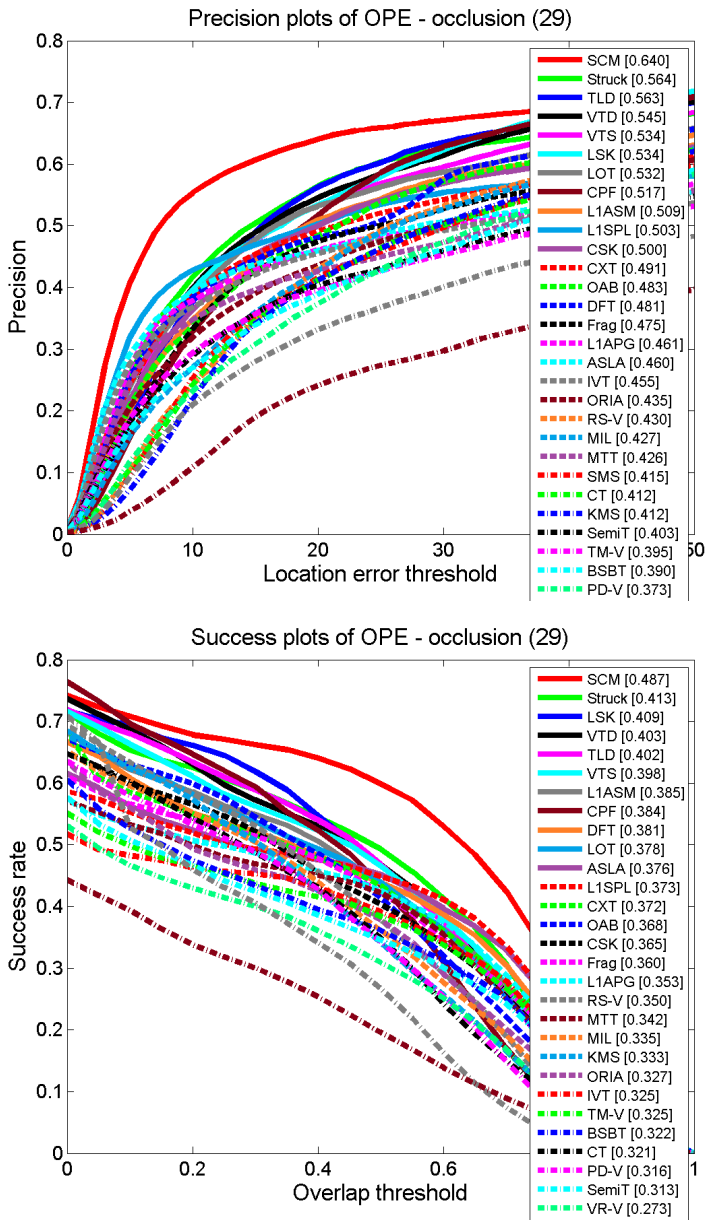

Figure 4.3: Visual tracking performance plots - Occlusion

is several orders faster than the ℓ_1-norm minimization. The speed of the whole procedure is 7-8 times faster than the ℓ_1 tracker accelerated by APG. The performance of L1SPL is close to or even slightly better than those previous ℓ_1 trackers. L1ASM considers the tracking problem from a novel perspective, expressing the template appearance using a sparse linear combination of the candidate samples. With this novel sparsity model, and without any of local and background representation, L1ASM is creditably ranked in the top 10 out of 31 trackers on a recent benchmark. The per-

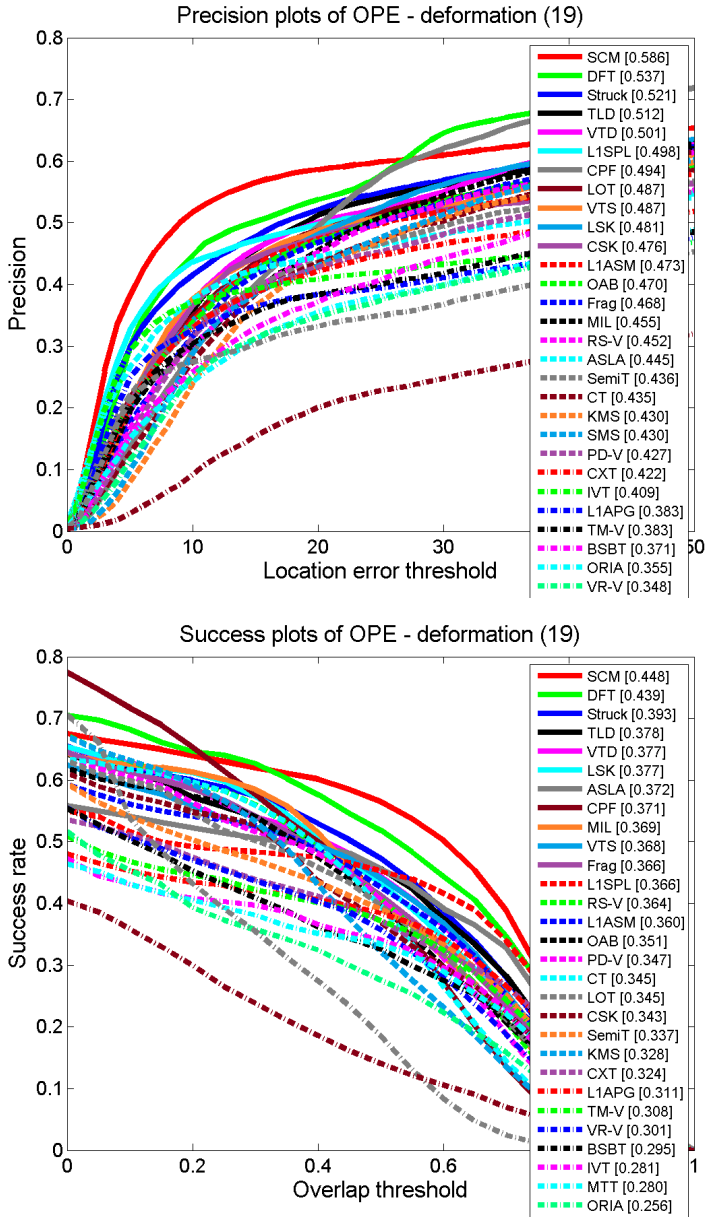

Figure 4.4: Visual tracking performance plots - Deformation

formance of L1ASM is better than all those sparsity based methods using only holistic sparse representation.

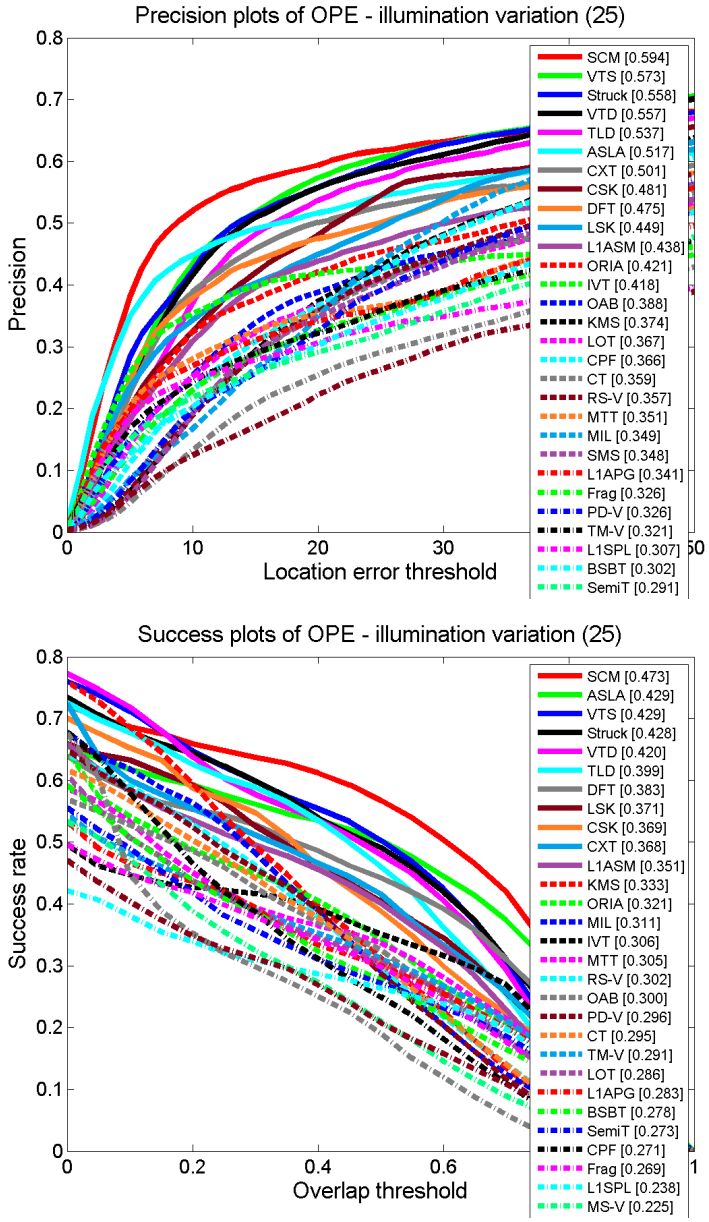

Figure 4.5: Visual tracking performance plots - Illumination Variation

Chapter 5

Conclusions and Future Works

In this thesis, we have considered three topics that exploit the low-dimensional structures of several motion problems, obtaining better low-dimensional models for them, solving the real world challenges often encountered in these problems, and achieving state-of-the-art performance on the standard datasets. Below we summarize the major contributions of the thesis and then put forth potential future works related to this thesis.

5.1 Summary of Contributions

For the 3D motion segmentation problem (Chapter 2), we propose a joint sparsity model, which combines affinities of the point correspondence in multiple image pairs. This novel sparsity model is capable of handling perspective effect because it is based on the epipolar constraint of two views, while simultaneously leveraging the rich information across multiple frames, avoiding ambiguities caused by a short observation duration of two frames. The nature of the joint sparsity model also leads to a simple means to handle missing data, without having to revise the optimization mechanism.

Embedded in any segmentation/clustering problem is the model selection problem (Chapter 3), which aims to estimate the number of clusters. For this problem, we propose a simultaneous low-rank and sparse model, where the rank function models the complexity of the model and the car-

dinality function is used to avoid the trivial solution and suppress small affinities. Under the ADMM framework, we solve two nonconvex subproblems, which are a more faithful representation of the original problem than that using the convex relaxation approach. Though it is hard to prove its convergence, it shows strong convergence behavior in our experiments. With these global and local costs and some other constraints, this model usually reveals the underlying structures of the affinity matrix, resulting in a perfect block diagonal matrix up to a permutation. This block diagonal matrix can then be directly used to indicate the data elements' clusters by factorization.

For the object tracking problem (Chapter 4), a common sparsity based approach models the candidate appearance using a sparse linear combination of the templates. However, this approach suffers from high computational cost that prevents its use in real-time applications. To speed up the method, we learn the patterns of the sparsity model and convert the high cost ℓ_1 norm minimization problems into the small scale ℓ_2 norm minimization problems. As a result, significant speedup is achieved without loss of performance. In addition, we propose an alternative sparsity model, which, reversing the roles of the candidates and templates, utilizes the candidate samples to sparsely represent the templates. This allows us to rapidly prune away large number of candidates which are not chosen for any template recovery, following which the observation likelihood is calculated based on the reconstruction error in a slightly different way. Finally the optimal candidate is chosen as per previous methods. This novel sparsity model outperforms other similar methods using holistic sparse representation and shows competitive results compared to other approaches augmented with additional information such as local sparse appearance model and background information.

5.2 Future Works

The works in this thesis are related to a couple of interesting problems, which might be worth further exploration.

5.2.1 Motion segmentation

Even after solving many real world challenges inherent in motion segmentation, it is still hard to segment motions in a very long sequence or for real-time applications. One reason is that the dimension of the data can be very high; the other reason is that the data are not processed in an online fashion. Both reasons suggest the use of dynamic subspace clustering approach. This approach solves the subspace clustering problem using temporal sliding windows; this brings with it gradual subspace variation, element changing and even vector dimension changing. It is possible to use the sparsity pattern learning and fast pattern update techniques proposed in Chapter 4 to solve this problem.

5.2.2 Clustering and model selection

One urgent work is to establish the theoretical guarantee for the convergence of ADMM applied to nonconvex problems. Existing work proves the Karush-Kuhn-Tucker (KKT) conditions and a preliminary convergence result indicating that whenever the algorithm converges, it must converge to a stationary point [90].

If the number of groups is given a prior, it is possible to design a fixed rank clustering representation as a competitive approach compared to spectral clustering.

Given the strong relationship between the QSAP problem and the Markov random field (MRF) labeling problem, another potential research direction is to cross-breed the latest research results between the clustering and labeling problems, both for speed and performance.

5.2.3 Visual tracking

The novel sparsity model, which models the template appearance using a linear combination of the candidate samples, has not been thoroughly exploited. Theoretical analysis of the model and deeper understanding of its performance should be established. Another possible research direction is to consider it as a filter and integrate it into the particle filter or other searching mechanism. The motion model or dynamic model can be similarly integrated. With all these components installed, it will be more likely to predict good locations to track, which will reduce the searching range and thus further improve the tracking efficiency and robustness.

Bibliography

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD*, 1993. 53

[2] H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–723, 1974. 4, 45

[3] C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel pca. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):335–347, 2010. 4, 45

[4] M. S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. 73

[5] M. S. Asif. *Dynamic Compressive Sensing: Sparse recovery algorithms for streaming signals and videos.* PhD thesis, Georgia Institute of Technology, 2013. 72

[6] M. S. Asif and J. K. Romberg. Dynamic updating for ℓ_1-norm. *J. Sel. Topics Signal Processing*, 4(2):421–434, 2010. 73

[7] M. S. Asif and J. K. Romberg. Fast and accurate algorithms for re-weighted ℓ_1-norm minimization. *IEEE Transactions on Signal Processing*, 61(23):5905–5916, 2013. 73

[8] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal.*

Mach. Intell., 33(8):1619–1632, 2011. 66, 72

[9] S. Bagon and M. Galun. Large scale correlation clustering optimization. *CoRR*, abs/1112.2903, 2011. 47

[10] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. Technical report, arXiv:1006.4046, 2011. 35

[11] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008. 46

[12] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *FOCS*, 2002. 46

[13] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Proc. CVPR*, pages 1830–1837, 2012. 5, 70, 71, 81

[14] R. Basri and D. Jacobs. Lambertian reflection and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(3):218–233, 2003. 62

[15] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997. 1, 2

[16] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, 2002. 45

[17] T. Boult and L. Brown. Factorization-based segmentation of motions. In *Proc. of the IEEE Workshop on Motion Understanding*, 1991. 10

[18] S. Boyd, N. Parikh, E. Chu, and B. Peleato. Distributed optimization and statistical learning via the alternating direction method of mul-

tipliers. *Foundations and Trends in Machine Learning*, 1(3):1–122, 2011. 16, 44, 49, 76

[19] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proc. ECCV*, 2010. 22

[20] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–591, 2008. 2

[21] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11, 2011. 2

[22] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Commun. ACM*, 55(6):111–119, 2012. 2

[23] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. 2, 66

[24] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2009. 14

[25] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012. 46

[26] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *International Journal of Computer Vision*, 80(1):125–142, 2008. 14, 35, 37

[27] A. M. Cheriyadat and R. J. Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *Proc. ICCV*, 2009. 19

[28] T.-J. Chin, D. Suter, and H. Wang. Multi-structure model selection via kernel optimisation. In *Proc. CVPR*, 2010. 21

[29] T.-J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *Proc. ICCV*, 2009. 20, 35

[30] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003. 66

[31] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998. 10, 11, 18

[32] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014. 2

[33] E. Demaine and N. Immorlica. Correlation clustering with partial information. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 2764, pages 1–13. 2003. 47

[34] T. B. Dinh, N. Vo, and G. G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Proc. CVPR*, pages 1177–1184, 2011. 72

[35] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *aide-memoire of a lecture at ams conference on math challenges of 21st century*, 2000. 1

[36] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 2, 66

[37] A. Doucet, D. N. Freitas, and N. Gordon. *Sequential Monte Carlo Methods In Practice*. Springer-Verlag New York, 2001. 73

[38] R. Dragon, B. Rosenhahn, and J. Ostermann. Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In *Proc. ECCV*, 2012. 20, 21, 22, 35

[39] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013. 2, 3, 10, 12, 13, 18, 24, 25, 33, 35, 43, 55, 76

[40] Z. Gao, L.-F. Cheong, and M. Shan. Block-sparse rpca for consistent foreground detection. In *Proc. ECCV*, pages 690–703, 2012. 2

[41] P. Garrigues and L. E. Ghaoui. An homotopy algorithm for the lasso with online observations. In *Proc. NIPS*, pages 489–496, 2008. 72, 73

[42] C. W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998. 10

[43] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):643–660, 2001. 61

[44] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satifisability probelms using semidefinite programming. *Journal of ACM*, 42(6):1115–21145, 1995. 44

[45] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *Proc. CVPR*, 2004. 10, 18

[46] L. W. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(9):1074–1085, 1992. 42

[47] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *Proc. ICCV*, pages 263–270, 2011. 66, 72

[48] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 12, 22, 23

[49] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):328–340, 2005. 2

[50] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 66

[51] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Proc. CVPR*, pages 1822–1829, 2012. 70, 71, 83

[52] Y. Jian and C. Chen. Two-view motion segmentation with model selection and outlier removal by ransac-enhanced dirichlet process mixture models. *International Journal of Computer Vision*, 88(3):489–2501, 2010. 12, 20, 21

[53] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag New York, 1986. 2

[54] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1409–1422, 2012. 72

[55] K. Kanatani. Geometric information criterion for model selection. *International Journal of Computer Vision*, 26(3):171–189, 1998. 4, 45

[56] K. Kanatani. Motion segmentation by subspace separation and model selection. In *Proc. ICCV*, 2001. 10, 18

[57] J. Kim, J. Choi, J. Yi, and M. Turk. Effective representation using ica for face recognition robust to local distortion and partial occlusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1977–1981, 2005. 2

[58] J. Kwon and K. M. Lee. Visual tracking decomposition. In *Proc. CVPR*, pages 1269–1276, 2010. 71

[59] T. Lange, V. Roth, M. Braun, and J. Buhmann. Stability based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, 2004. 45

[60] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motions cues. In *Proc. CVPR*, 2011. 22

[61] H. Li. Two-view motion segmentation from linear programming relaxation. In *Proc. CVPR*, 2007. 12, 20, 21

[62] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Proc. CVPR*, 2001. 2

[63] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *Proc. CVPR*, 2007. 12, 19

[64] Z. Li, L.-F. Cheong, and S. Z. Zhou. Scams: Simultaneous clustering and model selection. In *Proc. CVPR*, 2014. 7

[65] Z. Li, J. Guo, L.-F. Cheong, and S. Z. Zhou. Perspective motion segmentation via collaborative clustering. In *Proc. ICCV*, pages 1369–1376, 2013. 7

[66] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, Technical report, UILU-ENG-09-2215, 2009. 16, 44, 49, 71, 76

[67] B. Liu, J. Huang, C. A. Kulikowski, and L. Yang. Robust visual tracking using local sparse appearance model and k-selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2968–2981, 2013. 71, 72

[68] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern*

Anal. Mach. Intell., 35(1):171–184, 2013. 2, 3, 4, 10, 12, 18, 19, 20, 24, 28, 33, 35, 43, 45, 55

[69] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, pages 2720–2727, 2013. 73

[70] F. Lu, S. Keles, S.Wright, and G.Wahba. Framework for kernel regularization with application to protein clustering. In *Proceedings of the National Academy of Sciences*, 2005. 4

[71] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 4, 42, 55

[72] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1546–1562, 2007. 4, 45

[73] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(11):2259–2272, 2011. 2, 5, 66, 70, 73, 75, 78, 81

[74] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai. Efficient minimum error bounded particle resampling l1 tracker with occlusion detection. *IEEE Transactions on Image Processing*, 22(7):2661–2675, 2013. 5, 70, 71, 75, 81

[75] P. Miettinen. *Matrix Decomposition Methods for Data Mining: Computational Complexity and Algorithms.* PhD thesis, University of Helsinki, 2009. 52

[76] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. *IEEE Trans. Knowl. Data Eng.*, 20(10):1348–1362, 2008. 44, 52

[77] K. Mohan, M. J.-Y. Chung, S. Han, D. M. Witten, S.-I. Lee, and M. Fazel. Structured learning of gaussian graphical models. In *Proc. NIPS*, pages 629–637, 2012. 46

[78] A. Montanari and S. Oh. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 56(6):2988–2998, 2010. 35

[79] R. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7(1):33–46, 1991. 20

[80] Y. Ni, J. Sun, X. Yuan, S. Yan, and L. F. Cheong. Robust low-rank subspace segmentation with semidefinite guarantees. In *ICDM Workshops*, 2010. 51, 107

[81] A. Rakotomamonjy. Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Processing*, 91(7):1505–1526, 2011. 26

[82] W. M. Rand. Objective criteria for the evaluation of clustering methods. *American Statistical Association*, 66(336):846–850, 1971. 55

[83] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(10):1832–1845, 2010. 3, 10, 17, 18, 33, 35

[84] S. Rao, A. Yang, S. Sastry, and Y. Ma. Robust algebraic segmentation of mixed rigid-body and planar motions from two views. *International Journal of Computer Vision*, 88(3):425–446, 2010. 12, 20

[85] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978. 4, 45

[86] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 66, 71

[87] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Computational and Applied Math*, 20(1):53–65, 1987. 4, 45

[88] K. Schindler and D. Suter. Two-view multibody structure-and-motion with outliers through model selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):983–995, 2006. 12, 20, 21

[89] K. Schindler, J. U, and H. Wang. Perspective n-view multibody structure-and-motion through model selection. In *Proc. ECCV*, 2006. 20, 21, 22, 34

[90] Y. Shen, Z. Wen, and Y. Zhang. Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software*, 29(2):239–263, 2014. 52, 91

[91] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 4, 42

[92] D. Sinclair. Motion segmentation and local structure. In *Proc. ICCV*, 1993. 20

[93] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2011. 4, 16, 17, 20, 21, 29, 30, 35, 45, 56, 58

[94] S. Still and W. Bialek. How many clusters? an information theoretic perspective. *Neural Computation*, 16(12):2483–2506, 2004. 4, 45

[95] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. ECCV*, 1996. 12

[96] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Workshop on Statistical Methods in Video Processing*, 2004. 10, 18

[97] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *J. Royal Statistical Soc. B*, 63(2):411–423, 2001. 4, 45

[98] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *International Journal of Computer Vision*, 9(2):137–154, 1992. 10

[99] P. Torr and D. Murray. Stochastic motion clustering. In *Proc. ECCV*, 1994. 20

[100] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002. 4, 12, 20, 45

[101] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Proc. CVPR*, 2005. 11

[102] B. Triggs. Factorization methods for projective structure and motion. In *Proc. CVPR*, 1996. 12

[103] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Proc. CVPR*, 2007. 12, 60

[104] M. Turk and A. Pentland. Eigenfaces for recognition. In *Proc. CVPR*, 1991. 2

[105] R. Vidal and R. Hartley. Motion segmentation with missing data by power factorization and generalized pca. In *Proc. CVPR*, 2004. 10, 17, 18, 33

[106] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 68(1):7–25, 2006. 12, 20, 35

[107] S. N. P. Vitaladevuni and R. Basri. Co-clustering of image segments using convex optimization applied to em neuronal reconstruction. In *Proc. CVPR*, pages 2203–2210, 2010. 47

[108] B. Wang and Z. Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *Proc. CVPR*, 2012. 43

[109] D. Wang, H. Lu, and M.-H. Yang. Online object tracking with sparse prototypes. *IEEE Transactions on Image Processing*, 22(1):314–325, 2013. 71

[110] N. Wang, J. Wang, and D.-Y. Yeung. Online robust non-negative dictionary learning for visual tracking. In *Proc. ICCV*, pages 657–664, 2013. 71, 72

[111] T.-C. P. William B. Thompson. Detecting moving objects. *International Journal of Computer Vision*, 4(1):39–57, 1990. 20

[112] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *Proc. CVPR*, 2001. 12, 20

[113] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009. 2, 66

[114] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Proc. ACCV*, 2010. 2

[115] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proc. CVPR*, pages 2411–2418, 2013. 70, 80, 81, 82, 83

[116] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. In *Proc. ICCV*, pages 1100–1107, 2011. 71

[117] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *Proc. CVPR*, 2005. 11

[118] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proc. ECCV*, 2006. 10, 11, 18, 33

[119] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *Proc. ECCV*, pages 864–877, 2012. 72

[120] S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46(7):1772–1788, 2013. 71

[121] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, 101(2):367–383, 2013. 70, 71, 72

[122] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012. 18, 35, 55

[123] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Proc. CVPR*, pages 1838–1845, 2012. 70, 71, 72, 83

[124] D. Zhou and C. Burges. Spectral clustering and transductive learning with multiple views. In *Proc. ICML*, 2007. 26

Appendix A

A.1 Proof of Theorem 1

Similar to the proof of Theorem 16 in [80], we can first transfer problem (3.7) to the following equivalent problem:

$$\begin{aligned} \mathbf{G}^* = &\quad arg\min_{\mathbf{G}} \|\mathbf{G} - \widehat{\mathbf{S}}\|_F^2 + \lambda rank(\mathbf{G}), \\ s.t. &\quad \mathbf{G} \in \mathbf{S}_+. \end{aligned} \tag{1}$$

Note that $\widehat{\mathbf{S}} = (\mathbf{S} + \mathbf{S}^T)/2$.

Then following the proof of Theorem 14 in [80], we can similarly transfer problem (1) to

$$\begin{aligned} \{\xi_i^*\}_{i=1}^N = &\quad arg\min_{\{\xi_i\}_{i=1}^N} \sum_i^N \|\xi_i - \lambda_i\|_F^2 + \lambda\|\xi_i\|_0, \\ s.t. &\quad \forall i, \quad \xi_i > 0, \end{aligned} \tag{2}$$

where $\{\lambda_i\}_{i=1}^N$ are the diagonal entries of $\mathbf{\Lambda}$. Note that $\mathbf{\Lambda}$ is from the spectrum(eigen-) decomposition of $\widehat{\mathbf{S}} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. Then the proof will be identical to the first part of the proof of Theorem 2.

A.2 Proof of Theorem 2

Since optimization of the elements of $\mathbf{H}$ in Equation 3.10 are separable, each elements h_{ij} can be optimized individually:

$$\sum_{i,j} \min_{h_{ij}} (h_{ij} - m_{ij})^2 + \gamma\|h_{ij}\|_0 + g(h_{ij}). \tag{3}$$

If $h_{ij} \neq 0$, the best $(h_{ij} - m_{ij})^2 + \gamma\|h_{ij}\|_0$ can achieve is γ with $h_{ij} = m_{ij}$; if $h_{ij} = 0$, $(h_{ij} - m_{ij})^2 + \gamma\|h_{ij}\|_0 = m_{ij}^2$. Thus, the minimum of $(h_{ij} - m_{ij})^2 + \gamma\|h_{ij}\|_0$ is $\min(m_{ij}^2, \gamma)$, with each term achieved by $h_{ij} = 0$ and $h_{ij} = m_{ij}$ respectively. Thus, if $m_{ij}^2 \leq \gamma$, $h_{ij} = 0$; otherwise, $h_{ij} = m_{ij}$. With the additional box constraint $[0, 1]$, if the minimum is achieved by $h_{ij} = m_{ij} < 0$, we project h_{ij} to 0, because 0 is the closest value to $m_{ij} < 0$ in $[0, 1]$, and the cost of $\gamma\|h_{ij}\|_0$ is 0; if the minimum is achieved by $h_{ij} = m_{ij} > 1$, we should also project it to $[0, 1]$. In this case, the minimum is $\min(m_{ij}^2, (1 - m_{ij})^2 + \gamma)$, with each term achieved by $h_{ij} = 0$ and $h_{ij} = 1$ respectively. Thus, if $2m_{ij} > \gamma + 1$, $h_{ij} = 1$; otherwise, $h_{ij} = 0$.

Appendix B

B.1 Blockwise Inversion

Matrices can be inverted blockwise by using the following analytic inversion formula:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}. \tag{4}$$

B.2 Sherman - Morrison Formula

Suppose $\mathbf{A}$ is an invertible square matrix and $\mathbf{u}$, $\mathbf{v}$ are vectors. Suppose furthermore that $1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u} \neq 0$. Then the Sherman - Morrison formula states that

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}. \tag{5}$$

Appendix C

C.1 More Visual Tracking Results based on Attributes

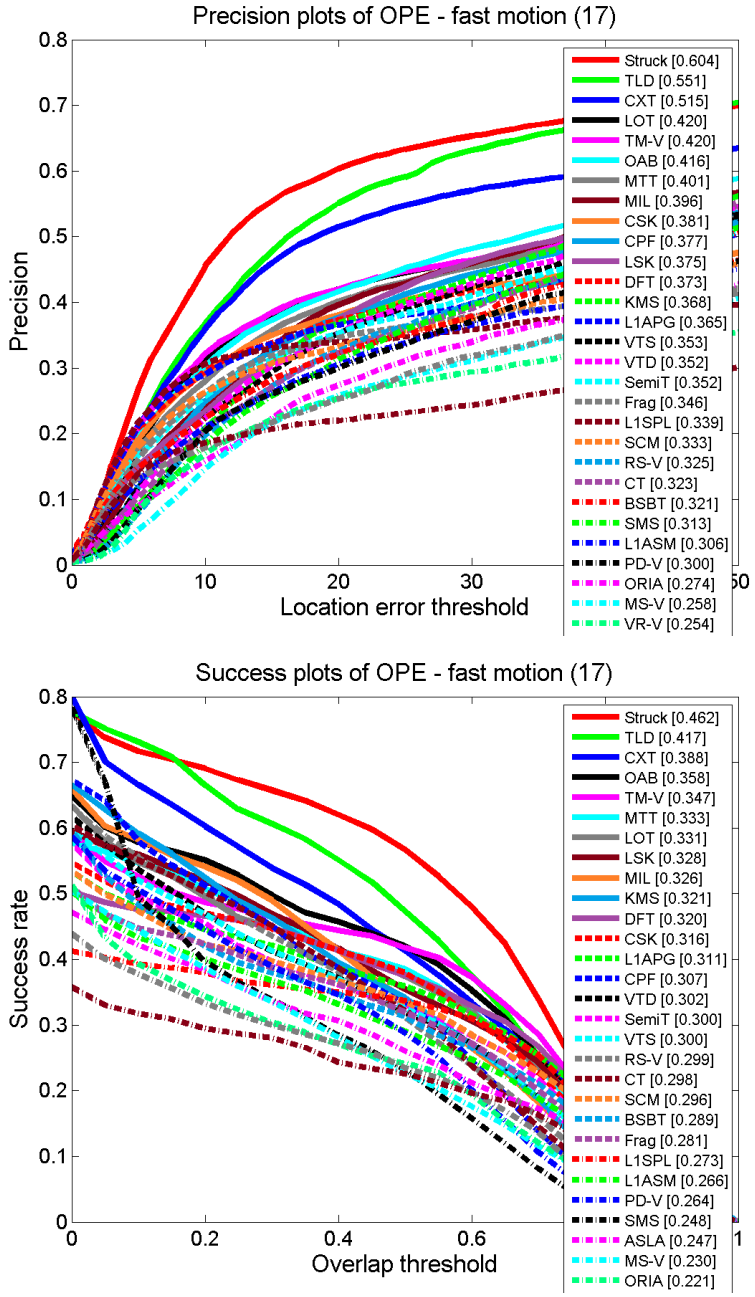

Figure C.1: Visual tracking performance plots - Fast Motion

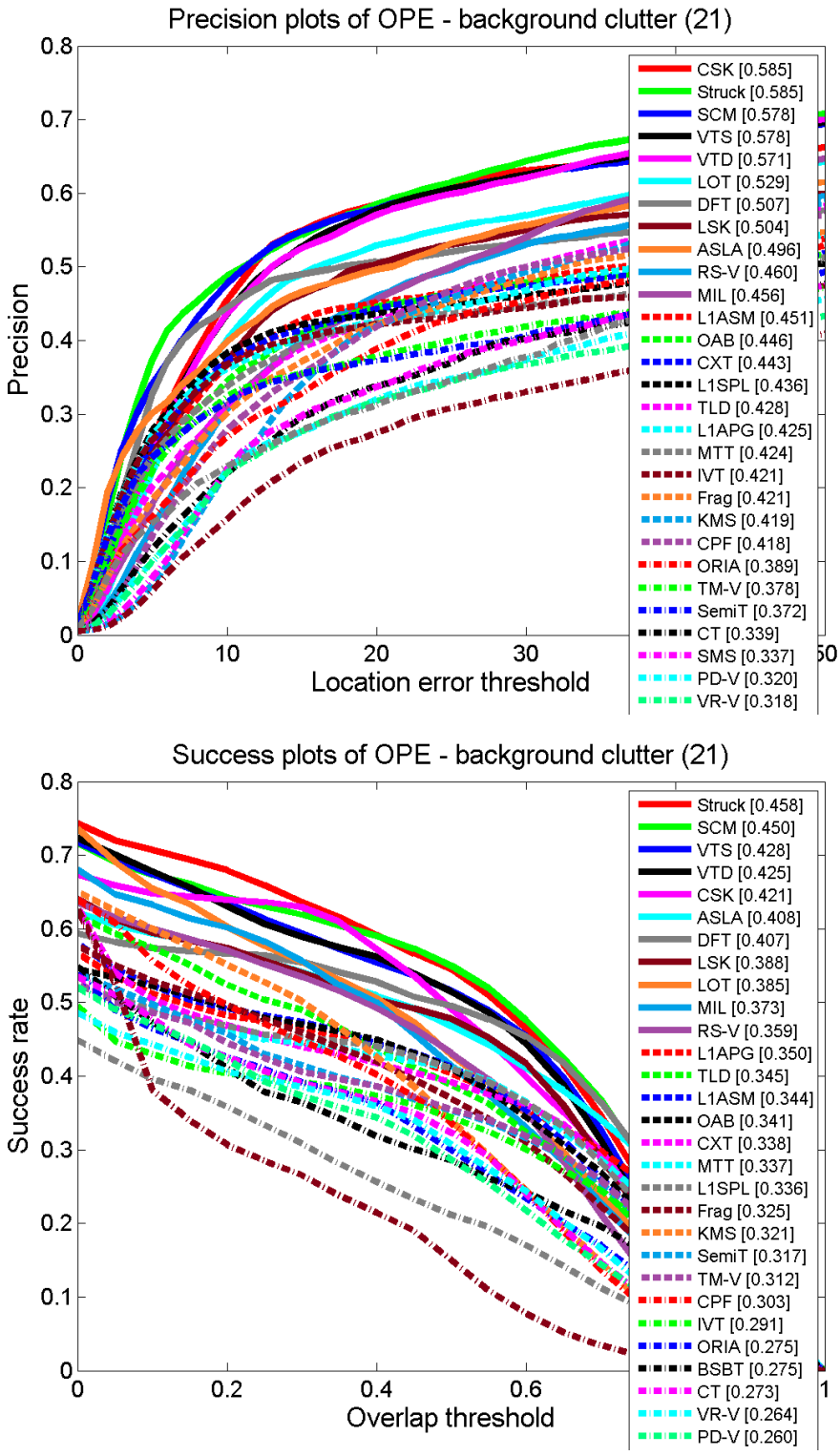

Figure C.2: Visual tracking performance plots - Background Clutter

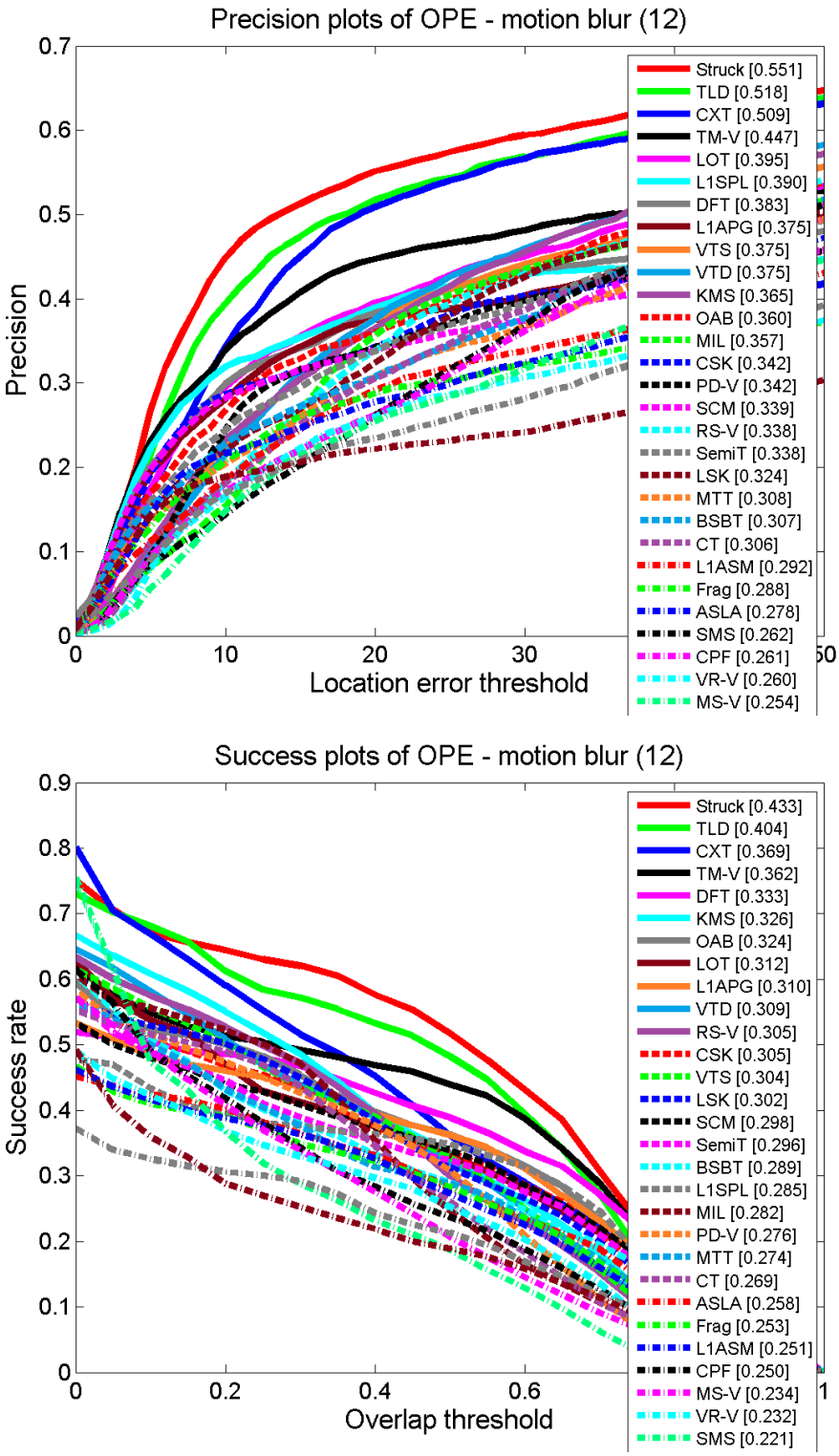

Figure C.3: Visual tracking performance plots - Motion Blur

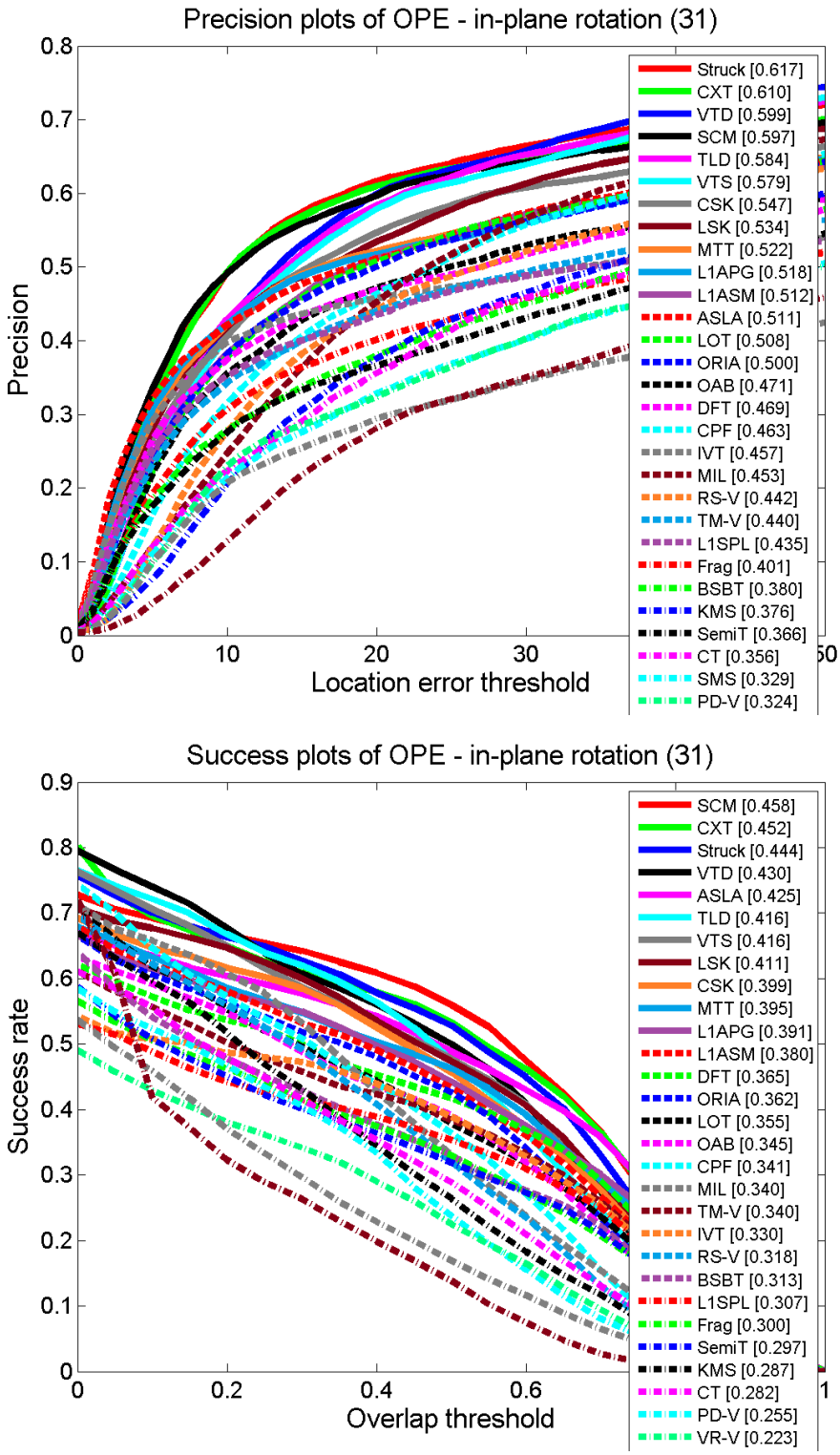

Figure C.4: Visual tracking performance plots - In-Plane Rotation

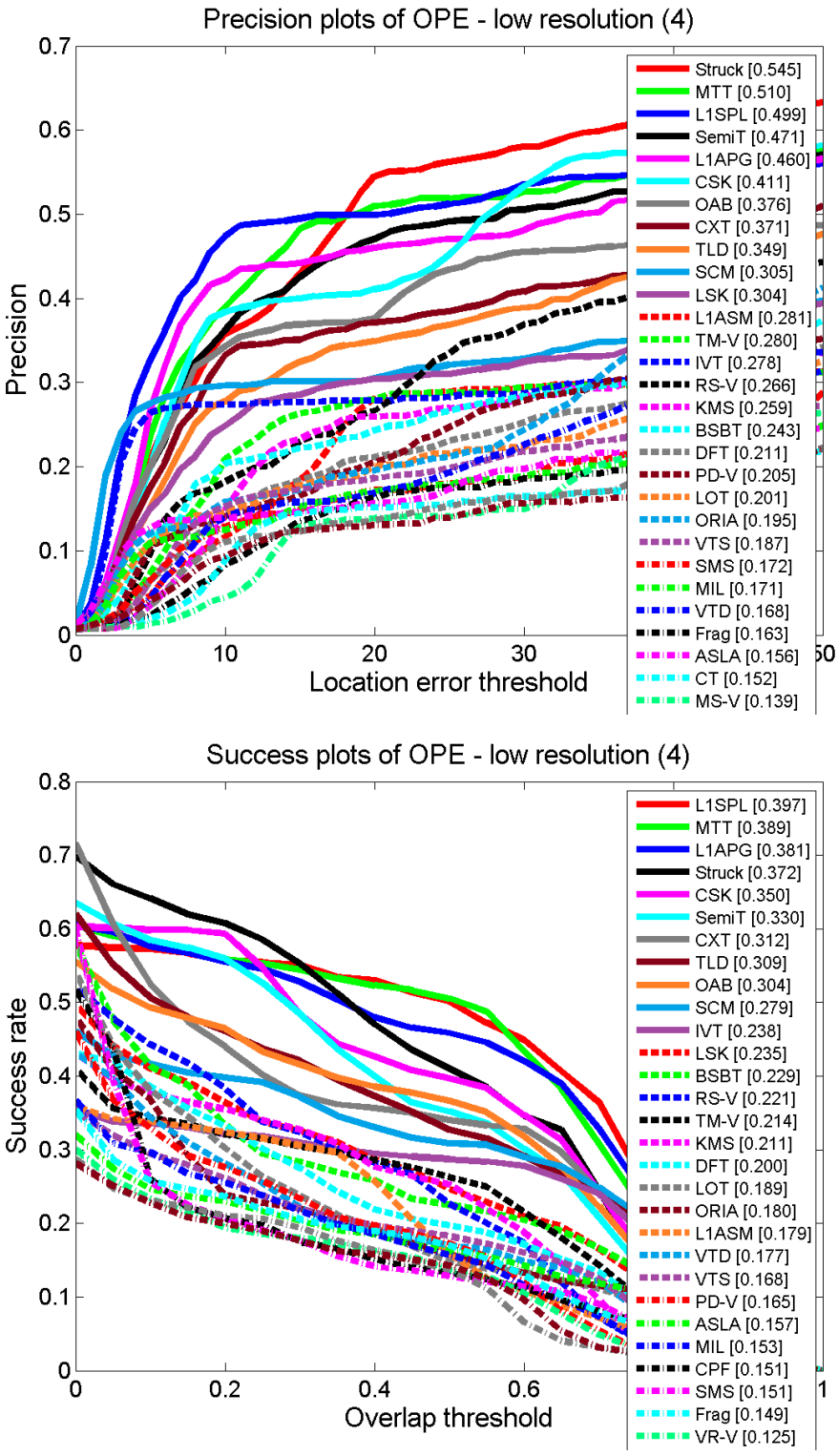

Figure C.5: Visual tracking performance plots - Low Resolution

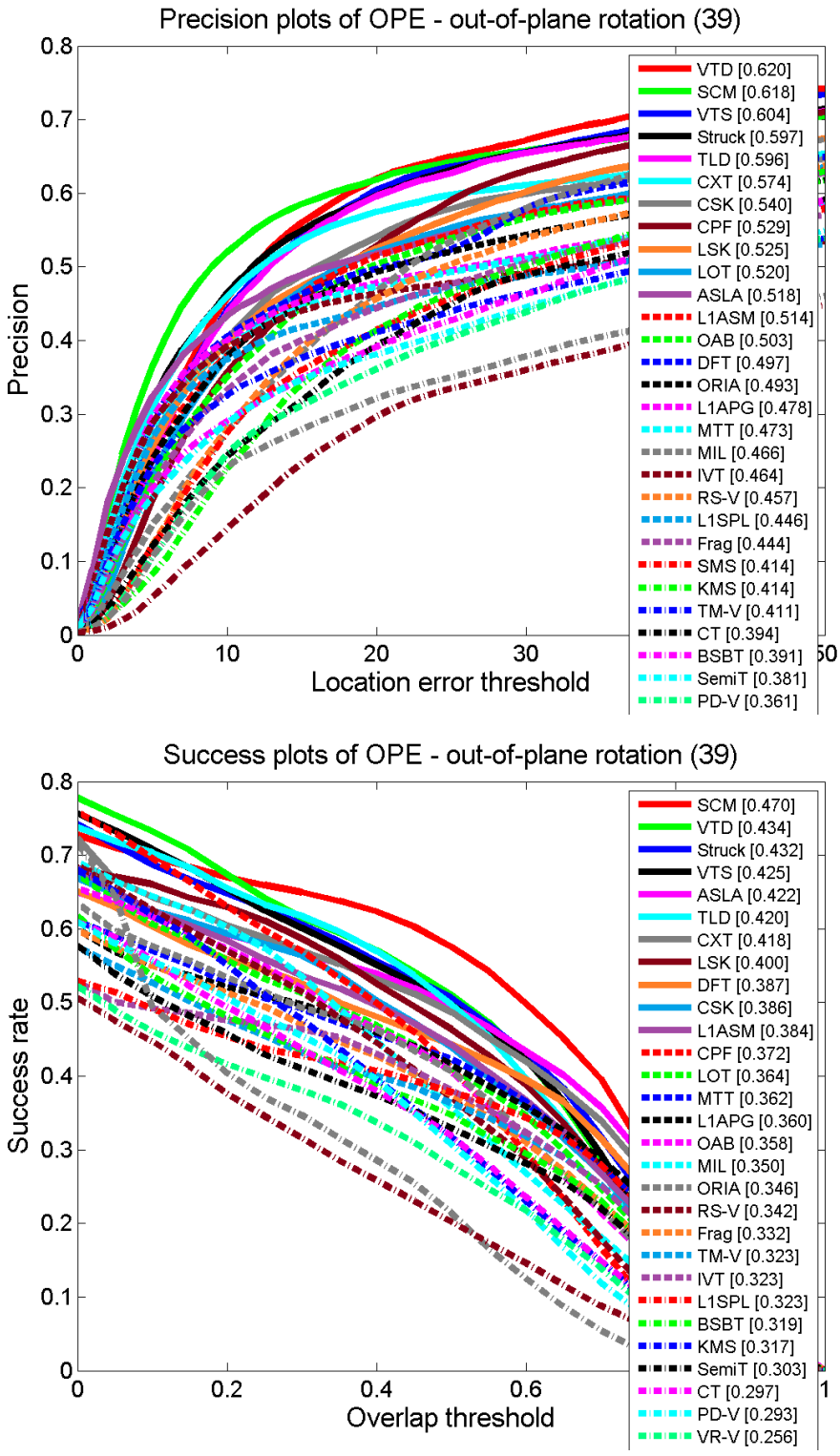

Figure C.6: Visual tracking performance plots - Out-of-Plane Rotation

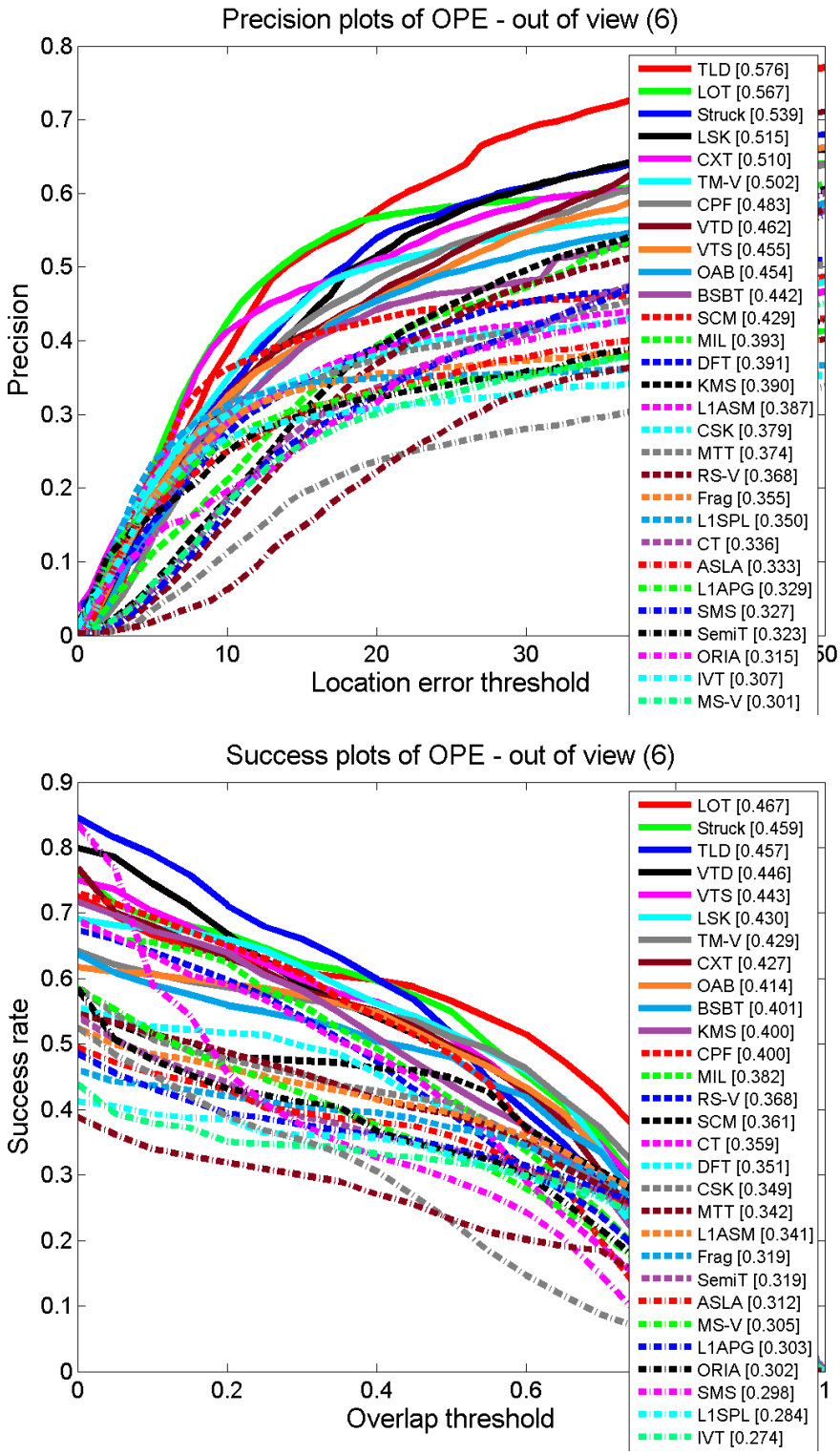

Figure C.7: Visual tracking performance plots - Out-of-View

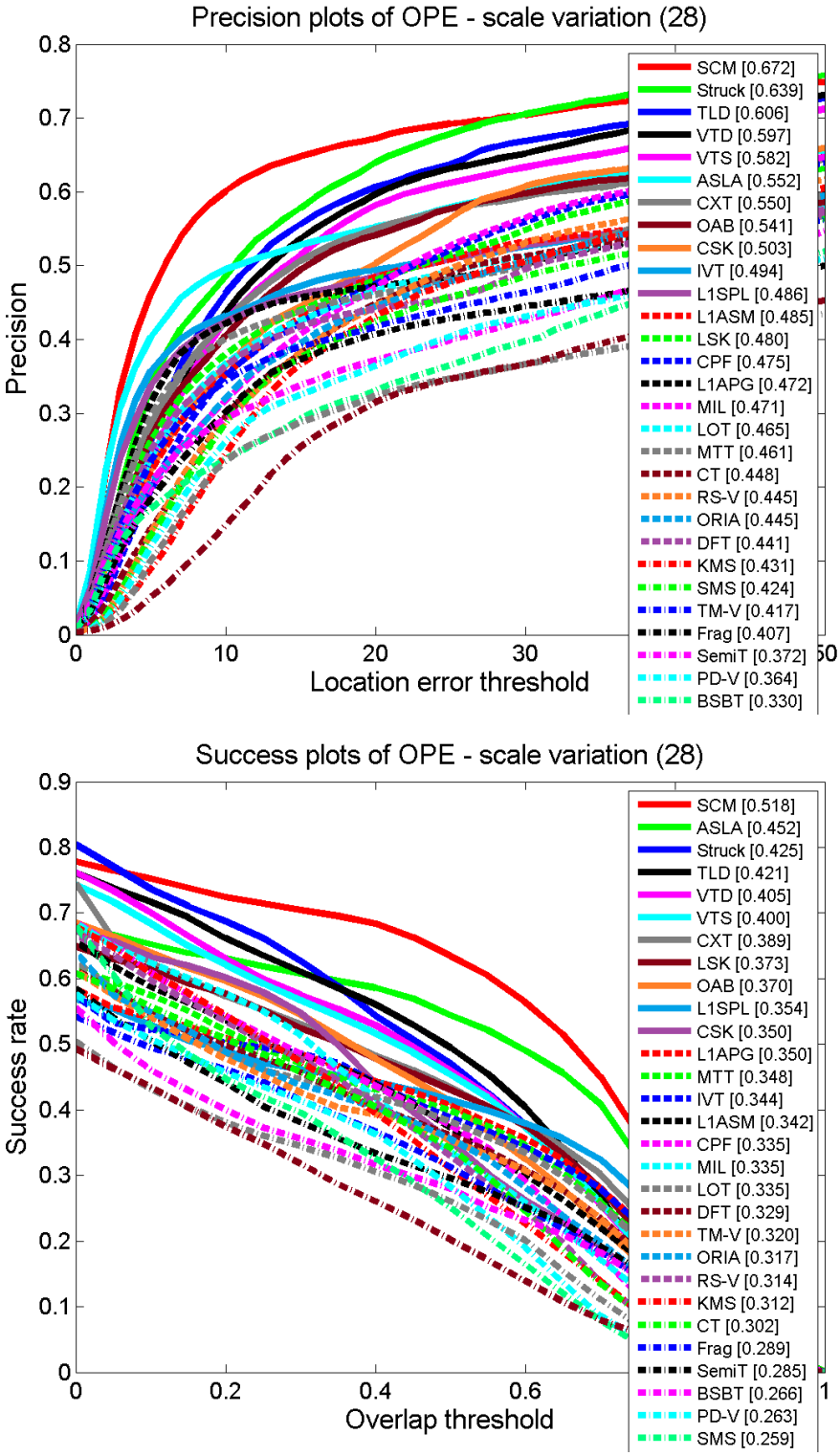

Figure C.8: Visual tracking performance plots - Scale Variation