

Sparse coding based image restoration and recognition: algorithms and analysis

Chenglong Bao

(B.Sc., Sun Yat-sen University, China)

**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Department of Mathematics
National University of Singapore**

2014

To my family.

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.

Chenglong Bao

2014

Handwritten signature in Chinese characters: 包承龙

Acknowledgements

This dissertation would never be completed or written without the guidance of my committee members, help from friends, and support from my family and my wife.

First and foremost I would like to express my deepest gratitude to my supervisors Associate Professor Hui Ji and Professor Defeng Sun for their fundamental roles in my doctoral work. They have been supportive since the days I began working with them and thanks to them I had the opportunity to dive into the applied mathematics. Ever since, they have not only provided me great help on my research projects, but also inspired me by their wisdom, knowledge and commitment with high standard.

Besides my supervisors, I would like to thank Professor Zuowei Shen for his precious suggestions, insightful comments and great encouragement. I remember he used to say something like "you need to connect all things together" to encourage my current and future research. I would also like to acknowledge all members in the NUS Wavelet group, Chaoqiang Liu, Kang Wang, Zhitao Fan, Zheng Gong, Likun Hou, Sibin Huang, Jia Li, Ming Li, Yuhui Quan, Peichu Xie, Yufei Zhao, Yu Luo, Yuping Sun, Jianbin Yang, Chunlin Wu, Heinecke Andreas. The numerous discussions with them helped me improved my knowledge in the research topics.

I am also grateful to my badminton coach Jiandee Chew and friends in badminton team, Xiaoxia Ye, Weiming Miao, Liangliang Wang, Xinyue Liu, Jiabin Wu, Qiushi Zhuang, Ruilun Cai, Aiqiang Zhang, Sengkee Chua, Yuzhi Shi, Gongyun Zhao, Zhengqi Huang, Shengjie Sun, Xin Zhong, Meng Ren, who made my life remarkable. I would also like to thank my friends, Xin Wang, Wenwen Huang, Jiayin Ye, Weiming Miao, Weijia Gu, Jingyi Chen, for their valuable friendship.

Finally, I would like to thank my family for all their love and encouragement. My parents raised me and supported me in my pursuits. And most of all, my loving wife Linlin Miao was always there cheering me up and stood by me through good and bad times.

Contents

Contents	ix
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Background	3
1.1.1 Dictionary learning for image restoration and recognition	4
1.1.2 Dictionary learning algorithms	7
1.1.3 Proximal methods	9
1.2 Motivations and contributions of the dissertation	12
1.2.1 Data-driven tight frame construction	13
1.2.2 Redundant dictionary learning	15
1.2.3 Incoherent dictionary learning	16
1.2.4 L1 visual tracker	18
1.3 Notation	18
2 Data-driven tight frame construction for image restoration	21
2.1 Introduction	21
2.2 Brief review on data-driven tight frame construction and related works	23
2.2.1 Tight frames and data-driven tight frames	23
2.2.2 Data-driven tight frame construction scheme	25
2.2.3 Related works	26
2.3 Sub-sequence convergence property of Algorithm 1	27
2.4 A modified algorithm for (2.7) with sequence convergence	32
2.4.1 Convergence analysis of Algorithm 2	33
2.5 Experiments on image denoising	39

2.6	Extensions	41
2.6.1	Problem formulation	41
2.6.2	Numerical method	42
2.6.3	Complexity analysis of Algorithm 3	45
2.6.4	Applications in image restoration	46
2.6.5	Experiments	48
2.6.6	Discussion and conclusion	53
3	Redundant dictionary learning for image restoration and recognition	55
3.1	Introduction	55
3.1.1	Motivation	56
3.1.2	Main contributions	57
3.2	Related work	58
3.2.1	ℓ_0 norm based methods	58
3.2.2	Convex relaxation methods	59
3.2.3	Non-convex relaxation methods	59
3.3	Algorithm and convergence analysis	60
3.3.1	Problem formulation	60
3.3.2	Alternating proximal method	61
3.4	Global convergence of Algorithm 6	65
3.5	Experiments	68
3.5.1	Image denoising	69
3.5.2	Face recognition	70
3.6	Summary	74
4	Incoherent dictionary learning for image recognition	75
4.1	Introduction	75
4.1.1	Motivation and main contributions	77
4.1.2	Related work	79
4.2	Incoherent dictionary learning algorithm	80
4.2.1	Problem formulation	81
4.2.2	A hybrid alternating proximal algorithm	81
4.3	Convergence analysis of Algorithm 7	87
4.4	Experiments	87
4.4.1	Experimental setting	87
4.4.2	Experimental results	89

4.5	Summary and conclusions	91
5	Sparse coding based visual tracking	99
5.1	Introduction	99
5.2	Related work	100
5.3	Introduction to L1 Tracker	102
5.4	Real time L1 Tracker	104
5.4.1	A modified ℓ_1 norm related minimization model	105
5.4.2	Fast numerical method for solving (5.9)	106
5.5	Experiments	112
5.5.1	Comparison with the existing L1 Tracker	112
5.5.2	Qualitative comparison with other methods	112
5.5.3	Quantitative comparison with other methods	114
5.6	Conclusion	114
	Bibliography	117

Summary

Image restoration and recognition are basic tasks in imaging and vision science. One key question in image recovery or recognition is how to effectively express the essential characteristic of images. In the last decade, the sparse representation or approximation of images has been one popular approach to regularize images in recovery or characterize images in recognition. The basic idea in sparse representation of images is that most images are compressible in some domain, i.e., an image of interest can be effectively expressed by the linear combination of very few atoms in some system (so-called dictionary). Owing to significant variations of image content, the dictionary used for sparsely expressing images has been adaptive to images of interest. As a whole, such procedure is the so-called *sparse coding*.

The sparse coding contains two coupled parts: one is how to compute sparse coefficients of the input under the dictionary and the other is how to find the dictionary that can generate optimal sparse coefficients. Therefore, in most applications, it leads to a challenging non-convex optimization problem. Many numerical methods have been proposed to solve such a non-convex optimization problem. However, most existing methods are derived from the heuristic arguments and often there are not convergence results provided for these methods. In this dissertation, we aim at developing fast numerical methods to solve variational problems often seen in practical sparse coding problems. Furthermore, the convergence analysis of these proposed methods are also established in this dissertation.

This dissertation begins by investigating the convergence behavior for iterative data-driven tight frame construction scheme [19] that is a solver for the dictionary learning problem with orthogonal constraint on the learned dictionary. We established the sub-sequence convergence property of the iteration scheme proposed in [19], and further showed that the method proposed in [19] can be modified to have sequence convergence property. In addition, an extension of the above orthogonal dictionary learning is proposed by fixing part of atoms of learned dictionary. This extension can further accelerate the dictionary learning process with satisfactory results in image restoration.

The second part of this dissertation is devoted to developing fast and convergent numerical methods to solve ℓ_0 norm based dictionary learning problem [1] which is to learn a

redundant dictionary without the orthogonal constraint on the learned dictionary. Based on proximal methods, our proposed method is theoretically proved to generate a convergent sequence that converges to a stationary point of the original non-convex minimization problem with comparable results in image restoration and face recognition. Moreover, our proposed method is much faster than the K-SVD method [1], which is validated in experiments.

The third part of this dissertation developed a hybrid proximal method for solving the incoherent dictionary learning problem as the low mutual coherence of a dictionary is an important property that ensures the optimality of the sparse code generated from this dictionary. The proposed incoherent dictionary learning method is not only of proved convergence, but also can benefit many sparse coding based face and object recognition methods, as shown in the experiments.

The final part of this dissertation applied the sparse representation to the visual tracker by modeling the target appearance using a sparse approximation over a template set. We proposed a modified the ℓ_1 tracker to improve the tracking accuracy and a fast numerical solver for the resulting ℓ_1 norm related minimization problem, using accelerated proximal gradient method. The real time performance and tracking accuracy of the proposed tracker is validated with a comprehensive evaluation involving eight challenging sequences and five alternative state-of-the-art trackers.

List of Figures

1.1	Pre-defined dictionary, learned dictionary and their denoising results	2
1.2	Image inpainting result.	5
1.3	Some exemplar face images from Extend Yale face database B.	5
1.4	The increments of $\ \mathbf{C}^{k+1} - \mathbf{C}^k\ _F$ of algorithm in [19] and the modified algorithm	13
1.5	Convergence behavior: the norms of the increments of the coefficient sequence \mathbf{C}^k generated by the K-SVD method and the proposed method.	14
1.6	Demonstration of the improvement of APG-L1 tracker (red) over BPR-L1 (blue) on tracking accuracy.	18
2.1	Convergence behavior of Algorithm 1 and Algorithm 2. (a) The ℓ_2 norm of the increments of the framelet coefficient vector at each iteration; and (b) the PSNR values of the intermediate results at each iteration when denoising the image "boat" with noise level $\sigma = 20$	39
2.2	Six test images	40
2.3	The dictionaries learned from the image "Barbara" with noise level $\sigma = 20$ using the K-SVD method and Algorithm 3. The atom size is 8×8	50
2.4	Test images.	51
2.5	Comparison of text removal. (a) image with overlapped texts; (b-e) correspond to the results from [11], two over-complete dictionary learning method with ℓ_1 norm sparsity penalty and MC penalty ([78]), and Algorithm 5.	51
2.6	Image inpainting with 50% random missing pixels. (a) Original image; (b) corrupted image; (c-e) the results from from two over-complete dictionary learning method with ℓ_1 norm sparsity penalty and MC penalty ([78]), and Algorithm 5.	51

3.1	Convergence behavior: the norm of the increments of the coefficient sequence \mathbf{C}^k generated by the K-SVD method and the proposed method. . . .	56
3.2	Test images.	69
3.3	The dictionaries learned from the image "Lena512" with noise level $\sigma = 30$ using the K-SVD method and Alg.6. The atom size is 8×8	70
3.4	Visual illustration of noisy images and denoised results	71
3.5	Overall running time of our method and the K-SVD de-noising method with comparable PSNR values.	72
4.1	The increments of the sequences generated by the methods.	90
4.2	The normalized histograms on the coherence matrices shown in Fig. 4.3. . . .	90
4.3	The mutual coherence matrices of the dictionaries learned from the YaleB face dataset using the K-SVD method and Alg.7. The i th-column and j th-row element in each matrix represents the mutual coherence between the i th and j -th atom.	90
5.1	Illustration of the L1 tracker on the sequence <i>lemming</i> using the model (5.3) and the L1 tracker using the proposed model (5.9). The first and the second row: results using (5.3) and using (5.9) respectively. Last row: the energy ratio $\ \mathbf{a}_I\ _2/\ \mathbf{a}\ _2$. The left graph is from (5.3) and the right is from (5.9). . . .	107
5.2	Demonstration of the improvement of APG-L1 tracker (red) over BPR-L1 (blue) on tracking accuracy.	112
5.3	The tracking error for each test sequence. The error is measured the same as in Table 5.1 and the legend as in Fig.5.4.	115
5.4	Tracking results of different algorithms for sequences <i>jump(a)</i> , <i>car(b)</i> , <i>singer(c)</i> , <i>woman(d)</i> , <i>pole(e)</i> , <i>sylv(f)</i> , <i>deer(g)</i> and <i>face(h)</i>	116

List of Tables

2.1	PSNR values of the denoised results	40
2.2	Complexity analysis for one iteration	46
2.3	Running time (seconds) breakdown on one iteration of the K-SVD method, approximated K-SVD method and the implementation of Algorithm 3 with patch size 8×8 and 16×16	49
2.4	Running time of the K-SVD method, approximated K-SVD method with 15 iterations and Algorithm 3 with 30 iterations.	49
2.5	PSNR values of the denoised results	52
3.1	PSNR values of the denoised results	72
3.2	Training time (seconds) on two face datasets.	72
3.3	Classification accuracies (%) on two face datasets.	73
4.1	Classification accuracies (%) on two face datasets and one object dataset.	89
5.1	The average tracking errors. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. The last row is the average error for each tracker over all the test sequences.	114

Chapter 1

Introduction

Recently, image restoration and recognition have become more and more important in image processing, visual tracking, object recognition, etc. Usually, image restoration aims at recovering a corrupted image by enhancing image features without introducing artifacts while image recognition is to identify and detect objects or features in an image or video sequence. The main difficulty for image restoration and recognition is to find the "good" representation for the input images. The so-called *sparse coding* method is now a well-established and powerful tool to provide good representation of input images, which represents given data by the linear combination of few elements of certain set. Such a set can be a *system* or a *dictionary* and elements in the set are called *atoms*. More specifically, let $\mathbf{D} = \{\mathbf{d}_k\}_{k=1}^m \subseteq \mathbb{R}^n$ denotes a set with m atoms, given an input signal $\mathbf{y} \in \mathbb{R}^n$, the *sparse approximation* over \mathbf{D} is to find a linear expansion $\mathbf{D}\mathbf{c} = \sum_{k=1}^m c_k \mathbf{d}_k$ using fewest atoms of \mathbf{D} that approximates \mathbf{y} within an error bound ε . Mathematically, the *sparse approximation* can be formulated as the following minimization problem:

$$\min_{\mathbf{c}} \|\mathbf{c}\|_0, \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2 \leq \varepsilon, \quad (1.1)$$

where $\|\mathbf{c}\|_0$ counts the number of nonzero elements in \mathbf{c} . The problem (1.1) is a challenging NP-hard problem and only sub-optimal solutions can be found in polynomial time. Most existing algorithms either use greedy algorithms to iteratively select locally optimal solu-

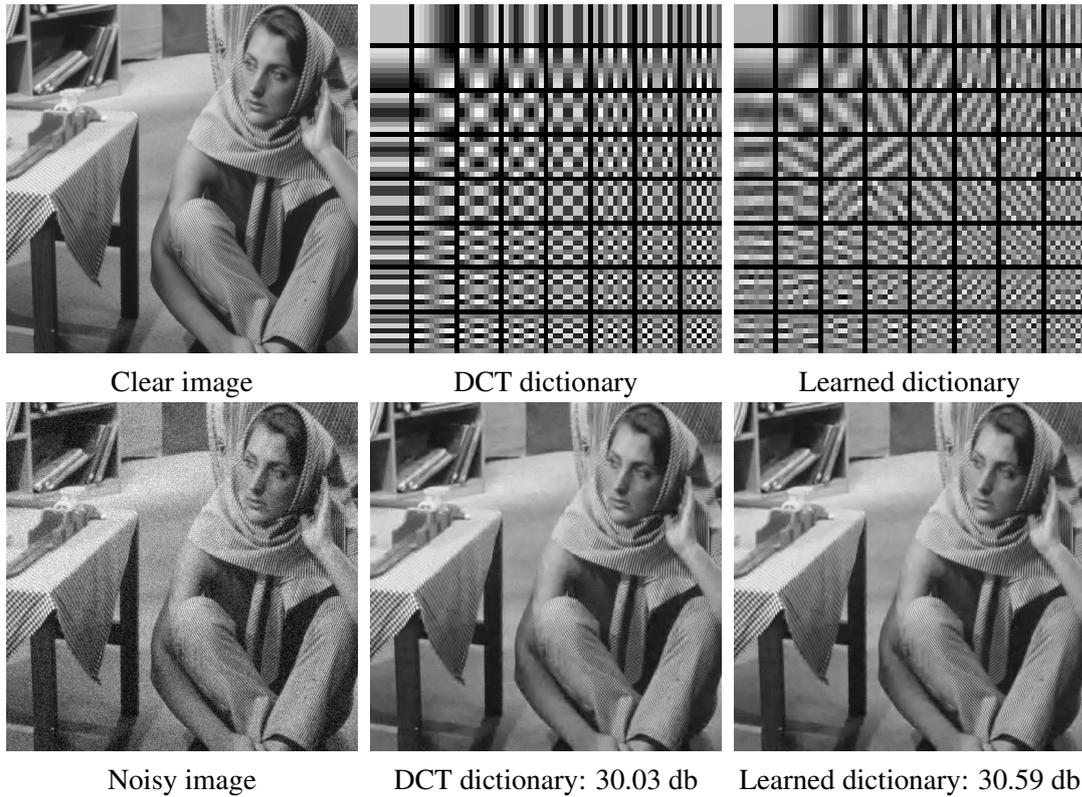


Figure 1.1: Pre-defined dictionary, learned dictionary and their denoising results

tions (e.g. orthogonal matching pursuit (OMP) [81]), or replace the non-convex ℓ_0 norm by its convex relaxation ℓ_1 norm (e.g. basis pursuit [24]). Besides the numerical difficulty of solving minimization (1.1), another fundamental problem for the sparse coding of \mathbf{y} is how to define the set \mathcal{D} such that the signal \mathbf{y} has an optimal sparse approximation.

The earliest work is focused on designing orthonormal bases, e.g. *discrete cosine transform* [69], *wavelets* [29, 59]. Owing to better performance in practice, over-complete systems have been more recognized in sparsity-based image processing problems. In particular, as a redundant extension of orthonormal bases, tight frames are now wide-spread in many applications as they have the same efficient and simple decomposition and reconstruction schemes as orthonormal bases. Many types of tight frames has been proposed for sparse image modeling including *shift-invariant wavelets* [25], *framelets* [30, 72], *curvelets* [20] and many others. These tight frames are optimized for signals with certain functional properties, which do not always hold true for natural images. Therefore, a more efficient approach to

sparsely approximate images of interest, the so-called *dictionary learning*, is to construct the certain set that adaptive to the inputs. See Figure 1.1 as an illustration. The basic idea is to construct a dictionary from training samples that maximizes the sparsity of the approximation. More concretely, given training samples $\mathbf{Y} := \{\mathbf{y}_i\}_{i=1}^p \subseteq \mathbb{R}^n$, the *dictionary learning* is formulated as the following optimization problem:

$$\min_{\mathbf{D} \in \mathcal{D}, \{\mathbf{c}_k\}_{k=1}^p} \sum_{k=1}^p \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \lambda \|\mathbf{c}_k\|_0, \quad \text{s.t.} \quad \|\mathbf{d}_k\|_2 = 1, k = 1, \dots, p, \quad (1.2)$$

where $\mathbf{C} = \{\mathbf{c}_k\}_{k=1}^p$ denotes the sparse coefficients of training set \mathbf{Y} , \mathbf{D} denotes the learned dictionary and the set \mathcal{D} denotes the desired property of the learned dictionary \mathbf{D} . The minimization (1.2) is an NP-hard problem where the challenge comes from two sources: the non-convex and non-smooth of ℓ_0 norm and the bi-linearity of the dictionary \mathbf{D} and the sparse coefficients \mathbf{C} .

In this thesis, based on different structures of the learned dictionary, we investigated the following three types of dictionary learning problems: orthogonal dictionary learning, redundant dictionary learning and incoherent dictionary learning. Using proximal methods, we rigorously proved the proposed numerical methods generate convergent sequences. The resulting numerical methods not only achieve comparable performance as existing sparse coding based methods in image restoration and recognition, but also significantly outperform other methods in terms of computational efficiency.

1.1 Background

Before moving to the main body of this thesis, we first introduce the background related to this thesis including the dictionary learning based image restoration and recognition, dictionary learning algorithms and proximal methods.

1.1.1 Dictionary learning for image restoration and recognition

In this section, we introduce some applications of dictionary learning problems including image restoration and recognition, which motivate our research.

Dictionary learning for image denoising. The first successful application of dictionary learning is image denoising when the observed image is corrupted by white Gaussian noise [1]. Let $\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_q\} \subseteq \mathbb{R}^n$ be the collection of patches from the observed image, the denoising procedure in [1] is as follows.

1. Generate the training data $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_p\} \subseteq \mathbb{R}^n$ where each column corresponding to a vectored image patch. There are two ways to generate the training data: one is to get image patches from a large natural image dataset, the other is to select image patches from the noisy image itself.
2. Learn the dictionary \mathbf{D} via solving the ℓ_0 norm related minimization (1.2). A detailed review of dictionary learning algorithms will be given in section 1.1.2.
3. Find the sparse approximation \mathbf{c}_k for each patch \mathbf{g}_k by solving the minimization:

$$\min_{\mathbf{c}_k} \|\mathbf{c}_k\|_0, \text{ s.t. } \|\mathbf{g}_k - \mathbf{D}\mathbf{c}_k\|_2^2 \leq \varepsilon,$$

where \mathbf{D} is the learned dictionary and ε is some pre-defined approximation accuracy.

4. Reconstruct the estimated image. First, we reconstruct the estimation of the image patches $\hat{\mathcal{G}} = \{\hat{\mathbf{g}}_1, \dots, \hat{\mathbf{g}}_q\}$ using the product of learned dictionary \mathbf{D} and sparse coefficients $\mathbf{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_q\}$. Then, we average all the image patches to obtain the restored image. That is, taking out all i -th pixel estimations $\hat{x}_1, \dots, \hat{x}_r$ from restored image patches $\{\hat{\mathbf{g}}_k\}_{k=1}^q$, the i -th pixel estimation is given by $\frac{1}{r} \sum_{j=1}^r \hat{x}_j$.

It is shown in figure 1.1 that the learned dictionary has some oriented atoms and obtains better denoising result. Besides, it is also worth to note that the choice of the training data and the averaging process are two key steps in the above denoising approach. It is reported in [1] that generating training data from the noisy image always obtain better denosing results

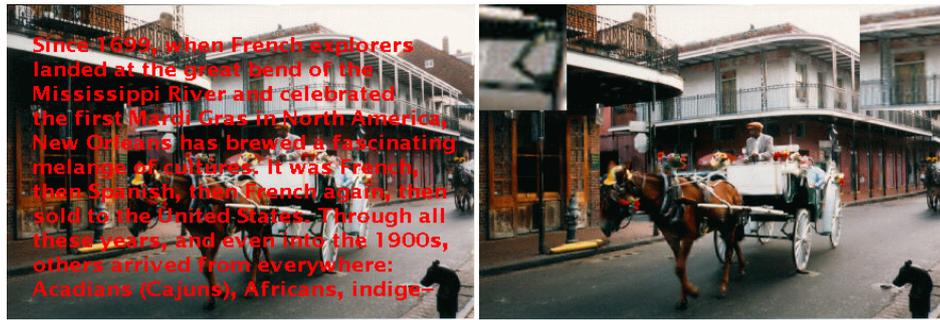


Figure 1.2: Image inpainting result.

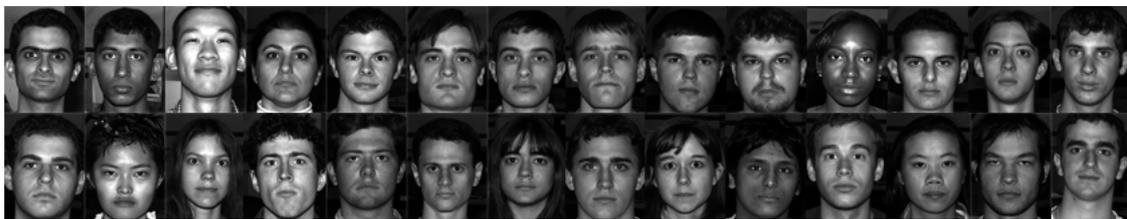


Figure 1.3: Some exemplar face images from Extend Yale face database B.

than generating training data from a general image dataset. The averaging process can be viewed as a reconstruction of a shift-invariant operator to image patches [19].

Dictionary learning for image inpainting. The dictionary learning for image denoising is generalized to solve image inpainting problem [54] where some pixels of the observed image are missing. The procedure for image inpainting proposed in [54] is the same as image denoising results except the step of learning dictionary step. In the dictionary learning stage, it attempts to solve the ℓ_1 norm related minimization:

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{C}} \sum_{k=1}^P \frac{1}{2} \|\mathbf{M}_k \odot (\mathbf{y}_k - \mathbf{D}\mathbf{c}_k)\|_2^2 + \lambda \|\mathbf{c}_k\|_1, \quad (1.3)$$

where \mathbf{M}_k is the mask of image patch \mathbf{y}_k . Compared to (1.8), the addition of the mask \mathbf{M}_k does not significantly change the dictionary learning problem, it also applies an alternating scheme to solve (1.3). Another approach for solving image inpainting problems is to learn a dictionary from a interpolated estimation of the image and refined it on the estimation of the in-painted image. See figure 1.2 is one image inpainting result from [7].

Dictionary learning for image recognition. Different from image restoration problems,

image recognition aims at identifying images of different categories. See some exemplar face images in figure 1.3 from Extended Yale face database B [38]. As a consequence, it requires discriminative representations of images as well as good approximations.

The dictionary learning method has been proven to have good reconstruction ability in image restoration, and been extended for image recognition problem [45, 53, 55, 65, 86, 87, 95] by imposing the discrimination of the learned dictionary. One approach to learn a discriminative dictionary is to construct a separate dictionary for each class [53]. Another more promising approach is to unify the dictionary learning and classifies training into a mixed re-constructive and discriminative formulation [45, 55, 65, 86, 87, 95]. As the second approach is used in this thesis, we introduce the main procedure in [45, 95] for image recognition as follows.

1. Select training samples $\{(\mathbf{y}_k, \mathbf{H}_k)\}_{k=1}^p$ from database, where \mathbf{y}_k is the training image or image feature and \mathbf{H}_k is a binary vector denotes the associated label. For instance, $\mathbf{H}_k = (0, \dots, 0, 1, 0, \dots, 0)$ denotes i -th category label if all the entries of \mathbf{H}_k are zero except the i -th entry. A usual way to generate training data is to randomly choose a fixed number of images from each class of database.
2. Learn a discriminative dictionary \mathbf{D} and a linear classifier \mathbf{W} simultaneously via solving non-convex minimization problems. In [95], it combines the discriminative ability and representative ability into a single minimization which is formulated as

$$\min_{\mathbf{D}, \mathbf{W}, \mathbf{C}} \sum_{k=1}^p \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \gamma \|\mathbf{H}_k - \mathbf{W}\mathbf{c}_k\|_2^2 + \beta \|\mathbf{W}\|_2^2, \text{ s.t. } \|\mathbf{c}_k\|_0 \leq s, \forall k, \quad (1.4)$$

In [45], an additional label consistent constraint is imposed in (1.4) which is formulated as

$$\min_{\mathbf{D}, \mathbf{W}, \mathbf{A}, \mathbf{C}} \sum_{k=1}^p \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \alpha \|\mathbf{Q} - \mathbf{A}\mathbf{C}\|_F^2 + \beta \|\mathbf{H} - \mathbf{W}\mathbf{C}\|_F^2, \text{ s.t. } \|\mathbf{c}_k\|_0 \leq s, \forall k, \quad (1.5)$$

where $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_N)$ are pre-defined some "discriminative" sparse code. For example, $\mathbf{q}_i = (0, \dots, 0, 1, 1, 0, \dots, 0)^\top$ is the discriminative sparse pattern for the i -th

training sample \mathbf{y}_i . Both problems (1.4) and (1.5) are solved by the K-SVD method [1] which will be introduced in section 1.1.2.

3. Normalize the atoms in learned dictionary \mathbf{D} and adjust the weights of linear classifier \mathbf{W} . In concrete, this step is done by

$$\mathbf{D} = \left(\frac{\mathbf{d}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{d}_2}{\|\mathbf{d}_2\|_2}, \dots, \frac{\mathbf{d}_m}{\|\mathbf{d}_m\|_2} \right);$$

$$\mathbf{W} = \left(\frac{\mathbf{w}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{d}_2\|_2}, \dots, \frac{\mathbf{w}_m}{\|\mathbf{d}_m\|_2} \right).$$

4. Identify the category of a new image \mathbf{y} . Compute the sparse coding \mathbf{c} of test image \mathbf{y} via solving the ℓ_0 norm minimization problem:

$$\min_{\mathbf{c}} \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 + \alpha \|\mathbf{c}\|_0$$

The category of the test image \mathbf{y} is determined by $j = \arg \max_i \{(\mathbf{W}\mathbf{c})_i, \forall i\}$.

The above dictionary learning based image recognition has demonstrated impressive recognition accuracy in face recognition and object recognition, as shown in [45, 95].

1.1.2 Dictionary learning algorithms

Based on the different sparsity prompting functions, dictionary learning algorithms can be divided into the following three categories: ℓ_0 norm regularization, ℓ_1 norm regularization and non-convex norm regularization. In the next, we will introduce numerical methods related to the above three kinds of regularizations.

ℓ_0 norm regularization. The ℓ_0 norm based dictionary learning can be formulated as solving the following minimization:

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{C}} \sum_{k=1}^p \frac{1}{2} \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2, \text{ s.t. } \|\mathbf{c}_k\|_0 \leq s, \forall k = 1, \dots, p, \quad (1.6)$$

where s is the sparsity level. The first approach for solving (1.6) is the so-called MOD (method of optimal directions) which is proposed by Engan et al. in [35]. It takes an

alternative minimization between the dictionary \mathbf{D} and sparse coefficients \mathbf{C} . In concrete, it updates $(\mathbf{C}^k, \mathbf{D}^k)$ via the following two steps.

- **Sparse approximation for \mathbf{C} .** Fix the dictionary \mathbf{D}^k , it directly solves the ℓ_0 norm related minimization problem:

$$\min_{\mathbf{c}} \frac{1}{2} \sum_{k=1}^p \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2, \text{ s.t. } \|\mathbf{c}\|_0 \leq s. \quad (1.7)$$

The minimization (1.7) is an NP-hard problem and only sub-optimal solutions are obtained via greedy algorithms including matching pursuit, orthogonal matching pursuit (OMP) [81] and modified version of OMP [82].

- **Dictionary update for \mathbf{D} .** Fix the sparse coefficients \mathbf{C}^{k+1} and the dictionary \mathbf{D} is updated via

$$\mathbf{D}^{k+1} = \Pi_{\mathcal{D}}(\mathbf{Y}\mathbf{C}(\mathbf{C}\mathbf{C}^T)^{-1}),$$

where Π is the orthogonal projection operator.

Another more promising approach for solving (1.6) is the K-SVD method [1]. It also takes an alternating approach between \mathbf{D} and \mathbf{C} . When the dictionary \mathbf{D} is fixed, it uses the OMP to update sparse coefficients. When sparse coefficients \mathbf{C} is fixed, it updates the dictionary \mathbf{D} column by column via the singular value decomposition (SVD). Despite its great success in practice, there is no available convergence analysis of the ℓ_0 norm based dictionary learning.

ℓ_1 norm regularization. The ℓ_1 norm regularization method [63] is first proposed by Olshausen et al. to approximate vectors that most of entries have small amplitude. In recent years, owing to the fundamental progress in compressed sensing, a replacement of non-convex ℓ_0 norm by convex ℓ_1 norm has been proposed to the ℓ_1 norm based dictionary learning [44, 54, 57, 58] which is formulated as:

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{C}} \sum_{k=1}^p \frac{1}{2} \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \lambda \|\mathbf{c}_k\|_1. \quad (1.8)$$

Although the minimization (1.8) is a bi-linear minimization problem, it is convex when

fixing \mathbf{D} or \mathbf{C} . A straightforward way to solve (1.8) is also takes an alternative way between \mathbf{D} and \mathbf{C} . In the sparse coding stage, a number of efficient numerical solvers have been applied to different applications such as homotopy method [33] in [57]; the accelerated gradient method [84] or fast iterative shrinkage thresholding algorithm [10] in [44]; the fixed point method [42] in [56]. In the dictionary update stage, In the stage of dictionary update, the atoms in the dictionary either are updated one by one or are simultaneously updated. One-by-one atom updating is implemented in [44, 57] as it has the closed form solution. The projection gradient method is used in [56] to update the whole dictionary together. Recently, a convergent algorithm for solving (1.8) is proposed based on the proximal methods.

Non-convex norm regularization. As shown in [43, 94], the ℓ_1 norm penalty tends to have biased estimation for large coefficients and sometimes results in over-penalization. Thus, several non-convex relaxations of ℓ_0 norm are proposed for better accuracy in sparse coding. For example, the non-convex minimax concave (MC) penalty is used in [78] as the replacement of ℓ_0 norm and gives a convergent algorithm for sparse coding. For other non-convex relaxations (e.g. smoothly clipped absolute deviation [43], log penalty[37]), the proximal-based algorithms have been proposed in [40, 67, 79] to solve the minimization problem with these non-convex regularization terms. The convergence analysis of these non-convex relaxation methods is only limited to subsequence convergence. It is not clear whether they are globally convergent or not.

1.1.3 Proximal methods

Nowadays, proximal methods are widely applied for solving non-smooth, constrained minimization problems. In this section, we briefly review these methods closely related this thesis (see [64] for a detailed review). Let t be a positive constant and $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ be a proper and lower semi-continuous function bounded below, the *proximal operator* $\text{Prox}_t^f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ of f is defined as

$$\text{Prox}_t^f(\mathbf{x}) := \arg \min_{\mathbf{u}} f(\mathbf{u}) + \frac{1}{2t} \|\mathbf{u} - \mathbf{x}\|_2^2. \quad (1.9)$$

It is worth to note that the range of the proximal operator (1.9) is nonempty and compact for any $t \in (0, +\infty)$ [15] without the convexity assumption of f . In the following, we review proximal methods for solving both convex minimization problems and non-convex minimization problems.

Proximal methods for convex minimization problems. Consider the minimization

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) \quad (1.10)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ are closed proper convex and f is differentiable.

The *proximal gradient method* updates \mathbf{x}^{k+1} via

$$\mathbf{x}^{k+1} := \text{Prox}_{\lambda^k}^g(\mathbf{x}^k - \lambda^k \nabla f(\mathbf{x}^k)), \quad (1.11)$$

where $\lambda^k > 0$ is a step size. If ∇f is Lipschitz continuous with constant L , then \mathbf{x}^k generated by (1.11) converges to the global minimizer with rate $O(1/k)$ when $\lambda^k = \lambda \in (0, 1/L]$. It has been further proved in [27] that the scheme (1.11) converges if $\lambda^k \in (0, 2/L)$.

The accelerated version of proximal gradient method, the so-called *accelerated proximal gradient* method, is proposed by introducing an extrapolation step. It updates \mathbf{x}^{k+1} via

$$\begin{aligned} \mathbf{y}^{k+1} &:= \mathbf{x}^k + \omega^k(\mathbf{x}^k - \mathbf{x}^{k-1}), \\ \mathbf{x}^{k+1} &:= \text{Prox}_{\lambda^k}^g(\mathbf{y}^{k+1} - \lambda^k \nabla f(\mathbf{y}^{k+1})), \end{aligned} \quad (1.12)$$

where $\omega^k \in [0, 1)$ is the extrapolation parameter and λ^k is the step size. It reduces to the *proximal gradient* method if $\omega^k = 0$. These parameters must be chosen carefully to accelerate the convergence. Typically, in [10, 84], it takes

$$\omega^k := (t_{k-1} - 1)/t_k, \quad (1.13)$$

where $t_k := \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}$ and $t_0 = t_{-1} = 1$. When ∇f is Lipschitz continuous with constant L , the objective value at \mathbf{x}^k which is generated by (1.12) converges with rate $O(1/k^2)$ which is optimal among all first order methods if $\lambda^k = \lambda \in (0, 1/L]$ and ω^k is chosen as (1.13).

Proximal methods (1.11) and (1.12) can be easily applied to solve the following ℓ_1 regularized minimization

$$\min_{\mathbf{c}} \frac{1}{2} \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (1.14)$$

by setting $f(\mathbf{c}) = \frac{1}{2} \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2$ and $g(\mathbf{c}) = \lambda \|\mathbf{c}\|_1$. The proximal operator in (1.11) and (1.12) have closed form solutions which are given by the following lemma:

Lemma 1.1.1. *The minimization $\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1$ has unique minimizer*

$$\mathbf{x}^* = \text{sign}(\mathbf{y}) \odot \max(|\mathbf{y}| - \lambda, 0),$$

where \odot denotes the Hadamard product.

Proximal methods for non-convex minimization problems. In recent years, proximal methods [2, 3, 15] have been proposed to solve the non-convex minimization problem of the form:

$$\min_{\mathbf{x}, \mathbf{y}} H(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}) + Q(\mathbf{x}, \mathbf{y}) + G(\mathbf{y}), \quad (1.15)$$

where F, G are lower semi-continuous and Q is Lipschitz continuous with constant L .

The *proximal alternating* method [2] updates $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1})$ via

$$\begin{aligned} \mathbf{x}^{k+1} &\in \arg \min_{\mathbf{x}} F(\mathbf{x}) + Q(\mathbf{x}, \mathbf{y}^k) + G(\mathbf{y}^k) + \frac{\mu^k}{2} \|\mathbf{x} - \mathbf{x}^k\|_F^2; \\ \mathbf{y}^{k+1} &\in \arg \min_{\mathbf{y}} F(\mathbf{x}^{k+1}) + Q(\mathbf{x}^{k+1}, \mathbf{y}) + G(\mathbf{y}) + \frac{\lambda^k}{2} \|\mathbf{y} - \mathbf{y}^k\|_F^2, \end{aligned} \quad (1.16)$$

It has been proved that the sequence generated by the scheme (1.16) converges to the stationary point of (1.15) if $(\lambda^k, \mu^k) = (\lambda, \mu) \in \mathbb{R}_+^2$ and $H(\mathbf{x}, \mathbf{y})$ is a KL-function [14].

In general, the scheme (1.16) requires solving the non-smooth and non-convex minimization problems in each step which often has no closed form solutions. Therefore, the *proximal linearized alternating* method [15] has been proposed such that each subproblem has a closed form solution. Instead of solving the subproblems as (1.16), the alternating proximal linearized algorithm replaces the smooth term Q in (1.16) by its first order linear approxi-

mation:

$$\begin{cases} \mathbf{x}^{k+1} \in \arg \min_{\mathbf{x}} F(\mathbf{x}) + \hat{Q}_{(\mathbf{x}^k, \mathbf{y}^k)}(\mathbf{x}) + G(\mathbf{y}^k) + \frac{\mu^k}{2} \|\mathbf{x} - \mathbf{x}^k\|_F^2; \\ \mathbf{y}^{k+1} \in \arg \min_{\mathbf{y}} F(\mathbf{x}^{k+1}) + \hat{Q}_{(\mathbf{x}^{k+1}, \mathbf{y}^k)}(\mathbf{y}) + G(\mathbf{y}) + \frac{\lambda^k}{2} \|\mathbf{y} - \mathbf{y}^k\|_F^2. \end{cases} \quad (1.17)$$

where

$$\begin{aligned} \hat{Q}_{(\mathbf{x}^k, \mathbf{y}^k)}(\mathbf{x}) &= Q(\mathbf{x}^k, \mathbf{y}^k) + \langle \nabla_{\mathbf{x}} Q(\mathbf{x}^k, \mathbf{y}^k), \mathbf{x} - \mathbf{x}^k \rangle, \\ \hat{Q}_{(\mathbf{x}^k, \mathbf{y}^k)}(\mathbf{y}) &= Q(\mathbf{x}^k, \mathbf{y}^k) + \langle \nabla_{\mathbf{y}} Q(\mathbf{x}^k, \mathbf{y}^k), \mathbf{y} - \mathbf{y}^k \rangle, \end{aligned}$$

and μ^k, λ^k are carefully chosen step sizes. For instance, if $(\mu^k, \lambda^k) = (\mu, \lambda) \in \mathbb{R}_+^2$ and $\mu, \lambda > L$, it has been proved in [15] that the sequence $(\mathbf{x}^k, \mathbf{y}^k)$ generated by the scheme (1.17) converges to the stationary point of (1.15) when $H(\mathbf{x}, \mathbf{y})$ is a KL function.

1.2 Motivations and contributions of the dissertation

This thesis brings two main contributions to sparse coding based image restoration and recognition problems. Firstly, we systemically investigated ℓ_0 norm based dictionary learning problems for image restoration and recognition by imposing several structures on the learned dictionary. Secondly, we developed some proximal methods for solving the resulting non-convex minimization problems. Compared to the existing ℓ_0 norm based dictionary learning methods, our proposed methods have the following two main advantages: one is its theoretical guarantee of the generated sequence, which is the first available theoretical convergence analysis for ℓ_0 norm based dictionary learning problem; the other is its great gain in computational efficiency which might make our methods more scalable for big data. Additionally, based on the accelerated proximal gradient method, we developed a real time visual tracker that uses the sparse approximation of the target. In the next, we present these results with more details.

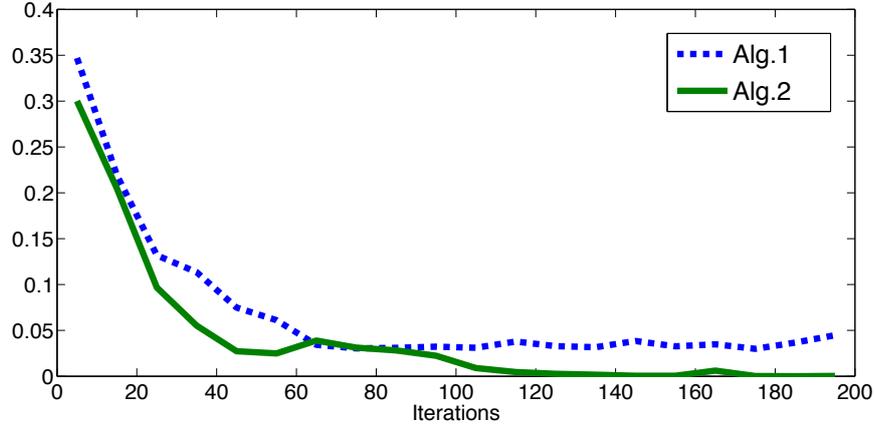


Figure 1.4: The increments of $\|\mathbf{C}^{k+1} - \mathbf{C}^k\|_F$ of algorithm in [19] and the modified algorithm

1.2.1 Data-driven tight frame construction

Recently, Cai et al. [19] proposed a variational model to learn a tight frame system that is adaptive to the input image in terms of sparse approximation. The tight frame construction scheme proposed in [19] requires solving an ℓ_0 norm related non-convex minimization problem:

$$\min_{\mathbf{D} \in \mathbb{R}^{m \times m}, \mathbf{C} \in \mathbb{R}^{m \times n}} \|\mathbf{C} - \mathbf{D}^\top \mathbf{Y}\|_F^2 + \lambda_0^2 \|\mathbf{C}\|_0, \quad \text{s.t.} \quad \mathbf{D}^\top \mathbf{D} = \mathbf{I}_m, \quad (1.18)$$

where \mathbf{D} contains framelet filters and \mathbf{C} contains the canonical frame coefficients. An alternating iteration is proposed in [19] for solving (1.18), which is very fast as both sub-problems in each iteration have closed-form solutions. It is shown that, with comparable performance in image denoising, the proposed adaptive tight frame construction runs much faster than other generic dictionary learning methods (e.g. the K-SVD method [1]). However, Cai et al. [19] did not provide any convergence analysis of the proposed method.

As a sequel to [19], chapter 2 provides the convergence analysis of the alternating iterative method proposed in [19] for solving (1.18). In that chapter, we showed that the algorithm provided by [19] has sub-sequence convergence property. In other words, we showed that there exists at least one convergent sub-sequence of the sequence generated by the algorithm in [19] and any convergent sub-sequence converges a stationary point of

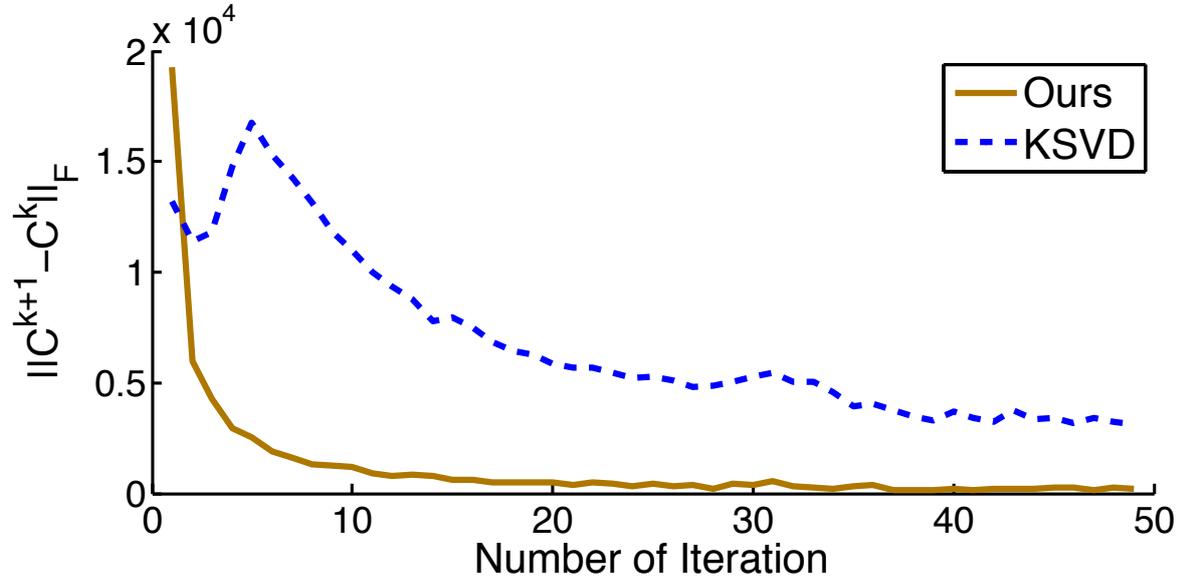


Figure 1.5: Convergence behavior: the norms of the increments of the coefficient sequence \mathbf{C}^k generated by the K-SVD method and the proposed method.

(1.18). Moreover, we empirically observed that the sequence generated by the algorithm proposed in [19] itself is not convergent. See figure 1.4 as an illustration. Motivated by the theoretical interest, we modified the algorithm proposed in [19] by adding a proximal term in the iteration scheme, and then showed that the modified algorithm has sequence convergence. In other words, the sequence generated by the modified method converges to a stationary point of (1.18). Moreover, we extended the data-driven tight frame construction when some of filters are fixed which is formulated as the following minimization:

$$\min_{\mathbf{D} \in \mathbb{R}^{m \times r}, \mathbf{C} \in \mathbb{R}^{m \times p}} \|\mathbf{Y} - [\mathbf{A}, \mathbf{D}]\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_0, \text{ s.t. } [\mathbf{A}, \mathbf{D}]^\top [\mathbf{A}, \mathbf{D}] = \mathbf{I}_m, \quad (1.19)$$

where $r \leq m$ and $\mathbf{A} \in \mathbb{R}^{m \times (m-r)}$ is the predefined filters. The extension of (1.19) can further accelerate the dictionary learning process as reported in experiments.

1.2.2 Redundant dictionary learning

Compared to the orthogonal dictionary learning (1.18), a more general approach is to learn a redundant dictionary that is maximizing the sparse degree of the approximation. Mathematically, the redundant dictionary learning is formulated as the following non-convex minimization problem:

$$\min_{\mathbf{D} \in \mathbb{R}^{m \times n}, \mathbf{C} \in \mathbb{R}^{n \times p}} \|\mathbf{Y} - \mathbf{DC}\|_F^2 + \lambda \|\mathbf{C}\|_0, \text{ s.t. } \|\mathbf{d}_k\|_2 = 1, \forall k. \quad (1.20)$$

where $n > m$. The non-convexity of minimization (1.20) comes from two sources: the sparsity-prompting functional ℓ_0 norm and the bi-linearity between the dictionary \mathbf{D} and the codes $\{\mathbf{c}_k\}_{k=1}^p$. Most existing approaches (e.g. [1, 44, 56, 57]) take an alternating iteration between two modules: sparse approximation for updating $\{\mathbf{c}_k\}_{k=1}^p$ and dictionary learning for updating dictionary \mathbf{D} . Despite the success of these alternating iterative methods in practice, none of them established the global convergence property, i.e., the whole sequence generated by the method converges to a stationary point of (1.20). These schemes can only guarantee that the functional values are decreasing over the iterations, and thus there exists a convergent sub-sequence as the sequence is always bounded. Indeed, the sequence generated by the popular K-SVD method [1] is not convergent as its increments do not decrease to zero. See figure. 1.5 for an illustration. The global convergence property is not only of great theoretical importance, but also likely to be more efficient in practical computation as many intermediate results are useless for a method without global convergence property.

In chapter 3, we proposed an alternating proximal linearized method for solving (1.20). The main contribution of the proposed algorithm lies in its theoretical contribution to the open question regarding the convergence property of ℓ_0 norm based dictionary learning methods. In that chapter, we showed that the whole sequence generated by the proposed method converges to a stationary point of (1.20). Moreover, we also showed that the convergence rate of the proposed algorithm is at least sub-linear. To the best of our knowledge, this is the first algorithm with global convergence for solving ℓ_0 norm based dictionary learning problems. The proposed method can also be used to solve other variations of (1.20) with

small modifications, e.g. the ones used in discriminative K-SVD based recognition methods [45, 95]. Compared to many existing methods including the K-SVD method, the proposed method also has its advantage on computational efficiency. The experiments showed that the implementation of the proposed algorithm has comparable performance to the K-SVD method in two applications: image de-noising and face recognition, but is noticeably faster.

1.2.3 Incoherent dictionary learning

In chapter 4, we considered the problem of sparse coding that explicitly imposes additional regularization on the mutual coherence of the dictionary, which can be formulated as the following minimization problem:

$$\begin{aligned} \min_{\mathbf{D}, \{\mathbf{c}_i\}_{i=1}^p} \quad & \sum_i \left(\frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_0 \right) + \frac{\alpha}{2} \|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2, \\ \text{s.t.} \quad & \|\mathbf{d}_j\|_2 = 1, \quad 1 \leq j \leq m. \end{aligned} \tag{1.21}$$

where \mathbf{D} is the learned dictionary, and \mathbf{C} is the sparse coefficients. Our motivations and contributions are presented in the following.

The need of an incoherent dictionary for sparse coding. Once a dictionary is learned, the sparse code for each input is then computed via some pursuit methods, e.g. *orthogonal matching pursuit* [81], *basis pursuit* [24]. The success of these methods for finding the optimal sparse code depends on the incoherence property of the dictionary. In [81], Tropp showed that the OMP can recover the exact support of the coefficients whenever mutual coherence μ is less than $1/(2S - 1)$ where S is the number of nonzero entries of the correct coefficients. It is further proved in [75] that the similar requirement on the mutual coherence is also needed for ensuring the correctness of the thresholding-based sparse coding algorithms. In practice, it is also observed that a dictionary with high mutual coherence will impact the performance of sparse coding based methods; see e.g [13, 68, 87].

The need of a variational model that explicitly regularizes mutual coherence. In a quick glance, the widely used K-SVD method [1] for sparse coding considered a variational model which has no explicit functional on minimizing the mutual coherence of the result,

i.e., it considered a special case of (1.21) with $\alpha = 0$. However, the implementation of the K-SVD method implicitly controlled the mutual coherence of the dictionary by discarding the "bad" atom which is highly correlated to the ones already in the dictionary. Such an ad-hoc approach certainly is not optimal for lowering the overall mutual coherence of the dictionary. In practice, the K-SVD method may still give a dictionary that contains highly correlated atoms, which will lead to poor performance in sparse approximation, see [28] for more details.

The need of a convergent algorithm. The minimization problem (1.21) is a challenging non-convex problem. Most existing methods that used the model (1.21) or its extensions, e.g. [45, 56, 95], simply call some generic non-linear optimization solvers such as the *projected gradient* method. Such a scheme is slow and not stable in practice. Furthermore, all these methods at most can be proved that the functional value is decreasing at each iteration. The sequence itself may not be convergent. From the theoretical perspective, a non-convergent algorithm certainly is not satisfactory. From the application perspective, the divergence of the algorithm also leads to troublesome issues such as when to stop the numerical solver, which often requires manual tune-up.

In chapter 4, we proposed a hybrid alternating proximal scheme for solving (1.21). Compared to the K-SVD method that controls the mutual coherence of the dictionary in an ad-hoc manner, the proposed method is optimized for learning an incoherent dictionary for sparse coding. Compared to the generic numerical scheme for solving (1.21) adopted in the existing applications, the convergence property of the proposed method is rigorously established in the chapter. We showed that the whole sequence generated by the proposed method converges to a stationary point. As a comparison, only sub-sequence convergence can be proved for existing numerical methods. The whole sequence convergence of an iteration scheme is not only of theoretical interest, but also important for applications as the number of iterations does not need to be empirically chosen to keep the output stable.

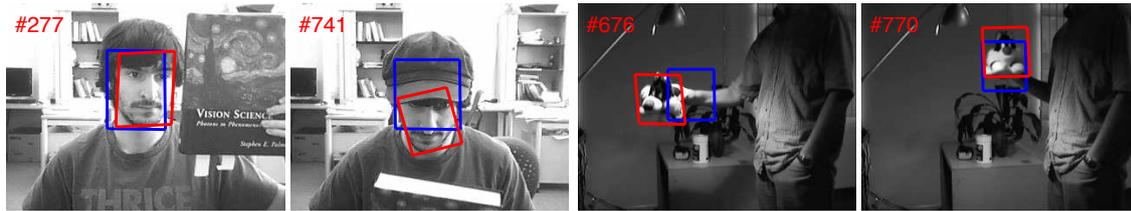


Figure 1.6: Demonstration of the improvement of APG-L1 tracker (red) over BPR-L1 (blue) on tracking accuracy.

1.2.4 L1 visual tracker

In chapter 5, the sparse representation has been applied to visual tracker by modeling the target appearance using sparse approximation over a template set, which leads to the so-called L1 tracker as it needs to solve a ℓ_1 norm related minimization problem for many times. While these L1 trackers showed impressive tracking accuracies, they are very computationally demanding and the speed bottleneck is the solver to ℓ_1 norm related minimizations. In Appendix, we developed an L1 tracker that not only runs in real times but also enjoys better robustness than other L1 trackers. In our proposed L1 tracker, a new ℓ_1 norm related minimization model is proposed to improve the accuracy by adding an ℓ_2 norm regularization on the coefficients associated with trivial templates. See figure 1.6 as an example. Moreover, based on the accelerated proximal gradient method, a fast numerical solver is developed to solve the resulting ℓ_1 norm related minimization problem.

1.3 Notation

The following definitions and notations are used in this thesis for discussion. For example, we denote $\mathbf{Y} \in \mathbb{R}^{m \times n}$ be a $m \times n$ matrix, \mathbf{Y}_{ij} be the entry at row i and column j of \mathbf{Y} , \mathbf{y}_j be the j -th column of the matrix \mathbf{Y} and \mathbf{y}_i be the i -th element of the vector \mathbf{y} .

Given a vector \mathbf{y} , for $q \geq 2$, its ℓ_q -norm and ℓ_0 pseudo-norm are defined as

$$\|\mathbf{y}\|_q = \left(\sum_j |\mathbf{y}_j|^q \right)^{1/q}, \quad \|\mathbf{y}\|_0 = \#\{j, \mathbf{y}_j \neq 0\}.$$

Given a matrix \mathbf{Y} , its Frobenius norm $\|\mathbf{Y}\|_F$, ℓ_0 pseudo-norm $\|\mathbf{Y}\|_0$ and uniform norm

$\|\mathbf{Y}\|_\infty$ are defined as

$$\|\mathbf{Y}\|_F = \left(\sum_{i,j} |\mathbf{Y}_{ij}|^2\right)^{1/2}, \quad \|\mathbf{Y}\|_0 = \#\{(i,j) : \mathbf{Y}_{ij} \neq 0\}, \quad \|\mathbf{Y}\|_\infty = \max_{i,j} |\mathbf{Y}_{ij}|.$$

Given a set \mathcal{X} , the indicator function $I_{\mathcal{X}}(\cdot)$ is defined as

$$I_{\mathcal{X}}(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \in \mathcal{X}; \\ +\infty, & \text{if } \mathbf{x} \notin \mathcal{X}. \end{cases}$$

Given $\lambda > 0$ and matrix \mathbf{Y} , the *hard thresholding* operator $T_\lambda(\mathbf{Y})$ is defined as

$$[T_\lambda(\mathbf{Y})]_{ij} = \begin{cases} \mathbf{Y}_{ij}, & \text{if } |\mathbf{Y}_{ij}| > \lambda; \\ 0, & \text{if } |\mathbf{Y}_{ij}| < \lambda; \\ \{0, \mathbf{Y}_{ij}\}, & \text{if } |\mathbf{Y}_{ij}| = \lambda. \end{cases}$$

Chapter 2

Data-driven tight frame construction for image restoration

2.1 Introduction

It is now well established that sparse modelling is a very powerful tool for many image recovery tasks, which models an image as the linear combination of only a small number of elements of some system. Such a system can be either a basis or an over-complete system. When using the sparsity prior of images to regularize image recovery, the performance largely depends on how effective images of interest can be sparsely approximated under the given system. Therefore, a fundamental question in sparsity-based image regularization is how to define a system such that the target image has an optimal sparse approximation. Earlier work on sparse modelling focuses on the design of orthonormal bases, such as *discrete cosine transform* [70], *wavelets* [29, 59]. Owing to their better performance in practice, over-complete systems have been more recognized in sparsity-based image recovery methods. In particular, as a redundant extension of orthonormal bases, tight frames are now wide-spread in many applications as they have the same efficient and simple decomposition and reconstruction schemes as orthonormal bases. Many types of tight frames has been proposed for sparse image modelling including *shift-invariant wavelets* [25], *framelets* [30, 72], *curvelets* [20] and many others. These tight frames are optimized for the signals with certain

functional properties, which do not always hold true for natural images. As a consequence, a more effective approach to sparsely approximate images of interest is to construct tight frames that are adaptive to the inputs.

In recent years, the concept of data-driven systems has been exploited to construct adaptive systems for sparsity-based modelling (see e.g. [1, 19, 49, 54]). The basic idea is to construct the system that is adaptive to the input so as to obtain a better sparse approximation than the pre-defined ones. Most sparsity-based dictionary learning methods ([1, 49, 54]) treat the input image as the collection of small image patches, and then construct an over-complete dictionary for sparsely approximating these image patches. Despite the impressive performance in various image restoration tasks, the minimization problems proposed by these methods are very challenging to solve. As a result, the numerical methods proposed in past for these models not only lack rigorous analysis on their convergence and stability, but also are very computational demanding.

Recently, Cai et al. [19] proposed a variational model to learn a tight frame system that is adaptive to the input image in terms of sparse approximation. Different from the existing over-complete dictionary learning methods, the adaptive systems constructed in [19] are tight frames that have *perfect reconstruction property*, a property ensures that any input can be perfectly reconstructed by its canonical coefficients in a simple manner. The tight frame property of the system constructed in [19] not only is attractive to many image processing tasks, but also leads to very efficient construction scheme. Indeed, by considering a special class of tight frames, the construction scheme proposed in [19] only requires solving an ℓ_0 norm related non-convex minimization problem:

$$\min_{\mathbf{D} \in \mathbb{R}^{m \times m}, \mathbf{C} \in \mathbb{R}^{m \times n}} \|\mathbf{C} - \mathbf{D}^\top \mathbf{Y}\|_F^2 + \lambda_0^2 \|\mathbf{C}\|_0, \quad \text{s.t.} \quad \mathbf{D}^\top \mathbf{D} = m^{-1} \mathbf{I}_m, \quad (2.1)$$

where \mathbf{D} contains framelet filters and \mathbf{C} contains the canonical frame coefficients. An alternating iteration is proposed in [19] for solving (2.1), which is very fast as both sub-problems in each iteration have closed-form solutions. It is shown that, with comparable performance in image denoising, the proposed adaptive tight frame construction runs much faster than

other generic dictionary learning methods (e.g. the K-SVD method [1]). However, Cai et al. [19] did provide any convergence analysis of the proposed method.

As a sequel to [19], this chapter provides the convergence analysis of the alternating iterative method proposed in [19] for solving (2.1). In this chapter, we showed that the algorithm provided by [19] has sub-sequence convergence property. In other words, we showed that there exists at least one convergent sub-sequence of the sequence generated by the algorithm [19] and any convergent sub-sequence converges a stationary point of (2.1). Moreover, we empirically observed that the sequence generated by the algorithm proposed in [19] itself is not convergent. Motivated by the theoretical interest, we modified the algorithm proposed in [19] by adding a proximal term in the iteration scheme, and then showed that the modified algorithm has sequence convergence. In other words, the sequence generated by the modified method converges to a stationary point of (2.1).

2.2 Brief review on data-driven tight frame construction and related works

In this section, we gave a brief review on tight frames, data-driven tight frames proposed in [19] and some most related works. Interesting readers are referred to [18, 76] for more details.

2.2.1 Tight frames and data-driven tight frames

For a Hilbert space \mathcal{H} , a sequence $\{x_n\} \subset \mathcal{H}$ is a *tight frame* for \mathcal{H} if

$$\|x\|^2 = \sum_n |\langle x, x_n \rangle|^2, \quad \text{for any } x \in \mathcal{H},$$

or equivalently, $x = \sum_n \langle x, x_n \rangle x_n, x \in \mathcal{H}$. The sequence $\{\langle x, x_n \rangle\}$ is called the canonical frame coefficient sequence. A tight frame $\{x_n\}$ is an orthonormal basis for \mathcal{H} if and only if $\|x_n\| = 1$ for all x_n . A tight frame has two associated operators: the *analysis operator* \mathbf{W}

defined by

$$\mathbf{W} : x \in \mathcal{H} \longrightarrow \{\langle x, x_n \rangle\} \in \ell_2(\mathbb{N})$$

and its adjoint operator \mathbf{W}^\top (often called the *synthesis operator*):

$$\mathbf{W}^\top : \{a_n\} \in \ell_2(\mathbb{N}) \longrightarrow \sum_n a_n x_n \in \mathcal{H}.$$

Then, the sequence $\{x_n\} \subset \mathcal{H}$ is a tight frame if and only if $\mathbf{W}^\top \mathbf{W} = I$, where I denotes the identity operator of \mathcal{H} . The tight frames considered in [19] are single-level un-decimal discrete wavelet systems generated by all integer shifts of a set of filters $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$. For any filter $\mathbf{a} \in \ell_2(\mathbb{Z})$, let $\mathcal{S}_{\mathbf{a}} : \ell_2(\mathbb{Z}) \rightarrow \ell_2(\mathbb{Z})$ denote its associated convolution operator defined by

$$[\mathcal{S}_{\mathbf{a}}(\mathbf{v})](n) := [\mathbf{a} \star \mathbf{v}](n) = \sum_{k \in \mathbb{Z}^2} \mathbf{a}(n-k) \mathbf{v}(k), \quad \forall \mathbf{v} \in \ell_2(\mathbb{Z}). \quad (2.2)$$

Then, for a given set of framelet filters, we define its associated analysis operator \mathbf{W} by

$$\mathbf{W} = [\mathcal{S}_{\mathbf{a}_1}^\top(\cdot), \mathcal{S}_{\mathbf{a}_2}^\top(\cdot), \dots, \mathcal{S}_{\mathbf{a}_m}^\top(\cdot)]^\top. \quad (2.3)$$

The rows of \mathbf{W} form a tight frame for $\ell_2(\mathbb{Z})$ if and only if $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, and the corresponding synthesis operator is the transpose of \mathbf{W} , denoted by \mathbf{W}^\top .

The data-driven tight frame construction proposed in [19] constructs the set of framelet filters $\{\mathbf{a}_j\}_{j=1}^m$ via solving the following problem:

$$\min_{\mathbf{v}, \{\mathbf{a}_i\}_{i=1}^m} \|\mathbf{v} - \mathbf{W}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m) \mathbf{g}\|_F^2 + \lambda_0^2 \|\mathbf{v}\|_0, \quad \text{s.t.} \quad \mathbf{W}^\top \mathbf{W} = \mathbf{I}. \quad (2.4)$$

where \mathbf{g} denotes the input signal, $\{\mathbf{a}_j\}_{j=1}^m$ denotes the set of framelet filters of the adaptive tight frame, and \mathbf{v} denotes the canonical coefficient vector of \mathbf{g} . Here and throughout this chapter, $\|\mathbf{v}\|_0$ stands for the number of non-zero elements of \mathbf{v} and $\|\cdot\|_F$ denotes the Frobenius norm.

2.2.2 Data-driven tight frame construction scheme

For general framelet filters, the minimization problem (2.4) is very challenging to solve. Therefore, a special class of framelet filters are considered in [19], which is composed by m^2 2D real-valued framelet filters $\{\mathbf{a}_j\}_{j=1}^{m^2} \subset \mathbb{R}^{m \times m}$. Let D denote the associated filter matrix defined by

$$A = [\vec{\mathbf{a}}_1, \vec{\mathbf{a}}_2, \dots, \vec{\mathbf{a}}_{m^2}],$$

where $\vec{\mathbf{a}}_j$ denotes the vector form of \mathbf{a}_j by concatenating all columns of \mathbf{a}_j to a column vector. It is shown in [19, Proposition 3] that the rows of \mathbf{W} defined by $\{\mathbf{a}_j\}_{j=1}^{m^2} \subset \mathbb{R}^{m \times m}$ form a tight frame for $\ell^2(\mathbb{Z})$, provided that $A^\top A = \frac{1}{m} I_{m^2}$. Thus, the minimization problem (2.4) for general tight frame construction is simplified to the following one:

$$\min_{\mathbf{v}, \{\mathbf{a}_i\}_{i=1}^{m^2}} \|\mathbf{v} - \mathbf{W}(A)\mathbf{g}\|_F^2 + \lambda_0^2 \|\mathbf{v}\|_0, \quad \text{s.t.} \quad A^\top A = \frac{1}{m} I_{m^2}. \quad (2.5)$$

The problem (2.5) can be re-formulated in terms of image patches as follows. Let $\{\vec{\mathbf{g}}_\ell\}_{\ell=1}^L \subset \mathbb{R}^{m^2}$ denotes the set of all image patches of size $m \times m$ densely sampled from the image \mathbf{g} . For each patch vector $\vec{\mathbf{g}}_\ell$, let $\vec{\mathbf{v}}_n = A^\top \vec{\mathbf{g}}_\ell \in \mathbb{R}^{m^2}$ denotes the vector generated by the inner product between $\vec{\mathbf{g}}_n$ and all m^2 framelet filters $\{\vec{\mathbf{a}}_j\}_{j=1}^{m^2}$. Define three matrices as follows,

$$\begin{cases} \mathbf{Y} := \frac{1}{\sqrt{m}} [\vec{\mathbf{g}}_1, \vec{\mathbf{g}}_2, \dots, \vec{\mathbf{g}}_L] \in \mathbb{R}^{m^2 \times L}; \\ \mathbf{D} := \sqrt{m} A = \sqrt{m} [\vec{\mathbf{a}}_1, \vec{\mathbf{a}}_2, \dots, \vec{\mathbf{a}}_{m^2}] \in \mathbb{R}^{m^2 \times m^2}; \\ \mathbf{C} := [\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2, \dots, \vec{\mathbf{v}}_{m^2}] \in \mathbb{R}^{m^2 \times L}. \end{cases} \quad (2.6)$$

Then, it is shown in [19] that the minimization (2.5) is equivalent to

$$\min_{\mathbf{D} \in \mathbb{R}^{m^2 \times m^2}, \mathbf{C} \in \mathbb{R}^{m^2 \times L}} \|\mathbf{C} - \mathbf{D}^\top \mathbf{Y}\|_F^2 + \lambda^2 \|\mathbf{C}\|_0, \quad \text{s.t.} \quad \mathbf{D}^\top \mathbf{D} = I_{m^2 \times m^2}, \quad (2.7)$$

where λ denotes some pre-defined regularization parameter.

The minimization model (2.7) is solved in [19] via an alternating scheme between \mathbf{D} and \mathbf{C} . More specifically, given the current estimate $(\mathbf{D}_k, \mathbf{C}_k)$, the next iteration updates it

via the following scheme:

$$\begin{cases} \mathbf{D}_{k+1} \in \arg \min_{\mathbf{D} \in \mathbb{R}^{m^2 \times m^2}} \|\mathbf{C}_k - \mathbf{D}^\top \mathbf{Y}\|_F^2, & \text{s.t. } \mathbf{D}^\top \mathbf{D} = \mathbf{I}; \\ \mathbf{C}_{k+1} \in \arg \min_{\mathbf{C} \in \mathbb{R}^{m^2 \times L}} \|\mathbf{C} - \mathbf{D}_{k+1}^\top \mathbf{Y}\|_F^2 + \lambda^2 \|\mathbf{C}\|_0. \end{cases} \quad (2.8)$$

It is shown in [19] that both sub-problems in (2.8) have closed-form solutions given by

$$\mathbf{D}_{k+1} := \mathbf{U}_k \mathbf{V}_k^\top; \quad \mathbf{C}_{k+1} \in T_\lambda(\mathbf{D}_{k+1}^\top \mathbf{Y}), \quad (2.9)$$

where \mathbf{U}_k and \mathbf{V}_k are given by the singular value decomposition (SVD) of $\mathbf{Y}\mathbf{C}_k^\top$ such that $\mathbf{Y}\mathbf{C}_k^\top = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$. See Algorithm 1 for the summary of the alternating iteration scheme [19].

Algorithm 1 Alternating iteration scheme [19] for solving (2.7).

- 1: **INPUT:** Input image \mathbf{g} ;
 - 2: **OUTPUT:** Adaptive filter set \mathbf{D} ;
 - 3: **Main Procedure:**
 - i. Set initial filter matrix \mathbf{D}_0 and coefficient matrix \mathbf{C}_0 .
 - ii. Construct the patch matrix \mathbf{Y} as (2.6).
 - iii. For $k = 0, 1, \dots$,
 1. compute the SVD of $\mathbf{Y}\mathbf{C}_k^\top = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$;
 2. $\mathbf{D}_{k+1} := \mathbf{U}_k \mathbf{V}_k^\top$ and $\mathbf{C}_{k+1} \in T_\lambda(\mathbf{D}_{k+1}^\top \mathbf{Y})$.
-

2.2.3 Related works

The minimization (2.7) is an ℓ_0 norm related non-convex problem with quadratic constraints. Algorithm 1 proposed in [19] for solving (2.7) alternately updates the filter matrix \mathbf{D} by the SVD and updates the coefficient matrix \mathbf{C} by hard thresholding the coefficients from the last estimate. Such an iterative hard thresholding on wavelet frame coefficients approach has been used in solving various linear inverse problems in image recovery, see e.g. the wavelet frame based image super-resolution methods [22, 23].

As a sparsity prompting functional, the ℓ_0 norm is also used in other sparse approximation based dictionary learning methods. The popular K-SVD method [1] proposed the following minimization model for learning an over-complete dictionary $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m\} \subset \mathbb{R}^n$ with $m > n$:

$$\min_{\mathbf{D} \in \mathbb{R}^{n \times m}, \mathbf{C} \in \mathbb{R}^{m \times p}} \frac{1}{2} \|\mathbf{Y} - \mathbf{DC}\|_F^2 + \lambda \|\mathbf{C}\|_0, \quad \text{s.t.} \quad \|\mathbf{d}_i\|_2 = 1, i = 1, 2, \dots, n. \quad (2.10)$$

An alternating iteration scheme between \mathbf{D} and \mathbf{C} is used in the K-SVD method for solving (2.10). Different from the model (2.7) proposed in [19], the ℓ_0 norm related minimization problem for updating the code \mathbf{C} is a challenging one. The greedy algorithm, such as orthogonal matching pursuit, is used in [1] for estimating the code. Therefore, the computational cost of the K-SVD method is much higher than Algorithm 1.

Both the K-SVD method and Algorithm 1 perform noticeably better in image denoising than other wavelet frame based methods. The advantage of Algorithm 1 over the K-SVD method lies in its computational efficiency. Despite their impressive performances in practice, both methods lack the convergence analysis. Indeed, it is empirically observed that the sequences generated by both methods are not convergent. In this chapter, we first provided the convergence analysis for Algorithm 1 by showing that the sequence generated by Algorithm 1 has sub-sequence convergence. Then we proposed a modified version of Algorithm 1 for solving (2.7) and established the sequence convergence of the new algorithm.

2.3 Sub-sequence convergence property of Algorithm 1

In this section, we will show that the sequence generated by Algorithm 1 has sub-sequence convergence property, i.e., there exists at least one convergent subsequence and every convergent subsequence converges to a stationary point of (2.7). Before establishing the main result, we first introduce the definition of the stationary point of non-convex and non-smooth functions.

Definition 2.3.1. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ be a proper lower semi-continuous function.*

1 The domain of f is defined by $\text{dom} f := \{x \in \mathbb{R}^n : f(x) < +\infty\}$.

2 For each $x \in \text{dom} f$, x is called the coordinate-wise minimum of f if it satisfies

$$f(x + (0, \dots, d_k, \dots, 0)) \leq f(x), \quad \forall d_k, 1 \leq k \leq n,$$

where $x = (x_1, x_2, \dots, x_n)$.

3 The Fréchet subdifferential $\partial_F f$ is defined by

$$\partial_F f(x) = \{z : \liminf_{y \rightarrow x} \frac{f(y) - f(x) - \langle z, x - y \rangle}{\|y - x\|} \geq 0\} \quad (2.11)$$

for any $x \in \text{dom} f$ and $\partial_F f(x) = \emptyset$ if $x \notin \text{dom} f$.

4 For each $x \in \text{dom} f$, x is called the stationary point of f if it satisfies $0 \in \partial_F f(x)$.

Remark There are several definitions for stationary points of proper lower semi-continuous functions. In [83], the stationary point x is defined as

$$\liminf_{\lambda \downarrow 0} \frac{f(x + \lambda y) - f(x)}{\lambda} \geq 0, \quad \forall y \in \mathbb{R}^n.$$

In [3], the stationary point x of f is defined by $0 \in \partial f(x)$, where ∂f is the limiting subdifferential given by

$$\partial f(x) = \{z : \exists x_n \rightarrow x, f(x_n) \rightarrow f(x), z_n \in \partial_F f(x_n) \rightarrow z\}.$$

The definition of stationary points used in this chapter is different from the definitions used in [83] and [3]. Indeed, ours is stronger than the other two definitions.

To simplify notations, define $\mathcal{X} = \{D \in \mathbb{R}^{m^2 \times m^2} : D^\top D = I_{m^2}\}$ and define $\Omega_C = \mathbb{R}^{m^2 \times N}$, $\Omega_D = \mathbb{R}^{m^2 \times m^2}$, $\Omega_z = (\Omega_C, \Omega_D)$. Define

$$f(\mathbf{C}) = \lambda^2 \|\mathbf{C}\|_0, \quad Q(\mathbf{C}, \mathbf{D}) = \|\mathbf{D}^\top \mathbf{Y} - \mathbf{C}\|_F^2, \quad g(\mathbf{D}) = I_{\mathcal{X}}(\mathbf{D}), \quad (2.12)$$

where $I_{\mathcal{X}}(\mathbf{D}) = 0$, if $\mathbf{D} \in \mathcal{X}$ and $+\infty$ otherwise. Then, the minimization (2.7) can be re-written as

$$\min_{\mathbf{C} \in \Omega_{\mathbf{C}}, \mathbf{D} \in \Omega_{\mathbf{D}}} L(\mathbf{C}, \mathbf{D}) := f(\mathbf{C}) + Q(\mathbf{C}, \mathbf{D}) + g(\mathbf{D}). \quad (2.13)$$

Before proving the sub-sequence convergence property of Algorithm 1, we first establish some facts and results related to (2.13). Firstly, the function g is a lower semi-continuous function, as \mathcal{X} is a compact set. Secondly, it can be seen that for any $\mathbf{Z} = (\mathbf{C}, \mathbf{D})$, the function $Q(\mathbf{Z})$ satisfies the following properties:

$$\begin{cases} Q(\mathbf{C}, \mathbf{D}) = Q(\mathbf{C}_1, \mathbf{D}) + \langle \nabla_{\mathbf{C}} Q(\mathbf{C}_1, \mathbf{D}), \mathbf{C} - \mathbf{C}_1 \rangle + o(\|\mathbf{C} - \mathbf{C}_1\|_F), \quad \forall \mathbf{C}_1 \in \Omega_{\mathbf{C}}; \\ Q(\mathbf{C}, \mathbf{D}) = Q(\mathbf{C}, \mathbf{D}_1) + \langle \nabla_{\mathbf{D}} Q(\mathbf{C}, \mathbf{D}_1), \mathbf{D} - \mathbf{D}_1 \rangle + o(\|\mathbf{D} - \mathbf{D}_1\|_F), \quad \forall \mathbf{D}_1 \in \Omega_{\mathbf{D}}; \\ Q(\mathbf{C}, \mathbf{D}) = Q(\mathbf{C}_1, \mathbf{D}_1) + \langle \nabla Q(\mathbf{C}_1, \mathbf{D}_1), \mathbf{Z} - \mathbf{Z}_1 \rangle + o(\|\mathbf{Z} - \mathbf{Z}_1\|_F), \quad \forall \mathbf{Z}_1 \in \Omega_{\mathbf{Z}}, \end{cases} \quad (2.14)$$

where $o(\|x\|_F)$ is defined by $\lim_{\|x\|_F \rightarrow 0} \frac{o(\|x\|_F)}{\|x\|_F} = 0$.

Lemma 2.3.2. *The sequence $\mathbf{Z}_k := (\mathbf{C}_k, \mathbf{D}_k)$ generated by Algorithm 1 is a bounded sequence. For any convergent sub-sequence $\mathbf{Z}_{k'}$ with limit point $\mathbf{Z}^* = (\mathbf{C}^*, \mathbf{D}^*)$, we have*

$$\lim_{k' \rightarrow +\infty} f(\mathbf{C}_{k'}) = f(\mathbf{C}^*), \quad \text{and} \quad \lim_{k' \rightarrow +\infty} L(\mathbf{Z}_{k'}) = L(\mathbf{Z}^*).$$

Proof. By the definition of (2.9), we have

$$L(\mathbf{Z}_k) \leq L(\mathbf{C}_{k-1}, \mathbf{D}_k) \leq L(\mathbf{C}_{k-1}, \mathbf{D}_{k-1}) \leq \dots \leq L(\mathbf{Z}_0),$$

which implies

$$\|\mathbf{C}_k\|_F - \|\mathbf{D}_k^\top \mathbf{Y}\|_F \leq \|\mathbf{D}_k^\top \mathbf{Y} - \mathbf{C}^k\|_F \leq \sqrt{L(\mathbf{Z}_0)}, \quad k = 1, 2, \dots \quad (2.15)$$

Together with (2.15) and the fact that $\mathbf{D}_k \in \mathcal{X}$, we have \mathbf{Z}_k is bounded. Next, by the definition of (2.9), we also have

$$Q(\mathbf{C}_{k'}, \mathbf{D}_{k'}) + f(\mathbf{C}_{k'}) \leq Q(\mathbf{C}, \mathbf{D}_{k'}) + f(\mathbf{C}), \quad \forall \mathbf{C} \in \Omega_{\mathbf{C}}. \quad (2.16)$$

By substituting \mathbf{C} by \mathbf{C}^* and taking $k' \rightarrow \infty$ in (2.16), we have $\liminf_{k' \rightarrow +\infty} f(\mathbf{C}_{k'}) \leq f(\mathbf{C}^*)$. Together with the fact that $f(\mathbf{C}) = \lambda^2 \|\mathbf{C}\|_0$ is lower semi-continuous and $\mathbf{C}_{k'} \rightarrow \mathbf{C}^*$ as $k' \rightarrow +\infty$, we have

$$\liminf_{k' \rightarrow +\infty} f(\mathbf{C}_{k'}) = f(\mathbf{C}^*).$$

Since $\mathbf{D}_{k'} \in \mathcal{X}$ for all k' and \mathcal{X} is a compact subset, $\mathbf{D}^* \in \mathcal{X}$ and $g(\mathbf{D}^*) = g(\mathbf{D}_{k'}) = 0$ for all k' . It can be seen that $Q(\mathbf{C}_{k'}, \mathbf{D}_{k'}) \rightarrow Q(\mathbf{C}^*, \mathbf{D}^*)$ as $k' \rightarrow +\infty$, as Q is a continuous function. In addition, $L(\mathbf{Z}_k)$ is decreasing by (2.15) and $L \geq 0$, which implies that $L(\mathbf{Z}_k)$ is a convergent sequence. Consequently, we have

$$\lim_{k' \rightarrow +\infty} f(\mathbf{C}_{k'}) = f(\mathbf{C}^*),$$

since $f(\mathbf{C}) = L(\mathbf{Z}) - Q(\mathbf{Z}) - g(\mathbf{D})$. Moreover, we have

$$\begin{aligned} \lim_{k' \rightarrow +\infty} L(\mathbf{Z}_{k'}) &= \lim_{k' \rightarrow +\infty} f(\mathbf{C}_{k'}) + \lim_{k' \rightarrow +\infty} Q(\mathbf{C}_{k'}, \mathbf{D}_{k'}) + \lim_{k' \rightarrow +\infty} g(\mathbf{D}_{k'}) \\ &= f(\mathbf{C}^*) + Q(\mathbf{C}^*, \mathbf{D}^*) + g(\mathbf{D}^*). \end{aligned}$$

Thus, $\lim_{k' \rightarrow +\infty} L(\mathbf{Z}_{k'}) = L(\mathbf{Z}^*)$. ■

Lemma 2.3.3. *Let $\mathbf{Z}_k := (\mathbf{C}_k, \mathbf{D}_k)$ denote the sequence generated by Algorithm 1 and let Ω_* denote the set that contains all limit points of \mathbf{Z}_k . Then Ω_* is not empty and*

$$L(\mathbf{C}^*, \mathbf{D}^*) = \inf_k L(\mathbf{C}_k, \mathbf{D}_k), \quad \forall (\mathbf{C}^*, \mathbf{D}^*) \in \Omega_*.$$

Proof. By Lemma 2.3.2, \mathbf{Z}_k is a bounded sequence. Thus, the set Ω_* is a non-empty set. Moreover, the set Ω_* is also a compact set as $\Omega_* = \bigcap_{j \in \mathbb{N}} \overline{\bigcup_{k \geq j} \{\mathbf{Z}_k\}}$. Notice that $L(\mathbf{Z}_k)$ is a decreasing sequence and $L(\mathbf{Z}) \geq 0$. Then, there exists some constant ρ such that $\inf_k L(\mathbf{Z}_k) = \rho$. Take any $\mathbf{Z}^* \in \Omega_*$ and assume $\mathbf{Z}_{k'} \rightarrow \mathbf{Z}^*$ as $k' \rightarrow +\infty$. By lemma 2.3.2, we have that

$$\lim_{k' \rightarrow +\infty} L(\mathbf{Z}_{k'}) = L(\mathbf{Z}^*) = \rho. \quad \blacksquare$$

At last, we show that the sequence generated by Algorithm 1 has sub-sequence convergence property.

Theorem 2.3.4. *The sequence $\mathbf{Z}_k := (\mathbf{C}_k, \mathbf{D}_k)$ generated by Algorithm 1 has at least one limit point, and any limit point of the sequence \mathbf{Z}_k is a stationary point of (2.13).*

Proof. By Lemma 2.3.3, the sequence $\mathbf{Z}_k := (\mathbf{C}_k, \mathbf{D}_k)$ generated by Algorithm 1 has at least one limit point. For any limit point $\mathbf{Z}^1 = (\mathbf{C}^1, \mathbf{D}^1)$ of the sequence \mathbf{Z}_k , let $\{\mathbf{Z}_{k'}\}$ be the sub-sequence of \mathbf{Z}_k that converges to \mathbf{Z}^1 . Without loss of generality, assume the sub-sequence $\{\mathbf{Z}_{k'+1}\}$ converges to $\mathbf{Z}^2 = (\mathbf{C}^2, \mathbf{D}^2)$. By the definition of the second step in (2.9), we have

$$Q(\mathbf{C}_{k'}, \mathbf{D}_{k'}) + f(\mathbf{C}_{k'}) \leq Q(\mathbf{C}, \mathbf{D}_{k'}) + f(\mathbf{C}), \quad \forall \mathbf{C} \in \Omega_C. \quad (2.17)$$

Taking $k' \rightarrow +\infty$ in (2.17). By Lemma 2.3.2, we have

$$g(\mathbf{D}^1) + Q(\mathbf{C}^1, \mathbf{D}^1) + f(\mathbf{C}^1) \leq g(\mathbf{D}^1) + Q(\mathbf{C}, \mathbf{D}^1) + f(\mathbf{C}), \quad \forall \mathbf{C} \in \Omega_C, \quad (2.18)$$

which implies

$$L(\mathbf{C}^1, \mathbf{D}^1) \leq L(\mathbf{C}, \mathbf{D}^1), \quad \forall \mathbf{C} \in \Omega_C. \quad (2.19)$$

As $\mathbf{Z}_{k'+1}$ is defined from $\mathbf{Z}_{k'}$ by (2.9), we have

$$\begin{cases} Q(\mathbf{C}_{k'}, \mathbf{D}_{k'+1}) + g(\mathbf{D}_{k'+1}) \leq Q(\mathbf{C}_{k'}, \mathbf{D}) + g(\mathbf{D}), \quad \forall \mathbf{D} \in \Omega_D; \\ Q(\mathbf{C}_{k'+1}, \mathbf{D}_{k'+1}) + f(\mathbf{C}_{k'+1}) \leq Q(\mathbf{C}, \mathbf{D}_{k'+1}) + f(\mathbf{C}), \quad \forall \mathbf{C} \in \Omega_C. \end{cases}$$

The summation of the first inequality and the second inequality with $\mathbf{C} = \mathbf{C}_{k'}$ gives

$$g(\mathbf{D}_{k'+1}) + Q(\mathbf{C}_{k'+1}, \mathbf{D}_{k'+1}) + f(\mathbf{C}_{k'+1}) \leq g(\mathbf{D}) + Q(\mathbf{C}_{k'}, \mathbf{D}) + f(\mathbf{C}_{k'}). \quad (2.20)$$

Taking $k' \rightarrow +\infty$ in (2.20). By Lemma 2.3.2 and Lemma 2.3.3, we have

$$L(\mathbf{C}^1, \mathbf{D}^1) = L(\mathbf{C}^2, \mathbf{D}^2) \leq L(\mathbf{C}^1, \mathbf{D}). \quad (2.21)$$

Thus, the combination of (2.19) and (2.21) shows that the point $(\mathbf{C}^1, \mathbf{D}^1)$ is a coordinate-

wise minimum point of (2.13). Therefore, for any $\delta_Z = (\delta_C, \delta_D)$, we have

$$\begin{aligned}
& \liminf_{\|\delta_Z\| \rightarrow 0} \frac{L(\mathbf{Z}^1 + \delta_Z) - L(\mathbf{Z}^1)}{\|\delta_Z\|} \\
&= \liminf_{\|\delta_Z\| \rightarrow 0} \frac{Q(\mathbf{Z}^1 + \delta_Z) - Q(\mathbf{Z}^1) + f(\mathbf{C}^1 + \delta_C) - f(\mathbf{C}^1) + g(\mathbf{D}^1 + \delta_D) - g(\mathbf{D}^1)}{\|\delta_Z\|} \\
&\geq \liminf_{\|\delta_Z\| \rightarrow 0} \frac{\langle \nabla Q(\mathbf{Z}^1), \delta_Z \rangle + f(\mathbf{C}^1 + \delta_C) - f(\mathbf{C}^1) + g(\mathbf{D}^1 + \delta_D) - g(\mathbf{D}^1)}{\|\delta_Z\|} \\
&= \liminf_{\|\delta_Z\| \rightarrow 0} \left(\frac{Q(\mathbf{C}^1 + \delta_C, \mathbf{D}^1) - Q(\mathbf{C}^1, \mathbf{D}^1) - o(\|\delta_C\|) + f(\mathbf{C}^1 + \delta_C) - f(\mathbf{C}^1)}{\|\delta_Z\|} \right. \\
&\quad \left. + \frac{Q(\mathbf{C}^1, \mathbf{D}^1 + \delta_D) - Q(\mathbf{C}^1, \mathbf{D}^1) - o(\|\delta_D\|) + g(\mathbf{D}^1 + \delta_D) - g(\mathbf{D}^1)}{\|\delta_Z\|} \right) \\
&\geq \liminf_{\|\delta_Z\| \rightarrow 0} \frac{-o(\|\delta_C\|) - o(\|\delta_D\|)}{\|\delta_Z\|} = 0,
\end{aligned}$$

where the first inequality is from (2.14) and the second inequality is from the fact that $\mathbf{Z}^1 := (\mathbf{C}^1, \mathbf{D}^1)$ is the coordinate-wise minimum point of (2.13). By Definition (2.3.1), the point \mathbf{Z}^1 is a stationary point of (2.13). \blacksquare

2.4 A modified algorithm for (2.7) with sequence convergence

In the previous section, we showed that the sequence generated by Algorithm 1 has subsequence convergence property. The next question is whether the sequence itself is convergent or not. The experiments show that it is not the case; see Fig.2.1 (a) for the increments of the sequence \mathbf{C}_k . The lack of sequence convergence is not crucial to the applications in image recovery, as the result we are seeking for is not the frame coefficient vector but the image synthesized from the coefficients. See Fig. 2.1 (b) for an illustration. However, the divergence of the coefficient sequence could cause severe stability issue when the coefficient set is the one needed, e.g. in the case of sparse coding based recognition tasks. Motivated by both theoretical interest and the needs from applications, we proposed a modified version of Algorithm (1) with sequence convergence property, i.e., the sequence generated by the

new algorithm converges to a stationary point of (2.13).

The modification on Algorithm 1 for gaining sequence convergence is done by adding a proximal term in each iteration, a technique which has been used in other alternating iterative methods to ensure the convergence. For example, the proximal method proposed in [3] for solving a class of non-convex and non-smooth functions. The modified version of Algorithm 1 updates the estimates of C and D via solving the following problems:

$$\begin{cases} \mathbf{D}_{k+1} \in \arg \min_{\mathbf{D}} L(\mathbf{C}_k, \mathbf{D}) + \lambda_k \|\mathbf{D} - \mathbf{D}_k\|_F^2; \\ \mathbf{C}_{k+1} \in \arg \min_{\mathbf{C}} L(\mathbf{C}, \mathbf{D}_{k+1}) + \mu_k \|\mathbf{C} - \mathbf{C}_k\|_F^2, \end{cases} \quad (2.22)$$

where $\lambda_k, \mu_k \in (a, b)$ and $a, b > 0$. It can be seen that the new iteration (2.22) adds two additional proximal terms, $\lambda_k \|\mathbf{D} - \mathbf{D}_k\|_F^2$ and $\mu_k \|\mathbf{C} - \mathbf{C}_k\|_F^2$, to the original iteration (2.8). Same as (2.8), both minimization problems in (2.22) also have closed-form solutions.

Proposition 2.4.1. *The solution of (2.22) is given by*

$$\begin{cases} \mathbf{D}_{k+1} = \mathbf{U}_k \mathbf{V}_k^\top, \\ \mathbf{C}_{k+1} \in T_{\lambda/\sqrt{\mu_k+1}}\left(\frac{\mathbf{D}_{k+1}^\top \mathbf{Y} + \mu_k \mathbf{C}_k}{1 + \mu_k}\right), \end{cases} \quad (2.23)$$

where $\mathbf{U}_k, \mathbf{V}_k$ is given by the SVD of $\mathbf{Y}\mathbf{C}_k^\top + \lambda_k \mathbf{D}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$.

Proof. The proof is exactly the same as that of (2.9) provided in [19]. ■

See Algorithm 2 for the summary of the modified algorithm for solving (2.13).

2.4.1 Convergence analysis of Algorithm 2

In this section, we first establish the sub-convergence property of Algorithm 2. Then we establish the sequence convergence of the algorithm by showing that the sequence is a Cauchy sequence and converges to a stationary point of (2.13). The main proof is built on the results presented in [3] about the convergence analysis of proximal methods for solving a class of non-smooth and non-convex problems.

Algorithm 2 Proximal alternating iteration scheme for solving (2.7).

- 1: **INPUT:** Input image \mathbf{g} ;
 - 2: **OUTPUT:** Adaptive filter set \mathbf{D} ;
 - 3: **Main Procedure:**
 - i. Set initial filter matrix \mathbf{D}_0 and coefficient matrix \mathbf{C}_0 .
 - ii. Construct the patch matrix \mathbf{Y} as (2.6).
 - iii. For $k = 0, 1, \dots$,
 1. compute the SVD of $\mathbf{Y}\mathbf{C}_k^\top + \lambda_k\mathbf{D}_k = \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{V}_k^\top$;
 2. $\mathbf{D}_{k+1} = \mathbf{U}_k\mathbf{V}_k^\top$ and $\mathbf{C}_{k+1} \in T_{\lambda\sqrt{\mu_{k+1}}}(\frac{\mathbf{D}_{k+1}^\top\mathbf{Y} + \mu_k\mathbf{C}_k}{1 + \mu_k})$.
-

Theorem 2.4.2. Let $\mathbf{Z}_k := (\mathbf{C}_k, \mathbf{D}_k)$ denote the sequence generated by Algorithm 2. Then, \mathbf{Z}_k has at least one convergent subsequence and every convergent subsequence of \mathbf{Z}_k converges to a stationary point of (2.13).

Proof. By the definition of (2.22), we have

$$\begin{cases} L(\mathbf{C}_k, \mathbf{D}_{k+1}) + \lambda_k \|\mathbf{D}_{k+1} - \mathbf{D}_k\|_F^2 \leq L(\mathbf{C}_k, \mathbf{D}_k), \\ L(\mathbf{C}_{k+1}, \mathbf{D}_{k+1}) + \mu_k \|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F^2 \leq L(\mathbf{C}_k, \mathbf{D}_{k+1}). \end{cases}$$

Sum up both inequalities and by the fact that $a \leq \mu_k, \lambda_k \leq b$, we have

$$L(\mathbf{Z}_k) - L(\mathbf{Z}_{k+1}) \geq a \|\mathbf{Z}_k - \mathbf{Z}_{k+1}\|_F^2 \geq 0. \quad (2.24)$$

By the same argument for (2.15), we have \mathbf{Z}_k is bounded and has at least one limit point.

By (2.24), we obtain

$$L(\mathbf{Z}_0) - L(\mathbf{Z}_{k+1}) \geq \sum_{j=0}^k a \|\mathbf{Z}_j - \mathbf{Z}_{j+1}\|_F^2. \quad (2.25)$$

let $k \rightarrow +\infty$ in (2.25), together with the facts that $L(\mathbf{Z}_k) \geq 0$ and $L(\mathbf{Z}_k)$ is a decreasing sequence, we have

$$\sum_{k=1}^{+\infty} \|\mathbf{Z}_k - \mathbf{Z}_{k+1}\|_F^2 < +\infty,$$

which implies that

$$\lim_{k \rightarrow +\infty} \|\mathbf{Z}_k - \mathbf{Z}_{k+1}\|_F = 0. \quad (2.26)$$

Let $\mathbf{Z}^1 := (\mathbf{C}^1, \mathbf{D}^1)$ denote any limit point of \mathbf{Z}_k , i.e., there exists a sub-sequence $\mathbf{Z}_{k'}$ converges to \mathbf{Z}^1 . In the next, we prove that the sub-sequence $\mathbf{Z}_{k'+1}$ also converges to \mathbf{Z}^1 . For any $\varepsilon > 0$, there exists N_0 such that $\|\mathbf{Z}_{k'} - \mathbf{Z}^1\|_F < \varepsilon/2$ and $\|\mathbf{Z}_{k'} - \mathbf{Z}_{k'+1}\|_F < \varepsilon/2$ for all $k' > N_0$. The first inequality is from the fact that $\mathbf{Z}_{k'}$ converges to \mathbf{Z}^1 and the second one is from (2.26). Thus, for all $k' > N_0$,

$$\|\mathbf{Z}_{k'+1} - \mathbf{Z}^1\|_F \leq \|\mathbf{Z}_{k'} - \mathbf{Z}_{k'+1}\|_F + \|\mathbf{Z}_{k'} - \mathbf{Z}^1\|_F < \varepsilon. \quad (2.27)$$

Consequently, we have $\mathbf{Z}_{k'+1} \rightarrow \mathbf{Z}^1$ as $k' \rightarrow +\infty$.

By the definition of (2.22), we have that, for any $\mathbf{C} \in \Omega_C$,

$$L(\mathbf{C}_{k'+1}, \mathbf{D}_{k'+1}) + a\|\mathbf{C}_{k'+1} - \mathbf{C}_{k'}\|_F^2 \leq L(\mathbf{C}, \mathbf{D}_{k'+1}) + b\|\mathbf{C} - \mathbf{C}_{k'}\|_F^2.$$

Similar to the derivation of (2.16), by setting $\mathbf{C} = \mathbf{C}^1$ and taking $k' \rightarrow +\infty$ in the inequality above, we have $\liminf_{k' \rightarrow +\infty} f(\mathbf{C}_{k'+1}) \leq f(\mathbf{C}^1)$. As f is a lower semi-continuous function, we have

$$\liminf_{k' \rightarrow +\infty} f(\mathbf{C}_{k'+1}) = f(\mathbf{C}^1).$$

By the same arguments in the proof of Lemma 2.3.2, we have $\lim_{k' \rightarrow +\infty} f(\mathbf{C}_{k'+1}) = f(\mathbf{C}^1)$. Again, by using the same arguments for $\lim_{k' \rightarrow +\infty} f(\mathbf{C}_{k'+1})$, we also have $\lim_{k' \rightarrow +\infty} f(\mathbf{C}_{k'}) = f(\mathbf{C}^1)$. Notice that $\mathbf{D}_k \in \mathcal{X}$, $k = 1, 2, \dots$, and \mathcal{X} is a compact set. Thus, $g(\mathbf{D}_{k'}) = g(\mathbf{D}_{k'+1}) = g(\mathbf{D}^1) = 0$ and Q is continuous, which leads to

$$\lim_{k' \rightarrow +\infty} L(\mathbf{Z}_{k'}) = \lim_{k' \rightarrow +\infty} L(\mathbf{Z}_{k'+1}) = L(\mathbf{C}^1, \mathbf{D}^1). \quad (2.28)$$

By the definition of \mathbf{C}_k in (2.22), we have

$$L(\mathbf{C}_{k'+1}, \mathbf{D}_{k'+1}) + a\|\mathbf{C}_{k'+1} - \mathbf{C}_{k'}\|_F^2 \leq L(\mathbf{C}, \mathbf{D}_{k'+1}) + b\|\mathbf{C} - \mathbf{C}_{k'}\|_F^2, \forall \mathbf{C} \in \Omega_C.$$

Taking $k' \rightarrow +\infty$ in the inequality above, together with (2.28) and (2.26), we have

$$L(\mathbf{C}^1, \mathbf{D}^1) \leq L(\mathbf{C}^1 + \mathbf{C}, \mathbf{D}^1) + b\|\mathbf{C}\|_F^2, \forall \mathbf{C} \in \Omega_C. \quad (2.29)$$

Again, by the definition of (2.22), we have

$$\begin{cases} L(\mathbf{C}_{k'}, \mathbf{D}_{k'+1}) + \lambda_{k'} \|\mathbf{D}_{k'+1} - \mathbf{D}_{k'}\|_F^2 \leq L(\mathbf{C}_{k'}, \mathbf{D}) + \lambda_{k'} \|\mathbf{D} - \mathbf{D}_{k'}\|_F^2; \\ L(\mathbf{C}_{k'+1}, \mathbf{D}_{k'+1}) + \mu_{k'} \|\mathbf{C}_{k'+1} - \mathbf{C}_{k'}\|_F^2 \leq L(\mathbf{C}, \mathbf{D}_{k'+1}) + \mu_{k'} \|\mathbf{C} - \mathbf{C}_{k'}\|_F^2. \end{cases} \quad (2.30)$$

Recall that $\lambda_{k'}, \mu_{k'} \in (a, b)$. Then,

$$L(\mathbf{Z}_{k'+1}) + a\|\mathbf{Z}_{k'+1} - \mathbf{Z}_{k'}\|_F^2 \leq L(\mathbf{C}_{k'}, \mathbf{D}) + b\|\mathbf{D} - \mathbf{D}_{k'}\|_F^2, \forall \mathbf{D} \in \Omega_D.$$

Taking $k' \rightarrow +\infty$ in above, together with (2.28) and (2.26), we have

$$L(\mathbf{C}^1, \mathbf{D}^1) \leq L(\mathbf{C}^1, \mathbf{D}^1 + \mathbf{D}) + b\|\mathbf{D}\|_F^2, \forall \mathbf{D} \in \Omega_D.$$

Consequently, for any $\mathbf{d} = (\delta_C, \delta_D) \in (\Omega_C, \Omega_D)$, we have

$$\begin{aligned} & \liminf_{\|\mathbf{d}\| \rightarrow 0} \frac{L(\mathbf{Z}^1 + \mathbf{d}) - L(\mathbf{Z}^1)}{\|\mathbf{d}\|} \\ &= \liminf_{\|\mathbf{d}\| \rightarrow 0} \frac{Q(\mathbf{Z}^1 + \mathbf{d}) - Q(\mathbf{Z}^1) + f(\mathbf{C}^1 + \delta_C) - f(\mathbf{C}^1) + g(\mathbf{D}^1 + \delta_D) - g(\mathbf{D}^1)}{\|\mathbf{d}\|} \\ &\geq \liminf_{\|\mathbf{d}\| \rightarrow 0} \frac{\langle \nabla Q(\mathbf{Z}^1), \mathbf{d} \rangle + f(\mathbf{C}^1 + \delta_C) - f(\mathbf{C}^1) + g(\mathbf{D}^1 + \delta_D) - g(\mathbf{D}^1)}{\|\mathbf{d}\|} \\ &= \liminf_{\|\mathbf{d}\| \rightarrow 0} \left(\frac{Q(\mathbf{C}^1 + \delta_C, \mathbf{D}^1) - Q(\mathbf{C}^1, \mathbf{D}^1) - o(\|\delta_C\|) + f(\mathbf{C}^1 + \delta_C) - f(\mathbf{C}^1)}{\|\mathbf{d}\|} \right. \\ &\quad \left. + \frac{Q(\mathbf{C}^1, \mathbf{D}^1 + \delta_D) - Q(\mathbf{C}^1, \mathbf{D}^1) - o(\|\delta_D\|) + g(\mathbf{D}^1 + \delta_D) - g(\mathbf{D}^1)}{\|\mathbf{d}\|} \right) \\ &\geq \liminf_{\|\mathbf{d}\| \rightarrow 0} \frac{-o(\|\delta_C\|) - o(\|\delta_D\|) - b(\|\delta_C\|_F^2 + \|\delta_D\|_F^2)}{\|\mathbf{d}\|} = 0. \end{aligned}$$

By Definition 2.3.1, we have \mathbf{Z}^1 is a stationary point of (2.13). ■

In the next, we will establish the convergence of the sequence $\mathbf{Z}_k = (\mathbf{C}_k, \mathbf{D}_k)$ generated by (2.22) by showing that it satisfies the so-called *finite length property*, i.e.,

$$\sum_{k=1}^{+\infty} \|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F < +\infty.$$

Clearly, a sequence with finite length property is a Cauchy sequence. Together with Theorem 2.4.2, we have the sequence \mathbf{Z}_k converges to a stationary point of (2.13). The proof is based on the convergence analysis developed in a series of papers ([3, 14, 15]), which studied the convergence of the iteration scheme (2.22) for solving (2.13) with respect to a class of objective functions.

Theorem 2.4.3. [3, Theorem 9] *The sequence $\mathbf{Z}_k = (\mathbf{C}_k, \mathbf{D}_k)$ generated by the iteration (2.22) has finite length property if the following conditions hold:*

1. $L(\mathbf{C}, \mathbf{D})$ is a K-L function;
2. $\mathbf{Z}_k, k = 1, 2, \dots$ is a bounded sequence and there exists some positive constants a, b such that $\lambda_k, \mu_k \in (a, b), k = 1, 2, \dots$;
3. $\nabla Q(\mathbf{C}, \mathbf{D})$ has Lipschitz constant on any bounded set.

In Theorem 2.4.3, there are three conditions to ensure the sequence satisfies the finite length property. The first condition requires that the objective function L satisfies the so-called *Kurdyka-Lojasiewicz (K-L) property* in its effective domain; see [15, Definition 3] for more details on K-L property. Given a function, it is often not easy to check whether it satisfies the K-L property. Nevertheless, it is shown in [14, Remark 5] and [14, Theorem 11] that any so-called *semi-algebraic* function satisfies the K-L property.

Definition 2.4.4. [15] *A subset S of \mathbb{R}^n is called a semi-algebraic set if there exists a finite number of real polynomial functions g_{ij}, h_{ij} such that*

$$S = \bigcup_j \bigcap_i \{\mathbf{u} \in \mathbb{R}^n : g_{ij}(\mathbf{u}) = 0, h_{ij}(\mathbf{u}) < 0\}.$$

A function $f(\mathbf{u})$ is called a semi-algebraic function if its graph $\{(\mathbf{u}, t) \in \mathbb{R}^n \times \mathbb{R}, t = f(\mathbf{u})\}$ is a semi-algebraic set.

Theorem 2.4.5. *Let $\mathbf{Z}_k = (\mathbf{C}_k, \mathbf{D}_k)$ denote the sequence generated by (2.22). Then, the sequence \mathbf{Z}_k has the finite length property and thus is a Cauchy sequence.*

Proof. The proof is done by showing that Theorem 2.4.3 is applicable to the objective function (2.13) and the sequence \mathbf{Z}_k generated by (2.22). Thus, we only need to verify all three conditions in Theorem 2.4.3.

The first condition in Theorem 2.4.3 is verified by showing that all three terms in the objective function L given by (2.13) are semi-algebraic functions. The second term $Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2} \|\mathbf{D}^\top \mathbf{Y} - \mathbf{C}\|_F^2$ is clearly a semi-algebraic function as it is a real polynomial. Next, it can be seen that the set $\mathcal{X} = \{\mathbf{D} \in \mathbb{R}^{m^2 \times m^2} : \mathbf{D}^\top \mathbf{D} = \mathbf{I}\} = \bigcap_{j=1}^m \bigcap_{k=1}^m \{\mathbf{D} : \sum_{i=1}^m \mathbf{d}_{ki} \mathbf{d}_{ji} = \delta_{j,k}\}$ is a semi-algebraic set. Thus, the last term $g(\mathbf{D}) = I_{\mathcal{X}}(\mathbf{D})$ is also a semi-algebraic function, as it is shown in [2] that indicator functions of semi-algebraic sets are semi-algebraic functions. Regarding the first term $f(\mathbf{C}) = \lambda^2 \|\mathbf{C}\|_0$. The graph of $F = \|\mathbf{C}\|_0$ is $S = \bigcup_{k=0}^{m^2 L} L_k \triangleq \{(\mathbf{C}, k) : \|\mathbf{C}\|_0 = k\}$. For each $k = 0, \dots, m^2 L$, let $\mathcal{S}_k = \{J : J \subseteq \{1, \dots, m^2 L\}, |J| = k\}$, then $L_k = \bigcup_{J \in \mathcal{S}_k} \{(\mathbf{C}, k) : \mathbf{C}_{J^c} = 0\}$. It can be seen that the set $\{(\mathbf{C}, k) : \mathbf{C}_{J^c} = 0\}$ is a semi-algebraic set in $\mathbb{R}^{m^2 \times L} \times \mathbb{R}$. Thus, $F(\mathbf{C}) = \|\mathbf{C}\|_0$ is a semi-algebraic function, as the finite union of the semi-algebraic set is still semi-algebraic.

Regarding the second condition in theorem 2.4.3, the boundness of the sequence $\mathbf{Z}_k = (\mathbf{C}_k, \mathbf{D}_k)$ is ensured by Theorem 2.4.2. Moreover, by the definition of (2.22), there exist two positive constants $a, b > 0$ such that $\lambda_k, \mu_k \in (a, b)$ for $k = 1, 2, \dots$

For the last condition in theorem 2.4.3, notice that the function $Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2} \|\mathbf{C} - \mathbf{D}^\top \mathbf{Y}\|_F^2$ is a smooth function. Thus, for any bounded set \mathcal{M} , there exists a constant $M > 0$ such that

$$\|\nabla Q(\mathbf{C}_1, \mathbf{D}_1) - \nabla Q(\mathbf{C}_2, \mathbf{D}_2)\|_F \leq M \|(\mathbf{C}_1, \mathbf{D}_1) - (\mathbf{C}_2, \mathbf{D}_2)\|_F$$

for any $(\mathbf{C}_1, \mathbf{D}_1) \in \mathcal{M}$ and $(\mathbf{C}_2, \mathbf{D}_2) \in \mathcal{M}$. ■

In summary, we have the following result regarding the convergence of Algorithm 2.

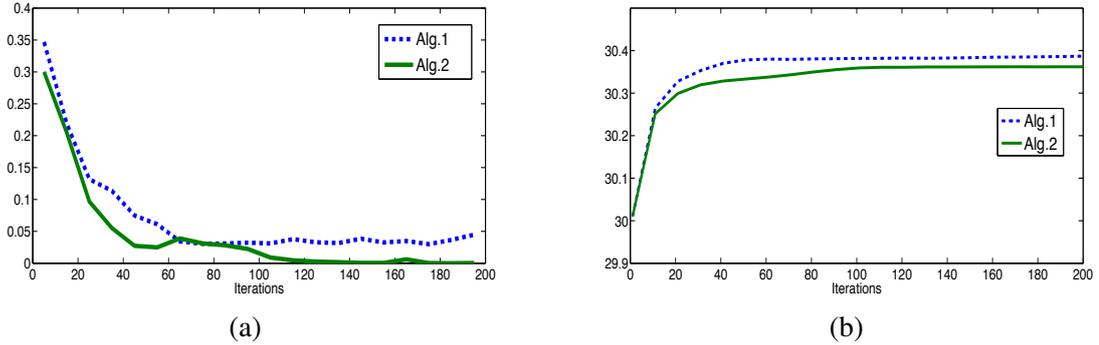


Figure 2.1: Convergence behavior of Algorithm 1 and Algorithm 2. (a) The ℓ_2 norm of the increments of the framelet coefficient vector at each iteration; and (b) the PSNR values of the intermediate results at each iteration when denoising the image "boat" with noise level $\sigma = 20$.

Corollary 2.4.6. *The sequence $\mathbf{Z}_k := (\mathbf{C}_k, \mathbf{D}_k)$ generated by Algorithm 2 converges to a stationary point of (2.13).*

2.5 Experiments on image denoising

There are two main parts in this section: one is the convergence analysis of the method proposed in [19] and the other is the modifications of the original algorithm for gaining stronger convergence property. The later is more of theoretical interest and for potential benefit to other applications. Thus, the experimental evaluation done in this chapter for image denoising is not as comprehensive as [19]. The data-driven tight frame based image denoising is done as follows. Let $f = g + \varepsilon(\sigma)$ denote some noisy observation of g , where $\varepsilon(\sigma)$ is the additive i.i.d. Gaussian noise with zero mean and standard deviation σ . Taking f as the input and using 8×8 DCT as the initial guess, the filters of data-drive tight frame $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{64}\}$ are constructed using Algorithm 1 (or Algorithm 2). Then the denoised result, denoted by \tilde{g} , is obtained via hard thresholding:

$$\tilde{g} = \mathbf{W}^\top (T_{\tilde{\lambda}}(\mathbf{W}f)),$$

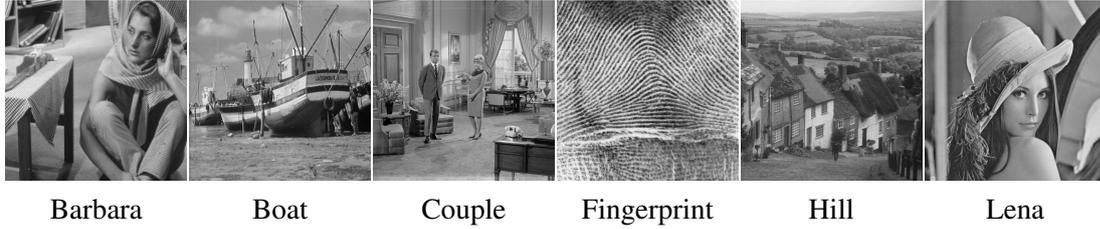


Figure 2.2: Six test images

Image	Babara					Boat				
σ	10	20	30	40	50	10	20	30	40	50
Alg. 1; 8	34.36	30.60	28.42	26.88	25.67	33.62	30.38	28.39	27.06	25.99
Alg. 1; 16	34.63	31.07	29.07	27.60	26.48	33.59	30.41	28.45	27.18	26.08
Alg. 2; 8	34.34	30.58	28.34	26.89	25.74	33.61	30.29	28.39	26.94	25.87
Alg. 2; 16	34.63	31.14	29.02	27.58	26.41	33.58	30.39	28.48	27.16	26.13
Image	Fingerprint					Hill				
σ	10	20	30	40	50	10	20	30	40	50
Alg. 1; 8	32.23	28.32	26.18	24.67	23.52	33.28	30.22	28.56	27.36	26.48
Alg. 1; 16	32.25	28.40	26.34	24.95	23.88	33.28	30.30	28.61	27.52	26.63
Alg. 2; 8	32.20	28.27	26.13	24.66	23.46	33.26	30.20	28.45	27.25	26.38
Alg. 2; 16	32.24	28.38	26.33	24.93	23.87	33.22	30.23	28.64	27.50	26.65
Image	Couple					Lena				
Alg. 1; 8	33.63	30.09	28.16	26.72	25.68	35.52	32.25	30.22	28.80	27.60
Alg. 1; 16	33.55	30.19	28.27	26.95	25.87	35.65	32.56	30.58	29.16	28.14
Alg. 2; 8	33.49	30.05	28.02	26.64	25.61	35.47	32.29	30.25	28.77	27.57
Alg. 2; 16	33.52	30.10	28.25	26.93	25.89	35.64	32.53	30.51	29.16	28.06

Table 2.1: PSNR values of the denoised results

where \mathbf{W} denotes the analysis operator determined by $\{\mathbf{a}_j\}_{j=1}^{64}$ and $\tilde{\lambda}$ is thresholding parameter determined by noise level. Throughout all experiments, the parameter $\tilde{\lambda}$ is fixed at $\tilde{\lambda} = 2.7\sigma$ for both Algorithm 1 and Algorithm 2. The other settings for Algorithm 1 are the same as [19]. For Algorithm 2, we set the maximum number of iteration to 70 and set $\lambda_k = 0.047, \mu_k = 0.024$ for all k .

We start with the demonstration of convergence behavior of Algorithm 1 proposed in [19] and Algorithm 2 proposed in this chapter. See Fig. 2.1 (a) for the comparison of the ℓ_2 norm of the increments of the frame coefficient vectors C^k generated by two algorithms. It can be seen that the coefficient sequence generated by Algorithm 1 does not converge while the one generated by Algorithm 2 converges. However, the lack of sequence convergence of Algorithm 1 does not impact its performance of image denoising, as shown in Fig. 2.1. The PSNR values of the denoised results from both algorithms are summarized in Table 2.1 with

respect to different images and different noise levels. It can be seen that the performances of both algorithms in image denoising are very close in terms of PSNR value.

2.6 Extensions

In this section, we extended the data-driven tight frame construction scheme when part of the filters are fixed during the learning process.

2.6.1 Problem formulation

Given an image g , let $\mathbf{G} = \{\mathbf{g}_1, \dots, \mathbf{g}_m\} \in \mathbb{R}^{n \times m}$ denote the training set of image patches of size $\sqrt{n} \times \sqrt{n}$ collected from the image after vectorization. The image patches for the training can be selected randomly or regularly. Now we consider the sparse approximation problem for the set \mathbf{G} under an orthogonal dictionary $\hat{\mathbf{D}} := [\mathbf{A}, \mathbf{D}] \in \mathbb{R}^{n \times n}$ whose columns refer to dictionary atoms. The dictionary has two sub-dictionaries in our implementation: one is $\mathbf{A} \in \mathbb{R}^{n \times n-r}$ which contains the input orthogonal atoms known as good ones from other sources; the other is $\mathbf{D} \in \mathbb{R}^{n \times r}$ which denotes the set of atoms need to be learned from the input image. The orthogonal constraint on the dictionary says that

$$\hat{\mathbf{D}}^\top \hat{\mathbf{D}} = I_n \Rightarrow \mathbf{A}^\top \mathbf{A} = I_{n-r}; \mathbf{D}^\top \mathbf{D} = I_r; \mathbf{A}^\top \mathbf{D} = \mathbf{0}.$$

We propose to learn the orthogonal dictionary \mathbf{D} via solving the following minimization model

$$\begin{aligned} \min_{\mathbf{D} \in \mathbb{R}^{n \times r}, \mathbf{V} \in \mathbb{R}^{n \times m}} & \|\mathbf{G} - [\mathbf{A}, \mathbf{D}]\mathbf{V}\|_F^2 + \lambda^2 \|\mathbf{V}\|_0, \\ \text{s.t.} & \quad \mathbf{D}^\top \mathbf{D} = I_r; \mathbf{A}^\top \mathbf{D} = \mathbf{0}. \end{aligned} \quad (2.31)$$

It is noted that $r = n$ if the set \mathbf{A} is empty.

The minimization (2.1) is quite similar to the model (3.2) used in the K-SVD method, except some additional linear and bi-linear constraints on \mathbf{D} . In the next, we will show that the minimization (2.31) is much easier to solve than (3.2).

2.6.2 Numerical method

Same as the K-SVD method, we take an alternating iterative scheme to solve (2.31). More specifically, let $\mathbf{D}^{(0)}$ be the initial dictionary to start (e.g. the DCT dictionary or multi-scale wavelet dictionary). Then for $k = 0, 1, \dots, K-1$,

1. **sparse coding:** given the dictionary $\mathbf{D}^{(k)}$ with orthogonal columns, find the sparse code $\mathbf{V}^{(k)}$ via solving

$$\mathbf{V}^{(k)} := \arg \min_{\mathbf{V} \in \mathbb{R}^{n \times m}} \|\mathbf{G} - [\mathbf{A}, \mathbf{D}^{(k)}]\mathbf{V}\|_F^2 + \lambda^2 \|\mathbf{V}\|_0. \quad (2.32)$$

2. **dictionary updating:** given the sparse code $\mathbf{V}^{(k)}$, update the dictionary via solving the minimization:

$$\begin{aligned} \mathbf{D}^{(k+1)} &:= \arg \min_{\mathbf{D} \in \mathbb{R}^{n \times r}} \|\mathbf{G} - [\mathbf{A}, \mathbf{D}]\mathbf{V}^{(k)}\|_F^2, \\ \text{s.t. } &\mathbf{D}^\top \mathbf{D} = \mathbf{I}_r, \mathbf{A}^\top \mathbf{D} = \mathbf{0}. \end{aligned} \quad (2.33)$$

Then, we show that both the minimization (2.32) for sparse coding and (2.33) for dictionary update are trivial to solve. Indeed, each of them has an explicit solution. Define $\hat{\mathbf{D}} = [\mathbf{A}, \mathbf{D}^{(k)}]$. Then by the definition of \mathbf{A} and $\mathbf{D}^{(k)}$, we have $\hat{\mathbf{D}}^\top \hat{\mathbf{D}} = \mathbf{I}_n$. The next proposition gives an explicit solution to (2.32).

Proposition 2.6.1. (sparse coding) *Suppose that $\hat{\mathbf{D}}^\top \hat{\mathbf{D}} = \mathbf{I}_n$. The following minimization problem,*

$$\min_{\mathbf{V}} \|\mathbf{G} - \hat{\mathbf{D}}\mathbf{V}\|_F^2 + \lambda^2 \|\mathbf{V}\|_0, \quad (2.34)$$

has a unique solution given by $\mathbf{V}^ = T_\lambda(\hat{\mathbf{D}}^\top \mathbf{G})$.*

Proof. By the fact that $\hat{\mathbf{D}}^\top \hat{\mathbf{D}} = \mathbf{I}_n$, the minimization (2.34) is the equivalent to the following minimization

$$\min_{\mathbf{V}} \|\hat{\mathbf{D}}^\top \mathbf{G} - \mathbf{V}\|_F^2 + \lambda^2 \|\mathbf{V}\|_0, \quad (2.35)$$

which can be rewritten as

$$\min_{\{\mathbf{V}_{i,j}\}} \sum_{i,j} (\mathbf{V}_{i,j} - (\hat{\mathbf{D}}^\top \mathbf{G})_{i,j})^2 + \lambda |\mathbf{V}_{i,j}|,$$

or equivalently the summation of multiple independent univariate minimization problems

$$\sum_{i,j} \min_{\{\mathbf{V}_{i,j}\}} (\mathbf{V}_{i,j} - (\hat{\mathbf{D}}^\top \mathbf{G})_{i,j})^2 + \lambda |\mathbf{V}_{i,j}|.$$

Recall that minimization problem $\min_{x \in \mathbb{R}} \|x - y\|_2^2 + \lambda^2 \|x\|_0$ has a unique solution $x^* = T_\lambda(y)$. Thus, the unique minimizer for (2.34) is $T_\lambda(\hat{\mathbf{D}}^\top \mathbf{G})$. ■

For dictionary update, let $\mathbf{V}^{(k)} = [\mathbf{V}_A^{(k)\top}, \mathbf{V}_D^{(k)\top}]^\top$, where $\mathbf{V}_A^{(k)}$ denotes the codes associated with \mathbf{A} and $\mathbf{V}_D^{(k)}$ associated with $\mathbf{D}^{(k)}$. Let \mathcal{P}_A denote the orthogonal projection operator from \mathbb{R}^n to the space spanned by the columns of \mathbf{A} : $\mathcal{P}_A \mathbf{v} = \mathbf{A}(\mathbf{A}^\top \mathbf{v})$, $\forall \mathbf{v} \in \mathbb{R}^n$. Then, the next proposition gives the explicit solution to the minimization (2.33).

Proposition 2.6.2. (dictionary updating) *The following constrained minimization*

$$\begin{aligned} \min_{\mathbf{D} \in \mathbb{R}^{n \times r}} \|\mathbf{G} - (\mathbf{A}\mathbf{V}_A + \mathbf{D}\mathbf{V}_D)\|_F^2 \\ \text{s.t. } \mathbf{D}^\top \mathbf{D} = \mathbf{I}_r, \mathbf{A}^\top \mathbf{D} = \mathbf{0} \end{aligned} \quad (2.36)$$

has a unique solution given by $\mathbf{D}^* = \mathbf{P}\mathbf{Q}^\top$, where \mathbf{P} and \mathbf{Q} denote the orthogonal matrices defined by the following SVD

$$(\mathbf{I}_n - \mathcal{P}_A)\mathbf{G}\mathbf{V}_D^\top = \mathbf{P}\Sigma\mathbf{Q}^\top.$$

Proof. The objective function in (2.36) is equal to

$$\begin{aligned} \|\mathbf{G} - \mathbf{A}\mathbf{V}_A - \mathbf{D}\mathbf{V}_D\|_F^2 \\ = \|\mathbf{G} - \mathbf{A}\mathbf{V}_A\|_F^2 + \|\mathbf{D}\mathbf{V}_D\|_F^2 - \text{Tr}((\mathbf{G} - \mathbf{A}\mathbf{V}_A)^\top \mathbf{D}\mathbf{V}_D). \end{aligned} \quad (2.37)$$

If $\mathbf{D}^\top \mathbf{D} = \mathbf{I}$ and $\mathbf{A}^\top \mathbf{D} = \mathbf{0}$, then the first two terms in (2.37) are constant and $\text{Tr}((\mathbf{A}\mathbf{V}_A)^\top \mathbf{D}\mathbf{V}_D) =$

0. Therefore, the minimization (2.36) is equivalent to

$$\max_{\mathbf{D}} \text{Tr}(\mathbf{D}^\top \mathbf{G} \mathbf{V}_D^\top), \text{ s.t. } \mathbf{D}^\top \mathbf{D} = \mathbf{I}_r, \mathbf{A}^\top \mathbf{D} = \mathbf{0}. \quad (2.38)$$

Consider the following SVD: $(\mathbf{I}_n - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^\top = \mathbf{P} \mathbf{\Sigma} \mathbf{Q}^\top$. From the Theorem 4 in [96], $\mathbf{D} = \mathbf{P} \mathbf{Q}^\top$ is the minimizer of the following minimization problem

$$\max_{\mathbf{D} \in \mathbb{R}^{n \times r}} \text{Tr}(\mathbf{D}^\top (\mathbf{I} - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^\top), \text{ s.t. } \mathbf{D}^\top \mathbf{D} = \mathbf{I}_r. \quad (2.39)$$

Notice that the space spanned by the columns \mathbf{P} is equal to the one spanned by the columns of $(\mathbf{I} - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^\top$ which is orthogonal to the space spanned by \mathbf{A} . Therefore, $\mathbf{A}^\top \mathbf{D} = \mathbf{A}^\top \mathbf{P} \mathbf{Q}^\top = \mathbf{0}$. Put all together, we have $\mathbf{D} = \mathbf{P} \mathbf{Q}^\top$ is the minimizer to the following minimization problem

$$\begin{aligned} \max_{\mathbf{D} \in \mathbb{R}^{r \times p}} \text{Tr}(\mathbf{D}^\top (\mathbf{I} - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^\top), \\ \text{ s.t. } \mathbf{D}^\top \mathbf{D} = \mathbf{I}_p, \mathbf{A}^\top \mathbf{D} = \mathbf{0}. \end{aligned} \quad (2.40)$$

Together with the fact

$$\begin{aligned} \mathbf{D}^\top \mathbf{G} \mathbf{V}_D^\top &= \mathbf{D}^\top \mathcal{P}_A \mathbf{G} \mathbf{V}_D^\top + \mathbf{D}^\top (\mathbf{I} - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^\top \\ &= \mathbf{D}^\top (\mathbf{I} - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^\top \end{aligned} \quad (2.41)$$

The last equality in (2.41) holds when the constraint $\mathbf{A}^\top \mathbf{D} = \mathbf{0}$ is satisfied. ■

Therefore, each iteration in the proposed alternative iteration scheme is very simple. There is no need to use any iterative scheme for solving any minimization problem when doing the sparse coding and dictionary updating. The sparse coding is done via a hard thresholding operation and the dictionary updating is done via a single SVD. See Algorithm 3 for the complete description of the algorithm.

Algorithm 3 Extended orthogonal dictionary learning

Input: image patches \mathbf{G} , input orthogonal atoms \mathbf{A} **Output:** learned dictionary \mathbf{D} **Main procedure:**

1. Set the initial guess $\mathbf{D}^{(0)}$.
2. For $k = 0, 1, \dots, K$,
 - (a) $\mathbf{V}_D^{(k)} := T_\lambda(\mathbf{D}^{(k)\top} \mathbf{G})$;
 - (b) run the SVD for the matrix

$$(\mathbf{I}_n - \mathcal{P}_A) \mathbf{G} \mathbf{V}_D^{(k)\top} = \mathbf{P} \mathbf{\Sigma} \mathbf{Q}^\top;$$

- (c) $\mathbf{D}^{(k+1)} := \mathbf{P} \mathbf{Q}^\top$.
 3. $\mathbf{D} := \mathbf{D}^{(K+1)}$.
-

2.6.3 Complexity analysis of Algorithm 3

In this section, we give a detailed analysis on the computational complexity of Algorithm 3 for sparsity-based orthogonal dictionary learning. Let m denotes the number of training patches in \mathbf{G} and consider the worst scenario in which no pre-defined atoms are provided, i.e. $\mathbf{D} \in \mathbb{R}^{n \times n}$.

The sparse coding of Alg. 3 uses $2mn^2$ operations to obtain the matrix product $\mathbf{D}^\top \mathbf{G}$ and mn^2 operations in hard thresholding. Let K denote the average number of non-zero entries in each column of \mathbf{V} . For dictionary update of Alg. 3, the number of operations required to calculate the multiplications $\hat{\mathbf{G}} \mathbf{V}^\top$ is $2mnK$. The standard algorithm to obtain the singular value decomposition of $\hat{\mathbf{G}} \mathbf{V}^\top \in \mathbb{R}^{n \times n}$ takes $21n^3$ operations [40]. So, the total number of operations in one iteration of Alg. 3 is

$$T = 3mn^2 + 2mnK + 21n^3 \quad (2.42)$$

The K-SVD method [34] is very computationally demanding. The OMP used for sparse coding is known to be slow. The dictionary update of the K-SVD method need to call SVD operators for $4n$ times. Thus, a fast approximate K-SVD method is developed in [74] which use batch-OMP for sparse coding and replacing SVD by matrix-vector multiplication. The complexity analysis of the approximate K-SVD method (the dimension of dictionary is set

	Sparse Coding	Dictionary learning	Total
Approx. K-SVD [74]	$m(8n^2 + 4K^2n + 12Kn) + mK^3 + 16n^3$	$20mKn + 64n^3$	$m(8n^2 + 4nK^2 + 32Kn + K^3) + 80n^3$
Alg. 3	$3mn^2$	$2mKn + 21n^3$	$m(3n^2 + 2Kn) + 21n^3$

Table 2.2: Complexity analysis for one iteration

$4n$ by default), together with ours are listed in table 2.2. Clearly, Algorithm 3 requires far less operations. The computational efficiency in real applications will be further investigated in the section of experiments.

2.6.4 Applications in image restoration

The sparsity-based online orthogonal dictionary learning Algorithm 3 is very simple to implement and also very computationally efficient. To evaluate its performance in image restoration in terms of recovery quality and computational efficiency, we applied Algorithm 3 on two sample image restoration tasks: image denoising and image inpainting.

Image denoising. Algorithm 3 can be directly applied on de-noising by taking the noisy image as the input image for training. It is known in signal processing that most noise are in the high-pass channels. Thus, we fix a low-pass filter in the dictionary and only learn $n - 1$ high-pass filters from the input image. That is, we define $\mathbf{A} = [\boldsymbol{\alpha}_0] \in \mathbb{R}^{n \times 1}$, where

$$\boldsymbol{\alpha}_0 = n^{-1/2}[1, 1, \dots, 1]^\top.$$

Clearly, the orthogonal constraint $\boldsymbol{\alpha}_0^\top \mathbf{D} = 0$ on \mathbf{D} ensures that all atoms in $\mathbf{D} \in \mathbb{R}^{n \times n-1}$ are high-pass filters. After generating the training matrix \mathbf{G} by randomly sampling the image patches of size $\sqrt{n} \times \sqrt{n}$ from the noisy image, the dictionary D is learned from Algorithm 3. Then the de-noised image is reconstructed from the de-noised patch matrix $\hat{\mathbf{D}}T_{\lambda_1}(\hat{\mathbf{D}}^\top \mathbf{G})$ by averaging the overlapping pixels, where $\hat{\mathbf{D}} = [\boldsymbol{\alpha}_0, \mathbf{D}]$. For computational efficiency, the training patches are uniformly selected from the image at random. See Algorithm 3 for the

outline of the algorithm.

Image inpainting. Image in-painting is about recovering the missing values of image pixels or removing unwanted content from the image, which can be formulated as solving the following under-determined linear inverse problem:

$$f(k) = g(k) + \varepsilon, \quad k \in \Lambda^c.$$

where g denote the image for recovery, Λ denotes the region for in-painting and Λ^c denotes its complement, and ε denotes noise. Using a dictionary \mathbf{D} generated from wavelet frame filters, Cai *et al.* [16] proposed the following iteration scheme for in painting f :

$$\mathbf{G}^{(k+1)} = (I - P_\Lambda)\mathbf{F} + P_\Lambda\mathbf{D}^{-1}(T_\lambda\mathbf{D}\mathbf{G}^{(k)}), \quad (2.43)$$

where P_Λ is the diagonal projection matrix whose diagonal element equals to 1 if in Λ and 0 otherwise, $\mathbf{G}^{(k)}$ are image patch matrices from $g^{(k)}$ and f respectively. In our implementation, we use the same iteration scheme. Different from image denoising, during each iteration of Algorithm 3, we use the newest estimate $g^{(k)}$ to generate the training patch matrix. See Algorithm 5 for details.

Algorithm 4 Denoising via orthogonal dictionary learning

Input: noisy image g

Output: denoised image g^*

Main procedure:

1. Initialization.
 - (a) synthesizing image patch matrix \mathbf{G} from g ;
 - (b) defining $\mathbf{A} = [\mathbf{a}_0]$ for some low-pass filter \mathbf{a}_0 .
 2. Learning a dictionary \mathbf{D}^* using Algorithm 3 with input \mathbf{G} and $\mathbf{A} = [\mathbf{a}_0]$.
 3. De-noising patch matrix $\mathbf{G}^* := \hat{\mathbf{D}}T_{\lambda_1}(\hat{\mathbf{D}}^\top \mathbf{G})$ with $\hat{\mathbf{D}} = [\mathbf{A}, \mathbf{D}^*]$.
 4. Synthesizing the denoised image g^* from \mathbf{G}^* by averaging the overlapping pixels.
-

Algorithm 5 Inpainting via orthogonal dictionary learning**Input:** image g and inpainting region Λ **Output:** inpainted image g^* **Main procedure:**

1. Initialization.
 - (a) initializing an in-painted image $g^{(0)}$ by interpolation;
 - (b) synthesizing patch matrix $\mathbf{G}^{(0)}$ from $g^{(0)}$; and defining $\mathbf{A} = [\mathbf{a}_0]$.
2. For $k = 0, 1, \dots, K$,
 - (a) learning a dictionary $\mathbf{D}^{(k)}$ using one iteration of Algorithm 3 with input $\mathbf{G}^{(k)}$ and $\mathbf{A} = [\mathbf{a}_0]$;
 - (b) synthesizing the image $h^{(k+1)}$ from the denoised patch matrix $\mathbf{G}^* := \hat{\mathbf{D}}T_{\lambda_1}(\hat{\mathbf{D}}^\top \mathbf{G})$;
 - (c) defining $g^{(k+1)} := (I - P_\Lambda)(g) + P_\Lambda h^{(k+1)}$.
3. $g^* := g^{(K+1)}$.

2.6.5 Experiments

In this section, we evaluate the performance of the proposed orthogonal dictionary learning on image denoising and image in-painting. The experiments are conducted in MATLAB R2011b (64bit) Linux version on a PC workstation with an INTEL CPU (2.4GHZ) and 48G memory. The initial dictionary is generated by the local DCT transform: either 8×8 or 16×16 . The image patches for training are uniformly selected from the input image at random. For image size 512×512 and patch size 16×16 , about $4 * 10^4$ patches are used for training.

Computational efficiency. Under the same software and hardware environment, Algorithm 3 is compared to the widely used over-complete dictionary learning: the K-SVD algorithm [34] and its fast version, the approximated K-SVD algorithm [74] with the implementations from the original authors¹. Table 2.6.5 listed the detailed running time of each module in K-SVD method, approximated K-SVD method and Algorithm 3. For each iteration, clearly Algorithm 3 is much faster than both the K-SVD method and the approximate K-SVD method.

The shorter running time for each iteration does not imply the algorithm run faster,

¹<http://www.cs.technion.ac.il/~ronrubin/software.html>

implementation	module	8×8	16×16
K-SVD [34]	dictionary update	8.60	24.87
	sparse coding	1.19	2.18
Approx. K-SVD [74]	dictionary update	0.56	1.45
	sparse coding	1.44	3.50
Algorithm 3	dictionary update	0.02	0.15
	sparse coding	0.04	0.18

Table 2.3: Running time (seconds) breakdown on one iteration of the K-SVD method, approximated K-SVD method and the implementation of Algorithm 3 with patch size 8×8 and 16×16 .

Patch size \ method	running time (sec.) vs. PSNR (dB)	K-SVD	Approx. K-SVD	Alg. 3
8×8	time	202.75	98.35	2.02
	PSNR	28.51	28.61	28.44
16×16	time	484.25	206.49	12.11
	PSNR	27.86	27.84	28.93

Table 2.4: Running time of the K-SVD method, approximated K-SVD method with 15 iterations and Algorithm 3 with 30 iterations.

as it might has slow convergence. Thus, we conduct another test on the overall running time when applying the three methods on image de-noising. The tested image is the image "Barbara" of 512×512 in the presence of i.i.d. Gaussian noise with s.t.d. $\sigma = 30$. Totally 15 iterations are used in the K-SVD method and the approximate K-SVD method as more iterations do no improve the PSNR value anymore. Table 2.6.5 listed the total running time of the two K-SVD methods and Alg. 3. While all three methods have comparable PSNR values, our method is much faster that the other two.

Image denoising. Algorithm 4 for image denoising is evaluated on several tested images shown in Fig. 2.4 with different noise levels. In our experiments, we set $\lambda = 3.5\sigma$ and $\lambda_1 = 2.7\lambda$ as the thresholding value for the dictionary learning process. Our results are compared against two fixed transform based thresholding methods: linear spline framelet [30] and 8×8 DCT, the PCA-based non-local hierarchical method [31] and the K-SVD denoising method [34] with patch size of both 8×8 and 16×16 . See Table 2.5 for the list

of PSNR values of the results and Fig.2.3 for a visual illustration.

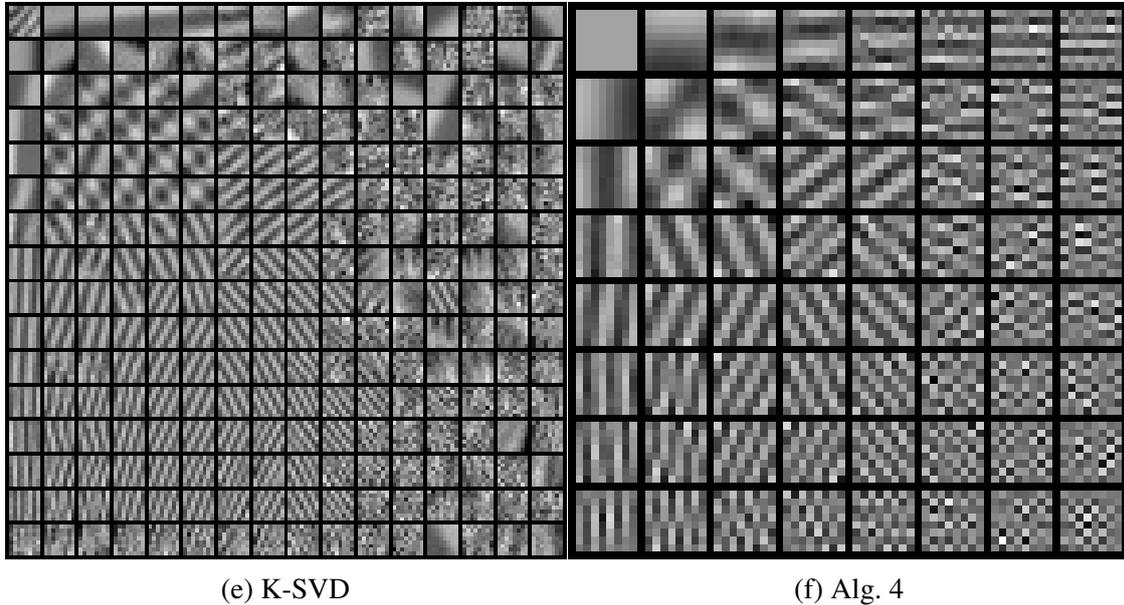


Figure 2.3: The dictionaries learned from the image "Barbara" with noise level $\sigma = 20$ using the K-SVD method and Algorithm 3. The atom size is 8×8 .

Image Inpainting. Algorithm 5 is only tested on two sample image in-painting problems. The first example is the text removal from the image ([11]). The second example is to filling missing pixels in the image ([78]). In the first example, the results are compared to the classic in-painting method [11], and two dictionary learning based methods derived from the K-SVD method ([78]). The main difference between two dictionary learning methods lies in the choice of sparsity promoting functional: one uses the ℓ_1 norm and the other one uses MC penalty. The results are shown in Fig. 2.5, together with two zoom-in regions shown in the top-left and top-right corner of the image for easier inspection. It is seen that the result from Algorithm 5 has less artifacts than others. In the second example, the values of 50% of image pixels are missing at random. Algorithm 5 and two dictionary learning methods [78] are applied to recover the missing pixel values. See Figure. 2.6 for the visual illustration of the results. It is seen that Algorithm 5 outperformed the methods derived from the K-SVD methods.

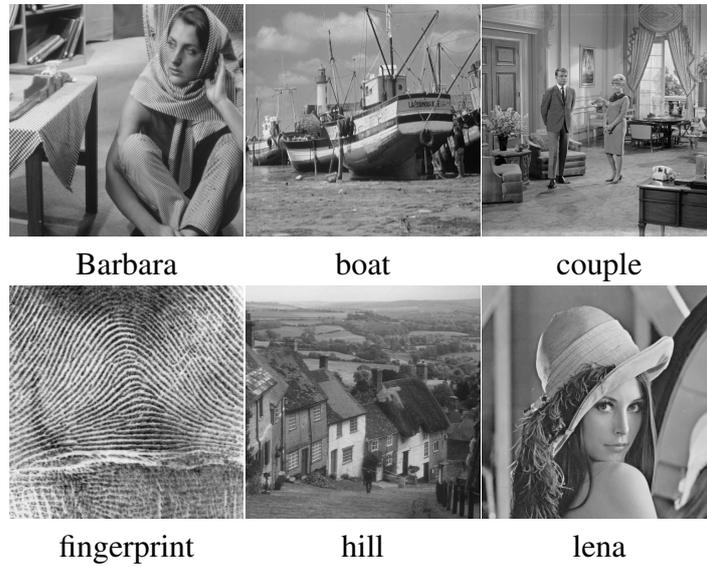


Figure 2.4: Test images.

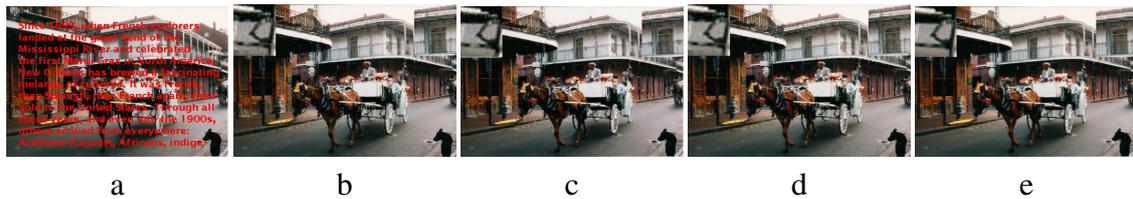
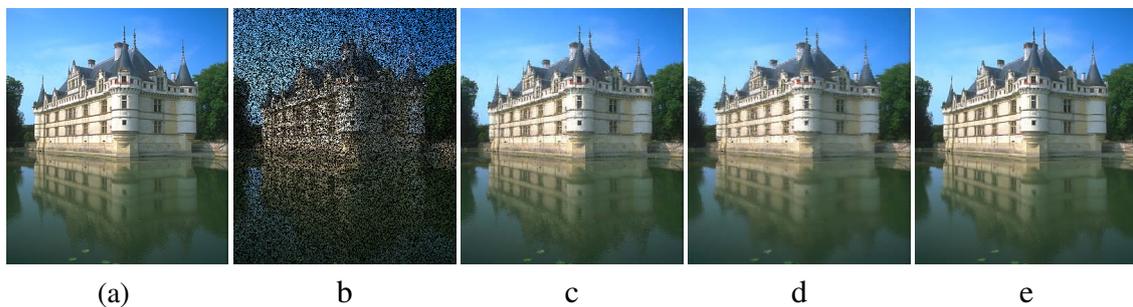
Figure 2.5: Comparison of text removal. (a) image with overlapped texts; (b-e) correspond to the results from [11], two over-complete dictionary learning method with ℓ_1 norm sparsity penalty and MC penalty ([78]), and Algorithm 5.Figure 2.6: Image inpainting with 50% random missing pixels. (a) Original image; (b) corrupted image; (c-e) the results from from two over-complete dictionary learning method with ℓ_1 norm sparsity penalty and MC penalty ([78]), and Algorithm 5.

Image	Babara					Boat				
σ	10	20	30	40	50	10	20	30	40	50
DCT; 8×8	34.13	30.24	27.96	26.41	25.15	33.49	30.01	27.96	26.51	25.42
linear framelet	32.08	27.98	25.76	24.25	23.18	32.80	29.36	27.25	25.74	24.48
hierarchical PCA	34.52	30.85	28.92	27.38	26.00	33.65	30.23	28.24	26.75	25.57
K-SVD; 8×8	34.48	30.86	28.57	26.92	25.47	33.67	30.41	28.44	27.04	25.94
K-SVD; 16×16	34.09	30.27	27.81	26.09	24.78	33.06	29.48	27.28	25.89	24.86
Alg.4; 8×8	34.34	30.58	28.44	26.94	25.75	33.64	30.33	28.38	27.00	25.95
Alg.4; 16×16	34.56	31.00	28.94	27.44	26.31	33.51	30.26	28.36	27.00	25.99
Image	Fingerprint					Hill				
σ	10	20	30	40	50	10	20	30	40	50
DCT; 8×8	32.25	28.29	26.08	24.49	23.27	33.24	30.02	28.26	26.94	25.91
linear framelet	30.44	26.49	24.26	22.70	21.45	32.69	29.46	27.58	26.12	24.96
hierarchical PCA	32.33	28.38	26.31	24.83	23.62	33.41	30.20	28.59	27.34	26.31
K-SVD; 8×8	32.40	28.47	26.29	24.70	23.19	33.38	30.20	28.39	27.15	26.28
K-SVD; 16×16	31.88	27.69	25.26	23.49	22.22	32.81	29.38	27.38	25.99	24.94
Alg.4; 8×8	32.24	28.33	26.17	24.68	23.47	33.27	30.21	28.51	27.31	26.43
Alg.4; 16×16	32.25	28.35	26.25	24.86	23.81	33.18	30.19	28.54	27.40	26.54
Image	Couple					Lena				
σ	10	20	30	40	50	10	20	30	40	50
DCT; 8×8	33.41	29.86	27.79	26.33	25.25	35.29	31.86	29.74	28.17	26.90
linear framelet	33.06	29.42	27.24	25.60	24.39	34.22	30.69	28.52	26.83	25.50
hierarchical PCA	33.56	29.95	27.86	26.41	25.32	35.39	32.25	30.47	29.03	27.70
K-SVD; 8×8	33.55	30.01	27.90	26.40	25.31	35.56	32.45	30.49	29.03	27.82
K-SVD; 16×16	32.87	29.10	26.85	25.19	24.08	35.02	31.71	29.57	28.06	26.78
Alg.4; 8×8	33.57	30.04	28.06	26.62	25.57	35.52	32.31	30.32	28.84	27.66
Alg.4; 16×16	33.40	29.97	28.05	26.70	25.71	35.52	32.40	30.49	29.09	27.95

Table 2.5: PSNR values of the denoised results

2.6.6 Discussion and conclusion

In this subsection, we proposed an extended orthogonal dictionary learning for image restoration, as an replacement of the widely used K-SVD method. The performance of the proposed orthogonal dictionary learning method is comparable to the K-SVD method, but it runs much faster than the K-SVD method. Such a significant improvement on the speed could be very important to many image restoration application when dealing with image of very large size or processing many images. In future, we would like to study how to effectively combine the non-local scheme and the proposed orthogonal dictionary learning method to develop better image restoration methods.

Chapter 3

Redundant dictionary learning for image restoration and recognition

3.1 Introduction

In recent years, sparse coding has been widely used in many applications [80], e.g. image recovery, machine learning, and recognition. The goal of sparse coding is to represent given data by the linear combination of few elements taken from a set learned from given training samples. Such a set is called *dictionary* and the elements of the set are called *atoms*. Let $\mathbf{D} = \{\mathbf{d}_k\}_{k=1}^m \subset \mathbb{R}^n$ denote an over-complete dictionary composed of $m (\geq n)$ atoms. Then, for a signal $\mathbf{y} \in \mathbb{R}^n$, its *sparse approximation* over \mathbf{D} is about finding a linear expansion $\mathbf{D}\mathbf{c} = \sum_{k=1}^m c_k \mathbf{d}_k$ using the fewest elements that approximates \mathbf{y} with an error bound ε . The sparse approximation for an input signal can be formulated as the following optimization problem:

$$\min_{\mathbf{c} \in \mathbb{R}^m} \|\mathbf{c}\|_0, \quad \text{subject to } \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 \leq \varepsilon. \quad (3.1)$$

The problem (3.1) is a challenging NP-hard problem and only sub-optimal solutions can be found in polynomial time. Most existing algorithms either use greedy algorithms to iteratively select locally optimal solutions (e.g. orthogonal matching pursuit (OMP) [81]), or replace the non-convex ℓ_0 norm by its convex relaxation ℓ_1 norm (e.g. basis pursuit [24]).

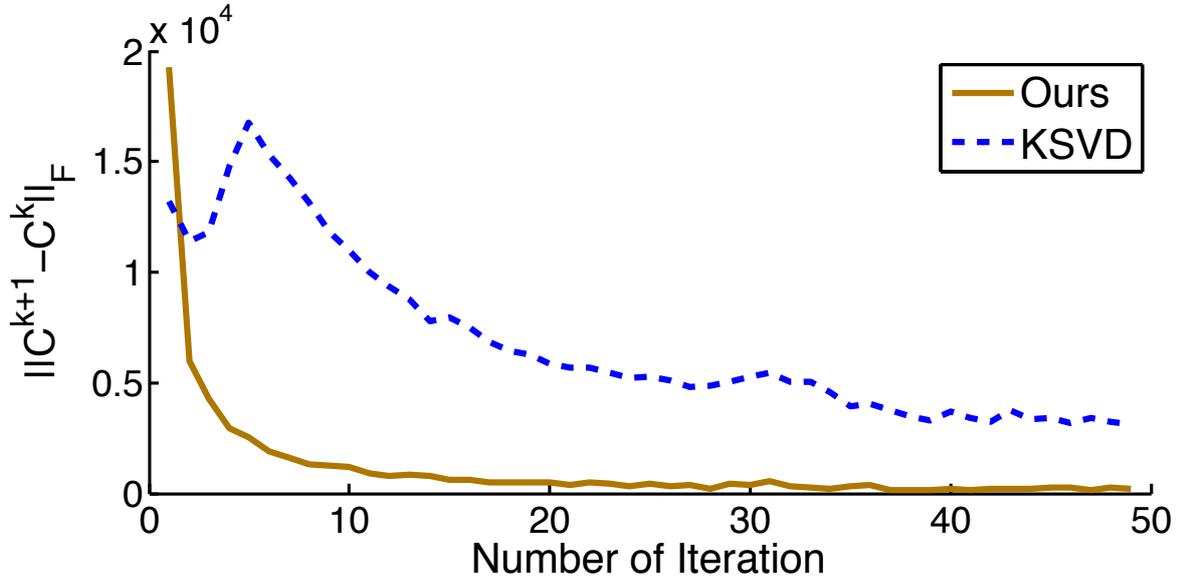


Figure 3.1: Convergence behavior: the norm of the increments of the coefficient sequence \mathbf{C}^k generated by the K-SVD method and the proposed method.

The dictionary for sparse approximation is usually learned from given training sample to maximize the efficiency of sparse approximation in terms of sparsity degree. More concretely, given a training set of p signals $\mathbf{Y} := \{\mathbf{y}_k\}_{k=1}^p \subset \mathbb{R}^n$, the *dictionary learning* is often formulated as the following minimization problem:

$$\min_{\mathbf{D}, \{\mathbf{c}_k\}_{k=1}^p} \sum_{k=1}^p \frac{1}{2} \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \lambda \|\mathbf{c}_k\|_0, \quad (3.2)$$

subject to $\|\mathbf{d}_k\|_2 = 1, k = 1, 2, \dots, m$, where $\mathbf{c} = \{\mathbf{c}_k\}_{k=1}^p$ denotes the sparse coefficients of training set \mathbf{Y} and \mathbf{D} denotes the learned dictionary.

3.1.1 Motivation

The minimization problem (3.2) is a non-convex problem whose non-convexity comes from two sources: the sparsity-prompting functional ℓ_0 norm and the bi-linearity between the dictionary \mathbf{D} and the codes $\{\mathbf{c}_k\}_{k=1}^p$. Most existing approaches (e.g. [1, 44, 56, 57]) take an alternating iteration between two modules: sparse approximation for updating $\{\mathbf{c}_k\}_{k=1}^p$ and

dictionary learning for updating dictionary D .

Despite the success of these alternating iterative methods in practice, to best of our knowledge, none of them established the global convergence property, i.e., the whole sequence generated by the method converges to a stationary point of (3.2). These schemes can only guarantee that the functional values are decreasing over the iterations, and thus there exists a convergent sub-sequence as the sequence is always bounded. Indeed, the sequence generated by the popular K-SVD method [1] is not convergent as its increments do not decrease to zero. See Fig. 3.1 for an illustration. The global convergence property is not only of great theoretical importance, but also likely to be more efficient in practical computation as many intermediate results are useless for a method without global convergence property.

3.1.2 Main contributions

In this chapter, we proposed an alternating proximal linearized method for solving (3.2). The main contribution of the proposed algorithm lies in its theoretical contribution to the open question regarding the convergence property of ℓ_0 norm based dictionary learning methods. In this chapter, we showed that the whole sequence generated by the proposed method converges to a stationary point of (3.2). Moreover, we also showed that the convergence rate of the proposed algorithm is at least sub-linear. To the best of our knowledge, this is the first algorithm with global convergence for solving ℓ_0 norm based dictionary learning problems.

The proposed method can also be used to solve other variations of (3.2) with small modifications, e.g. the ones used in discriminative K-SVD based recognition methods [45, 95]. Compared to many existing methods including the K-SVD method, the proposed method also has its advantage on computational efficiency. The experiments showed that the implementation of the proposed algorithm has comparable performance to the K-SVD method in two applications: image de-noising and face recognition, but is noticeably faster.

3.2 Related work

In this section, we give a brief review on dictionary learning and related applications. Based on the used sparsity prompting functional, the existing dictionary learning methods can be classified into the following three categories.

3.2.1 ℓ_0 norm based methods

The most popular ℓ_0 norm based dictionary learning method is the K-SVD method [1] which used the model (3.2) for image denoising. Using many image patches from the input image as the training set, the K-SVD method alternatively iterates between sparse approximation and dictionary updating. The sparse approximation is based on the OMP method and the dictionary is estimated via sequential column-wise SVD updates.

The K-SVD method showed good performance in image de-noising and is also used in face/object recognition by adding some additional fidelity term in (3.2). For example, the so-called discriminative K-SVD method in [45, 95] seeks the sparse code that minimizes both reconstruction error and classification error as follows,

$$\begin{aligned}
 \min_{\mathbf{D}, \mathbf{W}, \{\mathbf{c}_k\}_{k=1}^p} & \sum_{k=1}^p \frac{1}{2} \|\mathbf{y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \sum_{k=1}^p \frac{\beta}{2} \|\boldsymbol{\ell}_k - \mathbf{W}\mathbf{c}_k\|_2^2, \\
 \text{s.t.} & \quad \|\mathbf{c}_j\|_0 \leq \tau, j = 1, 2, \dots, p, \\
 & \quad \|\mathbf{w}_k\|_2 \leq 1, \|\mathbf{d}_k\|_2 \leq 1, k = 1, 2, \dots, m,
 \end{aligned} \tag{3.3}$$

where $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_m]$ denotes the linear classifier learned from the training set and $\boldsymbol{\ell}_k$ denotes the binary encoded class label of the k -th sample. Both dictionary update and sparse approximation is done via calling the K-SVD method. Also using the ℓ_0 norm related optimization model, a fast method is proposed in [19] for learning a tight frame, which has closed form solutions for both sparse approximation and dictionary update.

3.2.2 Convex relaxation methods

As a convex relaxation of ℓ_0 norm, the ℓ_1 norm has been used in many dictionary learning methods to improve the computational feasibility and efficiency of sparse coding; see e.g. [44, 56, 57, 90]. All these methods also take an alternating scheme between sparse coding and dictionary updating. In the stage of sparse approximation which requires solving a ℓ_1 norm related minimization problem, various methods have been used in different applications, including the accelerated gradient method [84] or fast iterative shrinkage thresholding algorithm [10] in [44]; the fixed point method [42] in [56]. In the stage of dictionary update, the atoms are either updated one by one or are simultaneously updated. One-by-one atom updating is implemented in [44, 57] as it has closed form solutions. The projection gradient method is used in [56] to update the whole dictionary together. The convergence analysis is provided for the proximal method proposed in [90] for the ℓ_1 norm based dictionary learning.

3.2.3 Non-convex relaxation methods

As shown in [43, 94], the ℓ_1 norm penalty tends to have biased estimation for large coefficients and sometimes results in over-penalization. Thus, several non-convex relaxations of ℓ_0 norm are proposed for better accuracy in sparse coding. For example, the non-convex minimax concave (MC) penalty [94] is used in [78] for sparse dictionary learning. For other non-convex relaxations, e.g. smoothly clipped absolute deviation [43] and log penalty [37], the proximal methods have been proposed in [67, 79] to solve the minimization problems with these non-convex regularization terms. The convergence property of these methods is limited to the subsequence convergence.

3.3 Algorithm and convergence analysis

3.3.1 Problem formulation

The original model (3.2) does not impose any constraint on the code $\{\mathbf{c}_k\}$. When a dictionary with high redundancy is adopted, some elements of the sparse coefficient vector could have unusual large values, which in general are not correct. Thus, we slightly modify the model (3.2) by adding a bound constraint on $\{\mathbf{c}_k\}$. Then, the minimization model considered in this chapter is defined as follows,

$$\begin{aligned} \min_{\mathbf{D}, \{\mathbf{c}_k\}_{k=1}^p} \quad & \frac{1}{2} \sum_{k=1}^p \|\mathbf{Y}_k - \mathbf{D}\mathbf{c}_k\|_2^2 + \lambda \|\mathbf{c}_k\|_0 \\ \text{s.t.} \quad & \|\mathbf{d}_k\|_2 = 1, 1 \leq k \leq m; \|\mathbf{c}_k\|_\infty < M, 1 \leq k \leq p, \end{aligned} \quad (3.4)$$

where M is a pre-defined upper-bound for all elements of $\{\mathbf{c}_k\}$. It is noted that the bound constraint on $\{\mathbf{c}_k\}_k$ is mainly for improving the stability of the model (3.2), which can be set to a sufficiently large value to avoid any negative impact on the accuracy of the coefficients. For the simplicity of discussion, let $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_p]$ denote the training sample matrix and let $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_p]$ denote the coefficient matrix. Let $\mathcal{X} = \{\mathbf{D} \in \mathbb{R}^{n \times m} : \|\mathbf{d}_k\|_2 = 1, 1 \leq k \leq m\}$ denote the feasible set for the dictionary \mathbf{D} , and let $\mathcal{C} = \{\mathbf{C} \in \mathbb{R}^{m \times p} : \|\mathbf{c}_k\|_\infty \leq M, 1 \leq k \leq p\}$ denote the feasible set for the coefficient matrix \mathbf{C} . Then the model (3.4) can be expressed in the following compact form:

$$\min_{\mathbf{D} \in \mathbb{R}^{n \times m}, \mathbf{C} \in \mathbb{R}^{m \times p}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_0, \quad \text{s.t.} \quad \mathbf{D} \in \mathcal{X}, \mathbf{C} \in \mathcal{C}. \quad (3.5)$$

In the next, we will present an alternating proximal method for solving (3.5), as well as the convergence analysis.

3.3.2 Alternating proximal method

The proposed algorithm is based on the proximal method [15] for solving the following non-convex problem:

$$\min_{\mathbf{x}, \mathbf{y}} H(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}) + Q(\mathbf{x}, \mathbf{y}) + G(\mathbf{y}), \quad (3.6)$$

where $F(\mathbf{x})$, $G(\mathbf{y})$ are proper lower semi-continuous functions, and $Q(\mathbf{x}, \mathbf{y})$ is a smooth function with Lipschitz gradient on any bounded set. The proximal method proposed in [15] updates the estimate of (\mathbf{x}, \mathbf{y}) via solving the following proximal problems:

$$\begin{aligned} \mathbf{x}^{k+1} &\in \arg \min_{\mathbf{x}} F(\mathbf{x}) + \langle \mathbf{x} - \mathbf{x}^k, \nabla_{\mathbf{x}} Q(\mathbf{x}^k, \mathbf{y}^k) \rangle + \frac{t_k^1}{2} \|\mathbf{x} - \mathbf{x}^k\|_2^2; \\ \mathbf{y}^{k+1} &\in \arg \min_{\mathbf{y}} G(\mathbf{y}) + \langle \mathbf{y} - \mathbf{y}^k, \nabla_{\mathbf{y}} Q(\mathbf{x}^{k+1}, \mathbf{y}^k) \rangle + \frac{t_k^2}{2} \|\mathbf{y} - \mathbf{y}^k\|_2^2, \end{aligned} \quad (3.7)$$

where t_k^1 and t_k^2 are two appropriately chosen step sizes. Using the so-called *proximal operator* [71] defined as

$$\text{Prox}_t^F(\mathbf{x}) := \arg \min_{\mathbf{u}} F(\mathbf{u}) + \frac{t}{2} \|\mathbf{u} - \mathbf{x}\|_2^2, \quad (3.8)$$

the minimizations (3.7) are equivalent to the following proximal problem:

$$\begin{aligned} \mathbf{x}^{k+1} &\in \text{Prox}_{t_k^1}^F\left(\mathbf{x}^k - \frac{1}{t_k^1} \nabla Q(\mathbf{x}^k, \mathbf{y}^k)\right), \\ \mathbf{y}^{k+1} &\in \text{Prox}_{t_k^2}^G\left(\mathbf{y}^k - \frac{1}{t_k^2} \nabla Q(\mathbf{x}^{k+1}, \mathbf{y}^k)\right). \end{aligned} \quad (3.9)$$

The minimization problem (3.5) can be expressed in the form of (3.6) by setting

$$\begin{cases} F(\mathbf{C}) = \|\mathbf{C}\|_0 + I_{\mathcal{C}}(\mathbf{C}); \\ Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{DC}\|_F^2; \\ G(\mathbf{D}) = I_{\mathcal{D}}(\mathbf{D}), \end{cases} \quad (3.10)$$

where $I_{\mathcal{X}}(\mathbf{D})$ denotes the indicator function of \mathbf{D} that satisfies $I_{\mathcal{X}}(\mathbf{D}) = 0$ if $\mathbf{D} \in \mathcal{X}$ and $+\infty$ otherwise. Then using proximal operators, we propose the following alternating iterative

scheme for solving (3.5): let $\mathbf{D}^{(0)}$ be the initial dictionary, then for $\ell = 0, 1, \dots$,

1. **sparse approximation:** given the dictionary $\mathbf{D}^{(\ell)}$, find the sparse code $\mathbf{C}^{(\ell)}$ that satisfies

$$\mathbf{C}^{(\ell)} \in \text{Prox}_{\lambda_\ell/\lambda}^F(\mathbf{C}^{(\ell-1)} - \frac{1}{\lambda_\ell} \nabla_{\mathbf{C}} Q(\mathbf{C}^{(\ell-1)}, \mathbf{D}^{(\ell)})), \quad (3.11)$$

where λ_ℓ is an estimated step size (more on this later).

2. **dictionary update:** given the sparse code $\mathbf{C}^{(\ell)}$, update the dictionary $\mathbf{D}_k^{(\ell)} = \{\mathbf{d}_k^{(\ell+1)}\}_{k=1}^m$ atom by atom:

$$\mathbf{d}_k^{(\ell+1)} \in \text{Prox}_{\mu_k^\ell}^{G(\hat{\mathbf{D}}_k^{(\ell)})}(\mathbf{D}_k^{(\ell)} - \frac{1}{\mu_k^\ell} \nabla_{\mathbf{D}_k} Q(\mathbf{C}^{(\ell)}, \tilde{\mathbf{D}}_k^{(\ell)})), \quad (3.12)$$

where

$$\begin{cases} \hat{\mathbf{D}}_k^{(\ell)} = [\mathbf{d}_1^{(\ell+1)}, \dots, \mathbf{d}_{k-1}^{(\ell+1)}, \mathbf{d}_k, \mathbf{d}_{k+1}^{(\ell)}, \dots, \mathbf{d}_n^{(\ell)}]; \\ \tilde{\mathbf{D}}_k^{(\ell)} = [\mathbf{d}_1^{(\ell+1)}, \dots, \mathbf{d}_{k-1}^{(\ell+1)}, \mathbf{d}_k^{(\ell)}, \mathbf{d}_{k+1}^{(\ell)}, \dots, \mathbf{d}_n^{(\ell)}], \end{cases}$$

and μ_k^ℓ is a step size need to be estimated.

Each iteration above requires solving two optimization problems (3.11) and (3.12). In the next, we show that both have closed form solutions. Define

$$\begin{cases} \mathbf{T}_{\mathbf{C}}^{(\ell)} = \mathbf{C}^{(\ell-1)} - \frac{1}{\lambda_\ell} \nabla_{\mathbf{C}} Q(\mathbf{C}^{(\ell-1)}, \mathbf{D}^{(\ell)}); \\ \mathbf{s}_k^{(\ell)} = \mathbf{d}_k^{(\ell)} - \frac{1}{\mu_k^\ell} \nabla_{\mathbf{d}_k} Q(\mathbf{C}^{(\ell)}, \tilde{\mathbf{D}}_k^{(\ell)}). \end{cases}$$

Then by a direct calculation, two optimization problems (3.11) and (3.12) are equivalent to

$$\begin{cases} \mathbf{C}^{(\ell)} \in \arg \min_{\mathbf{C} \in \mathcal{C}} \frac{\lambda_\ell}{2\lambda} \|\mathbf{C} - \mathbf{T}_{\mathbf{C}}^{(\ell)}\|_F^2 + \|\mathbf{C}\|_0, \\ \mathbf{d}_k^{(\ell)} \in \arg \min_{\|\mathbf{d}_k\|_2=1} \frac{1}{2} \|\mathbf{d}_k - \mathbf{s}_k^{(\ell)}\|_F^2, 1 \leq k \leq m. \end{cases} \quad (3.13)$$

Proposition 3.3.1. *Suppose that M is chosen such that $M > \sqrt{2\lambda/\lambda_\ell}$, two minimization*

problems in (3.13) have the closed form solutions given by

$$\begin{cases} \mathbf{C}^{(\ell)} = \min\{T_{\sqrt{2\lambda/\lambda_\ell}}(\mathbf{T}_\mathbf{C}^{(\ell)}), M\}; \\ \mathbf{d}_k^{(\ell)} = \mathbf{s}_k^{(\ell)} / \|\mathbf{s}_k^{(\ell)}\|_2, 1 \leq k \leq m. \end{cases} \quad (3.14)$$

Proof. The proof of the solution to the second problem in (3.13) is trivial. The first is easy to obtain as it can be decomposed into the summation of independent minimization problems with respect to each variable. \blacksquare

Setting of step sizes. There are two step sizes, λ_ℓ in (3.11) and μ_ℓ^k in (3.12), need to be set during the iteration. The step size λ_ℓ can be chosen as $\lambda_\ell = \max\{\rho L(\mathbf{D}^{(\ell)}), \underline{\ell}\}$ where $\underline{\ell} > 0$ is a constant, $\rho > 1$ and $L(\mathbf{D}^{(\ell)})$ satisfies

$$\|\nabla_{\mathbf{C}}(Q(\mathbf{C}_1, \mathbf{D}^{(\ell)})) - \nabla_{\mathbf{C}}(Q(\mathbf{C}_2, \mathbf{D}^{(\ell)}))\| \leq L(\mathbf{D}^{(\ell)}) \|\mathbf{C}_1 - \mathbf{C}_2\|.$$

The step size μ_ℓ^k can be chosen as $\mu_\ell^k = \max\{\rho L(\mathbf{Z}_k^{(\ell)}), \underline{\ell}\}$ where $\mathbf{Z}_k^{(\ell)} = (\mathbf{C}^{(\ell)}, \mathbf{D}^{(\ell)}) \setminus \mathbf{d}_k^{(\ell)}$, $\underline{\ell} > 0$, $\rho > 1$ and $L(\mathbf{Z}_k^{(\ell)})$ satisfies

$$\|\nabla_{\mathbf{d}_k}(Q(\mathbf{Z}_k^{(\ell)}, \mathbf{d}_k^1) - \nabla_{\mathbf{d}_k}(Q(\mathbf{Z}_k^{(\ell)}, \mathbf{d}_k^2))\| \leq L(\mathbf{Z}_k^{(\ell)}) \|\mathbf{d}_k^1 - \mathbf{d}_k^2\|,$$

for any pair $\mathbf{d}_k^1, \mathbf{d}_k^2$. Consequently, we can choose $L(\mathbf{D}^{(\ell)}) = \|\mathbf{D}^{(\ell)\top} \mathbf{D}^{(\ell)}\|_F$ and $L(\mathbf{Z}_k^{(\ell)}) = [\mathbf{C}^{(\ell)} \mathbf{C}^{(\ell)\top}]_{k,k}, \forall k = 1, 2, \dots, m$. It can be seen that the sequence $L(\mathbf{D}^{(\ell)})$ is a bounded sequence since each column in \mathbf{D} is of unit norm. Moreover, the sequence $L(\mathbf{Z}_k^{(\ell)})$ is also a bounded sequence since both \mathbf{C} and \mathbf{D} are bounded. See Alg.6 for the outline of the proposed dictionary learning method that solves (3.5).

Iteration complexity. The main computational cost of our algorithm 6 lies in the matrix product in the sparse coding stage. So, the algorithm 6 has $O(mnp)$ iteration complexity which is less than $O(mnp + K^2mp)$, the iteration complexity of the accelerated version of the K-SVD method [74], where K is the predefined sparsity level.

Remark Algorithm 6 can be further accelerated by updating its associated coefficients right after one dictionary item is updated. The coefficient update can be done using least squares

Algorithm 6 Proximal method for dictionary learning1: **INPUT:** Training signals \mathbf{Y} 2: **OUTPUT:** Learned Dictionary \mathbf{D} 3: **Main Procedure:**1. Initialization: set dictionary $\mathbf{D}^{(0)}$, $\rho > 1, \underline{\ell} > 0$.2. For $\ell = 0, 1, \dots$

(a) Sparse approximation:

$$\begin{aligned}\lambda_\ell &= \max\{\rho L(\mathbf{D}^{(\ell)}), \underline{\ell}\}; \\ \mathbf{T}_{\mathbf{C}}^{(\ell)} &= \mathbf{C}^{(\ell)} f - \nabla_{\mathbf{C}} Q(\mathbf{C}^{(\ell)} f, \mathbf{D}^{(\ell)}); \\ \mathbf{C}^{(\ell)} &= \min\{T_{\sqrt{2\lambda/\lambda_\ell}}(\mathbf{T}_{\mathbf{C}}^{(\ell)}), M\}.\end{aligned}\tag{3.15}$$

(b) for $k = 1, \dots, m$,

$$\mathbf{V}^{(\ell)} = \mathbf{C}^{(\ell)} \mathbf{C}^{(\ell)\top}, \quad L(\mathbf{z}_k^{(\ell)}) = V_{kk}^{(\ell)}.$$

(c) Dictionary update: for $k = 1, \dots, p$,

$$\begin{aligned}\mu_l^k &= \max\{\rho L(\mathbf{z}_k^{(\ell)}), \underline{\ell}\}; \\ \mathbf{s}_k^{(\ell)} &= \mathbf{d}_k^{(\ell)} - \frac{1}{\mu_l^k} \nabla_{\mathbf{d}_k} Q(\mathbf{C}^{(\ell)}, \tilde{\mathbf{D}}_k^{(\ell)}); \\ \mathbf{d}_k^{(\ell+1)} &= \mathbf{s}_k^{(\ell)} / \|\mathbf{s}_k^{(\ell)}\|_2.\end{aligned}\tag{3.16}$$

(d) $L(\mathbf{D}^{(\ell+1)}) = \|\mathbf{D}^{(\ell+1)\top} \mathbf{D}^{(\ell+1)}\|_F$.

regression on the same support of the previous one ¹.

3.4 Global convergence of Algorithm 6

Before proving the global convergence of Alg. 6, we first introduce the definition of the critical points of a non-convex function given in [15].

Definition Given the non-convex function $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is a proper and lower semi-continuous function and $\text{dom}f = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) < +\infty\}$.

- For $\mathbf{x} \in \text{dom}f$, its *Fréchet sub-differential* of f is defined as

$$\hat{\partial}f(\mathbf{x}) = \{\mathbf{u} : \liminf_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \neq \mathbf{x}} \frac{f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle}{\|\mathbf{y} - \mathbf{x}\|} \geq 0\}$$

and $\hat{\partial}f(\mathbf{x}) = \emptyset$ if $\mathbf{x} \notin \text{dom}f$.

- The *Limiting Sub-differential* of f at \mathbf{x} is defined as

$$\partial f(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^n : \exists \mathbf{x}^k \rightarrow \mathbf{x}, f(\mathbf{x}^k) \rightarrow f(\mathbf{x}) \text{ and } \mathbf{u}^k \in \hat{\partial}f(\mathbf{x}^k) \rightarrow \mathbf{u}\}.$$

- The point \mathbf{x} is a *critical point* of f if $0 \in \partial f(\mathbf{x})$.

Remark

- If \mathbf{x} is a local minimizer of f then $0 \in \partial f(\mathbf{x}) \subseteq \hat{\partial}f(\mathbf{x})$.
- If (\mathbf{C}, \mathbf{D}) is the critical point of (3.5), then we have

$$(\mathbf{D}^\top \mathbf{D} \mathbf{C})[i, j] = (\mathbf{D}^\top \mathbf{Y})[i, j] \text{ if } \mathbf{C}[i, j] \neq 0.$$

Theorem 3.4.1. [*Sequence convergence*] The sequence generated by the algorithm 6, $\{(\mathbf{C}^{(\ell)}, \mathbf{D}^{(\ell)})\}$, is a Cauchy sequence and converges to a critical point of (3.5).

¹The convergence analysis for the accelerated implementation can be done using similar arguments for Alg. 6

Proof. The proof is built upon Theorem 1 from [15].

Theorem 3.4.2. [15] *The sequence $\mathbf{Z}^{(\ell)} = (\mathbf{x}^{(\ell)}, \mathbf{y}^{(\ell)})$ generated by the iteration (3.7) converges to the critical point of (3.6), if the following conditions hold:*

1. $H(\mathbf{x}, \mathbf{y})$ is a KL function;
2. $\mathbf{Z}^{(\ell)}, \ell = 1, 2, \dots$ is a bounded sequence and there exists some positive constant $\underline{\ell}, \bar{\ell}$ such that $t_\ell^1, t_\ell^2 \in (\underline{\ell}, \bar{\ell}), \ell = 1, 2, \dots$;
3. $\nabla Q(\mathbf{x}, \mathbf{y})$ has Lipschitz constant on any bounded set.

The first condition requires that the objective function satisfies the so-called *Kurdyka-Lojasiewicz (KL)* properties in its effective domain; see Definition 3 in [15] for more details on KL properties. It is shown in Remark 5 and Theorem 11 in [14] that any so-called *semi-algebraic* function satisfy the Kurdyka-Lojasiewicz property. In the next, we first give the definition of the semi-algebraic sets and functions, followed by the proof that the objective function (3.6) defined via (3.10) is a semi-algebraic function.

The next lemma establishes that the objective function (3.6) defined via (3.10) is a semi-algebraic function.

Lemma 3.4.3. *Each term in the function (3.6) defined via (3.10) is a semi-algebraic function, and thus the function (3.6) defined via (3.10) is a semi-algebraic function.*

Proof. For $Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{DC}\|_F^2$ is a real polynomial function, $Q(\mathbf{C}, \mathbf{D})$ is a semi-algebraic function [15].

It is easy to notice that the set $\mathcal{X} = \{\mathbf{D} \in \mathbb{R}^{n \times m} : \|\mathbf{d}_k\|_2 = 1, 1 \leq k \leq m\} = \bigcap_{k=1}^m \{\mathbf{D} : \sum_{j=1}^n \mathbf{D}_{kj}^2 = 1\}$ is a semi-algebraic set. And the set $\mathcal{C} = \{\mathbf{C} \in \mathbb{R}^{m \times p} : \|\mathbf{c}_k\|_\infty \leq M\} = \bigcup_{j=1}^M \bigcup_{k=1}^p \{\mathbf{C} : \|\mathbf{c}_k\|_\infty = j\}$ is a semi-algebraic set. Therefore, the indicator functions $I_{\mathcal{C}}(\mathbf{C})$ and $I_{\mathcal{X}}(\mathbf{D})$ are semi-algebraic functions from the fact that the indicator function for semi-algebraic sets are semi-algebraic functions [2].

For the function $F(\mathbf{C}) = \|\mathbf{C}\|_0$. The graph of F is $S = \bigcup_{k=0}^{mp} L_k \triangleq \{(\mathbf{C}, k) : \|\mathbf{C}\|_0 = k\}$. For each $k = 0, \dots, mp$, let $\mathcal{S}_k = \{J : J \subseteq \{1, \dots, mp\}, |J| = k\}$, then $L_k = \bigcup_{J \in \mathcal{S}_k} \{(\mathbf{C}, k) : \mathbf{C}_{J^c} =$

$0, \mathbf{C}_J \neq 0\}$. It is easy to know the set $\{(\mathbf{C}, k) : \mathbf{C}_{J^c} = 0, \mathbf{C}_J \neq 0\}$ is a semi-algebraic set in $\mathbb{R}^{m \times p} \times \mathbb{R}$. Thus, $F(\mathbf{C}) = \|\mathbf{C}\|_0$ is a semi-algebraic function since the finite union of the semi-algebraic set is still semi-algebraic [15]. \blacksquare

For the second condition in theorem 3.4.2, $\mathbf{C}^{(\ell)} \in \mathcal{C}$ and $\mathbf{D}^{(\ell)} \in \mathcal{X}$ for any $\ell = 1, 2, \dots$, which implies $\mathbf{Z}^{(\ell)} = (\mathbf{C}^{(\ell)}, \mathbf{D}^{(\ell)})$ is a bounded sequence. In addition, for $\ell = 1, 2, \dots$, the step size $\lambda_\ell = \max(\rho L(\mathbf{D}^{(\ell)}), \underline{l})$ is bounded above since $L(\mathbf{D}^{(\ell)}) = \|\mathbf{D}^{(\ell)\top} \mathbf{D}^{(\ell)}\|_F$ and $\mathbf{D} \in \mathcal{X}$. The same holds for the step size $\{\mu_k^\ell\}_{k=1}^m$ since $\mu_k^\ell = \max(\rho L(\mathbf{Z}_k^{(\ell)}), \underline{l})$ where $L(\mathbf{Z}_k^{(\ell)}) = [\mathbf{C}^{(\ell)} \mathbf{C}^{(\ell)\top}]_{k,k}$ is bounded above. Consequently, there exists $\underline{l}, \bar{l} > 0$ such that $\lambda_\ell, \mu_k^\ell \in (\underline{l}, \bar{l})$ for any k, ℓ .

For the last condition in theorem 3.4.2, notice that the function $Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{DC}\|_F^2$ is a smooth function. More specifically, $\nabla Q(\mathbf{C}, \mathbf{D}) = (\mathbf{D}^\top (\mathbf{DC} - \mathbf{Y}), (\mathbf{DC} - \mathbf{Y}) \mathbf{C}^\top)$ has Lipschitz constant on any bounded set. In other words, for any bounded set \mathcal{M} , there exists a constant $M > 0$, such that for any $\{(\mathbf{C}_1, \mathbf{D}_1), (\mathbf{C}_2, \mathbf{D}_2)\} \subseteq \mathcal{M}$,

$$\|\nabla Q(\mathbf{C}_1, \mathbf{D}_1) - \nabla Q(\mathbf{C}_2, \mathbf{D}_2)\| \leq M \|(\mathbf{C}_1, \mathbf{D}_1) - (\mathbf{C}_2, \mathbf{D}_2)\|. \quad \blacksquare$$

Remark Different from the subsequence convergence property, the global convergence property is defined as: $(\mathbf{C}, \mathbf{D}) \rightarrow (\bar{\mathbf{C}}, \bar{\mathbf{D}})$, as $\ell \rightarrow +\infty$.

Next, we show that Algorithm 1 has at least of sub-linear convergent rate.

Theorem 3.4.4. [Sub-linear convergence rate] *The sequence generated by the Alg. 6, $\{(\mathbf{C}^{(\ell)}, \mathbf{D}^{(\ell)})\}$, converges to a critical point $(\bar{\mathbf{C}}, \bar{\mathbf{D}})$ of (3.5) at least in the sub-linear convergence rate, i.e. there exist some $\omega > 0$, such that*

$$\|(\mathbf{C}^{(\ell)}, \mathbf{D}^{(\ell)}) - (\bar{\mathbf{C}}, \bar{\mathbf{D}})\| \leq \omega \ell^{-\frac{1-\theta}{2\theta-1}} \quad (3.17)$$

where $\theta \in (\frac{1}{2}, 1)$.

Proof. The proof is a direct application of the following theorem established in [2].

Proposition 3.4.5 ([2]). *For a given semi-algebraic function $f(\mathbf{u})$, for all $\mathbf{u} \in \text{dom}f$, there exists $\theta \in [0, 1)$, $\eta \in (0, +\infty]$ a neighborhood U of \mathbf{u} and a concave and continuous function $\phi(s) = cs^{1-\theta}$, $s \in [0, \eta]$ such that for all $\bar{\mathbf{u}} \in U$ and satisfies $f(\bar{\mathbf{u}}) \in (f(\mathbf{u}), f(\mathbf{u}) + \eta)$, the following inequality holds*

$$\phi'(f(\bar{\mathbf{u}}) - f(\mathbf{u}))\text{dist}(0, \partial f(\bar{\mathbf{u}})) \geq 1 \quad (3.18)$$

where $\text{dist}(0, \partial f(\bar{\mathbf{u}})) = \max\{\|\mathbf{u}^*\| : \mathbf{u}^* \in \partial f(\bar{\mathbf{u}})\}$.

Theorem 3.4.6 ([2]). *If the objective function is semi-algebraic, $\mathbf{Z}^{(\ell)} = (\mathbf{x}^{(\ell)}, \mathbf{y}^{(\ell)})$ generated by the iteration (3.7), and $\bar{\mathbf{Z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is its limit point. Then*

- *If $\theta = 0$, $\mathbf{Z}^{(\ell)}$ converges to $\bar{\mathbf{Z}}$ in finite steps.*
- *If $\theta \in (0, 1/2]$, then $\exists \omega > 0$ and $\tau \in [0, 1)$, such that $\|\mathbf{Z}^{(\ell)} - \bar{\mathbf{Z}}\| \leq \omega \tau^\ell$*
- *If $\theta \in (1/2, 1)$, then $\exists \omega > 0$ such that $\|\mathbf{Z}^{(\ell)} - \bar{\mathbf{Z}}\| \leq \omega \ell^{-\frac{1-\theta}{2\theta-1}}$.*

where θ corresponding to the desingularizing function $\phi(s) = cs^{1-\theta}$ defined in proposition 3.4.5.

In the proposed Alg.6, notice that $\frac{\tau^\ell}{\ell^{-\frac{1-\theta}{2\theta-1}}} \rightarrow 0$ as $\ell \rightarrow +\infty$, where $\tau \in [0, 1)$ and $\theta \in (1/2, 1)$. Thus, the sequence $\mathbf{Z}^{(\ell)}$ converges to $\bar{\mathbf{Z}}$ at least in sub-linear rate. ■

3.5 Experiments

In this section, the practical performance and computational efficiency of the proposed approach is evaluated on two applications: image de-noising and face recognition. The experiments on these two applications showed that, using the same minimization model, the performance of our approach is comparable to the K-SVD based method, but is more computationally efficient with less running time.

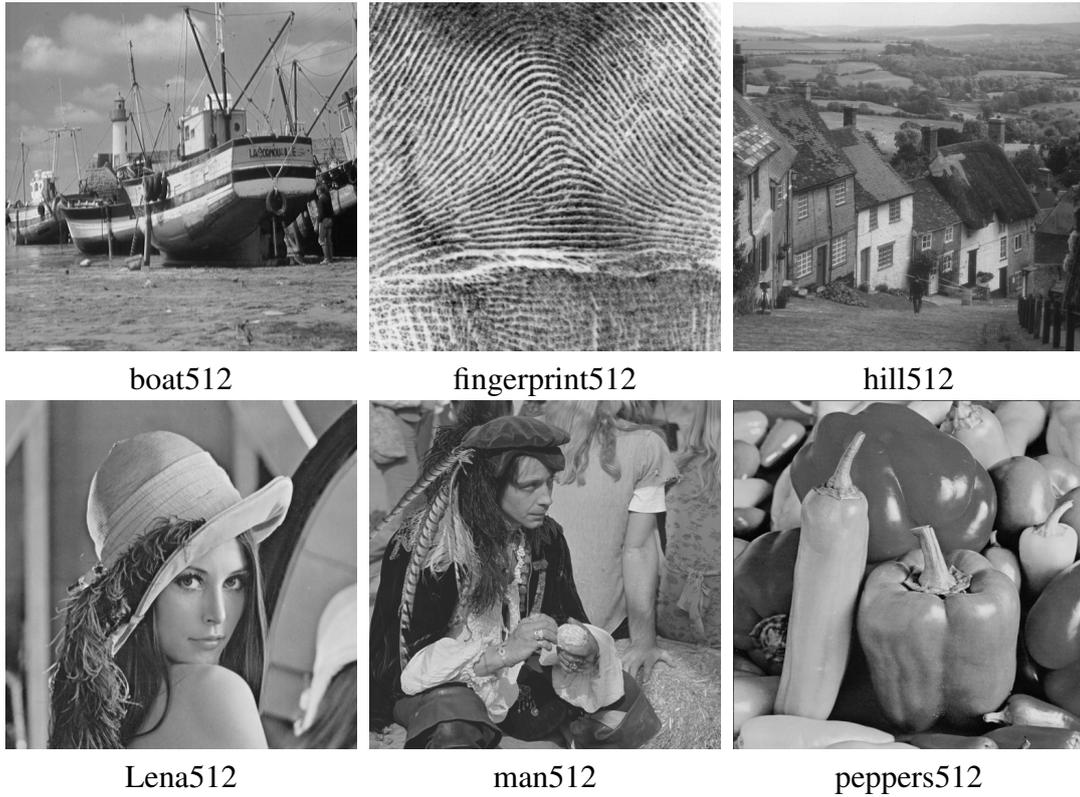


Figure 3.2: Test images.

3.5.1 Image denoising

Alg. 6 for image denoising is evaluated on tested images shown in Fig. 3.2 with different noise levels. Through all the experiments, we set $\lambda = 10\sigma^2$ as the thresholding value for dictionary learning process. Same as the K-SVD method [34], the dimension of the dictionary is set to $m = 4n$ and the initialization is done via filling in 8×8 local DCT transform and leaving others zero vectors. The maximum iteration of Alg. 6 is set as 30. After the dictionary is learned via training samples, the image is de-noised using the coefficients from the OMP method under the learned dictionary in one pass. The results is compared to the DCT-based thresholding method and the K-SVD denoising method [34] with patch size 8×8 . See Table 3.1 for the list of PSNR values of the results and Fig. 3.4 for a visual illustration of the denoised images. Fig. 3.3 shows the dictionaries learned from noise image by both the K-SVD method and the proposed method. It can be seen that the performance of our approach is comparable to the K-SVD method.

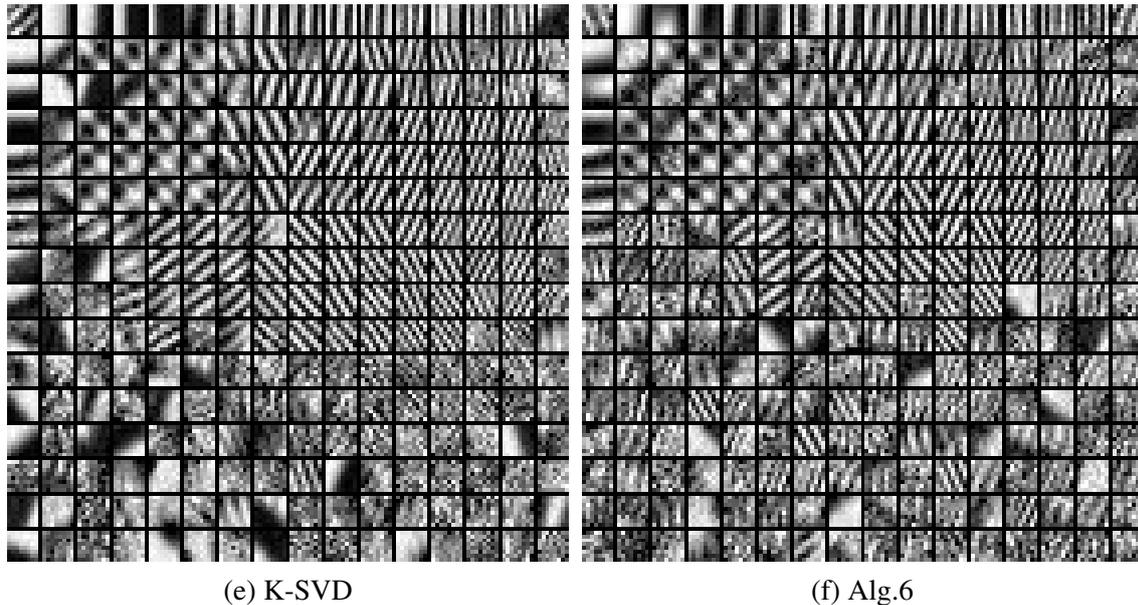


Figure 3.3: The dictionaries learned from the image "Lena512" with noise level $\sigma = 30$ using the K-SVD method and Alg.6. The atom size is 8×8 .

The computational efficiency of the proposed one is compared to the accelerated version of the K-SVD method, the approximated K-SVD Algorithm [74] with the implementation from the original authors². All two methods run on the same environment: MATLAB R2011b (64bit) Linux version on a PC workstation with an INTEL CPU (2.4GHZ) and 48G memory. The average running time of each iteration is: 2.81 seconds (K-SVD) vs. 0.71 seconds (ours). Fig. 3.5 shows the comparison of the overall running time of the accelerated implementation of Alg.6 and the K-SVD method to denoise image "Lena512" with noise level $\sigma = 25$. Clearly, Alg.6 is noticeably faster than the approximate K-SVD method when learning the dictionary of the same size.

3.5.2 Face recognition

Alg. 6 can also be applied to recognition tasks using the model (3.5) by simply replacing the K-SVD module by the proposed one. The performance is evaluated on two face datasets: Extended YaleB dataset [38] and AR face dataset [60]. The one used our approach is compared to three K-SVD based methods: LC-KSVD [45], D-KSVD [95] and K-SVD [1]. The

²<http://www.cs.technion.ac.il/~ronrubin/software.html>



Figure 3.4: Visual illustration of noisy images and denoised results

Image	Boat512					Fingerprint512				
σ	5	10	15	20	25	5	10	15	20	25
DCT; 8×8	36.79	33.49	31.34	29.96	28.90	36.34	32.25	29.68	28.29	26.85
K-SVD; 8×8	37.17	33.64	31.73	30.36	29.28	36.59	32.39	30.06	28.47	27.26
Ours; 8×8	37.02	33.57	31.62	30.20	29.16	36.59	32.35	29.97	28.28	27.03
Image	Lena512					Man512				
σ	5	10	15	20	25	5	10	15	20	25
DCT; 8×8	38.29	35.25	33.39	32.03	30.96	37.16	33.12	31.01	29.65	28.67
K-SVD; 8×8	38.59	35.47	33.70	32.38	31.32	37.61	33.62	31.45	30.13	29.11
Ours; 8×8	38.49	35.41	33.57	32.25	31.19	37.46	33.47	31.43	30.02	29.00
Image	Hill512					Peppers512				
σ	5	10	15	20	25	5	10	15	20	25
DCT; 8×8	36.54	32.93	31.11	30.02	29.00	37.06	34.48	33.02	31.89	30.95
K-SVD; 8×8	36.99	33.34	31.43	30.17	29.19	37.77	34.72	32.37	32.26	31.39
Ours; 8×8	36.94	33.31	31.29	30.02	29.06	37.68	34.64	33.22	32.14	31.18

Table 3.1: PSNR values of the denoised results

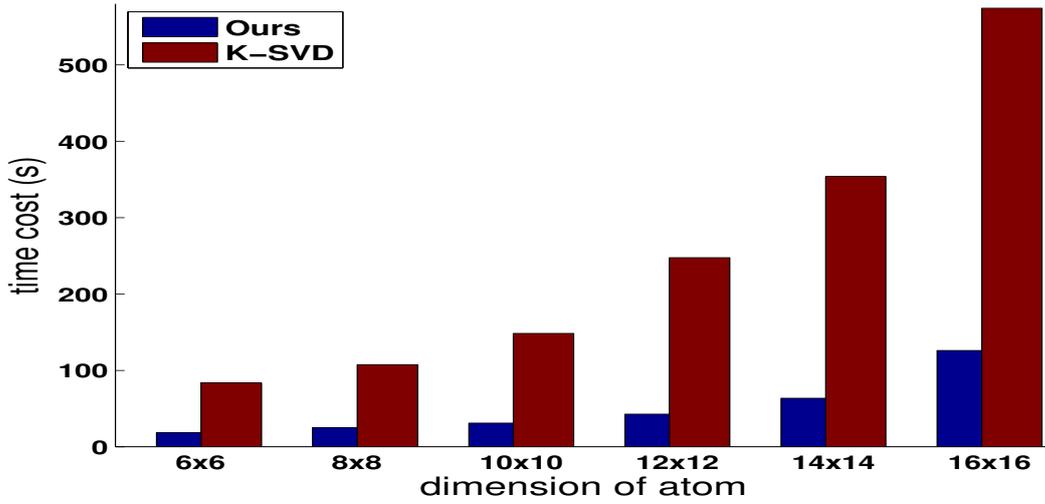


Figure 3.5: Overall running time of our method and the K-SVD de-noising method with comparable PSNR values.

Table 3.2: Training time (seconds) on two face datasets.

Dataset	K-SVD	D-KSVD	LC-KSVD	Ours
Extended YaleB	44.46	63.47	184.64	10.52
AR Face	55.03	70.43	256.12	22.75

Table 3.3: Classification accuracies (%) on two face datasets.

Dataset	K-SVD	D-KSVD	LC-KSVD	Ours
Extended YaleB	93.10	94.10	95.00	95.66
AR Face	86.50	88.80	93.70	94.41

experimental setting is set the same as [45, 95]:

Extended Yale B: The extended YaleB database [38] contains 2,414 images of 38 human frontal faces under about 64 illumination conditions and expressions. There are about 64 images for each person. The original images were cropped to 192×168 pixels. Following [95], we project each face image into a 504-dimensional feature vector using a random matrix of zero-mean normal distribution. The database is randomly split into two halves. One half which contains 32 images for each person was used for training the dictionary. The other half was used for testing.

AR Face Database: The AR face database [60] consists of over 4000 frontal images from 126 individuals. For each individual, 26 pictures were taken in two separate sessions. The main characteristic of the AR database is that it includes frontal views of faces with different facial expressions, lighting conditions and occlusion conditions. Following the standard evaluation procedure from [45, 95], we use a subset of the database consisting of 2,600 images from 50 male subjects and 50 female subjects. For each person, twenty images are randomly picked up for training and the remaining images are for testing. Each face image is cropped to 165×120 and then projected onto a 540-dimensional feature vector.

We set the thresholding parameter λ to be $10^{-4}/2$ and initialize the dictionary with identity matrix. Besides the classification accuracies, we also evaluate the training time of all compared approaches under the same environment. The results of all the tested methods are listed in Table 3.3 and Table 3.2. It can be seen that our approach performs consistently with the state-of-the-art methods while have noticeable advantages on computational efficiency.

3.6 Summary

In this chapter, we proposed an alternating proximal method iteration scheme for solving ℓ_0 norm based dictionary learning problems in sparse coding. The proposed one not only answered the open question regarding the existence of a convergent method for solving ℓ_0 norm based dictionary learning problems, but also showed the computational efficiency on two practical applications. In future, we will investigate the applications of the proposed framework for solving other non-convex minimization problems in computer vision.

Chapter 4

Incoherent dictionary learning for image recognition

4.1 Introduction

Recently, sparse coding has been one important tool in many applications ([80]) including image recovery, machine learning, recognition and etc. Given a set of input patterns, most existing sparse coding models aim at finding a small number of *atoms* (representative patterns) whose linear combinations approximate those input patterns well. More specifically, given a set of vectors $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_p\} \subset \mathbb{R}^n$, sparse coding is about determining a *dictionary* (the set of atoms)

$$\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m\} \subset \mathbb{R}^n,$$

together with a set of coefficient vectors $\{\mathbf{c}_1, \dots, \mathbf{c}_p\} \subset \mathbb{R}^m$ with most elements close to zero, so that each input vector \mathbf{y}_j can be approximated by the linear combination $\mathbf{y}_j \approx \sum_{\ell=1}^m \mathbf{c}_j(\ell) \mathbf{d}_\ell$. The typical sparse coding method, e.g. K-SVD [1], determines the dictionary $\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m\}$ via solving an optimization problem with sparsity-prompting functional on the coefficients:

$$\min_{\mathbf{D}, \{\mathbf{c}_i\}_{i=1}^p} \sum_{i=1}^p (\|\mathbf{y}_i - \mathbf{D}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_0), \quad \text{subject to } \|\mathbf{d}_j\|_2 = 1, \quad 1 \leq j \leq m, \quad (4.1)$$

where $\|\cdot\|_0$ counts the number of non-zero entries and $\mathbf{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_m\}$ is the dictionary for sparse coding. It is well known that the above minimization (4.1) is an NP-hard problem and only sub-optimal solution can be obtained in polynomial time. Most existing methods use an alternating iteration scheme to solve (4.1).

Despite the success of sparse coding in many applications, the sequence generated by most existing numerical solvers for solving the non-convex problem (4.1) can only guarantee that the functional value of (4.1) is decreasing at each iteration, which can not guarantee the generated sequence is convergent. Indeed, the sequence generated by the K-SVD method is not convergent; see Fig. 4.1 for an illustration. Moreover, as it has been mentioned in the literature, good performance of sparse coding in various recognition tasks requires imposing some additional constraints of the dictionary. One of such essential dictionary properties is the so-called *mutual coherence*:

$$\mu(\mathbf{D}) = \max_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|, \quad (4.2)$$

which further increases the technical difficulty of designing an effective numerical method with theoretical soundness. Although there is no such term in (4.1), the existing implementation of the K-SVD method implicitly tries to avoid learning a dictionary with high mutual coherence by discarding the learned atom which has large mutual coherence with the existing ones in each iteration.

In this chapter, we consider the problem of sparse coding that explicitly imposes additional regularization on the mutual coherence of the dictionary, which can be formulated as the following minimization problem:

$$\begin{aligned} \min_{\mathbf{D}, \{\mathbf{c}_i\}_{i=1}^p} & \sum_i \left(\frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\mathbf{c}_i\|_F^2 + \lambda \|\mathbf{c}_i\|_0 \right) + \frac{\alpha}{2} \|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2, \\ \text{s.t.} & \quad \|\mathbf{d}_j\|_2 = 1, \quad 1 \leq j \leq m. \end{aligned} \quad (4.3)$$

The minimization models similar to (4.3) have been used in several sparse coding based systems; see e.g. [9, 52, 68]. As a more general optimization problem which contains the K-SVD model (4.1) by setting $\alpha = 0$, the optimization problem (4.3) is a even harder problem

to solve.

This chapter aims at developing a fast alternating iteration scheme specifically designed for solving (4.3). As shown in the experiments, compared to the generic dictionary generated by the K-SVD method, the dictionary generated by the proposed method has much lower mutual coherence and it provides better performance in several sparse coding based recognition tasks. Moreover, in contrast to the existing numerical solvers for (4.3), we provided the rigorous analysis on the convergence of the proposed method. It is mathematically proved that the whole sequence generated by the proposed method converges to a stationary point of the problem, while the existing analysis of all other solvers only shows that the functional values of the sequence is decreasing or equivalently only a sub-sequence is convergent. The whole sequence convergence of an iteration scheme is not only of theoretical interest, but also important for applications, e.g. the number of iterations does not need to be empirically chosen for obtaining stability.

4.1.1 Motivation and main contributions

The main motivation of this chapter is two-fold: one is the need for learning an incoherent dictionary for sparse coding in many applications, and the other is the need of a numerical solver for solving (4.3) with proved convergence property.

Motivation

The need of an incoherent dictionary for sparse coding. Once a dictionary is learned, the sparse code for each input is then computed via some pursuit methods, e.g. *orthogonal matching pursuit* [81], *basis pursuit* [24]. The success of these methods for finding the optimal sparse code depends on the incoherence property of the dictionary. In [81], Tropp showed that that the OMP can recover the exact support of the coefficients when mutual coherence μ is less than $1/(2S - 1)$ where S is the number of nonzero entries of the correct coefficients. It is further proved in [75] that the similar requirement on the mutual coherence is also needed for ensuring the correctness of the thresholding-based sparse coding algorithms. In practice, it is also observed that a dictionary with high mutual coherence will

impact the performance of sparse coding based methods; see e.g [13, 68, 87].

The need of a variational model that explicitly regularizes mutual coherence. In a quick glance, the widely used K-SVD method [1] for sparse coding considered a variational model which has no explicit functional on minimizing the mutual coherence of the result, i.e., it considered a special case of (4.3) with $\alpha = 0$. However, the implementation of the K-SVD method implicitly controlled the mutual coherence of the dictionary by discarding the "bad" atom which is highly correlated to the ones already in the dictionary. Such an ad-hoc approach certainly is not optimal for lowering the overall mutual coherence of the dictionary. In practice, the K-SVD method may still give a dictionary that contains highly correlated atoms, which will lead to poor performance in sparse approximation, see [28] for more details.

The need of a convergent algorithm. The minimization problem (4.3) is a challenging non-convex problem. Most existing methods that used the model (4.3) or its extensions, e.g. [45, 56, 95], simply call some generic non-linear optimization solvers such as the *projected gradient* method. Such a scheme is slow and not stable in practice. Furthermore, all these methods at most can be proved that the functional value is decreasing at each iteration. The sequence itself may not be convergent. From the theoretical perspective, a non-convergent algorithm certainly is not satisfactory. From the application perspective, the divergence of the algorithm also leads to troublesome issues such as when to stop the numerical solver, which often requires manual tune-up.

Main Contributions

In this chapter, we proposed a hybrid alternating proximal scheme for solving (4.3). Compared to the K-SVD method that controls the mutual coherence of the dictionary in an ad-hoc manner, the proposed method is optimized for learning an incoherent dictionary for sparse coding. Compared to the generic numerical scheme for solving (4.3) adopted in the existing applications, the convergence property of the proposed method is rigorously established in the chapter. We showed that the whole sequence generated by the proposed method converges to a stationary point. As a comparison, only sub-sequence convergence can be

proved for existing numerical methods. The whole sequence convergence of an iteration scheme is not only of theoretical interest, but also important for applications as the number of iterations does not need to be empirically chosen to keep the output stable.

4.1.2 Related work

In this section, we give a brief review on most related generic dictionary learning methods and incoherent dictionary learning methods for sparse coding.

Generic Dictionary Learning Methods

Among many existing dictionary learning methods, the so-called K-SVD method [1] is the most widely used one. The K-SVD method solves the problem (4.3) with $\alpha = 0$ by alternatively iterating between sparse code \mathbf{C} and the dictionary \mathbf{D} . The sparse code \mathbf{C} is estimated by using the OMP method [81]: at each step, one atom is selected such that it is most correlated with the current residuals and finally the observation is projected onto the linear space spanned by the chosen atoms. In the dictionary update stage for estimating \mathbf{D} , the atoms are updated sequentially by using the rank-1 approximation to current residuals which can be exactly solved by the SVD decomposition. Most other existing dictionary learning methods (e.g. [44, 56, 57, 67]) are also based on the similar alternating scheme between the dictionary update and sparse code estimation. In [44, 57], the atoms in the dictionary are updated sequentially with closed form solutions. The projection gradient descent method is used in [56] to update the whole dictionary. For the ℓ_0 norm related minimization problem in the stage of sparse code estimation, many relaxation methods have been proposed and the ℓ_1 norm based relaxation is the most popular one; see e.g. [44, 56, 57, 90]. Among these methods, the convergence analysis is provided in [90] for its proximal method. Recently, an proximal alternating linearized method is presented in [8] to directly solve the ℓ_0 norm based optimization problem for dictionary learning. The method proposed in [8] is mathematically proved to be globally convergent.

Incoherent Dictionary Learning Methods

There are two types of approaches to learn an incoherent dictionary for sparse coding. The first one is to add an additional process in the existing generic dictionary learning method to lower the mutual coherence, e.g. [9, 52]. Both [52] and [9] added the decorrelation step after the dictionary update stage in K-SVD method. In [52], the de-correlation is done via minimizing the distance between the learned dictionary generated by the K-SVD method and the space spanned by the dictionaries with certain mutual coherence level. However, this projection step doesn't consider the approximation error and may significantly increase the whole minimization functional value. Thus, in [9], the iterative projection method is introduced to lower the mutual coherence of the dictionary, together with an additional dictionary rotation step to improve the approximation error of the de-correlated dictionary. The other way to learn the incoherent dictionary is directly solving a minimization model that contains the functional related the mutual coherence of the dictionary, e.g. [7, 68]. In [68], an additional regularization term on mutual coherence is added to (4.1) when being applied in image classification and clustering. The approach presented in [9] used the OMP method in sparse code estimation and method of optimal coherence-constrained direction for dictionary update. In [7], the orthogonality constraints on the dictionary atoms are explicitly added in the variational model for dictionary learning such that its mutual coherence is always 0. With the performance comparable to the K-SVD method in image recovery, the orthogonal dictionary based method [7] is significantly faster than the K-SVD method. Such advantages on computational efficiency comes from the fact that both sparse code estimation and dictionary update have closed-form solutions in [7].

4.2 Incoherent dictionary learning algorithm

We first give an introduction to the definitions and notations used in this section. We define \mathbf{Y} be a matrix, \mathbf{y}_j be the j -th column of \mathbf{Y} and y_{ij} be the (i, j) -th element of \mathbf{Y} . Given the matrix \mathbf{Y} , the Frobenius norm of \mathbf{Y} is defined by $\|\mathbf{Y}\|_F = (\sum_{i,j} y_{ij}^2)^{1/2}$, its ℓ_0 norm $\|\mathbf{Y}\|_0$ is defined as the number of nonzero entries of \mathbf{Y} and the infinity norm of $\|\mathbf{Y}\|_\infty = \max_{i,j} \{|y_{ij}|\}$.

Define the *hard thresholding operator* $T_\lambda(\mathbf{D})[i, j] = d_{ij}$ if $|d_{ij}| > \lambda$ and $T_\lambda(\mathbf{D})[i, j] = 0$ otherwise.

4.2.1 Problem formulation

Given the training samples $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_p) \in \mathbb{R}^{n \times p}$, we consider the sparse approximation of \mathbf{Y} by the redundant dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$. Same as [68], we can introduce the regularization $\|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2$ to the variational model to minimize the mutual coherence. The variational model of incoherent dictionary learning model is given as follows,

$$\begin{aligned} \min_{\mathbf{D}, \mathbf{C}} \quad & \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_0 + \frac{\alpha}{2} \|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2, \\ \text{s.t.} \quad & \|\mathbf{d}_j\|_2 = 1, 1 \leq j \leq m; \|\mathbf{c}_i\|_\infty \leq M, 1 \leq i \leq m, \end{aligned} \quad (4.4)$$

where $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_m) \in \mathbb{R}^{n \times m}$, $\mathbf{C} = (\mathbf{c}_1^\top, \dots, \mathbf{c}_m^\top)^\top \in \mathbb{R}^{m \times p}$ and M is the predefined upper bound for the elements in \mathbf{C} . It is noted that the predefined upper bound M is mainly for the stability of the algorithm, which is allowed to be set arbitrarily large. For the simplicity of discussion, define $\mathcal{D} = \{\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_m) \in \mathbb{R}^{n \times m} : \|\mathbf{d}_j\|_2 = 1, 1 \leq j \leq m\}$ and $\mathcal{C} = \{\mathbf{C} = (\mathbf{c}_1^\top, \dots, \mathbf{c}_m^\top)^\top \in \mathbb{R}^{m \times p}, \|\mathbf{c}_i\|_\infty \leq M, 1 \leq i \leq m\}$. Then the model (4.4) can be reformulated as

$$\min_{\mathbf{D}, \mathbf{C}} \quad \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_0 + \frac{\alpha}{2} \|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2, \text{ s.t. } \mathbf{D} \in \mathcal{D}, \mathbf{C} \in \mathcal{C}. \quad (4.5)$$

In the next, we will propose the hybrid alternating proximal algorithm for solving (4.5) with the whole sequence convergence property.

4.2.2 A hybrid alternating proximal algorithm

The algorithm for solving (4.4) is based on a hybrid scheme that combines the alternating proximal method [3] and the alternating proximal linearized method [15], which are about tackling the non-convex minimization problem of the form:

$$\min_{z:=(x,y)} H(x, y) = F(x) + Q(z) + G(y), \quad (4.6)$$

where F, G are proper lower semi-continuous functions and Q is the smooth function with Lipschitz derivatives on any bounded set, that is, for the bounded set \mathcal{L} , there exists a constant $L > 0$, such that $\|\nabla Q(z_1) - \nabla Q(z_2)\|_F \leq L\|z_1 - z_2\|_F, z_1, z_2 \in \mathcal{L}$.

The alternating proximal method [3] updates the (x, y) via as follows,

$$\begin{cases} x_{k+1} \in \arg \min_x F(x) + Q(x, y_k) + G(y_k) + \frac{\mu^k}{2} \|x - x_k\|_F^2; \\ y_{k+1} \in \arg \min_y F(x_{k+1}) + Q(x_{k+1}, y) + G(y) + \frac{\lambda^k}{2} \|y - y_k\|_F^2, \end{cases} \quad (4.7)$$

where μ^k, λ^k are suitable step sizes. In general, the scheme (4.7) requires solving the non-smooth and non-convex minimization problems in each step which often has no closed form solutions. This motivates a linearized version of alternating proximal algorithm [15] such that each subproblem has a closed form solution. Instead of solving the subproblems as (4.7), the alternating proximal linearized algorithm replaces the smooth term Q in (4.7) by its first order linear approximation:

$$\begin{cases} x_{k+1} \in \arg \min_x F(x) + \hat{Q}_{(x_k, y_k)}(x) + G(y_k) + \frac{\mu^k}{2} \|x - x_k\|_F^2; \\ y_{k+1} \in \arg \min_y F(x_{k+1}) + \hat{Q}_{(x_{k+1}, y_k)}(y) + G(y) + \frac{\lambda^k}{2} \|y - y_k\|_F^2. \end{cases} \quad (4.8)$$

where $\hat{Q}_{(x_k, y_k)}(x) = Q(x_k, y_k) + \langle \nabla_x Q(x_k, y_k), x - x_k \rangle$, $\hat{Q}_{(x_k, y_k)}(y) = Q(x_k, y_k) + \langle \nabla_y Q(x_k, y_k), y - y_k \rangle$, and μ^k, λ^k are carefully chosen step sizes.

Although the proximal linearized method has closed form solutions for all sub-problems, it requires more iterations to converge than the proximal method as it only provides approximated solutions to two-subproblems in (4.7). The problem (4.5) we are solving is different from the generic model considered in the proximal method, as the first sub-problem for sparse code estimation in (4.7) has a closed-form solution while the second one does not. Motivated by this observation, we proposed a hybrid iteration scheme which uses the formulation of the proximal method for sparse code estimation and uses the formulation of the proximal linearized method for dictionary update. In other words, it is a hybrid version that combines both the proximal method and the proximal linearized method. As a result, the proposed one also has the closed form solutions for all sub-problems at each iteration, but

converges faster than the proximal linearized method.

Remark Although both (4.7) and (4.8) are the alternating schemes between two variables, they can be extended to the case of the alternating iteration among a finite number of blocks [4, 15].

The iterations (4.7) and (4.8) can be re-written by using the *proximal operator* [71]:

$$\text{Prox}_t^F(x) := \arg \min_u F(u) + \frac{t}{2} \|u - x\|_F^2.$$

Then, the minimization (4.7) can be re-written as

$$\begin{cases} x_{k+1} \in \text{Prox}_{\mu^k}^{F+Q(\cdot, y_k)}(x_k), \\ y_{k+1} \in \text{Prox}_{\lambda^k}^{G+Q(x_{k+1}, \cdot)}(y_k), \end{cases} \quad (4.9)$$

and the minimization (4.8) can be re-written as

$$\begin{cases} x_{k+1} \in \text{Prox}_{\mu^k}^F(x_k - \frac{1}{\mu^k} \nabla_x Q(x_k, y_k)), \\ y_{k+1} \in \text{Prox}_{\lambda^k}^G(y_k - \frac{1}{\lambda^k} \nabla_y Q(x_{k+1}, y_k)). \end{cases} \quad (4.10)$$

Remark It is shown in [15] that the proximal operator defined in (4.9), (4.10) are well defined, i.e., the solution sets of (4.7) and (4.8) are nonempty and compact.

The minimization (4.4) can be expressed in the form (4.6) by setting

$$\begin{cases} F(\mathbf{C}) = \lambda \|\mathbf{C}\|_0 + \delta_{\mathcal{C}}(\mathbf{C}), \\ Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{C}\|_F^2 + \frac{\alpha}{2} \|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2, \\ G(\mathbf{D}) = \delta_{\mathcal{D}}(\mathbf{D}), \end{cases} \quad (4.11)$$

where $\delta_{\mathcal{C}}(\mathbf{C})$ and $\delta_{\mathcal{D}}(\mathbf{D})$ are indicator functions, that is $\delta_{\mathcal{X}}(x) = 0$ if $x \in \mathcal{X}$ and $\delta_{\mathcal{X}}(x) = +\infty$ if $x \notin \mathcal{X}$. We propose the following alternating scheme to solve (4.4).

Sparse Code Estimator

given the dictionary $\mathbf{d}^{(k)}$, we update the sparse code $\mathbf{c}^{(k)} = \{\mathbf{c}_j^\top\}_{j=1}^m$ row by row as follows:

$$\mathbf{c}_j^{(k)} \in \text{Prox}_{\mu_j^k}^{F(\mathbf{U}_j^k) + Q(\mathbf{U}_j^k, \mathbf{D}^{(k)})}(\mathbf{c}_j^{(k-1)}), \quad 1 \leq j \leq m, \quad (4.12)$$

where $\mathbf{U}_j^k = (\mathbf{c}_1^{(k)\top}, \dots, \mathbf{c}_{j-1}^{(k)\top}, \mathbf{c}_j^\top, \mathbf{c}_{j+1}^{(k-1)\top}, \dots, \mathbf{c}_m^{(k-1)\top})^\top$ for $1 \leq j \leq m$. The minimization (4.12) is easy to solve as it has closed form solution. Define $\mathcal{S}_j^k = \{i | d_{ij} \neq 0, 1 \leq i \leq n\}$ and $\mathbf{R}^{j,k} = \mathbf{Y} - \sum_{i < j} \mathbf{d}_i^{(k)} \mathbf{c}_i^{(k)} - \sum_{i > j} \mathbf{d}_i^{(k)} \mathbf{c}_i^{(k-1)}$. By direct calculation, the minimization (4.12) is equivalent to

$$\mathbf{c}_j^{(k)} \in \arg \min_{\mathbf{c}_j \in \mathcal{C}} \frac{\mu_j^k}{2} \|\mathbf{c}_j - \mathbf{c}_j^{(k-1)}\|_F^2 + \frac{1}{2} \sum_{i \in \mathcal{S}_j^k} \|\mathbf{r}_i^{j,k} - d_{ij} \mathbf{c}_j\|_F^2 + \lambda \|\mathbf{c}_j\|_0, \quad (4.13)$$

where $\mathbf{R}^{j,k} = (\mathbf{r}_1^{j,k\top}, \dots, \mathbf{r}_n^{j,k\top})^\top \in \mathbb{R}^{n \times p}$.

Proposition 4.2.1. *Suppose M is chosen such that $M > \sqrt{\frac{2\lambda}{r_j^k}}$, where $r_j^k = \sum_{i \in \mathcal{S}_j^k} d_{ij}^2 + \mu_j^k$, the minimization (4.13) has the closed form solution for all $1 \leq j \leq m$, given by*

$$\mathbf{c}_j^{(k)} = \min(T \sqrt{2\lambda/r_j^k} ((\sum_{i \in \mathcal{S}_j^k} d_{ij} \mathbf{r}_i^{j,k} + \mu_j^k \mathbf{c}_j^{(k-1)})/r_j^k), M). \quad (4.14)$$

Proof. By direct calculation, it can be seen that the minimization (4.13) is equivalent to the following minimization.

$$\mathbf{c}_j^{(k)} \in \arg \min_{\mathbf{c}_j \in \mathcal{C}} r_j^k \|\mathbf{c}_j - (\sum_{i \in \mathcal{S}_j^k} d_{ij} \mathbf{r}_i^{j,k} + \mu_j^k \mathbf{c}_j^{(k-1)})/r_j^k\|_F^2 + 2\lambda \|\mathbf{c}_j\|_0. \quad (4.15)$$

The variables in the minimization (4.15) above are separable. Thus, it is easy to see that the solution of (4.15) is exactly the one defined by (4.14). ■

Dictionary Update

Given the sparse code $\mathbf{c}^{(k)}$, we update the dictionary $\mathbf{D}^{(k+1)} = \{\mathbf{d}_j\}_{j=1}^m$ atom by atom as follows:

$$\mathbf{d}_j^{(k+1)} \in \text{Prox}_{\lambda_j^k}^{G(\mathbf{S}_j^{(k)})}(\mathbf{d}_j^{(k)} - \frac{1}{\lambda_j^k} \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k)), \quad (4.16)$$

where

$$\begin{cases} \mathbf{S}_j^k = (\mathbf{d}_1^{(k+1)}, \dots, \mathbf{d}_{j-1}^{(k+1)}, \mathbf{d}_j, \mathbf{d}_{j+1}^{(k)}, \dots, \mathbf{d}_m^{(k)}), \\ \mathbf{V}_j^k = (\mathbf{d}_1^{(k+1)}, \dots, \mathbf{d}_{j-1}^{(k+1)}, \mathbf{d}_j^{(k)}, \mathbf{d}_{j+1}^{(k)}, \dots, \mathbf{d}_m^{(k)}). \end{cases}$$

Denote $\mathbf{d}^{j,k} = \mathbf{d}_j^{(k)} - \frac{1}{\lambda_j^k} \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k)$, Then (4.16) can be reformulated as:

$$\mathbf{d}_j^{(k+1)} \in \arg \min_{\|\mathbf{d}_j\|_2=1} \|\mathbf{d}_j - \mathbf{d}^{j,k}\|_2^2, \quad (4.17)$$

From (4.17), it is easy to know $\mathbf{d}_j^{(k+1)} = \mathbf{d}^{j,k} / \|\mathbf{d}^{j,k}\|_2$ for $1 \leq j \leq m$.

There are two step sizes, μ_j^k and λ_j^k needed to be set in the calculation. The step size μ_j^k can be set arbitrarily as long as there exists $a, b > 0$ such that $\mu_j^k \in (a, b)$, $\forall k = 1, 2, \dots, j = 1, \dots, m$. The step size λ_j^k can be chosen as $\lambda_j^k = \max(a, \rho L(\mathbf{d}_j^{(k)}))$, where the λ_j^k can be chosen so as to

$$\|\nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \bar{\mathbf{D}}_j^1) - \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \bar{\mathbf{D}}_j^2)\|_F \leq L(\mathbf{d}_j^k) \|\mathbf{d}_j^1 - \mathbf{d}_j^2\|_F, \quad (4.18)$$

for all $\mathbf{d}_j^1, \mathbf{d}_j^2 \in \mathbb{R}^n$ where $\bar{\mathbf{D}}_j^i = (\mathbf{d}_1^{(k+1)}, \dots, \mathbf{d}_{j-1}^{(k+1)}, \mathbf{d}_j^i, \mathbf{d}_{j+1}^{(k)}, \dots, \mathbf{d}_m^{(k)})$, $i = 1, 2$. Typically, we can choose $\mu_j^k = \mu_0$ and $L(\mathbf{d}_j^k) = \mathbf{c}_j^{(k)} \mathbf{c}_j^{(k)\top} + \alpha \|\mathbf{V}_j^k\|_2$ for all $j = 1, 2, \dots, m$ and $k = 1, 2, \dots$. It can be seen that $L(\mathbf{d}_j^k)$ is a bounded sequence since \mathbf{C} is bounded in the model (4.5). See the Alg. 7 for the outline of the proposed incoherent dictionary learning method that solves (4.5).

Algorithm 7 Incoherent dictionary learning algorithm via solving (4.5).

1: **INPUT:** Training signals \mathbf{Y} ;

2: **OUTPUT:** Learned Incoherent Dictionary \mathbf{D} ;

3: **Main Procedure:**

1. Set the initial dictionary $\mathbf{D}^{(0)}$, $\rho > 1$, $a > 0$ and $K \in \mathbb{N}$.

2. For $k = 0, 1, \dots, K$,

(a) Sparse Coding: for $j = 1, \dots, m$, let $\mathcal{S}_j^k = \{i : d_{ij}^{(k)} \neq 0, 1 \leq i \leq n\}$,

$$\begin{aligned} \mathbf{r}^{j,k} &= \mathbf{Y} - \sum_{i < j} \mathbf{d}_i^{(k)} \mathbf{c}_i^{(k)} - \sum_{i > j} \mathbf{d}_i^{(k)} \mathbf{c}_i^{(k-1)}, \\ \mathbf{c}^{j,k} &= \sum_{i \in \mathcal{S}_j^k} d_{ij} \mathbf{r}_i^{j,k} + \mu_j^k \mathbf{c}_j^{(k-1)}, \quad r_j^k = \sum_{i \in \mathcal{S}_j^k} d_{ij}^2 + \mu_j^k, \\ \mathbf{c}_j^{(k)} &= \min(T \sqrt{2\lambda/r_j^k} (\mathbf{c}^{j,k}/r_j^k), M). \end{aligned} \quad (4.19)$$

(b) Update the step size: for $j = 1, \dots, m$

$$\mathbf{V}^{(k)} = \mathbf{C}^{(k)} \mathbf{C}^{(k)\top}, \quad L(\mathbf{d}_j^{(k)}) = V_{j,j}^{(k)} + \alpha \|\mathbf{V}^k\|_2.$$

(c) Dictionary Update: let $\mu_j^k = \max\{\rho L(\mathbf{d}_j^k), a\}$, for $k = 1, \dots, m$,

$$\mathbf{d}_j^{j,k} = \mathbf{d}_j^{(k)} - \frac{1}{\mu_j^k} \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k); \quad \mathbf{d}_j^{(k+1)} = \mathbf{d}_j^{j,k} / \|\mathbf{d}_j^{j,k}\|_2. \quad (4.20)$$

4.3 Convergence analysis of Algorithm 7

Before proving the convergence property of the Alg.7, we define the critical points for the non-convex and non-smooth functions [15].

Theorem 4.3.1. [*Convergence Property*] *The sequence $\{(\mathbf{C}^{(k)}, \mathbf{D}^{(k)})\}$ generated by the algorithm 7, is a Cauchy sequence and converges to the critical point of (4.5).*

Proof. See Appendix A. ■

4.4 Experiments

We used the proposed incoherent dictionary learning method in sparse coding based recognition systems. The basic procedure is as follows. Firstly, the dictionary is learned from the training set using Alg. 7. Then, the sparse code C for each sample in the training set, as well as the test set, is calculated using the proximal alternating algorithm [64]. At last, a linear classifier is trained and tested on the sparse codes. Two applications are considered in the experiments: face recognition and object classification. The experimental results showed that using the incoherent dictionary learned from the proposed method, the sparse coding based recognition systems may have some additional performance gain.

4.4.1 Experimental setting

The performance is evaluated on two applications: face recognition on the Extended YaleB dataset [38] and the AR face dataset [60], and object classification on the Caltech-101 dataset [36]. Our approach is compared to two dictionary learning based methods:

- *K-SVD (Baseline)* [1] : The basic procedure is similar to ours, i.e., the dictionary is trained using K-SVD and the sparse codes are used to train a linear classifier. The dictionary learning process and the classifier training process are independent.
- *D-KSVD* [95] : This method is an extension of the above baseline method, which incorporates the classification error into the objective function of K-SVD dictionary

learning. The dictionary and the linear classifier are trained simultaneously.

Note that both methods are built upon the K-SVD dictionary learning method [1] which does not impose dictionary incoherence, and all the tested methods are based on a simple linear classifier. The experimental setting is as follows:

- *Extended Yale B* : The extended YaleB database [38] contains 2,414 images of 38 human frontal faces under about 64 illumination conditions and expressions. There are about 64 images for each person. The original images were cropped to 192×168 pixels. Each face image is projected into a 504-dimensional feature vector using a random matrix of zero-mean normal distribution. The database is randomly split into two halves. One half was used for training the dictionary which contains 32 images for each person, and the other half was used for testing.
- *AR Face* : The AR face database [60] consists of over 4000 frontal images from 126 individuals. For each individual, 26 pictures were taken in two separate sessions. The main characteristic of the AR database is that it includes frontal views of faces with different facial expressions, lighting conditions and occlusion conditions. A subset of the database consisting of 2,600 images from 50 male subjects and 50 female subjects is used. For each person, twenty images are randomly picked up for training and the remaining images are for testing. Each face image is cropped to 165×120 and then projected onto a 540-dimensional feature vector.
- *Caltech101* : The Caltech101 dataset [36] contains 9,144 images from 102 classes (i.e., 101 object categories with 8677 images and one additional “background category with 467 images) including vehicles, plants, animals, cartoon characters, and so on. The number of images in each category varies from 31 to 800. We use 20 samples per category for training the dictionary as well as the classifier and the rest for testing. The spatial pyramid feature presented in [95] is computed on each image as input.

To obtain reliable results, each experiment is repeated 30 times with different random splits of the training and testing images. The final classification accuracies are reported as

Table 4.1: Classification accuracies (%) on two face datasets and one object dataset.

Dataset	K-SVD	D-KSVD	Ours
Extended YaleB	93.10	94.10	95.72
AR Face	86.50	88.80	96.18
Caltech-101	68.70	68.60	72.29

the average of each run. Throughout the experiments, we fix the sparsity parameter λ to be 0.005 and the coherence parameter β to be 1. The iteration number K in Alg. 7 is fixed to be 10. The dictionary size is set 540 on the two face datasets and 3000 on the Caltech-101 dataset.

4.4.2 Experimental results

The results and the conclusions are summarized as follows.

- **Convergence behavior.** The convergence behaviors of the K-SVD method and Alg. 7 on the YaleB face dataset are compared in Fig. 4.1, which plots the Frobenius norm of the increments of the sparse codes generated by two algorithms at each iteration. It can be seen that the code sequence generated by the K-SVD method does not converge to zero, which means that the K-SVD method has at most sub-sequence convergence. In contrast, the increments of the code sequence generated by Alg. 7 converges to zero which shows that the whole sequence converges.
- **Mutual coherence of dictionary.** The matrices of the mutual coherence of the dictionaries learned from the YaleB dataset are shown in Fig. 4.3, and its normalized histograms are shown in Fig. 4.2. It can be seen that mutual coherence of the dictionary from our approach can be significantly lower than that from the K-SVD method when the regularization parameter β on mutual coherence is set sufficiently large.
- **Classification performance.** The classification results are listed in Table 4.1. It can be seen that our approach performs slightly better than the compared methods.

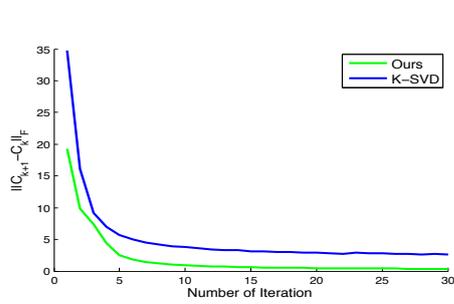


Figure 4.1: The increments of the sequences generated by the methods.

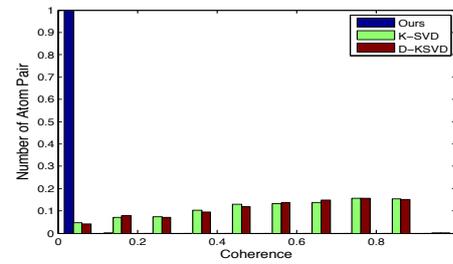


Figure 4.2: The normalized histograms on the coherence matrices shown in Fig. 4.3.

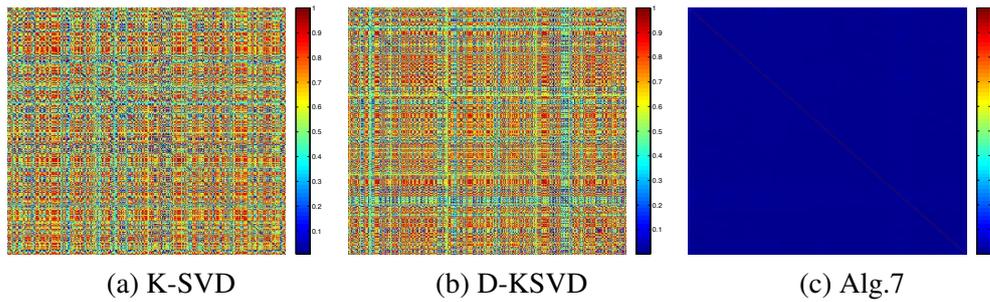


Figure 4.3: The mutual coherence matrices of the dictionaries learned from the YaleB face dataset using the K-SVD method and Alg.7. The i th-column and j th-row element in each matrix represents the mutual coherence between the i th and j -th atom.

4.5 Summary and conclusions

This chapter aims at developing an alternating iteration scheme for learning an incoherent dictionary, which is the first available incoherent dictionary learning method with proved sequence convergence. The proposed work not only is of theoretical interest from the viewpoint of optimization, but also might be useful to practical sparse coding based applications.

Appendix A

In this appendix, we give a detailed proof of Theorem 7. The proof of Theorem 7 is built upon Theorem 2.9 in [4].

Theorem 4.5.1. ([4]) *Assume $H(z)$ is a proper and lower semi-continuous function with $\inf H > -\infty$, the sequence $\{z^{(k)}\}_{k \in \mathbb{N}}$ is a Cauchy sequence and converges to the critical point of $H(z)$, if the following four conditions hold:*

(P1) **Sufficient decrease condition.** *There exists some positive constant ρ_1 , such that*

$$H(z^{(k)}) - H(z^{(k+1)}) \geq \rho_1 \|z^{(k+1)} - z^{(k)}\|_F^2, \quad \forall k = 1, 2, \dots$$

(P2) **Relative error condition.** *There exists some positive constant $\rho_2 > 0$, such that*

$$\|w^{(k+1)}\|_F \leq \rho_2 \|z^{(k+1)} - z^{(k)}\|_F, \quad w^{(k)} \in \partial H(z^{(k)}), \quad \forall k = 1, 2, \dots$$

(P3) **Continuity condition.** *There exists a subsequence $\{z^{(k_j)}\}_{j \in \mathbb{N}}$ and \bar{z} such that*

$$z^{(k_j)} \rightarrow \bar{z}, \quad H(z^{(k_j)}) \rightarrow H(\bar{z}), \quad \text{as } j \rightarrow +\infty.$$

(P4) **$H(z)$ is a KL function.** *$H(z)$ satisfies the Kurdyka-Lojasiewicz property in its effective domain.*

Let $\mathbf{Z}^{(k)} := (\mathbf{C}^{(k)}, \mathbf{D}^{(k)})$ denote the sequence generated by the algorithm 7. Firstly, it can be seen that the objective function $H(\mathbf{Z}) = F(\mathbf{C}) + Q(\mathbf{Z}) + G(\mathbf{D})$ is the proper, lower semi-continuous function and bounded below by 0 where F, Q, G are defined in (4.11). Secondly, the sequence $\{\mathbf{Z}^{(k)}\}_{k \in \mathbb{N}}$ generated by algorithm 7 is bounded since $\mathbf{D}^{(k)} \in \mathcal{D}$ and $\mathbf{C}^{(k)} \in \mathcal{C}$ for all $k = 1, 2, \dots$. In the next, we show that the sequence $\{\mathbf{Z}^{(k)}\}$ satisfies the condition (P1)-(P4) using the following four lemmas.

Lemma 4.5.2. *The sequence $\{\mathbf{Z}^{(k)}\}_{k \in \mathbb{N}}$ satisfies*

$$\begin{cases} H(\mathbf{T}_j^{(k+1)}, \mathbf{D}^{(k)}) \leq H(\mathbf{T}_{j-1}^{(k+1)}, \mathbf{D}^{(k)}) - \frac{\mu_j^k}{2} \|\mathbf{c}_j^{(k+1)} - \mathbf{c}_j^{(k)}\|_F^2, \\ H(\mathbf{C}^{(k+1)}, \mathbf{V}_j^{(k+1)}) \leq H(\mathbf{C}^{(k+1)}, \mathbf{V}_{j-1}^{(k+1)}) - \frac{\lambda_j^k - L(\mathbf{d}_j^{(k)})}{2} \|\mathbf{d}_j^{(k+1)} - \mathbf{d}_j^{(k)}\|_F^2, \end{cases} \quad (4.21)$$

for $1 \leq j \leq m$, where

$$\begin{cases} \mathbf{T}_j^{(k)} = (\mathbf{c}_1^{(k)\top}, \dots, \mathbf{c}_j^{(k)\top}, \mathbf{c}_{j+1}^{(k-1)\top}, \dots, \mathbf{c}_m^{(k-1)\top})^\top, \quad \mathbf{T}_0^{(k)} = \mathbf{C}^{(k-1)}, \\ \mathbf{V}_j^{(k)} = (\mathbf{d}_1^{(k)}, \dots, \mathbf{d}_j^{(k)}, \mathbf{d}_{j+1}^{(k-1)}, \dots, \mathbf{d}_m^{(k-1)}), \quad \mathbf{V}_0^{(k)} = \mathbf{D}^{(k-1)}. \end{cases} \quad (4.22)$$

Proof. From (4.12), we know

$$\mathbf{c}_j^{(k)} \in \arg \min_{\mathbf{c}_j \in \mathcal{C}} F(\bar{\mathbf{c}}_j^k) + Q(\mathbf{U}_j^k, \mathbf{D}^{(k-1)}) + \frac{\mu_j^k}{2} \|\mathbf{c}_j - \mathbf{c}_j^{(k-1)}\|_F^2, \quad (4.23)$$

By the optimality of $\mathbf{c}_j^{(k)}$ in (4.23), we have

$$F(\mathbf{c}_j^k) + Q(\mathbf{T}_j^{(k)}, \mathbf{D}^{(k-1)}) + \frac{\mu_j^k}{2} \|\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}\|_F^2 \leq F(\mathbf{c}_{j-1}^k) + Q(\mathbf{T}_{j-1}^{(k)}, \mathbf{D}^{(k-1)}).$$

Sum $G(\mathbf{D}^{(k-1)})$ on both sides of the above inequality, we have the first inequality in (4.21).

From (4.16), we know

$$\mathbf{d}_j^{(k)} \in \arg \min_{\mathbf{d}_j \in \mathcal{D}} G(\mathbf{S}_j^k) + \langle \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_{j-1}^{(k)}), \mathbf{d}_j - \mathbf{d}_j^{(k-1)} \rangle + \frac{\lambda_j^k}{2} \|\mathbf{d}_j - \mathbf{d}_j^{(k-1)}\|_F^2.$$

The above inequality implies

$$G(\mathbf{d}_j^k) + \langle \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_{j-1}^k), \mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)} \rangle + \frac{L(\mathbf{d}_j^k)}{2} \|\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}\|_F^2 \leq G(\mathbf{V}_{j-1}^k). \quad (4.24)$$

From (4.18), we have

$$Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k) \leq Q(\mathbf{C}^{(k)}, \mathbf{V}_{j-1}^k) + \langle \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_{j-1}^k), \mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)} \rangle + \frac{L(\mathbf{d}_j^k)}{2} \|\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}\|_F^2. \quad (4.25)$$

Together with (4.24), the second inequality in (4.21) is satisfied. \blacksquare

Sum up the above inequalities, we can obtain

$$\begin{aligned} & H(\mathbf{C}^{(k)}, \mathbf{D}^{(k)}) - H(\mathbf{C}^{(k+1)}, \mathbf{D}^{(k+1)}) \\ & \geq \sum_{j=1}^m \left(\frac{\mu_j^k}{2} \|\mathbf{c}_j^{(k+1)} - \mathbf{c}_j^{(k)}\|_F^2 + \frac{\lambda_j^k - L(\mathbf{d}_j^{(k)})}{2} \|\mathbf{d}_j^{(k+1)} - \mathbf{d}_j^{(k)}\|_F^2 \right). \end{aligned} \quad (4.26)$$

Using the fact that there exist $a, b > 0$ such that $a < \mu_j^k, \lambda_j^k < b$ and $\lambda_j^k > L(\mathbf{d}_j^{(k)})$, we can establish the sufficient decreasing property (P1) for $\{\mathbf{Z}^{(k)}\}_{k \in \mathbb{N}}$.

Lemma 4.5.3. Let $\mathbf{w}_{\mathbf{C}}^{(k)} = (\mathbf{w}_{\mathbf{C}}^{1\top}, \dots, \mathbf{w}_{\mathbf{C}}^{m\top})^\top$ and $\mathbf{w}_{\mathbf{D}}^{(k)} = (\mathbf{w}_{\mathbf{D}}^1, \dots, \mathbf{w}_{\mathbf{D}}^m)$ where

$$\begin{cases} \mathbf{w}_{\mathbf{C}}^j = \nabla_{\mathbf{c}_j} Q(\mathbf{Z}^{(k)}) - \nabla_{\mathbf{c}_j} Q(\mathbf{T}_j^{(k)}, \mathbf{D}^{(k-1)}) - \mu_j^k (\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}), \\ \mathbf{w}_{\mathbf{D}}^j = \nabla_{\mathbf{d}_j} Q(\mathbf{Z}^{(k)}) - \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^{(k)}) - \lambda_j^k (\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}), \end{cases} \quad (4.27)$$

and $(\mathbf{T}_j^{(k)}, \mathbf{V}_j^{(k)})$ is defined in (4.22). Then, $\mathbf{w}^k := (\mathbf{w}_{\mathbf{C}}^{(k)}, \mathbf{w}_{\mathbf{D}}^{(k)}) \in \partial H(\mathbf{Z}^{(k)})$ and there exists a constant $\rho > 0$, such that

$$\|\mathbf{w}^k\|_F \leq \rho \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F.$$

Proof. The optimality condition of (4.12) is

$$\nabla_{\mathbf{c}_j} Q(\mathbf{T}_j^k, \mathbf{D}^{(k-1)}) + \mu_j^k (\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}) + \mathbf{u}_j^k = 0, \quad (4.28)$$

where $u_j^k \in \partial_{\mathbf{c}_j} F(\mathbf{T}_j^k)$. Therefore, the following holds

$$\mathbf{u}_j^k = -(\nabla_{\mathbf{c}_j} Q(\mathbf{T}_j^k, \mathbf{D}^{(k-1)}) + \mu_j^k(\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)})) \quad (4.29)$$

Since $F(\mathbf{C}) = \|\mathbf{C}\|_0 = \sum_{j=1}^m \|\mathbf{c}_j\|_0$, we have $\mathbf{u}_j^k \in \partial_{\mathbf{c}_j} F(\mathbf{C}^{(k)})$. From (??), it is easy to know $\mathbf{u}_j^k + \nabla_{\mathbf{c}_j} Q(\mathbf{Z}^{(k)}) \in \partial_{\mathbf{c}_j} H(\mathbf{Z}^{(k)})$. Therefore, we have

$$\nabla_{\mathbf{c}_j} Q(\mathbf{Z}^{(k)}) - \nabla_{\mathbf{c}_j} Q(\mathbf{T}_j^k, \mathbf{D}^{(k-1)}) - \mu_j^k(\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}) \in \partial_{\mathbf{c}_j} H(\mathbf{Z}^{(k)}).$$

Similarly, by optimality condition of (4.16), we have

$$\nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k) + \lambda_j^k(\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}) + \mathbf{v}_j^k = 0, \quad (4.30)$$

where $\mathbf{v}_j^k \in \partial_{\mathbf{d}_j} G(\mathbf{V}_j^k)$. Since $\mathcal{D} = \bigcap_{j=1}^m \{\mathbf{D} : \|\mathbf{d}_j\|_2 = 1\}$, we have $\mathbf{v}_j^k \in \partial_{\mathbf{d}_j} G(\mathbf{D}^{(k)})$. From (??), we know $\mathbf{v}_j^k + \nabla_{\mathbf{d}_j} Q(\mathbf{Z}^{(k)}) \in \partial_{\mathbf{d}_j} H(\mathbf{Z}^{(k)})$. Consequently, we have

$$\nabla_{\mathbf{d}_j} Q(\mathbf{Z}^{(k)}) - \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k) - \lambda_j^k(\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}) \in \partial_{\mathbf{d}_j} H(\mathbf{Z}^{(k)}).$$

Since $\mathbf{C}^{(k)} \in \mathcal{C}$ and $\mathbf{D}^{(k)} \in \mathcal{D}$ for all $k \in \mathbb{N}$, the sequence $\{\mathbf{Z}^{(k)}\}_{k \in \mathbb{N}}$ is a bounded sequence. Let $\{\mathbf{Z}^{(k)}\} \subseteq \mathcal{Z}$, the following inequality holds: there exists $L > 0$, such that

$$\|\nabla_{\mathbf{Z}} Q(\mathbf{Z}_1) - \nabla_{\mathbf{Z}} Q(\mathbf{Z}_2)\|_F \leq L \|\mathbf{Z}_1 - \mathbf{Z}_2\|_F, \quad \forall \mathbf{Z}_1, \mathbf{Z}_2 \in \mathcal{Z}, \quad (4.31)$$

since Q has Lipschitz continuous gradient. Therefore, we have

$$\begin{aligned} \|\mathbf{w}_{\mathbf{C}}^j\| &\leq \mu_j^k \|\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}\|_F + \|\nabla_{\mathbf{c}_j} Q(\mathbf{Z}^{(k)}) - \nabla_{\mathbf{c}_j} Q(\mathbf{T}_j^k, \mathbf{d}^{(k-1)})\|_F \\ &\leq b \|\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}\|_F + L \left(\sum_{i=j}^m \|\mathbf{c}_i^{(k)} - \mathbf{c}_i^{(k-1)}\| + \|\mathbf{d}^{(k)} - \mathbf{d}^{(k-1)}\|_F \right) \\ &= (b + (m-j)L) \|\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}\|_F + L \|\mathbf{d}^{(k)} - \mathbf{d}^{(k-1)}\|_F \\ &\leq ((m+1)L + b) \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F \end{aligned} \quad (4.32)$$

Similarly, we also have

$$\begin{aligned}
\|\mathbf{w}_D^j\| &\leq \lambda_j^k \|\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}\|_F + \|\nabla_{\mathbf{d}_j} Q(\mathbf{Z}^{(k)}) - \nabla_{\mathbf{d}_j} Q(\mathbf{C}^{(k)}, \mathbf{V}_j^k)\|_F \\
&\leq b \|\mathbf{d}_j^{(k)} - \mathbf{d}_j^{(k-1)}\|_F + L \left(\sum_{i=j}^m \|\mathbf{d}_i^{(k)} - \mathbf{d}_i^{(k-1)}\|_F \right) \\
&\leq (mL + b) \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F
\end{aligned} \tag{4.33}$$

Therefore, by $\mathbf{w}^k = (\mathbf{w}_C^{(k)}, \mathbf{w}_D^{(k)})$, we have

$$\|\mathbf{w}^k\|_F = \sum_{j=1}^m \|\mathbf{w}_C^j\|_F + \|\mathbf{w}_D^j\|_F \leq \rho \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F, \tag{4.34}$$

where $\rho = m((2m + 1)L + 2b)$. ■

Lemma 4.5.4. *The sequence $\{\mathbf{Z}^{(k)}\}_{k \in \mathbb{N}}$ satisfies the Continuity condition (P3).*

Proof. Since $\mathbf{C}^{(k)} \in \mathcal{C}$ and $\mathbf{D}^{(k)} \in \mathcal{D}$ for all $k \in \mathbb{N}$, the sequence $\{\mathbf{Z}^{(k)}\}_{k \in \mathbb{N}}$ is a bounded sequence and there exists a sub-sequence $\{\mathbf{Z}^{(k_j)}\}_{j \in \mathbb{N}}$ such that $\mathbf{Z}^{(k_j)} \rightarrow \bar{\mathbf{Z}} = (\mathbf{U}, \bar{\mathbf{D}})$. Since $\mathbf{Z}^{(k_{j-1})}$ is also a bounded sequence, without loss of generality, assume $\mathbf{Z}^{(k_{j-1})} \rightarrow \bar{\mathbf{Z}}_1$. In the next, we first show that $\bar{\mathbf{Z}} = \bar{\mathbf{Z}}_1$. By the lemma 4.5.2, we have

$$H(\mathbf{Z}^{(k-1)}) - H(\mathbf{Z}^{(k)}) \geq \rho_1 \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F^2,$$

where $\rho_1 > b$. So, $H(\mathbf{Z}^{(k)})$ is a decreasing sequence and from the fact that $H(\mathbf{Z}^{(0)}) < +\infty, H(\mathbf{Z}) \geq 0$, we have $\lim_{k \rightarrow +\infty} H(\mathbf{Z}^{(k)}) = \bar{H}$, where \bar{H} is some constant. Summing from $k = 0$ to N , we have

$$H(\mathbf{Z}^{(0)}) - H(\mathbf{Z}^{(N)}) \leq \rho_1 \sum_{k=1}^N \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F^2,$$

let $N \rightarrow +\infty$ in the above, we have

$$\sum_{k=1}^{+\infty} \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F^2 \leq \frac{H(\mathbf{Z}^{(0)}) - \bar{H}}{\rho_1} < +\infty,$$

which implies $\lim_{k \rightarrow +\infty} \|\mathbf{Z}^{(k)} - \mathbf{Z}^{(k-1)}\|_F = 0$. So, for any $\varepsilon > 0$, there exists $J \in \mathbb{N}$, such that for all $j > N$, $\|\mathbf{Z}^{(k_j)} - \mathbf{Z}^{(k_j-1)}\|_F < \varepsilon/2$ and $\|\mathbf{Z}^{(k_j)} - \bar{\mathbf{Z}}\|_F < \varepsilon/2$. It implies

$$\|\mathbf{Z}^{(k_j-1)} - \bar{\mathbf{Z}}\|_F \leq \|\mathbf{Z}^{(k_j)} - \mathbf{Z}^{(k_j-1)}\|_F + \|\mathbf{Z}^{(k_j)} - \bar{\mathbf{Z}}\|_F < \varepsilon.$$

Consequently, $\mathbf{Z}^{(k_j-1)} \rightarrow \bar{\mathbf{Z}}$ as $j \rightarrow +\infty$.

Let $F(\mathbf{C}) = \sum_{j=1}^m f_j(\mathbf{c}_j)$, where $f_j(\mathbf{c}_j) = \|\mathbf{c}_j\|_0$. From (4.12), we have for all k ,

$$\mathbf{c}_j^{(k)} \in \arg \min_{\mathbf{c}_j \in \mathcal{C}} f_j(\mathbf{c}_j) + Q(\mathbf{U}_j^k, \mathbf{D}^{(k-1)}) + \frac{\mu_j^k}{2} \|\mathbf{c}_j - \mathbf{c}_j^{(k-1)}\|_F^2,$$

Let $\mathbf{c}_j = \mathbf{u}_j$ in the above inequality, we have

$$f_j(\mathbf{c}_j^{(k)}) + Q(\mathbf{T}_j^k, \mathbf{D}^{(k-1)}) + \frac{\mu_j^k}{2} \|\mathbf{c}_j^{(k)} - \mathbf{c}_j^{(k-1)}\|_F^2 \leq f_j(\mathbf{u}_j) + Q(\mathbf{U}_j^k, \mathbf{D}^{(k-1)}) + \frac{\mu_j^k}{2} \|\mathbf{u}_j - \mathbf{c}_j^{(k-1)}\|_F^2, \quad (4.35)$$

where $\mathbf{U}_j^k = (\mathbf{c}_1^{(k)\top}, \dots, \mathbf{c}_{j-1}^{(k)\top} \mathbf{u}_j^\top, \mathbf{c}_{j+1}^{(k-1)\top}, \dots, \mathbf{c}_m^{(k-1)\top})^\top$. Choose $k = k_j$ and let $j \rightarrow +\infty$ in (4.35), using the fact that $\mathbf{Z}^{(k_j-1)} \rightarrow \bar{\mathbf{Z}}$, we have

$$\limsup_{j \rightarrow +\infty} f_j(\mathbf{c}_j^{(k_j)}) \leq f_j(\mathbf{u}_j).$$

Since f_j is a lower semicontinuous function, we have $\lim_{j \rightarrow +\infty} f_j(\mathbf{c}_j^{(k_j)}) = f_j(\mathbf{u}_j)$. By the same argument, we have for all $j = 1, \dots, m$, $\lim_{j \rightarrow +\infty} f_j(\mathbf{c}_j^{(k_j)}) = f_j(\mathbf{u}_j)$. Since Q is a smooth function and $G(\mathbf{D}^{(k)}) = 0, \forall k \in \mathbb{N}$, we have

$$\lim_{j \rightarrow +\infty} Q(\mathbf{Z}^{(k_j)}) = Q(\bar{\mathbf{Z}}), \quad \lim_{j \rightarrow +\infty} G(\mathbf{D}^{(k_j)}) = G(\bar{\mathbf{D}}).$$

This implies

$$\lim_{j \rightarrow +\infty} H(\mathbf{Z}^{(k_j)}) = \lim_{j \rightarrow +\infty} F(\mathbf{C}^{(k_j)}) + Q(\mathbf{Z}^{(k_j)}) + G(\mathbf{D}^{(k_j)}) = H(\bar{\mathbf{Z}}).$$

■

For the property (P4), see [15] for the definition. An important class of functions that satisfies the Kurdyka-Lojasiewicz property is the so-called semi-algebraic functions [15].

Theorem 4.5.5. ([15]) *Let f is a proper and lower semicontinuous function. If f is semi-algebraic then it satisfies the K-L property at any point of $\text{dom}f$.*

Lemma 4.5.6. *All the function $F(\mathbf{C})$, $Q(\mathbf{Z})$ and $G(\mathbf{D})$ defined in (4.11) are semi-algebraic functions. Moreover, $H(\mathbf{Z}) = F(\mathbf{C}) + Q(\mathbf{Z}) + G(\mathbf{D})$ is the semi-algebraic function.*

Proof. For $Q(\mathbf{C}, \mathbf{D}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{C}\|_F^2 + \frac{\alpha}{2}\|\mathbf{D}^\top \mathbf{D} - \mathbf{I}\|_F^2$ is a real polynomial function, $Q(\mathbf{C}, \mathbf{D})$ is a semi-algebraic function [15].

It is easy to notice that the set $\mathcal{D} = \{\mathbf{Y} \in \mathbb{R}^{n \times m} : \|\mathbf{d}_k\|_2 = 1, 1 \leq k \leq m\} = \bigcap_{k=1}^m \{\mathbf{Y} : \sum_{j=1}^n y_{kj}^2 = 1\}$ is a semi-algebraic set. And the set $\mathcal{C} = \{\mathbf{C} \in \mathbb{R}^{m \times p} : \|\mathbf{c}_k\|_\infty \leq M\} = \bigcup_{j=1}^M \bigcup_{k=1}^p \{\mathbf{C} : \|\mathbf{c}_k\|_\infty = j\}$ is a semi-algebraic set. Therefore, the indicator functions $\delta_{\mathcal{D}}(\mathbf{C})$ and $\delta_{\mathcal{D}}(\mathbf{D})$ are semi-algebraic functions from the fact that the indicator function for semi-algebraic sets are semi-algebraic functions [2].

For the function $F(\mathbf{C}) = \|\mathbf{C}\|_0$. The graph of F is $S = \bigcup_{k=0}^{mp} L_k \triangleq \{(\mathbf{C}, k) : \|\mathbf{C}\|_0 = k\}$. For each $k = 0, \dots, mp$, let $\mathcal{S}_k = \{J : J \subseteq \{1, \dots, mp\}, |J| = k\}$, then $L_k = \bigcup_{J \in \mathcal{S}_k} \{(\mathbf{C}, k) : \mathbf{C}_{J^c} = 0, \mathbf{C}_J \neq 0\}$. It is easy to know the set $\{(\mathbf{C}, k) : \mathbf{C}_{J^c} = 0, \mathbf{C}_J \neq 0\}$ is a semi-algebraic set in $\mathbb{R}^{m \times p} \times \mathbb{R}$. Thus, $F(\mathbf{C}) = \|\mathbf{C}\|_0$ is a semi-algebraic function since the finite union of the semi-algebraic set is still semi-algebraic.

Consequently, $H(\mathbf{Z})$ is a semi-algebraic function since the finite summation of semi-algebraic functions are still semi-algebraic [15].

■

Chapter 5

Sparse coding based visual tracking

5.1 Introduction

Visual tracking has been an active research topic in computer vision community as it is widely applied in the automatic object identification, automated surveillance, vehicle navigation and many others. Despite great progresses in last two decades, due to many factors in real life, many challenging problems still remains when designing a practical visual tracking system. For example, sophisticated object shape or complex motion, illumination changes and occlusions all may cause serious stability issues for a visual tracker (see a more detailed discussion in [92]).

Recently, sparse representation and compressed sensing technique (e.g. [21, 32]) for finding a sparse solution of an under-determined linear system has drawn a great deal of attention in both mathematics and many applied fields, including visual tracking [47, 51, 61, 62, 89]. Similar to sparsity-based approach for face recognition developed in [86], these tracking methods express a target by a sparse linear combination of the templates in the template space, i.e., the target is well approximated by the linear combination of only a few templates. Benefitting from the stable recovery capability of sparse signal using the ℓ_1 norm minimization (e.g. [21]), these trackers have demonstrated good robustness in various tracking environments.

In the L1 tracker first proposed by [61], hundreds of ℓ_1 norm related minimization prob-

lems need to be solved for each frame during the tracking process. The solver for the ℓ_1 norm minimizations used in [61] is based on the interior point method which turns out to be too slow for tracking. A minimal error bounding strategy is introduced [62] to reduce the number of particles, equal to the number of the ℓ_1 norm minimizations for solving. A speed up by four to five times is reported in [62], but it is still far away from being real time. An efficient solver for the ℓ_1 norm related problems has been the key to use the L1 tracker in practice.

Moreover, in the existing L1 tracker, trivial templates are included in the template dictionary such that its sparse linear combination will present the occlusions and image noise in the target. However, as we empirically observed, the sparse linear combination of the trivial templates sometimes include parts of the object in the target, which will result in a loss of tracking accuracy in some sequences.

Built upon the same framework of the L1 tracker [61, 62], this chapter aims at developing a more robust L1 tracker which runs in real time. There are two main contributions in the proposed approach. One is the introduction of a new ℓ_1 norm related minimization model which empirically showed improvements on the tracking accuracy over the model used in [61]. The other more significant contribution is the introduction of a very fast numerical method to solve the resulting ℓ_1 norm minimization problems which leads to a real time L1 tracker. It is noted that the ℓ_1 minimization problem shown in [61] is just a special case of our ℓ_1 minimization problem. Thus, the proposed numerical method can also be applied to the original L1 tracker to make it a real-time tracker.

5.2 Related work

Among many approaches for real world visual tracking problem, discriminative tracking and generative tracking are two different categories with different formulations. Tracking problem is formulated as a binary classification problem in discriminative tracking methods. Discriminative trackers locate the object region by finding the best way to separate object from background; see e.g. [5, 6, 85, 93]. In [5], a feature vector is constructed for every

pixel in the reference image and an adaptive ensemble of classifiers is trained to separate pixels that belong to the object from the ones in the background. Online multiple instance learning is used in [6] to achieve robustness to occlusions and other image corruptions. Sparse Bayesian learning is used in [85]. Global mode seeking is used in [93] to detect the object after total occlusion and reinitialize the local tracker.

Generative tracking method is based on the appearance model of target object. Tracking is done via searching target location with best matching score by some metric; see e.g. eigentracker [12], mean shift tracker [26], incremental tracker [73] and covariance tracker [66]. To adapt to pose and illumination changes of the object, appearance model is often dynamically updated during the tracking.

Sparse representation have been applied to tracking problem in [61], and later exploited in [51]. In [61], a tracking candidate is sparsely represented by target templates and trivial templates. In [51], group sparsity is integrated and very high dimensional image features are used for improving tracking robustness. In these approaches, the sparse representation is obtained via solving a ℓ_1 -norm related minimization problem [61] or ℓ_0 -norm related minimization in [50, 51]. It is well known that ℓ_0 -norm related minimization is an NP-hard problem. The large-scale ℓ_1 -norm related minimization is also a challenging problem due to the non-differentiability of ℓ_1 norm. The numerical methods for solving ℓ_1 -norm related minimization in [61] is based on the interior point method [46], which is very slow when solving large-scale ℓ_1 -norm minimizations.

In recent years, there have been great progresses on fast numerical methods for solving large-scale ℓ_1 -norm related minimization problems arising in image science, such as Linearized Bregman iteration [17], Split Bregman method [39] etc. Meanwhile, Yang *et al.* [91] has done a comprehensive study of the ℓ_1 norm related minimization on robust face recognition. Among all these methods, one promising approach is the so-called accelerated proximal gradient (APG) method introduced by [84] for minimizing the summation of one smooth function and one non-differential function. The APG method is used in [77] to solve a unconstrained ℓ_1 norm related problem related to image restoration.

5.3 Introduction to L1 Tracker

Our tracker is closely related to the L1 tracker proposed by Mei and Ling [61]. The main differences lie in a different minimization model and a much faster numerical solver for the resulting ℓ_1 norm minimization problems. We first give a brief review on the L1 tracker within the particle filter framework proposed in [61, 62].

Particle Filter: The particle filter provides an estimate of posterior distribution of random variables related to Markov chain. In visual tracking, it gives an important tool for estimating the target of next frame without knowing the concrete observation probability. It consists of two steps: prediction and update. Specially, at the frame t , denote \mathbf{x}_t which describes the location and the shape of the target, $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{t-1}\}$ denotes the observation of the target from the first frame to the frame $t-1$. Particle filter proceeds two steps with following two probabilities:

$$p(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1},$$

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})}.$$

The optimal state for the frame t is obtained according to the maximal approximate posterior probability: $\mathbf{x}_t^* = \arg \max_{\mathbf{x}} p(\mathbf{x} | \mathbf{y}_{1:t})$.

The posterior probability (2) is approximated by using finite samples $\mathbf{S}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \dots, \mathbf{x}_t^N\}$ with different weights $\mathbf{W} = \{\mathbf{w}_t^1, \mathbf{w}_t^2, \dots, \mathbf{w}_t^N\}$ where N is the number of samples. The samples are generated by sequential importance distribution $\Pi(\mathbf{x}_t | \mathbf{y}_{1:t}, \mathbf{x}_{1:t-1})$ and weights are updated by:

$$\mathbf{w}_t^i \propto \mathbf{w}_{t-1}^i \frac{p(\mathbf{y}_t | \mathbf{x}_t^i) p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)}{\Pi(\mathbf{x}_t | \mathbf{y}_{1:t}, \mathbf{x}_{1:t-1})}. \quad (5.1)$$

In the case of $\Pi(\mathbf{x}_t | \mathbf{y}_{1:t}, \mathbf{x}_{1:t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1})$, the equation (5.1) has a simple form $\mathbf{w}_t^i \propto \mathbf{w}_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_t^i)$. Then, the weights of some particles maybe keep increasing and fall into the degeneracy case. To avoid such a case, in each step, samples are re-sampled to generate new sample set with equal weights according to their weights distribution.

Sparse Representation: The sparse representation model aims at calculating the observation likelihood for sample state \mathbf{x}_t , i.e. $p(\mathbf{z}_t|\mathbf{x}_t)$. At the frame t , given the target template set $\mathbf{T}_t = [\mathbf{t}_t^1, \mathbf{t}_t^2, \dots, \mathbf{t}_t^n]$, let $\mathbf{S}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \dots, \mathbf{x}_t^N\}$ denote the sampled states and let $\mathbf{O}_t = \{\mathbf{y}_t^1, \mathbf{y}_t^2, \dots, \mathbf{y}_t^N\}$ denote the corresponding candidate target patch in target template space. The sparse representation model is then:

$$\mathbf{y}_t^i = \mathbf{T}_t \mathbf{a}_T^i + I \mathbf{a}_I^i, \quad \forall \mathbf{y}_t^i \in \mathbf{O}_t, \quad (5.2)$$

where I is the trivial template set (identity matrix) and $\mathbf{a}_T^i = [\mathbf{a}_T^i; \mathbf{a}_I^i]$ is sparse. Additionally, nonnegative constraints are imposed on \mathbf{a}_T^i for the robustness of the L1 tracker [61]. Consequently, for each candidate target patch \mathbf{y}_t^i , the sparse representation of \mathbf{y}_t^i can be found via solving the following ℓ_1 -norm related minimization with nonnegative constraints:

$$\min_{\mathbf{a}} \frac{1}{2} \|\mathbf{y}_t^i - A\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1, \quad \mathbf{a} \succcurlyeq 0, \quad (5.3)$$

where $A = [\mathbf{T}_t, I, -I]$.

Finally, the observation likelihood of state \mathbf{x}_t^i is given as

$$p(\mathbf{z}_t|\mathbf{x}_t^i) = \frac{1}{\Gamma} \exp\{-\alpha \|\mathbf{y}_t^i - \mathbf{T}_t \mathbf{c}_T^i\|_2^2\}, \quad (5.4)$$

where α is a constant controlling the shape of the Gaussian kernel, Γ is a normal factor and \mathbf{c}_T^i is the minimizer of (5.3) restricted to \mathbf{T}_t . Then, the optimal state \mathbf{x}_t^* of frame t is obtained by

$$\mathbf{x}_t^* = \arg \max_{\mathbf{x}_t^i \in \mathbf{S}_t} p(\mathbf{z}_t|\mathbf{x}_t^i). \quad (5.5)$$

In addition, a template update scheme is adopted in [61] to overcome pose and illumination changes.

Minimal Error Bound: In [61], the ℓ_1 -norm related minimization problem (5.3) is solved by the interior point method which is very slow. A minimal error bounding method is then

proposed in [62] to reduce the number of needed ℓ_1 minimizations. Actually, their method is based on the following observation:

$$\|\mathbf{T}_t \mathbf{a} - \mathbf{y}\|_2^2 \geq \|\mathbf{T}_t \hat{\mathbf{a}} - \mathbf{y}\|_2^2, \quad \forall \mathbf{a} \in \mathbb{R}^N, \quad (5.6)$$

where

$$\hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \|\mathbf{T}_t \mathbf{a} - \mathbf{y}\|_2^2. \quad (5.7)$$

Consequently, for any samples \mathbf{x}_t^i , its observation likelihood has the following upper bound:

$$p(\mathbf{z}_t | \mathbf{x}_t^i) \leq \frac{1}{\Gamma} \exp\{-\alpha \|\mathbf{T}_t \hat{\mathbf{a}} - \mathbf{y}_t^i\|_2^2\} \triangleq q(\mathbf{z}_t | \mathbf{x}_t^i), \quad (5.8)$$

where $q(\mathbf{y}_t^i | \mathbf{x}_t^i)$ is the probability upper bound for state \mathbf{x}_t^i . It is seen that if $q(\mathbf{z}_t | \mathbf{x}_t^i) < \frac{1}{2N} \sum_{j=1}^{i-1} p(\mathbf{z}_t | \mathbf{x}_t^j)$, then the sample \mathbf{x}_t^i will not appear in the resample set. In other words, \mathbf{x}_t^i can be discarded without being processed. Thus, a two-stage resample method is proposed in [62] to significantly reduce the number of samples needed in tracking.

5.4 Real time L1 Tracker

Even though the minimal error bound [62] was proposed to reduce the computation load for L1 tracker, there are still many ℓ_1 -norm related minimizations for solving during the tracking process. For example, in the sequence *car* with 620 frames, around 80,000 ℓ_1 -norm related minimizations (5.3) needs to be solved with minimal error bound resampling scheme in [62]. Therefore, the speed bottleneck in the L1 tracker is how to solve the ℓ_1 -norm related minimization (5.3) much faster, in the scale of hundreds of times.

Also, as seen in the model (5.3), the trivial templates are included in the template dictionary such that its sparse linear combination will represent the occlusions and image noise in the target. However, as we observed in the experiments, the sparse linear combination will sometimes include parts of the object in the target which may lead to a loss of tracking accuracy in some sequences.

In this section, we first proposed a modified version of the minimization problem (5.3)

such that the sparse linear combination of trivial templates can represent the occlusions and image noise more accurately. Then, based on the accelerated proximal gradient approach [84], we proposed a fast numerical method for solving the resulting ℓ_1 norm related minimization problem such that the tracker runs in real time. It is noticed that the developed method is also applicable to original minimization problem in (5.3).

5.4.1 A modified ℓ_1 norm related minimization model

There are two types of templates in the template dictionary used by (5.3): target templates and trivial templates. The target templates are updated dynamically for representing target objects during the tracking process. The trivial templates (identity matrix I) is for representing occlusions, background and noise. However, since parts of objects may also be represented by the trivial templates, the region detected by the original tracker sometimes does not fit the target very accurately.

We take a modified version of (5.3) for improving tracking accuracy. The new model is based on the following observation. When there are no occlusions, the target in the next frame should be well approximated by a sparse linear combination of target templates with a small residual. Thus, the energy of the coefficients in \mathbf{a} associated with trivial templates, named *trivial coefficients*, should be small. On the other hand, when there exist noticeable occlusions, the target in the next frame cannot be well approximation by any sparse linear combination of target templates, the large residual (corresponding to occlusions, background and noise in an ideal situation) will be compensated by the part from the trivial templates, which leads to a large energy of the trivial coefficients. The minimization (5.3) is obviously not optimal since it does not differentiate these two cases.

In other words, to optimize the usage of the trivial templates in the tracking, we need to adaptively control the energy of the trivial coefficients. That is, when occlusions are negligible, the energy associated with trivial templates should be small. When there are noticeable occlusions, the energy should be allowed to be large. This motivation leads to

the following minimization model for L1 tracker

$$\min_{\mathbf{a}} \frac{1}{2} \|\mathbf{y} - A'\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 + \frac{\mu_t}{2} \|\mathbf{a}_I\|_2^2, \quad \text{s.t. } \mathbf{a}_T \succeq 0, \quad (5.9)$$

where $A' = [\mathbf{T}_t, I]$, $\mathbf{a} = [\mathbf{a}_T; \mathbf{a}_I]$ are the coefficients associated with target templates and trivial templates respectively, and the parameter μ_t is a parameter to control the energy in trivial templates. In our implementation, the value of μ_t for each state is automatically adjusted using the occlusion detection method [62]. That is, if occlusions are detected, $\mu_t = 0$; otherwise μ_t is set as some pre-defined constant.

The benefit of the additional ℓ_2 norm regularization term $\|\mathbf{a}_I\|_2^2$ is illustrated in 1. 5.1. In Fig. 5.1, about 30 percent of object energy is contained in trivial templates from minimization (5.3). In other words, trivial templates can not distinguish the object and background. On the other hand, we can see the trivial templates coefficients from minimization (5.9) are small and lead to better tracking results. At last, we note that the original minimization (5.3) is a special case of the minimization (5.9) by setting $\mu_t = 0$.

5.4.2 Fast numerical method for solving (5.9)

The proposed method for solving the minimization problem (5.9) is based on the accelerated proximal gradient (APG) approach [84].

APG approach. The APG method is originally designed for solving the following unconstrained minimization:

$$\min F(\mathbf{a}) + G(\mathbf{a}), \quad (5.10)$$

where $F(\mathbf{a})$ is an differentiable convex function with Lipschitz continuous gradient¹ and $G(\mathbf{a})$ is a non-smooth but convex function. The outline of the APG method is given in Algorithm 8. The efficiency of the APG method is justified by its quadratic convergence; see Theorem 5.4.1. However, we emphasize here that the APG method is fast only for

¹the gradient of F is Lipschitz continuous if $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, for some constant L .

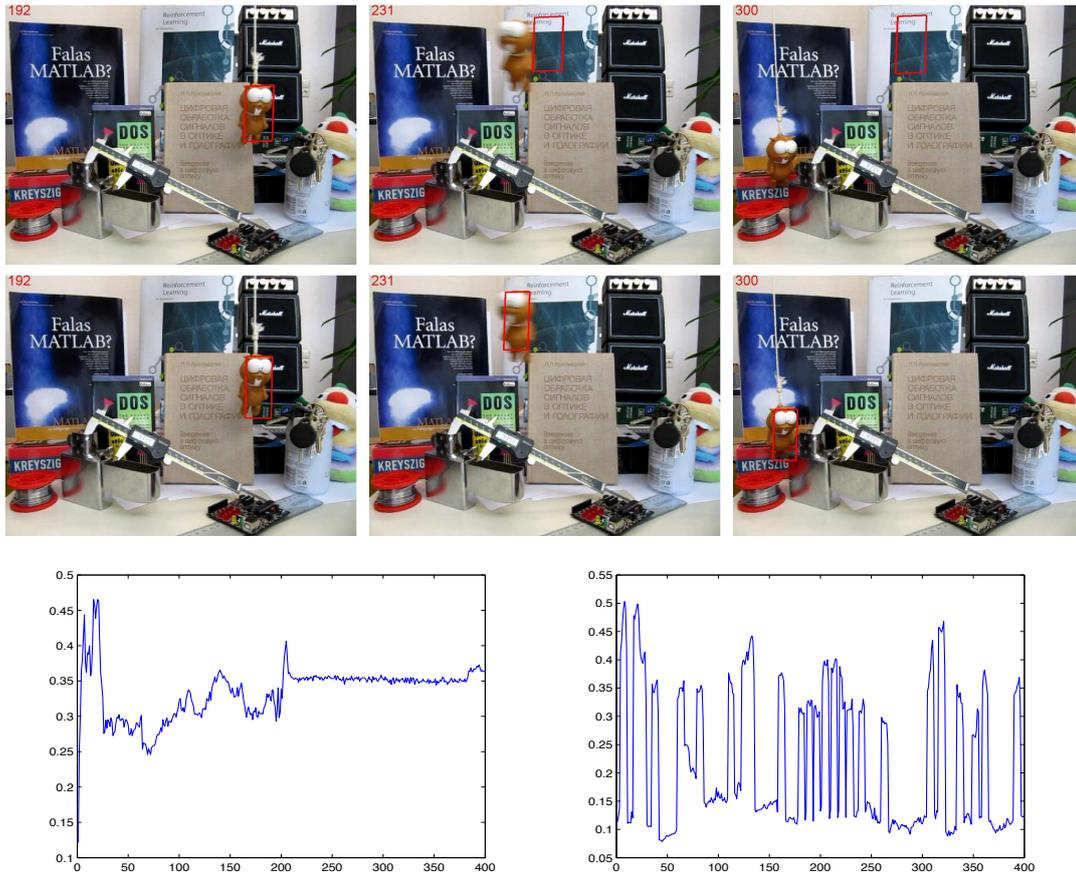


Figure 5.1: Illustration of the L1 tracker on the sequence *lemming* using the model (5.3) and the L1 tracker using the proposed model (5.9). The first and the second row: results using (5.3) and using (5.9) respectively. Last row: the energy ratio $\|\mathbf{a}_l\|_2/\|\mathbf{a}\|_2$. The left graph is from (5.3) and the right is from (5.9).

particular type of function G . During each iteration of Algorithm 1, we need to solve a minimization in Step 2. So, the quadratic convergence of APG is materialized only when the sub-problem in Step 2 has an analytic solution.

Theorem 5.4.1. ([84]) *Let $\{\alpha_k\}$ is the sequence generated by Algorithm 8. Then within $K = O(\sqrt{L/\varepsilon})$ iterations, $\{\alpha_k\}$ achieves ε -optimality such that $\|\alpha_K - \alpha^*\| < \varepsilon$, where α^* is one minimizer of (5.10).*

Algorithm 8 the generic APG approach in [84]

1. Set $\alpha_0 = \alpha_{-1} = \mathbf{0} \in \mathbb{R}^N$ and set $t_0 = t_{-1} = 1$.
2. For $k = 0, 1, \dots$, iterate until convergence

$$\begin{cases} \beta_{k+1} := \alpha_k + \frac{t_{k-1}-1}{t_k}(\alpha_k - \alpha_{k-1}); \\ \alpha_{k+1} := \arg \min_{\mathbf{a}} \frac{L}{2} \|\mathbf{a} - \beta_{k+1} + \frac{\nabla F(\beta_{k+1})}{L}\|_2^2 + G(\mathbf{a}); \\ t_{k+1} := \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \end{cases} \quad (5.11)$$

Reformulation of (5.9) for applying APG method. As we see, the original APG method is designed for unconstrained minimization problem which can not be directly applied to (5.9). Thus, we need to convert the constrained minimization model into an unconstrained problem. Let $\mathbf{1} \in \mathbb{R}^N$ denote the vector with all entries are equal to 1 and let $\mathbf{1}_{\mathbb{R}_+^N}(\mathbf{a})$ denote the indicator function defined by

$$\mathbf{1}_{\mathbb{R}_+^N}(\mathbf{a}) = \begin{cases} 0, & \mathbf{a} \succeq 0; \\ +\infty, & \text{otherwise.} \end{cases} \quad (5.12)$$

It is easy to see that the minimization (5.9) is equivalent to the following minimization problem:

$$\arg \min_{\mathbf{a}} \frac{1}{2} \|\mathbf{y} - A'\mathbf{a}\|_2^2 + \lambda \mathbf{1}_T^\top \mathbf{a}_T + \lambda \|\mathbf{a}_I\|_1 + \frac{\mu_t}{2} \|\mathbf{a}_I\|_2^2 + \mathbf{1}_{\mathbb{R}_+^n}(\mathbf{a}_T). \quad (5.13)$$

Then, the APG method can be applied to (5.13) with

$$\begin{aligned} F(\mathbf{a}) &= \frac{1}{2} \|\mathbf{y} - A'\mathbf{a}\|_2^2 + \lambda \mathbf{1}_T^\top \mathbf{a}_T + \frac{\mu_t}{2} \|\mathbf{a}_I\|_2^2, \\ G(\mathbf{a}) &= \|\mathbf{a}_I\|_1 + \mathbf{1}_{\mathbb{R}_+^n}(\mathbf{a}_T). \end{aligned} \quad (5.14)$$

All steps in Algorithm 8 are trivial except Step 2, in which we need to solve an optimization problem:

$$\alpha_{k+1} = \arg \min_{\mathbf{a}} \frac{L}{2} \|\mathbf{a} - \beta_{k+1} + \frac{\nabla F(\beta_{k+1})}{L}\|_2^2 + G(\mathbf{a}). \quad (5.15)$$

For general function G , it cannot be directly solved. However, in our setting, we have the analytic solution for (5.15); see Proposition 5.4.2. The algorithm for solving ℓ_1 -norm related minimization (5.9) is given in Algorithm 9.

Proposition 5.4.2. *If $F(\mathbf{a})$ and $G(\mathbf{a})$ are defined in (5.14), then the minimization problem (5.15) has the following solution:*

$$\begin{aligned} \alpha_{k+1}|_T &= \max(0, g_{k+1}|_T) \\ \alpha_{k+1}|_I &= \mathfrak{T}_{\lambda/L}(g_{k+1}|_I). \end{aligned} \quad (5.16)$$

where $g_{k+1} = \beta_{k+1} - \frac{\nabla F(\beta_{k+1})}{L}$ and \mathfrak{T} is the soft-thresholding operator: $\mathfrak{T}_\lambda(x) = \text{sign}(x) \max(|x| - \lambda, 0)$.

Proof. The optimization problem (5.15) is expressed as follows,

$$\min_{\mathbf{a}} \frac{L}{2} \|\mathbf{a} - g_{k+1}\|_2^2 + \mathbf{1}_{\mathbb{R}_+^n}(\mathbf{a}_T) + \|\mathbf{a}_I\|_1. \quad (5.17)$$

Since the variables of \mathbf{a} are independent, (5.17) is the same as

$$\begin{aligned} \min_{\mathbf{a}_T} \frac{L}{2} \|\mathbf{a}_T - g_{k+1}|_T\|_2^2 + \mathbf{1}_{\mathbb{R}_+^n}(\mathbf{a}_T), \\ \min_{\mathbf{a}_I} \frac{L}{2} \|\mathbf{a}_I - g_{k+1}|_I\|_2^2 + \lambda \|\mathbf{a}_I\|_1. \end{aligned} \quad (5.18)$$

It is easy to see the solution of first minimization in (5.18) is the projection of $g_{k+1}|_T$ to the \mathbb{R}_+^n space, i.e. $\max(0, g_{k+1}|_T)$. For the second minimization in (5.18), all the variables are

independent. So, we only need to solve the following minimization :

$$\min_x \frac{L}{2} \|y - x\|_2^2 + \lambda \|x\|_1 \triangleq f(x), \quad (5.19)$$

where $x, y \in \mathbb{R}$. The minimizer of (5.19) can be expressed as a soft thresholding operation:

$$x = \mathfrak{T}_{\lambda/L}(y) = \text{sgn}(y) * \max(|y| - L, 0). \quad (5.20)$$

Thus, we have $\mathbf{a}_I = \mathfrak{T}_{\lambda/L}(g_{k+1}|_I)$ as the minimizer of (5.18). ■

Algorithm 9 Real Time Numerical algorithm for solving the minimization (5.9)

(i) Set $\alpha_0 = \alpha_{-1} = \mathbf{0} \in \mathbb{R}^N$ and set $t_0 = t_{-1} = 1$.

(ii) For $k = 0, 1, \dots$, iterate until convergence

$$\left\{ \begin{array}{l} \beta_{k+1} := \alpha_k + \frac{t_{k-1}-1}{t_k} (\alpha_k - \alpha_{k-1}); \\ g_{k+1}|_T := \beta_{k+1}|_T - (A'^T (A' \beta_{k+1} - \mathbf{y}))|_T / L - \lambda \mathbf{1}_T; \\ g_{k+1}|_I := \beta_{k+1}|_I - (A'^T (A' \beta_{k+1} - \mathbf{y}))|_I / L \\ \quad - \mu \beta_{k+1}|_I / L; \\ \alpha_{k+1}|_T := \max(0, g_{k+1}|_T); \\ \alpha_{k+1}|_I := \mathfrak{T}_{\lambda/L}(g_{k+1}|_I); \\ t_{k+1} := (1 + \sqrt{1 + 4t_k^2})/2. \end{array} \right.$$

Tight Lipschitz constant L estimation. There is only one parameter, the Lipschitz constant L of ∇F , is involved in Algorithm 9. This Lipschitz constant L plays a crucial role in the above algorithm. Algorithm 8 with an wrong L will either diverges or converges very slowly. Next, we give a tight upper bound of L for F defined in (5.14) such that L is automatically set with optimal performance; see Proposition 5.4.3. The detailed description of the proposed real time L1 tracker, called *APG-LI* tracker, is given in algorithm 10.

Proposition 5.4.3. *Let F denote the function defined in (5.14) with $A' = [T, I]$, where T is template set and I is the identity matrix. The upper bound of the Lipschitz constant L for ∇F is given as follows,*

$$L \leq \lambda_{\max}^2 + \mu_t + 1, \quad (5.21)$$

Algorithm 10 APG-L1 Tracker

```

1: Input:
2: Current frame  $F_t$ ;
3: Sample Set  $\mathbf{S}_{t-1} = \{\mathbf{x}_{t-1}^i\}_{i=1}^N$ ;
4: Template set  $\mathbf{T} = \{\mathbf{t}_i\}_{i=1}^n$ .
5: for  $i = 1$  to  $N$  do
6:   Drawing the new sample  $\mathbf{x}_t^i$  from  $\mathbf{x}_{t-1}^i$ ;
7:   Preparing the candidate patch  $\mathbf{y}_t^i$  in template space;
8:   Solving the least square problem (5.7);
9:   Computing  $q_i$  according to (5.8);
10: end for
11: Sorting the samples in descent order according to  $q$ ;
12: Setting  $i = 1$  and  $\tau = 0$ .
13: while  $i < N$  and  $q_i \geq \tau$  do
14:   Solving the minimization (5.9) via Algorithm 9;
15:   Computing the observation likelihood  $p_i$  in (5.4);
16:    $\tau = \tau + \frac{1}{2N} p_i$ ;
17:    $i = i + 1$ ;
18: end while
19: Set  $p_j = 0, \forall j \geq i$ .
20: Output:
21: Finding the  $\mathbf{x}_t^*$  according to (5.5);
22: Detecting the occlusion [62] and update  $\mu$  in (5.9);
23: Updating the template set  $\mathbf{T}_{t-1}$  [62];
24: Updating the sample set  $\mathbf{S}_{t-1}$  with  $p$ .

```

where λ_{max} is the largest singular value of T .

Proof. From (5.14), we have

$$\nabla^2 F(x) = \begin{pmatrix} T^\top T & T^\top \\ T & (1 + \mu)I \end{pmatrix}. \quad (5.22)$$

Assume $T = U\Sigma V^\top$ by singular value decomposition, where U and V are orthonormal matrices, $\Sigma \in \mathbb{R}^{m \times N}$ ($m < N$) with $\Sigma_{ii} = \lambda_i$ and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$. It is easy to know $\nabla^2 F(x)$ is similar to $M \triangleq \begin{pmatrix} \Sigma^\top \Sigma & \Sigma^\top \\ \Sigma & (1 + \mu)I \end{pmatrix}$. So $\lambda_{Fmax} = \lambda_{Mmax} \leq \lambda_{max}^2 + 1 + \mu$, where λ_{Fmax} , λ_{Mmax} and λ_{max} are the largest singular values of $\nabla^2 F(x)$, M and T respectively. ■

5.5 Experiments

Through the experiments, APG algorithm is implemented with matlab, $\mu_t = 5$ in (5.9) when the occlusion is not detected and 0 otherwise, and $\lambda = 10^{-2}$ in Algorithm 9.

5.5.1 Comparison with the existing L1 Tracker

The computation efficiency and tracking accuracy of the proposed APG-L1 tracker is first compared to that of the BPR-L1 tracker [62] on ten sequences. The average running time of the proposed APG-based solver v.s. the interior point method used [62] is about 1 : 150. As a result, the average running time of the APG-L1 tracker v.s. the BPR-L1 tracker is around 1:20, with 600 particles. The APG-L1 tracker achieves about average 26 frames per second with 600 particles on a PC with Intel i7-2600 CPU (3.4GHz). The output bounding boxes of the target from the two tackers are similar in many sequences, while the results from APG-L1 are more accurate on some challenging sequences.

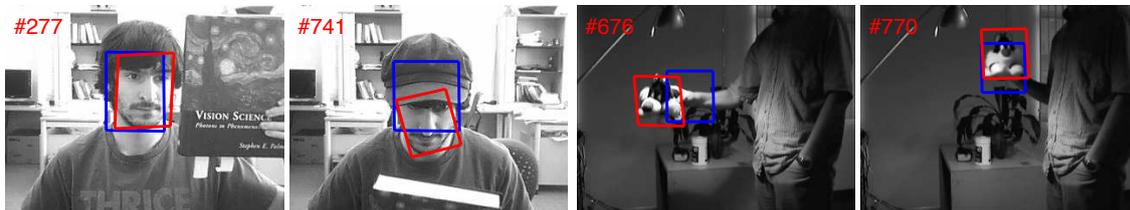


Figure 5.2: Demonstration of the improvement of APG-L1 tracker (red) over BPR-L1 (blue) on tracking accuracy.

5.5.2 Qualitative comparison with other methods

The performance of the proposed APG-L1 tracker is also evaluated on eight publicly available video sequences and is compared with five latest state-of-the-art trackers named Incremental Visual Tracking (IVT) [73], Multiple Instance Learning (MIL) [6], Visual Tracking Decomposition (VTD) [48], Incremental Covariance Tensor Learning (ICTL) [88], and On-line AdaBoost (OAB) [41]. The tracking results of the compared methods were obtained using the codes provided by the authors with the default parameters and using the same

initial positions in the first frame.

The sequence *jump* was captured outdoors. The target was jumping and the motion blurs are very severe. Results on several frames are presented in Fig. 5.4 (a). The APG-L1 tracker, IVT, OAB, and MIL tracks the target faithfully throughout the sequences. The other trackers fails track the target when there are abrupt motion and severe motion blur.

The sequence *car* shows a vehicle undergoes drastic illumination changes as it passes beneath a bridge and under trees. Tracking results on several frames are shown in Fig. 5.4 (b). The APG-L1 tracker and IVT can track the target well despite the drastic illumination changes, while the other trackers lose the target after it goes through the bridge.

Results of the sequence *singer* are shown in Fig. 5.4 (c). In this sequence, we show the robustness of our algorithm in severe illumination changes and large scale variations. Only our APG-L1 tracker and the VTD tracker can track the target throughout the sequence.

In the sequence *woman* (Fig. 5.4 (d)), only the APG-L1 tracker is able to track the target during the entire sequence. The other trackers drift to the man when he occludes the target due to his similar appearance as the target.

In the sequence *pole*, a person is walking away from the camera and is occluded by the pole for a short time (Fig. 5.4 (e)). The IVT loses the target from the start and the VTD starts to drift off the target at frame 274 and finally loses the target. All the rest successfully track the target but our APG-L1 tracker recovers the target scale better.

Results on the sequence *sylv* are shown in Fig. 5.4 (f), where a moving animal doll is undergoing challenging pose variations, lighting changes and scale variations. The IVT, and VTD eventually fails at frame 605 as a result of drastic pose and illumination changes. The rest trackers are able to track the target for this long sequence while our APG-L1 tracker performs with higher accuracy.

Results of the sequence *deer* are shown in Fig. 5.4 (g). In this sequence, we show the robustness of our algorithm in background clutters and the fast motion. Only our APG-L1 tracker and VTD can track the target through the sequence.

Fig. 5.4 (h) shows the results on the sequence *face*. Many trackers start drifting from the target when the man's face is severely occluded by the book. The APG-L1 tracker and IVT

	MIL	OAB	ICTL	VTD	IVT	ours
jump	0.030	0.030	0.198	0.221	0.020	0.025
car	0.749	0.786	0.326	0.313	0.049	0.048
singer	0.299	0.466	0.503	0.056	0.155	0.069
woman	0.361	0.179	0.323	0.339	0.148	0.032
pole	0.007	0.010	0.008	0.049	0.572	0.003
sylv	0.069	0.058	0.096	0.203	0.197	0.032
deer	0.022	0.060	0.306	0.027	0.110	0.017
face	0.120	0.144	0.137	0.209	0.053	0.062
Ave.	0.207	0.217	0.237	0.177	0.163	0.036

Table 5.1: The average tracking errors. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. The last row is the average error for each tracker over all the test sequences.

handle this very well and continue tracking the target when the occlusion disappears.

5.5.3 Quantitative comparison with other methods

To quantitatively evaluate the robustness of the APG-L1 tracker under challenging conditions, we manually annotated the target’s bounding box in each frame for all test sequences. The tracking error evaluation is based on the relative position errors (in pixels) between the center of the tracking result and that of the annotation. As shown in Fig.5.3 and Table 5.1, the APG-L1 tracker achieves comparable to the best performer on the sequence *jump*, *singer* and *face* to the best-performed trackers, and on all the other sequences it performs best.

5.6 Conclusion

In summary, based on the framework of L1 tracker [61, 62], we developed a real time L1 visual tracker with improved tracking accuracy. The accuracy improvement is achieved via a new minimization model for finding the sparse representation of the target and the real time performance is achieved by a new APG based numerical solver for the resulting ℓ_1 norm minimization problems. The experiments also validated the high computational efficiency and better tracking accuracy of the proposed APG-L1 tracker.

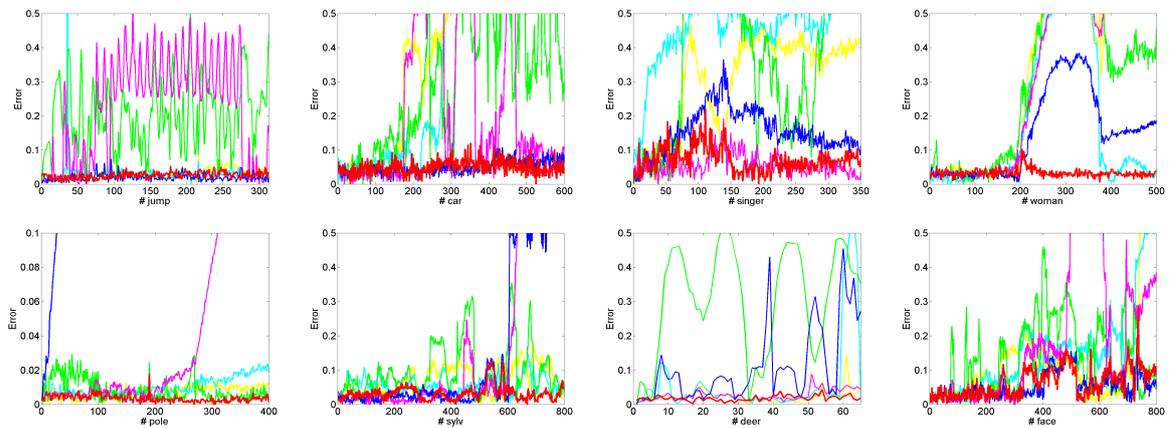


Figure 5.3: The tracking error for each test sequence. The error is measured the same as in Table 5.1 and the legend as in Fig.5.4.

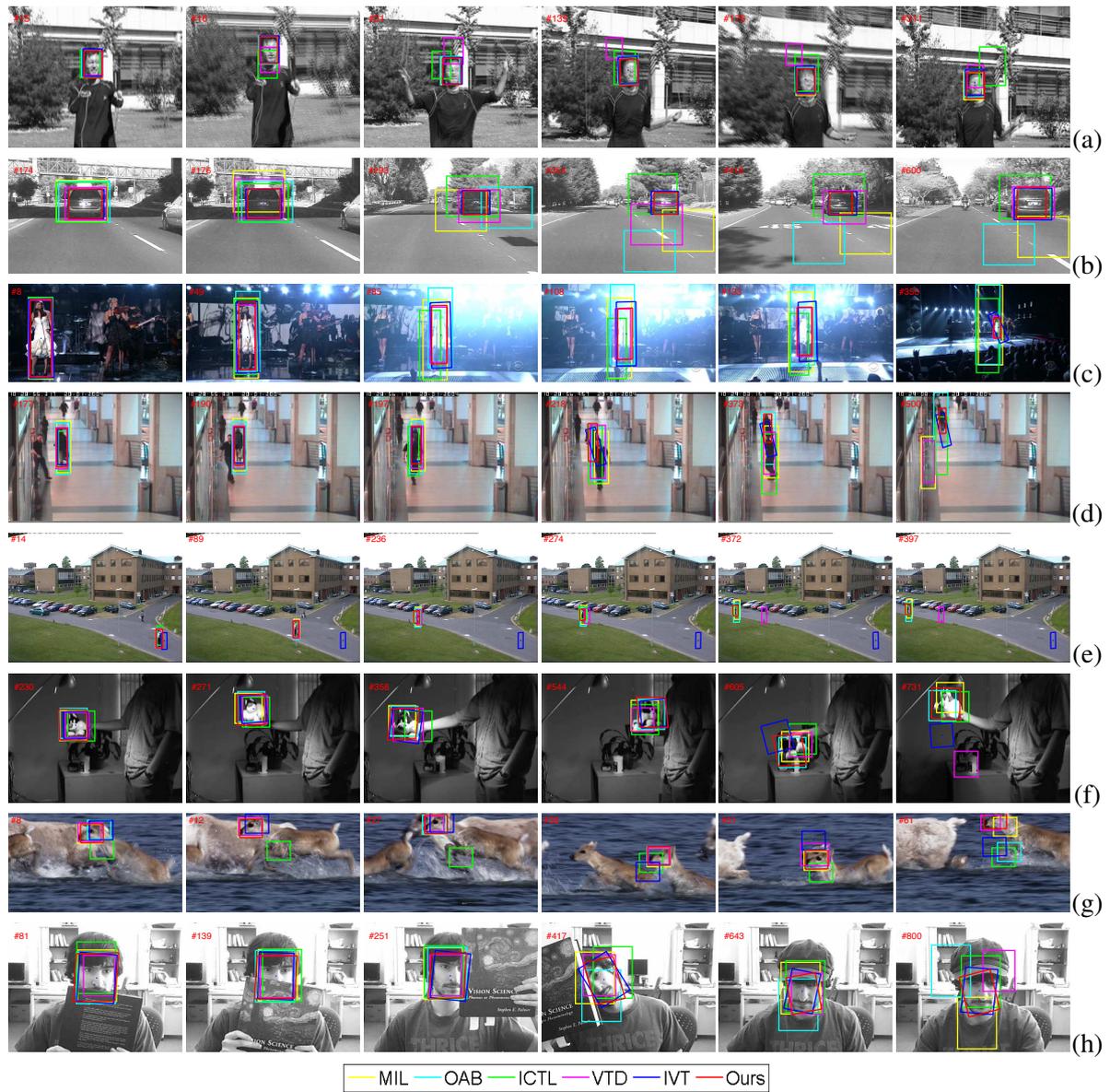


Figure 5.4: Tracking results of different algorithms for sequences *jump*(a), *car*(b), *singer*(c), *woman*(d), *pole*(e), *sylv*(f), *deer*(g) and *face*(h).

Bibliography

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing of over-complete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11): 4311–4322, 2006.
- [2] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Prog.*, 116(1-2):5–16, 2009.
- [3] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Math. Oper. Res.*, 35(2):438–457, 2010.
- [4] H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Math. Program.*, 137(1-2):91–129, 2013.
- [5] S. Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):261–271, 2007.
- [6] B. Babenko, M.H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.
- [7] C. Bao, J.-F. Cai, and H. Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *ICCV*, 2013.
- [8] C. Bao, H. Ji, Y. Quan, and Z. Shen. ℓ_0 norm based dictionary learning by proximal method with global convergence. In *CVPR*, 2014.

- [9] D. Barchiesi and M. D. Plumbley. Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. *IEEE Trans. Signal Process.*, 61(8):2055–2065, 2013.
- [10] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
- [11] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *ACM SIGGRAPH*, 2000.
- [12] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Comput. Vision*, 26(1):63–84, 1998.
- [13] J. Bobin, J.L. Starck, J.M. Fadili, Y. Moudden, and D.L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Trans. Image Process.*, 16(11):2675–2681, 2007.
- [14] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM J. Optimiz.*, 18(2):556–572, 2007.
- [15] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Prog.*, pages 1–36, 2013.
- [16] J. Cai, R. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Appl. Comp. Harm. Anal.*, 24(2):131–149, 2008.
- [17] J.-F. Cai, S. Osher, and Z. Shen. Linearized bregman iterations for compressed sensing. *Math. Comp*, 78(267):1515–1536, 2009.
- [18] J.-F. Cai, B. Dong, S. Osher, and Z. Shen. Image restoration: total variation, wavelet frames, and beyond. *J. Amer. Math. Soc.*, 25(4):1033–1089, 2012.
- [19] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye. Data-driven tight frame construction and image denoising. *Appl. Comput. Harmon. Anal.*, 37(1):89–105, 2014.

- [20] E.J. Candes and D.L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise- C^2 singularities. *Comm. Pure Appl. Math*, 57(2): 219–266, 2002.
- [21] E.J. Candes, J.K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math*, 59(8):1207–1223, 2006.
- [22] R.H. Chan, T.F. Chan, L. Shen, and Z. Shen. Wavelet algorithms for high-resolution image reconstruction. *SIAM J. Sci. Comput.*, 24(4):1408–1432, 2003.
- [23] R.H. Chan, S.D. Riemenschneider, L. Shen, and Z. Shen. Tight frame: An efficient way for high-resolution image reconstruction. *Appl. Comput. Harmon. Anal.*, 17(1): 91–115, 2004.
- [24] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1999.
- [25] R. Coifman and D.L. Donoho. Translation-invariant de-noising. In *Wavelet and Statistics*, Springer Lecture Notes in Statistics. Springer-Verlag., 1994.
- [26] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–577, 2003.
- [27] P. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.
- [28] W. Dai, T. Xu, and W. Wang. Dictionary learning and update based on simultaneous code-word optimization (simco). In *ICASSP*, 2012.
- [29] I. Daubechies. *Ten lectures on wavelets*, volume 61. CBMS-NSF Lecture Notes,SIAM, 1992.
- [30] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based constructions of wavelet frames. *Appl. Comp. Harm. Anal.*, 14(1):1–46, 2003.

-
- [31] C-A. Deledalle, J. Salmon, and A. S. Dalalyan. Image denoising with patch based pca: local versus global. In *BMVC*, 2011.
- [32] D.L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.
- [33] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.
- [34] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process*, 15(12):3736–3745, 2006.
- [35] K. Engan, S.O. Aase, and J.H. Husoy. Method of optimal directions for frame design. In *ICASSP*. IEEE, 1999.
- [36] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Comput. Vis. Image Und.*, 106(1):59–70, 2007.
- [37] J. H. Friedman. Fast sparse regression and classification. *Int. J. Forecasting*, 38(3): 722–738, 2012.
- [38] A. S. Georghiades, P. N. Belhumeur, and D. 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.
- [39] T. Goldstein and S. Osher. The split bregman method for l1 regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.
- [40] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*, 2013.
- [41] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *BMVC*, 2006.

- [42] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for ℓ_1 -regularized minimization with applications to compressed sensing. Technical report, CAAM, 2007.
- [43] Fan J. and Li. R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Assoc.*, 96(456):1348–1360, 2001.
- [44] R. Jenatton, J. Mairal, F.R. Bach, and G.R. Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010.
- [45] Z. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, 2011.
- [46] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1 -regularized least squares. *IEEE J. Sel. Topics Signal Process.*, 8(8): 1519–1555, 2007.
- [47] S. Kwak, W. Nam, B. Han, and J.H. Han. Learning occlusion with likelihoods for visual tracking. In *CVPR*, 2011.
- [48] J. Kwon and K.M. Lee. Visual tracking decomposition. In *CVPR*, 2010.
- [49] M. Lewicki and T. Sejnowski. Learning overcomplete representations. *Neural Comput.*, 12(2):337–365, 2000.
- [50] H. Li, C. Shen, and Q. Shi. Real-time visual tracking using compressive sensing. In *CVPR*, 2011.
- [51] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, 2010.
- [52] B. Mailhé, D. Barchiesi, and M.D. Plumbley. INK-SVD: Learning incoherent dictionaries for sparse representations. In *ICASSP*, 2012.
- [53] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*. IEEE, 2008.

-
- [54] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Process*, 17(1):53–69, 2008.
- [55] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2009.
- [56] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2009.
- [57] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010.
- [58] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.
- [59] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, third edition, 2008.
- [60] A.M. Martínez and R. Benavente. The AR face database. Technical report, Computer Vision Center, 1998.
- [61] X. Mei and H. Ling. Robust visual tracking using l_1 minimization. In *ICCV*, 2009.
- [62] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient l_1 tracker with occlusion detection. In *CVPR*, 2011.
- [63] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6593):607–609, 1996.
- [64] N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends optim.*, 1(3):123–231, 2013.
- [65] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *CVPR*, 2008.

-
- [66] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *CVPR*, 2006.
- [67] A. Rakotomamonjy. Direct optimization of the dictionary learning. *IEEE Trans. Signal Process.*, 2013.
- [68] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, 2010.
- [69] K. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages and Applications*. Academic Press, 1990.
- [70] K. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages and Applications*. Academic Press, 1990.
- [71] R.T. Rockafellar and R.J. Wets. *Variational analysis: grundlehren der mathematischen wissenschaften*, volume 317. Springer, 1998.
- [72] A. Ron and Z. Shen. Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operator. *J. Funct. Anal.*, 1997.
- [73] D.A. Ross, J. Lim, R.S. Lin, and M.H. Yang. Incremental learning for robust visual tracking. *Int. J. Comput. Vision*, 77(1-3):125–141, 2008.
- [74] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Technical report, CS Technion, 2008.
- [75] K. Schnass and P. Vandergheynst. Dictionary preconditioning for greedy algorithms. *IEEE Trans. Signal Process.*, 56(5):1994–2002, 2008.
- [76] Z. Shen. Wavelet frames and image restorations. In *Proceedings of the International Congress of Mathematicians*, 2010.

- [77] Z. Shen, K.C. Toh, and S. Yun. An accelerated proximal gradient algorithm for frame based image restorations via the balanced approach. *SIAM J. Imaging Sci.*, 4(2):573–596, 2011.
- [78] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang. A non-convex relaxation approach to sparse dictionary learning. In *CVPR*, 2011.
- [79] S. Sra. Scalable nonconvex inexact proximal splitting. In *NIPS*, 2012.
- [80] I. Tomic and P. Frossard. Dictionary learning. *IEEE Signal Process. Mag.*, 28(2):27–38, 2011.
- [81] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 50(10):2231–2242, 2004.
- [82] J.A. Tropp, A.C. Gilbert, and M.J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.
- [83] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001.
- [84] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM J. Optimiz.*, 2008.
- [85] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1292–1304, 2005.
- [86] 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, 2008.
- [87] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proc. IEEE*, 98(6):1031–1044, 2010.
- [88] Y. Wu, J. Cheng, J. Wang, and H. Lu. Real-time visual tracking via incremental covariance tensor learning. In *ICCV*, 2009.

-
- [89] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. In *ICCV*, 2011.
- [90] Y. Xu and W. Yin. A fast patch-dictionary method for the whole image recovery. Technical report, UCLA CAM report, 2013.
- [91] A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Fast l_1 -minimization algorithms and an application in robust face recognition: a review. In *ICIP*, 2010.
- [92] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv. (CSUR)*, 38(4):13, 2006.
- [93] Z. Yin and R.T. Collins. Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking. In *CVPR*, 2008.
- [94] C.H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.*, pages 894–942, 2010.
- [95] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, 2010.
- [96] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15(2):265–286, 2006.

