Data-Driven Representation, Learning and Applications:
From Compressed Sensing to Deep Neural Networks

by

Tao Xiong

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2017

© Tao Xiong 2017

All rights reserved
Abstract

As the development of high-density sensors, the compressed sensing (CS) and sparse representation have been successfully implemented for building power-efficient systems by reducing sampling, transmission bandwidth and storage capacity. CS-based systems essentially compress the original signals (e.g. neural spikes, images and videos) into a few measurements, and wirelessly transmit them to the receiver for reconstruction and analysis. Therefore, the hidden simplified sparse structure of the original signals from these measurements becomes the key indicator of the performance of such CS-based systems. On the other hand, recent advances in deep learning have dramatically improved the state-of-the-art and enabled significant progress in solving problems such as speech and visual object recognition. Taking advantage of its exceptional performance in solving computer vision tasks, we also exploit and incorporate deep neural networks to extract the inherent and hierarchy representation behind the large scale database. In future work, approaches to bridge these two techniques and more investigations of deep neural network architectures will be studied.
ABSTRACT

In this thesis, we first briefly review the background of compressed sensing, sparse representation and dictionary learning. Based on these theories, we describe a CS-based multi-channel system for neural recordings and spike sorting and a spatiotemporal CS pixel-wise control imaging system. Both systems are evaluated to demonstrate the better performance in terms of reconstruction quality, compression ratio and power efficiency compared to conventional and other state-of-the-art works. Hardware live demonstration for both systems are presented as well. After discussion on compressed sensing and sparse representation, we start to exploit the deep neural networks for human vertebrae localization and identification. From the perspective of deep learning, we propose an automatic and accurate deep image-to-image network with message passing schemes and shape basis learning for human vertebrae detection. Both convolutional and recurrent neural networks are investigated in this work. The proposed network has outperformed the state-of-the-art works on public challenging database. In addition, we also experimentally show the proposed network can thrive on large-scale databases and its extension to other landmark detection applications.

Primary Reader: Dr. Sang Peter Chin

Secondary Reader: Dr. Trac Duy Tran
Acknowledgments

First and foremost, I would like to thank my advisors, Dr. Sang Peter Chin, Dr. Trac Duy Tran and Dr. Ralph Etienne-Cummings for accepting me into their groups. They have been greatly supportive since the days I began working with them as a first-year graduate student. During my PhD, they contributed to an unique graduate school experience by giving me intellectual freedom in my research, engaging me in novel ideas, demanding a high quality of work in all my endeavors, supporting my trips at many conferences, and allowing me to do internships at various companies. Additionally, I would like to thank my Graduate Board Oral committee members, Dr. Theodore Poehler, Dr. Ernst Niebur, and Dr. Bruno Jedynak. I also would like to thank Dr. Jerry Prince and Dr. Howard Weinert for providing me the opportunities to serve as a teaching assistant. I am also very grateful to my academic advisors Dr. Amy Foster and Dr. Edward Scheinerman for mentoring me during my master programs.

Every result and paper presented in this thesis was accomplished with the help and support of fellow labmates and collaborators. Dr. Jie Zhang and I have been
ACKNOWLEDGMENTS

working together on several compressed sensing projects since I joined the group as a freshman PhD student. I gained a lot from his research skills and keen scientific insight. Without his help, my job in the projects would have definitely been more difficult. I was also fortunate to have the chance to work with Dong Yang and Dr. Daguang Xu in Siemens Corporate Research in Princeton. First of all, I am grateful to Dr. Daguang Xu for referring me to this internship. During the internship, Dong Yang and Dr. Daguang navigated me to the field of deep learning and medical image analysis. I am glad that we continue to work together even after I left Princeton. Without their effort, I would not have had published several papers on the most prestigious conferences in medical imaging.

I would like to thank the various members of the Digital Signal Processing Laboratory: Dr. Xiaoxia Sun, Dr. Shuai Huang, Qing Qu, Dung N. Tran, Sonia Joy, Luoluo Liu, Akshay Rangamani and Arun Nair. I also would like to acknowledge the fellow labmates of the Computational Sensory-Motor Systems Laboratory: Dr. Jamal Molin, Dr. Chetan Singh Thakur, Adam Khalifa, John Rattray, Eleonora Costa and Martina Leistner.

I also would like to acknowledge my mentors and colleagues in Siemens Corporate Research and LG Silicon Valley Lab: Dr. David Liu, Dr. Shaohua Kevin Zhou, Dr. Jenny Zhang, Dr. Peter Song, Dr. Mingqing Chen, Dr. Xiao Chen and many others.

I also would like to extend my gratitude to my classmates and friends in Johns Hopkins University, Peking University and high school.
ACKNOWLEDGMENTS

I am grateful for the funding resources that allowed me to pursue my graduate school studies: The Electrical and Computer Engineering Department Fellowship, The National Science Foundation and The Air Force Office of Scientific Research.

Finally, I would like to express my sincere gratitude to my parents, family and friends who supported me throughout the years. I would not be where I am today without your help and support. Thank you so very much for everything.
Dedication

To my parents.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.3.2</td>
<td>Unsupervised Dictionary Learning for Multi-channel Neural Recordings and Spike Sorting</td>
<td>57</td>
</tr>
<tr>
<td>2.2.3.3</td>
<td>Experiments</td>
<td>68</td>
</tr>
<tr>
<td>2.2.3.4</td>
<td>Conclusion</td>
<td>80</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Hardware Live Demonstration</td>
<td>81</td>
</tr>
<tr>
<td>2.2.4.1</td>
<td>Proposed Demonstration</td>
<td>82</td>
</tr>
<tr>
<td>2.2.4.2</td>
<td>Audience Interaction</td>
<td>84</td>
</tr>
<tr>
<td>2.3</td>
<td>Discussion and Conclusion</td>
<td>85</td>
</tr>
<tr>
<td>3</td>
<td>Compressed Sensing Pixel-Wise Exposure Control Imaging System</td>
<td>88</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>88</td>
</tr>
<tr>
<td>3.2</td>
<td>Methods</td>
<td>91</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Compressed Sensing Block-Wise Exposure Control Algorithm</td>
<td>91</td>
</tr>
<tr>
<td>3.2.1.1</td>
<td>Compressed Sensing Block-Wise Exposure Control</td>
<td>93</td>
</tr>
<tr>
<td>3.2.1.2</td>
<td>Experiments</td>
<td>98</td>
</tr>
<tr>
<td>3.2.1.3</td>
<td>Conclusion</td>
<td>100</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Spatiotemporal Compressed Sensing for Video Compression</td>
<td>102</td>
</tr>
<tr>
<td>3.2.2.1</td>
<td>Compressed Sensing</td>
<td>106</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>Spatiotemporal Compressed Sensing</td>
<td>107</td>
</tr>
<tr>
<td>3.2.2.3</td>
<td>Experiments</td>
<td>109</td>
</tr>
<tr>
<td>3.2.2.4</td>
<td>Conclusion</td>
<td>114</td>
</tr>
</tbody>
</table>
## CONTENTS

3.2.3  Hardware Live Demonstration ................................. 115

3.2.3.1 Proposed Demonstration .................................. 116

3.2.3.2 Audience Interaction ...................................... 116

3.3  Discussion and Conclusion ....................................... 117

4  Deep Learning for Medical Image Analysis .................. 120

4.1  Automatic Localization and Identification of Human Vertebrae using Deep Neural Network ................................ 120

4.1.1  Introduction ..................................................... 120

4.1.2  Methods ......................................................... 125

4.1.2.1 Deep Image-to-Image Neural Network (DI2IN) .......... 126

4.1.2.2 Message Passing and Structure Learning ................ 129

4.1.2.3 Post Refinement Schemes .................................. 135

4.1.3  Experiments ..................................................... 141

4.1.3.1 Deep Image-to-Image Network with Message Passing and Sparse Regularization ................................ 143

4.1.3.2 Deep Image-to-Image Recurrent Neural Network with Shape Basis Learning ........................................ 148

4.1.4  Discussion and Conclusion .................................... 150

5  Conclusion and Future Work .............................. 154

5.1  Conclusion ......................................................... 154
CONTENTS

5.2 Future Work .................................................. 158

Bibliography ...................................................... 162

Vita ................................................................. 180
List of Tables

2.1 Comparison of single channel recovery performance (in SNDR) between different CS methods. ........................................... 33
2.2 Comparison of multi-channel recovery performance (in SNDR) between different CS methods. ........................................... 34
2.3 Comparison of multi-channel classification performance (in CA) between different CS methods. ........................................... 34
2.4 Comparison of reconstruction performance (in SNDR) of “Easy” and “Difficult” spikes between different CS methods. .................. 48
2.5 Comparison of classification performance (in CA) of “Easy” and “Difficult” spikes between different CS methods. .................. 49
2.6 Comparison of reconstruction performance (in SNDR) of different CS methods on “Leicester”. ................................................. 73
2.7 Comparison of classification performance (in CA) of different CS methods on “Leicester”. ................................................. 73
2.8 Comparison of reconstruction and classification performance of different CS methods on “Pogo”. ............................................. 74
2.9 Comparison of reconstruction and classification performance of different CS methods on “Romeo”. ............................................. 74
2.10 Comparison of reconstruction performance (in SNDR) of different CS methods on “hc-1”. .................................................... 78
3.1 Comparison of reconstruction performance (in PSNR and RMSE) of different CS methods with compression ratio of 10. ................. 100
3.2 Comparison of reconstruction performance (in PSNR and RMSE) of different CS methods with compression ratio of 30. ................. 101
3.3 Comparison of reconstruction performance (in PSNR and CR) of “Car” without noise. .......................................................... 111
3.4 Comparison of reconstruction performance (in PSNR and CR) of “Pedestrian” without noise. .................................................... 111
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>Comparison of reconstruction performance (in PSNR and CR) of “Car” with low noise.</td>
<td>111</td>
</tr>
<tr>
<td>3.6</td>
<td>Comparison of reconstruction performance (in PSNR and CR) of “Pedestrian” with low noise.</td>
<td>112</td>
</tr>
<tr>
<td>3.7</td>
<td>Comparison of reconstruction performance (in PSNR and CR) of “Car” with high noise.</td>
<td>112</td>
</tr>
<tr>
<td>3.8</td>
<td>Comparison of reconstruction performance (in PSNR and CR) of “Pedestrian” with high noise.</td>
<td>113</td>
</tr>
<tr>
<td>3.9</td>
<td>Comparison of On-Chip Implementation (in Technology (nm) and Power (nJ/ Pixel))</td>
<td>114</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of localization errors in mm and identification rates among different methods for Set 1</td>
<td>145</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of localization errors in mm and identification rates among different methods for Set 2</td>
<td>146</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of localization errors in mm and identification rates among different methods for 0.7 mm X-ray Set.</td>
<td>147</td>
</tr>
<tr>
<td>4.4</td>
<td>Comparison of localization errors in mm and identification rates among different methods for 0.35 mm X-ray Set.</td>
<td>148</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of localization errors in mm and identification rates among different methods. Our method is trained and tested using default data setting in ”Set 1” and ”Set 2”, while ”+1000” indicates training with additional 1000 labeled spine data and evaluated on the same testing data.</td>
<td>149</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Development of Neural Recording Devices from 1950s to present.</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Schematic of the multi-channel CS neural recording framework for a tetrode setup. The multi-modal structured dictionary learning is the key contribution in this framework.</td>
<td>26</td>
</tr>
<tr>
<td>2.3</td>
<td>An illustration of the shape differences among spikes of three different neurons (color coded) using synthetic dataset from [1].</td>
<td>27</td>
</tr>
<tr>
<td>2.4</td>
<td>Illustration of group structure (color-coded blocks) and joint sparsity (red-dotted lines) for multi-modal structured dictionary learning.</td>
<td>29</td>
</tr>
<tr>
<td>2.5</td>
<td>An example of multi-channel signals (red) and CS recovered signals (blue) using our approach at CR of 50 times and SNDR = 9.17 dB.</td>
<td>35</td>
</tr>
<tr>
<td>2.6</td>
<td>A schematic of the proposed unsupervised CS approach for neural recordings. The unsupervised structured dictionary learning is our key contribution in this framework.</td>
<td>38</td>
</tr>
<tr>
<td>2.7</td>
<td>A illustration of the different classes (color coded spikes) of “Easy” and “Difficult” neural spikes data base from [2].</td>
<td>39</td>
</tr>
<tr>
<td>2.8</td>
<td>An illustration of discriminative group structures (color coded blocks) for unsupervised structured dictionary learning.</td>
<td>41</td>
</tr>
<tr>
<td>2.9</td>
<td>An intuitive schematic of joint prediction of class of neural spike using SRC [3] and Euclidean norm distance.</td>
<td>43</td>
</tr>
<tr>
<td>2.10</td>
<td>An illustration of unsupervised structured dictionary learning algorithm.</td>
<td>45</td>
</tr>
<tr>
<td>2.11</td>
<td>A reconstruction and classification example of “Difficult” spikes using proposed approach at CR of 50 times.</td>
<td>47</td>
</tr>
<tr>
<td>2.12</td>
<td>Basic block diagram of the proposed CS neural recording system. In the CS approach, multi-channel signals are randomly sampled by an on-chip sensing matrix $S$ and then wirelessly transmitted to an off-chip terminal for reconstruction and sorting. The design of the multi-modal dictionary learning $D$ for sparsifying signals is the major contribution of our proposed CS approach.</td>
<td>51</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>2.13</td>
<td>An illustration of the different clusters (color-coded spikes) of neural signals from the Leicester and MGH databases. In spike sorting, spikes are grouped into different clusters based on their distinct shapes.</td>
<td>54</td>
</tr>
<tr>
<td>2.14</td>
<td>Intuitive illustration of the proposed signal model with discriminative group structures (color-coded blocks) and joint-group sparsity (red filled) for multi-mode structured dictionary learning.</td>
<td>56</td>
</tr>
<tr>
<td>2.15</td>
<td>Illustration of different groups of spikes with distinct shapes. The red color-coded spikes indicate the centroids (mean shape) associated with the corresponding groups. The mean shape matching provides another perspective of similarity in the sparse coding stage.</td>
<td>60</td>
</tr>
<tr>
<td>2.16</td>
<td>An illustration of how the trust region performs in principal component analysis (PCA). As Algorithm 6 iterates from 1 to 10, the percentage of spikes in the trust region increases from 29.33% to 92.00%, indicating that most spikes in the training samples satisfy the predefined reconstruction quality after 10 iterations.</td>
<td>63</td>
</tr>
<tr>
<td>2.17</td>
<td>Illustration of the trust region and training error during the iterative training. (a) indicates the percentage change in trust region $S$ and (b) indicates average recovery error as dictionary learning iterates from 1 to 20.</td>
<td>65</td>
</tr>
<tr>
<td>2.18</td>
<td>An example of robustness of spike sorting from the perspective of sparse coding, visualized in the PCA domain. The iterative refinement helps correct the mistakenly sorted spikes generated from the initialization.</td>
<td>66</td>
</tr>
<tr>
<td>2.19</td>
<td>An example of $SNDR_x$ and $SNDR_y$ values for all the recording electrode over 15 weeks of recordings.</td>
<td>68</td>
</tr>
<tr>
<td>2.20</td>
<td>Examples of reconstruction performance of single-channel neural recordings. For (a)-(d), recovered signals (red) still preserve the major features of original signals (blue) at CR of 20:1 and 10:1, respectively. (a) and (b) demonstrate synthetic spikes from the Leicester database, while (c) and (d) demonstrate real spikes from the MGH database.</td>
<td>71</td>
</tr>
<tr>
<td>2.21</td>
<td>Examples of spike sorting performance shown in the PCA domain. (a) and (b) illustrate the spike sorting result of Leicester’s “Easy” and “Difficult” databases at a CR of 20:1. (c) and (d) illustrate the spike sorting result of MGH’s “Pogo” and “Romeo” databases at a CR of 10:1.</td>
<td>75</td>
</tr>
<tr>
<td>2.22</td>
<td>An example of reconstruction performance of multi-channel neural recordings on the hc-1 database at a CR of 8:1. Blue and red spikes indicate the original neural spikes and the recovered neural spikes, respectively.</td>
<td>76</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

2.23 An example of spike sorting performance of multi-channel neural recordings based on the hc-1 database. Figure 11 (a)-(d) indicate the clustering results of channels 1-4, respectively. Different colors represent different clusters. 79

2.24 Proposed demonstration setup. 83

2.25 Micrograph and specifications of the CS-ASIC. 84

2.26 Demonstration of in-vivo experiment using CS-ASIC. 85

2.27 Live demonstration using CS-ASIC on IEEE 2014 Biomedical Circuits and Systems Conference in Lausanne, Switzerland. 86

3.1 Conventional Global Exposure imaging vs. Pixel-wise Coded Exposure imaging. 89

3.2 Reconstructed video scenes using coded exposure images from the sensor. 90

3.3 Block diagram of the proposed compressed sensing CMOS image sensors using optical flow estimation. Video frames are sampled by the exposure-coded pixels. Optical flow pixels are placed within a block of exposure coded pixels. Frames are then compressed into one coded image through exposure controlled pixel. The optical flow pixel measures the scene motion and adjust the exposure time for optimal image quality. One coded image is transmitted wirelessly. The spatial temporal video scene is then reconstructed off-chip. 94

3.4 An intuitive demonstration of optical flow estimation using real dataset. 95

3.5 An intuitive example of block-wise sensing cube control using optical flow estimation. Red and blue arrows indicate the area with high-motion velocity and low-motion velocity respectively, which leads to short exposure duration in red block sensing cube and long exposure duration in blue block sensing cube. 97

3.6 The reconstruction example of one frame using different approaches with a compression ratio of 10 : 1. 101

3.7 The block diagram of spatiotemporal compressed sensing framework. 105

3.8 The demonstration of pixel-wise coded exposure in spatial and temporal domains. 108

3.9 The demonstration of coded images and reconstructed video frames at compression rate 20 : 1. A (red): without noise; B (green): low noise; C (blue): high noise. 113

3.10 Demonstration of (a) chip micrograph and (b) prototype. 117

3.11 Demonstration of GUI in ISCAS 2017. 118

3.12 Demonstration of CS spatiotemoral image sensor in ISCAS 2017. 118

4.1 Demonstration of pathological cases (a) Surgical Metal Implants (b) Spine Curvature (c) Limited FOV. 122
4.2 Proposed deep image-to-image network (DI2IN). The front part is a convolutional encoder-decoder network with feature concatenation, and the backend is multi-level deep supervision network. Numbers next to convolutional layers are channel numbers. Extra 26-channel convolution layers are implicitly used in deep supervision. 128

4.3 Left: CT image. Middle: Output of one channel from the network. Right: Overlaid display. The prediction in (a) is close to ground truth location. In (b), a false positive response exists remotely besides the response at the correct location. 130

4.4 (a) Chain-structure model for vertebra centers; (b) One iteration of message passing (landmarks represents vertebra centers): the neighbors’ landmark probability maps help compensate the missing response of landmark $i$. 131

4.5 The multi-layer ConvLSTM architecture for updating the vertebra response. 134

4.6 Maximum errors of vertebra localization before and after the shape-based refinement. 138

4.7 Probability map examples from DI2IN (left in each case) and ConvLSTM (right in each case). The prediction in “Good Cases” is close to ground truth location. In “Bad Cases”, some false positives exist remotely besides the response at the ground truth location. 140

4.8 Maximum errors of vertebra localization before and after the shape-based neural network in direction $z$. 142

4.9 Maximum errors of vertebra localization in challenging cases before and after the message passing and shape-based network refinement. 151

4.10 Maximum errors of vertebra localization in challenging cases before and after the message passing and shape-based network refinement. 152

4.11 Localization errors (in mm) for the vertebral centroids. The statistical results are for the testing datasets of set 1 and set 2 using the proposed methods with extra 1000 training volumes. ”C” is for cervical vertebrae, ”T” is for thoracic vertebrae, ”L” is for thoracic vertebrae, and ”S” is for sacral vertebrae. 152
Chapter 1

Introduction and Motivation

1.1 Introduction

In a few years, the global data information is projected to be more than 35 zettabytes, which is three times more than the projected storage capacity. Specifically, the wearable sensors and smart phones contribute a very large portion among the Big Data. By 2020, it is projected that every human being will contribute more than 1 terabyte of the sensor data individually. These sensors include but not limited to global position system, accelerometer, gyroscope, cell phone, digital camera and all kinds of biomedical devices. Several challenges arise from the scenario of the Big Data. First, transmitting and processing such a large amount of data requires high power-efficiency in corresponding sensors and devices. As Gordon Moore said “...the number of transistor in a dense integrated circuit doubles approximately every two years”,

1
the high density sensors have been rapidly developed over the decades. However, the battery and energy density does not follow Moore’s Law. Designing such a power efficient and high density sensors that acquire Big Data in power impoverished environments becomes quite challenging. Second, from another perspective, recent advances using deep learning and deep neural networks in the machine learning community have demonstrated that the huge data curse might turn out to be a blessing in the supervised manner: large carefully-labelled training samples combined with massive computational resources lead to numerous breakthroughs from speech recognition to object classification even to chess playing. Hence, taking advantage of such a Big Data and how to learn the structure from raw data in the supervised or unsupervised manner is one of the most basic challenges in machine learning and pattern recognition.

One of powerful tools to tackle the energy-efficient in high density sensors is compressed sensing (CS) technique based on sparse representation. Natural signals are inherently sparse in certain bases or dictionaries where they can be approximately represented by only a few significant components carrying the most relevant information. In other words, the intrinsic signal information is often encoded in the sparse representation. A sparse representation not only provides better signal compression for bandwidth, storage efficiency and power consumption, but also leads to faster processing algorithms as well as more effective signal separation for detection, classification, and other pattern recognition purposes because it focuses on the most
relevant property of the data itself. Sparse signal representation in CS framework allows us to capture the hidden simplified structure present in the data jungle, and thus minimizes the harmful of noises in practical settings. Elegant techniques such as dictionary learning have been employed to find the optimal sparse representations taking into account characteristics of the data. The details of compressed sensing and sparse representation will be discussed in the following section and chapter.

Recently, another technique to explore the intrinsic hidden structure in Big Data is deep neural networks. Deep neural networks are networks that consist of multiple linear and nonlinear layers that encode hierarchical representations of data. Over the last ten years, approaches based on deep neural networks have become state-of-the-arts in machine learning and pattern recognition tasks such as computer vision, speech recognition, machine translation, medical imaging, and many other fields. Most of these methods employ convolutional neural networks (CNNs) which are one kind of neural networks that have convolutional transformations followed by downsampling layers such as max-pooling and nonlinear layers such as rectified linear unit (ReLU) instead of generic linear transformations. Another methods use recurrent neural networks (RNNs) which are adopted to model sequential data. Benefiting from such a huge amount of global data information, deep neural networks have demonstrated its exceptional performance on many tasks. The details of how to take advantage of deep neural networks to model the Big Data will be discussed in the following sections and chapters.
1.2 Motivation of Compressed Sensing and Sparse Representation

High density systems have been widely employed in the modern life. Human beings are enjoying the benefits and convenience from the applications of such systems by using smart phones, laptops and even self-driving cars in common life everyday. In order to achieve the performance of high accuracy, the Big Data plays the most essential role in the evolution of pattern recognition systems. With the development of computer science and electronics industry, the amount of data we are using in our normal life, manufacturing industry and scientific research has increased from the order of megabyte to terabyte to zettabyte in over a few decades. In normal life, the random access memory (RAM) in our smart phone is no longer on the order of megabyte and the RAM of gigabyte has become the common storage for each smart phone nowadays. The amount of data that involved in scientific research also increases dramatically. For instance, in the field of neuroscience, the neural recording system has been developed in the last few decades from one electrode to high-density micro-electrodes array consisting of more than thousands of probes. In order to monitor the real-time brain activities, such a large amount of data to be processed is on the order of several megabyte per second, which raises the limitation for the design of power-efficient neural recording systems. What’s more, from the perspective of computer
vision, the visual data also explodes dramatically due to the development of high-speed and high-resolution imaging systems, which require much more data storage to record the data and more power consumption to transmit and process the data. Therefore, the amount of data on the order of megabyte, gigabyte or even terabyte generated by high-density, high speed and high-resolution imaging systems becomes the bottleneck for efficiently designing such an energy-efficient system.

For example, micro-electrodes array (MEA) neural recording system is designed to simultaneously collect large amount of neural signals by high-density electrodes during in-vivo experiments. It helps neuroscientists deeply explore the correlation between neurons and behaviour. In order to transmit the high-density neural signals in real-time for monitoring the neuron activities, the data needs to be compressed to alleviate the pressure of transmission. Another example is inserts-mounted small imaging system. Taking advantage of VSLI and CMOS technique, the high-speed imaging system is able to be fabricated on a tiny chip of size less than $10 \times 10 \ mm$, which paves a promising way for implementation on alive insects. However, it is not practical to transmit the whole imaging data recorded by insects-mounted imaging systems to the receiver wirelessly, due to large amount of power requirements. In order to address these challenges, the data needs to be compressed before it is transmitted. Compressed sensing is one of the approaches, where only a small amount of original signal is sensed and transmitted.

Compressed sensing (CS) unlike the conventional compression approach, has
been widely used in biomedical devices and systems. As an effective and novel compression approach, compressed sensing does not follow Nyquist-Shannon sampling theorem. Using a sensing matrix $S$, it is able to compress a signal $x$ into a measurement $y$ which is smaller than $x$ and then reconstruct the signal perfectly from the measurement if some conditions are satisfied. Benefiting from the compressed sensing theory, the high-density neural recording micro-system or high-speed imaging system can randomly sense the signal and transmit the information at a high compression ratio (> 10) online for further pattern recognition tasks. However, it is also challenging to extract the hidden feature of the original signal $x$ such as neural potentials or visual information from the limited measurement $y$ of which size is much less than the size of $x$. For instance, spike sorting, which groups neural signals into clusters based on similarity of their shapes, is one of pattern recognition tasks in the neural recording. But how to reconstruct the neural signals and correctly sort the spikes at high compression ratio is the essential challenge to be addressed in CS-based neural recording system. In modern video application, the image sensors also suffer from the same fundamental trade-offs: power consumption vs. frame rate, pixel resolution vs. frame rate and signal-to-noise ratio vs. motion blur. In order the leverage these trade-offs, the theory of CS has emerged to the imaging system to improve spatial and temporal resolution while reducing the power consumption. In such a system, how to sense the image or video effectively is the key guarantee to achieve high frame rate and pixel resolution given the limited power budget.
CHAPTER 1. INTRODUCTION AND MOTIVATION

To meet both requirements of power efficiency and accuracy in pattern recognition, the design of dictionary $D$ and sensing matrix $S$ is the essential contribution in CS-based systems. For example, the dictionary learning with group structure and joint sparsity could further improve reconstruction quality and spike sorting accuracy while achieving high compression ratio. In compressed sensing imaging system, the pixel-wise coded exposure could sense the image or video in a randomly spatial-temporal way but achieve high frame rate and pixel resolution. Our works have demonstrated its success on both CS-based neural recording system and imaging system. My major contribution is the design of dictionary learning and reconstruction approaches for the CS-based systems. The design of hardware is Dr. Jie Zhang’s thesis work. We built the entire systems together for real-time evaluation. The details will be presented in Chapter 2 and Chapter 3.

1.3 Motivation of Deep Neural Networks

Recently, deep learning dramatically improved the state-of-the-art and made significant progress on solving problems such as speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics.\cite{12}

Taking deep structure vision oriented layer design and efficient training scheme into account, current deep learning models have become deeper and deeper and even achieved better performance than human level accuracy on many tasks.
CHAPTER 1. INTRODUCTION AND MOTIVATION

In visual object classification, Krizhevsky et al.\textsuperscript{13} presented a convolutional neural network (CNN) that was composed of only five convolutional layers to classify the 1.2 million high-resolution images into the 1000 different classes. This network, also named AlexNet has achieved top-1 and top-5 error rates of 37.5\% and 17\% in the ImageNet LSVRC-2010 contest, which dominated the previous state-of-the-art at the time. Then, Simonyan et al.\textsuperscript{14} proposed a very deep convolutional networks consisting of 16 to 19 weight layers, which demonstrated around 7\% top-5 error and beat the performance of AlexNet on ILSVRC-2012. In ILSVRC-2014 contest, Google Inc. achieved the new state-of-the-art using GoogLeNet, a 22 layers deep network.\textsuperscript{15} Recently, Microsoft Research won the 1st place on the ILSVRC-2016. ResNet with a depth of up to 152 layers, proposed in He et al.\textsuperscript{16} outperforms other state-of-the-art methods by achieving 3.57\% error on the ImageNet test set.

The deeper convolutional neural networks have demonstrated their exceptional recognition performance, which benefit from the deep layer structure, the back propagation and the huge amount of training samples. The deeper convolutional layers increase the size of the receptive field and capture more contextual and spatial information from the 2D images or 3D videos, which dramatically improve the performance on variant pattern recognition tasks. Many applications especially computer vision problems immediately benefit from the “unreasonable” performance of deep convolutional neural network. Inspired by the unconventional paradigm of deep convolutional neural network, we plan to incorporate deep learning into our future work.
CHAPTER 1. INTRODUCTION AND MOTIVATION

to develop the fast and accurate pattern recognition framework. Our future work will mainly focus on solving computer vision tasks using deep convolutional neural network. Specifically, we develop the deep neural networks framework for automatic and accurate human vertebrae localization and identification in large-scale database. This work was originally inspired during the internship in Siemens Corporate Research. I have been working with Dong Yang and Dr. Daguang Xu on this topic since 2016. Dong and I contributed equally to this work. The details of this work will be discussed in Chapter 4.

Due to the scale of topics involved in this thesis, the background and literature reviews are discussed in each chapter individually, which will be a self-containing part with discussion on how it fits in the overall theme of this thesis. The key techniques and frameworks in which this thesis will address:

- Development and evaluation of compressed sensing multi-channel neural recording system in both supervised and unsupervised fashion in Chapter 2.

- Development and evaluation of spatiotemporal compressed sensing image system in the manner of a pixel-wise exposure control in Chapter 3.


Related papers of my work have been published in smaller parts over a course of
CHAPTER 1. INTRODUCTION AND MOTIVATION

The ultimate goal for this thesis is to explore the hidden structure and information behind the Big Data such as neural signals or medical images with sparse representation and deep neural networks and find the intrinsic relation between these techniques in future work. Chapter 2 and Chapter 3 will focus on the linear and one-level representation while Chapter 4 will discuss the nonlinear and multi-layer representation from the perspective of deep neural networks. As the thesis goes through, we shall see how each individual component contributes to the overall contributions as follows:

Chapter 2 is focusing on reviewing and developing several compressed sensing neural recording systems. We start from the background of compressed sensing theory and previous neural recording systems. Then, we move forward to introduce several compressed sensing based frameworks that have been developed for multi-channel neural recordings and spike sorting. More specifically, we describe three compressed sensing based neural recording systems including a supervised CS framework for multi-channel neural recording, an unsupervised CS framework for spike sorting and an unsupervised CS framework for multi-channel neural recording and spike sorting. We also evaluate the proposed frameworks comprehensively on both synthetic and real neural signal databases and demonstrate the performance compared to other works. In addition, we present the real-time hardware demonstration of proposed CS neural recording system.

Chapter 3 further presents a spatiotemporal compressed sensing imaging system.
CHAPTER 1. INTRODUCTION AND MOTIVATION

The novelty of compressed sensing technique in image systems is able to significantly reduce measurements in both spatial and temporal domain, which successfully balances the trade-offs in modern camera systems. We describe several frameworks in this chapter. First, we introduce the background and fundamental framework of spatiotemporal compressed sensing imaging systems. Then, we explore the advantage of pixel-wise exposure control camera system using optical flow estimation that can be regarded as a special case of spatio-temporal imaging systems. We also present a novel video compression approach from the perspective of spatiotemporal compressed sensing. Finally, we demonstrate the live spatiotemporal compressed sensing camera set up.

Having discussed the compressed sensing technique in neural recording and camera systems, Chapter 4 moves on to the deep learning and deep neural networks to exploit a higher level representation in data. Compared to the compressed sensing theory based on sparse representation, deep neural networks is able to learn more hierarchical representation of data and achieve better performance of recognition such as classification, segmentation and localization. In this chapter, we focus on developing a deep neural networks to solve the human vertebrae localization and labeling task in large-scale medical databases. The proposed deep neural networks is evaluated comprehensively on both large-scale 3D and 2D medical databases, which demonstrates its generic success in other related applications.

Finally, we draw the conclusion in Chapter 5 that summarizes the contribution of
CHAPTER 1. INTRODUCTION AND MOTIVATION

the thesis and discuss my future work.
Chapter 2

Compressed Sensing for Neural Recording

In the brain, neurons communicate with each other through generation of neural action potential, commonly known as spikes. These spikes are short-lasting events generated by neurons when the electrical membrane potential of a neuron rises and falls rapidly. In order to further study the behavior of neurons and brains, neuroscientists have been using neural recording devices such as multi-electrode arrays or high density silicon probes consisting of hundreds or even thousands of recording sites to collect spikes from neurons during \textit{in vivo} experiments. Typically, these experiments require not only the measurements of spikes from one neuron, but also from a group of neurons in a small area of the brain. As a result, multi-electrode arrays or high density silicon probes for neural recording have been designed.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

High-density multi-electrode neural recording microsystems have evolved over the years to become essential tools in neural electrophysiology experiments. As shown in Figure 2.1, the neural recording device has developed from single electrode to high-density multi-electrode in a few decades. These microsystems are able to monitor brain activities by collecting extracellular spikes from different areas of the brain. Using high-density multi-electrode arrays or tetrode drives, the action potential of each neuron can be recorded by multiple electrodes in its proximity. This redundancy of features can greatly improve the performance of neural recording systems such as reconstruction quality and spike sorting accuracy. However, the drawback of such a high-density system is that the large number of electrodes generate large amount of data. This presents itself as a huge challenge for the design of implantable recording system in terms of chip size and power consumption.

Typically, spikes are sampled at around 30 kHz at a resolution of more than 10 bits. Therefore, a multi-channel neural recording system containing up to thousands of channels generates data at the rate of 300 megabyte per second. This would cost around 50 mW to transmit wirelessly, which results in significant heat dissipation and impedes large-scale integration as the electronics are very close to the side of recording. In order to meet the both requirements of high acquisition and energy efficiency, compression techniques have to be utilized.

Recently, Compressed Sensing (CS) techniques have been proposed to address the challenge of dealing with large amount of data. For example, Mamaghanian et
al. incorporated CS in a real-time energy efficient framework for electrocardiogram (ECG) compression. This CS-based ECG compression outperformed the conventional digital wavelet transform (DWT)-based approach and was able to improve power efficiency. Another CS-based ECG compression system demonstrated that the signal could be compressed and reconstructed at a compression ratio of 4:1 to 16:1 with dynamic thresholding. Chen et al. also proposed a hardware-efficient CS architecture for data compression in wireless sensors, which had a power consumption of only 1.9 \(\mu\)W, thus significantly improving the power efficiency of such systems. Another CS-based system that exploits the rakeness approach to maximize the amount of information contained in the measurements demonstrated a superior performance with a compression ratio of 8:1 and 10:1. Furthermore, the CS-based system
has now been extended to multi-channel systems. Gangopadhyay et al.\textsuperscript{42} designed a 64-channel CS analog front-end for biosensor applications, which could recover and preserve most of features in the signal at a compression ratio of 2:1 to 6:1. Zhang et al.\textsuperscript{5} also proposed a 4-channel closed-loop CS neural recording system, which was able to achieve >10 times the compression ratio while consuming only 0.83 $\mu$W per channel. Li et al.\textsuperscript{43} presented a 256-channel digital signal processing system using CS technique, which has achieved a power consumption of 12.5 $\mu$W per channel at a data reduction of around 90%. Liu et al.\textsuperscript{22} designed a highly configurable 16-channel CS module for chronic recording and brain machine interface, featuring a compression ratio of 8:1.

Despite the successful application of CS technique, the CS-based systems developed previously for data compression suffer from several limitations. Most of the CS-based systems are non-adaptive, and use a signal-agnostic dictionary such as the identity matrix or the wavelet matrix to sparsify the signal. It has been shown that the use of a signal-dependent dictionary improves the reconstruction quality and compression ratio ($>10:1$) compared to the signal-agnostic dictionary.\textsuperscript{5,18,44,45} The signal-dependent dictionary helps increase the sparsity of the signal significantly in the signal-oriented basis. From the perspective of multi-channel neural recordings, the previous CS-based systems compresses the time varying neural signal on single electrode. They do not consider the signal characteristics and correlation at adjacent recording electrodes in the system design. As a result, the model does not take
these useful spatial information into account for signal compression and reconstruction. Additionally, the previous CS-based systems do not incorporate the feature of online analysis, such as spike sorting, in real-time experiments. In order to overcome these limitations in CS-based multi-channel neural recordings and to enable online analysis, we propose several CS-based frameworks that are more suitable for multi-channel recordings, and combine the post-processing such as spike sorting during the reconstruction process.

The rest of this chapter is organized as follows: In section 2.1, the CS theory and the background are recalled. In section 2.2, we introduce a multi-channel CS framework for neural recordings, an unsupervised dictionary learning algorithm for spike sorting and an unsupervised CS framework for multi-channel neural recording and spike sorting. In section 2.3, we end this chapter with a conclusion.

2.1 Background

2.1.1 Compressed Sensing and Sparse Representation

The CS theory\cite{10,11} demonstrates that an $S$-sparse signal $x$ of length $N$ is able to be compressed into a measurement vector $y$ of length $M$ by a matrix $S$ of dimension $M \times N$ satisfying the Restricted Isometry Property and $M \sim S \log (\frac{N}{S})$, where normally
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

$S << M < N$. Specifically, the $S$-sparse signal is defined as a signal of which only $S$ coefficients are non-zero elements in the entire length of $N$ or can be approximately represented by its largest $S$ coefficients.

By solving the $\ell_1$ norm optimization problem below, the sparse signal $x$ can be recovered with high probability.

$$\min_x ||x||_1 \text{ s.t. } y = Sx. \tag{2.1}$$

Normally, biomedical signals such as action potentials and electroencephalogram signals are not sparse in either time or frequency domains. Each neuron generates spikes with a characteristic shape and amplitude based on its morphology and proximity to electrodes. Spikes collected during neural recordings are generally stable over time. As a result, it is possible to construct a signal-dependent dictionary matrix $D$ of dimension $N$ by $L$ to represent spikes sparsely, which transforms the non-sparse signal $x$ of length $N$ into a $S$-sparse vector $a$ of length $L$ and normally $N << L$.

Therefore, the non-sparse signal $x$ can be represented as $x = Da$, which is defined as the linear combination of a few atoms from the dictionary. Now the original $\ell_1$ optimization problem becomes:
\[
\min_{a} ||a||_1 \quad s.t. \quad y = SDa. \tag{2.2}
\]

By solving the above optimization problem, we have the sparse vector \(a\) and the recovered non-sparse signal \(\hat{x} = Da\). Intuitively, the CS approach reconstructs the original spike \(x\) of length \(N\) from the measurement \(y\) of length \(M\), achieving a compression ratio of \(\frac{N}{M}\), which provides a promising way to compress the neural signal during data transmission. The sparsifying dictionary \(D\) is a transform domain where a signal \(x\) can be represented by only a few coefficients in \(a\). Therefore, the design of such a dictionary is the key to guarantee good performance of CS-based neural recording systems.

2.1.2 Dictionary Learning

In order to further reduce the number \(M\) of the measurement \(y\), a dictionary \(D\) should be designed to sparsely represent the signal \(x\) as much as possible, according to the CS theory. In common CS-based neural recording systems, there are two approaches of choosing a sparsifying dictionary. The first approach incorporates signal-agnostic dictionaries such as the identity or the wavelet dictionary, which can represent spikes in the time-frequency domain. The second approach trains a signal-dependent dictionary using prior information of spikes, since neural recording electrodes collect unique
and repetitive spikes from neurons. Previous works\cite{ref1, ref2, ref3} have demonstrated that the signal-dependent dictionary is superior to the signal-agnostic dictionary in terms of compression ratio, reconstruction quality, and spike sorting accuracy. Therefore, the ability to design a robust dictionary is key in determining the efficiency of a CS-based neural recording system.

The task of “dictionary learning” involves training a dictionary $D$ representing the training data samples $X$ as sparse compositions by optimizing the problem:

$$\min_{D, A} \|X - DA\|_2^2 \text{ s.t. } \forall i, \|a_i\|_0 \leq S.$$ (2.3)

$a_i$, which is a column item of $A$, indicates the $S$-sparse vector for $i$-th sample of training database and $A$ indicates the sparse coefficients matrix.

In order to train such a dictionary for sparse representation, K-SVD algorithm has been proposed and demonstrated its excellent performance on many tasks\cite{ref4}. Basically, K-SVD algorithm is composed of two stages: sparse coding stage and dictionary update stage. In the sparse coding stage, the dictionary $D$ is fixed and the optimization the problem:

$$\min_{a_i} \|x_i - Da_i\|_2^2 \text{ s.t. } \forall i, \|a_i\|_0 \leq S.$$ (2.4)
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

The second stage is the codebook update stage, where the sparse coefficients matrix $A$ is fixed and each atom of dictionary $D$ is updated by singular value decomposition method.

K-SVD algorithm provides us the inspiration as well as fundamental paradigm of dictionary learning for sparse representation. However, K-SVD algorithm only focuses on single mode and does not incorporate structures in dictionary learning, which means the correlation between measurements of multiple channels and distinct pattern between dictionary atoms for different neurons are not taken into account.

2.1.3 Multi-Channel Neural Recordings and Spike Sorting

The multi-channel neural recording systems collect spikes simultaneously from one neuron (using tetrodes), or from the entire neural region in the brain. By recording neural activities in close proximity, the multi-channel neural recording provides multiple perspectives for the neuroscientist for the post-processing and analysis of spikes of neurons. However, most of these systems simply focus on the recovery of individual neural signals and ignore the optimization of entire systems. For example, the correlation between channels has not been taken into account. Additionally, the user input required for spike sorting may introduce subjective bias, and has not been considered in the design of most neural recording systems.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Spike sorting, commonly used in neural recordings, groups spikes generated by neurons into different clusters based on their particular shapes. The clustering result helps neuroscientists study the activity of neurons and brains. However, the standard techniques for spike sorting are operated offline, after the neural recordings are complete. Given the raw sensor readings collected by neural recording device, neuroscientists often filter, detect and extract spikes first, and then cluster spikes using popular methods such as the Principal Component Analysis (PCA) or wavelets. This process might be trivial for a single channel neural recordings in the short term. But it is not time-efficient for long-term neural recordings, where large amounts of spikes are fired simultaneously by multiple neurons. Currently, none of the CS-based online methods allows neuroscientists do recovery and spike sorting at the same time. Neuroscientists are hard to benefit from the current CS based real-time neural recording microsystems since the offline spike sorting is independent of these recording systems. In order to take advantage of real-time neural recordings and monitor neuron activities online, a more efficient and hardware-friendly online spike sorting is required for the current high-density neural recording microsystems.
2.1.4 Compressed Sensing Neural Recording Systems

Many compression systems have incorporated the CS technique for processing biosignals. These CS-based neural recording systems are able to achieve a high power efficiency as well as a high density integration due to the implementation of a simple circuit. The sensing matrix $S$ can be implemented on chip to compress the signal in the front-end. Furthermore, the CS-based systems provide the flexibility of choosing a suitable dictionary $D$ as the on-chip random sensing mechanism independent from the sparse representation basis. Currently, there are two different methods in the design of the sparse representation basis. One method is to use a signal-agnostic dictionary, such as the identity and wavelet basis, which is independent from the signal itself. Another approach is to use a signal-dependent dictionary as the representation basis, which is adaptive and learns from the training samples. Previous works have demonstrated that the signal-dependent CS dictionaries have superior performance over other compression neural recording methods including spike detection, wavelet and other CS-based approaches in terms of compression ratio, reconstruction quality, spike sorting success rate, and chip power consumption.
2.2 Methods

In this section, we discuss the three frameworks that have been proposed to address the challenges in CS-based neural recording systems. Starting from the dictionary learning algorithm for multi-channel neural recordings, we describe a supervised CS-based framework using group structure and joint sparsity. Then, we introduce an unsupervised dictionary learning algorithm for neural recordings, which enables the online spike sorting without the prior information. Finally, we present an unsupervised CS algorithm for multi-channel neural recording and spike sorting, which is an extension of the first two frameworks. In details, each subsection is composed of signal model, algorithm, experiments and conclusion.

2.2.1 A Dictionary Learning Algorithm for Multi-Channel Neural Recordings

Multi-channel neural recording devices are widely used for \textit{in vivo} neuroscience experiments. Incurred by high signal frequency and large channel numbers, the acquisition rate could be on the order of hundred MB per second, which requires compression before wireless transmission to alleviate the pressure of power consumption. Therefore, we adopt the CS framework with a simple on-chip implementation. To improve the performance while reducing the number of measurements, we propose a multi-modal
structured dictionary learning algorithm that enforces both group sparsity and joint sparsity to learn sparsifying dictionaries for all channels simultaneously. When the data is compressed 50 times, this framework can achieve a gain of 4 dB and 10 percentage units over the state-of-art approaches in terms of the reconstruction quality and classification accuracy, respectively.

In this framework, we extend the signal dependent CS approach to multi-channel neural recordings such as tetrodes\textsuperscript{29} with following contributions, which can be utilized in other multi-electrode array due to the correlation among local electrodes.

a) Enable spike sorting using group sparsity: It is well known that spikes generated by different neurons have distinct shapes. By introducing group sparsity in the dictionary learning, we map each neuron to a specific group of dictionary atoms. Thus in reconstruction, we choose only atoms from the same group rather than different groups to recover each spike. This group sparsity constraint is shown to significantly improve the spike sorting performance.

b) Enhance reconstruction by joint sparsity: In multi-channel neural recordings, spikes from a single neuron is often picked up by all electrodes surrounding it. To take advantage of this correlation between measurements of different electrodes, we add a constraint of joint sparsity so that the recovered spikes for different channels strictly resemble the same neuron. This helps us achieve much better reconstruction quality with the same number of measurements.

Figure 2.2 presents the schematic of the multi-channel CS framework for a tetrodes
setup. The sensing matrix $S$ is implemented on-chip to compress the signal before transmission or readout. The sparsifying dictionary $D$ works as a transform domain where the signal can be represented with only a few atoms. The design of such a dictionary is the key to guarantee good reconstruction and spike sorting quality.

### 2.2.1.1 Signal Model

We assume there are $C$ channels and $G$ groups (classes) of neural signals in the database. We incorporate two key features in our signal model: group sparsity and joint sparsity. As shown in Figure 2.4, the neural signal $x_c$ for each channel $c$ (i.e., $c = 1, ..., 4$ in a tetrodes setup) is linearly represented using the corresponding dictionary
Figure 2.3: An illustration of the shape differences among spikes of three different neurons (color coded) using synthetic dataset from $^{11}$

$D_c$, which is a concatenation of sub-dictionaries $D_{c,g}$ for different groups of neural signals. For example, $D_{4,1}$ contains dictionary atoms for group 1 in channel 4. We let coefficient matrix $A = [a_1, a_2, \ldots, a_C]$, where $a_c$ denotes sparse coefficients vector of channel $c$ and $a_{c,g}$ is the sub-vector of $a_c$. For example, $a_{4,1}$ is the sub-vector in $a_4$ that contains coefficients for the sub-dictionary $D_{4,1}$.

As shown in Figure 2.3, different neurons fire spikes that are unique and repetitive in shapes. Intuitively, it is better to represent the signal using dictionary atoms belonging to a single group $g$ (same neuron) rather than atoms from different groups.
(different neurons). To achieve this goal, we enforce group sparsity by dividing our dictionary $D_t$ into different sub-dictionaries and choosing only the atoms from the same sub-dictionary $D_{t,g}$. As a result, the non-zero coefficients in $a_t$ will belong to same sub-vector $a_{t,g}$ as shown in Figure 2.4.

The aforementioned group sparsity works independently for each channel. However, neural spikes generated by the same neuron are recorded simultaneously by all nearby channels in similar patterns as in Figure 2.5. This indicates that a high correlation exists between measurements of different channels. To capture this correlation, we implement a joint sparsity using row-$\ell_0$ quasi-norm in the dictionary learning, which to choose the same group $g$ for all channels. As shown in Figure 2.4, sparse coefficients of all channels share the same support pattern in the same group as a result of the joint sparsity constraint.

2.2.1.2 Multi-Modal Structured Dictionary Learning

Based on the signal model, a multi-modal structured dictionary learning (MMSDL) algorithm is proposed. It incorporates both group sparsity and joint sparsity constraints. “Multi-modal” is defined here to emphasize the fact that the algorithm can handle the scenario that the dictionary is different for each individual channel as in a tetrodes setup. The algorithm iterates between two stages: sparse coding and code book update. In sparse coding stage, the objective function is defined as:
Note that the time stamps for training data $x_c$ from different channel $c$ need to be identical to guarantee our joint sparsity constraint. Here we enforce the data fidelity term $\|x_c - D_c a_c\|_2$ for all channels simultaneously. To embed the joint sparsity, we limit $\|A\|_{row,0}$, the number of nonzero rows, to be no greater than sparsity $S$. To enforce the group sparsity, we want the number of blocks with non-zero coefficients in $a_c$ to be equal to one. In our formulation, $I$ is the indicator function.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Then Algorithm 1 has been designed to train the dictionary. In sparse coding step, we decouple joint sparsity and group sparsity into sub-problems and solve them sequentially. We first represent the input signal \( x_c \) with the entire dictionary \( D_c \) using Simultaneous Orthogonal Matching Pursuit (SOMP)\(^{49}\). Then we determine the group \( g \) (type of neuron), which is chosen for all channels via sparse representation classifier (SRC)\(^{3}\). Finally, we decompose the signal via SOMP\(^{49}\) again but using only the chosen sub-dictionary \( D_{c,g} \). We denote this three-step procedure as Structured SOMP. For code book stage, we adopt the same approach as K-SVD\(^{48}\).

### 2.2.1.3 Reconstruction and Classification Approach

In this multi-channel CS framework, we use a Bernoulli sensing matrix \( S \) to compress the neural signals \( x_c \in \mathbb{R}^N \) into measurement vectors \( y_c \in \mathbb{R}^M \) because of its simple implementation using integrated circuits\(^{47}\). Given the compressed measurements \( y_c \) and the learned dictionary \( D_c \) from the MMSD, we adopt Structured SOMP (Step 4-6 in Algorithm\(^{1}\)) to solve the following problem,

\[
\hat{a}_c = \arg\min_{A} \|y_c - SD_c a_c\|_2
\]

s.t. \( \|A\|_{row,0} \leq S, \sum_{g=1}^{G} I(\|a_{c,g}\|_2 > 0) \leq 1, \forall c. \quad (2.6)\)

Then, the estimate of neural signal \( \hat{x}_c \) is defined as:

30
Algorithm 1 Multi-Modal Structured Dictionary Learning

Require: For each channel $c$, training data $X_c = [x_{c,1}, x_{c,2}, ..., x_{c,T}]$, where $x_{c,t}$ denotes the signal belonging to $c$-th channel at $t$-th time stamp. Number of groups $G$, and number of maximum iteration $maxIter$.

1: Initialization by randomly selection from training data.
2: while $iter \leq maxIter$ do
3:     for $i := 1$ to $T$ do
4:         Solve the representation problem via SOMP
        $\min_{A} ||x_c - D_c a_c||_2$
        s.t. $||A||_{row,0} \leq S, \quad \forall \ c.$
5:     Determine the class $g$ for the $t$-th signals of all channels using SRC.
6:     Solve the representation problem with the chosen sub-dictionary $D_{c,g}$ via SOMP
        $\min_{A} ||x_c - D_{c,g} a_{c,g}||_2$
        s.t. $||A||_{row,0} \leq S, \quad \forall \ c.$
7:     end for
8:     Code book update: we use the same method as in for updating each column.
9:     Set $iter = iter + 1.$
10: end while
11: Return $D_{c}, c = 1, 2, ..., C$

$\hat{x}_c = D_c \hat{a}_c. \quad (2.7)$

For our method and other methods used for comparison, we determine the group (class) $g$ via SRC using the entire dictionary $D_g$. 

31
2.2.1.4 Experiments

In this section, we first compare the single channel recovery performance of our MMSDL with the dictionary trained using K-SVD, the data dictionary and wavelet dictionary. The database used is the Leicester neural signal database. Then we compare the multi-channel recovery and classification performance of proposed approach with other approaches using the publicly available database hc-1 d14521 which collected by tetrodes from the neurons of rats. For all experiments, we randomly split the database into two halves with one part for training and the other for testing. We repeat our experiment ten different times and report the average results. The recovery performance is measured in terms of Signal-to-Noise and Distortion Ratio (SNDR), which is defined as:

\[
SNDR = 20 \log \frac{||x||_2}{||x - \hat{x}||_2}. \tag{2.8}
\]

and has the unit in dB. The classification performance is found in terms of classification accuracy (CA), which is the percentage of correctly classified spikes in evaluation.

We compare the MMSDL to the signal dependent dictionary, data dictionary and wavelet dictionary at different compression ratio (CR), which is defined as:
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Table 2.1: Comparison of single channel recovery performance (in SNDR) between different CS methods.

<table>
<thead>
<tr>
<th>Dictionary Learning &amp; Recovery Method</th>
<th>CR = 50</th>
<th>CR = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMSDL + Structured SOMP</td>
<td>8.84</td>
<td>13.61</td>
</tr>
<tr>
<td>K-SVD + OMP</td>
<td>4.87</td>
<td>12.44</td>
</tr>
<tr>
<td>Data Dictionary + OMP</td>
<td>4.89</td>
<td>10.48</td>
</tr>
<tr>
<td>Wavelet Dictionary + OMP</td>
<td>-0.85</td>
<td>-0.84</td>
</tr>
</tbody>
</table>

\[
CR = \frac{N}{M}. 
\]

The same random Bernoulli matrix \( S \) for all approaches to compress the neural signal. We use Structured SOMP in our framework and OMP for other approaches because this represents the fundamental difference in the signal models between our approach and other approaches. The reconstruction performance is shown in Table 2.1. Note that there is much less difference when the CR is small (i.e., 10 times). However, the proposed method works significantly better by about 4 dB compared to either signal dependent dictionary or data dictionary when the CR goes up to 50 times.

We also evaluate our method in a multi-channel setup. The SNDR used here is the average recovery performance of all four channels and the classification method is based on SRC. The reconstruction and classification performance is shown in
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Table 2.2: Comparison of multi-channel recovery performance (in SNDR) between different CS methods.

<table>
<thead>
<tr>
<th>Dictionary Learning &amp; Recovery Method</th>
<th>CR = 50</th>
<th>CR = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMSDL &amp; Structured SOMP</td>
<td>7.96</td>
<td>10.25</td>
</tr>
<tr>
<td>K-SVD + OMP</td>
<td>5.15</td>
<td>8.79</td>
</tr>
<tr>
<td>Data Dictionary + OMP</td>
<td>6.06</td>
<td>8.28</td>
</tr>
<tr>
<td>Wavelet Dictionary + OMP</td>
<td>-0.10</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Table 2.3: Comparison of multi-channel classification performance (in CA) between different CS methods.

<table>
<thead>
<tr>
<th>Dictionary Learning &amp; Recovery Method</th>
<th>CR = 50</th>
<th>CR = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMSDL &amp; Structured SOMP</td>
<td>92.55</td>
<td>98.50</td>
</tr>
<tr>
<td>K-SVD + OMP</td>
<td>82.50</td>
<td>88.10</td>
</tr>
<tr>
<td>Data Dictionary + OMP</td>
<td>83.10</td>
<td>84.25</td>
</tr>
<tr>
<td>Wavelet Dictionary + OMP</td>
<td>63.10</td>
<td>75.75</td>
</tr>
</tbody>
</table>

Table 2.2 and Table 2.3, respectively. The results demonstrate that the MMSDL improves both the reconstruction and classification performance. The reconstruction performance and classification performance are improved by more than 3 dB and 10 percentage units, respectively. To show the quality of recovered neural signals, we show an example of reconstruction at the CR of 50 times in Figure 2.5. The red curve indicates the ground truth and the blue curve indicates the recovered signal for each channel. It can be seen that the main structures of the neural signals are well preserved with only two measurements in this case.
Figure 2.5: An example of multi-channel signals (red) and CS recovered signals (blue) using our approach at CR of 50 times and SNDR = 9.17 dB.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

2.2.1.5 Conclusion

In this section, a multi-modal structured dictionary learning approach in a CS framework is presented. This framework combines group structure and joint sparsity to promote both reconstruction and classification performances of multi-channel neural recordings. This approach can be used in conjunction with a simple hardware implementation and applied to other multi-channel biological signal monitoring. Additionally, this framework can be extended to unsupervised dictionary learning for spike sorting.

2.2.2 An Unsupervised Dictionary Learning Algorithm for Neural Recordings

Neural action potentials (a.k.a. spikes), are short-lasting events generated by the neurons when the electrical membrane potential of a neuron rises and falls rapidly. To study the brain, neuroscientists have been using multi-electrode arrays (MEAs) or high density silicon probes consisting hundreds or even thousands of electrodes to collect spikes from neurons during experiments. The typical sampling rate by a single electrode is around 20 kHz at a resolution above 12 bits. Therefore, MEAs consisting of hundreds of channel typically generate data on the order of MB/s. Thus, to transmit such amount of data wirelessly, the system power consumption is on the
order of mW. In addition to an efficient data compression method, neuroscientists are also interested in an approach that allows an unsupervised classification of the neural spikes from the compressed measurement. Thus, they can identify different neurons recorded on single electrode in order to isolate multiple neuron activities. To address these needs, an efficient compression and an unsupervised sorting method are outlined in this session.

Recently, Compressed Sensing (CS) provides a promising approach for reconstruction and classification of neural recordings. Our previous work has demonstrated that supervised learning CS framework outperforms other traditional compression approaches such as spike detection, on-chip wavelet transformation and wavelet-based CS approach in reconstruction and classification performance. Figure 2.6 outlines the essential components in a signal dependent CS framework. The band-limited neural spikes are first compressed by multiplication with a sensing matrix $S$ implemented on-chip. The compressed measurements are transmitted off-chip through a wireless link. The off-chip recovery algorithm then uses a trained sparsifying dictionary, $D$, to recover the spike using these compressed measurements. In our previous works, we used a random Bernoulli matrix to implement $S$. The sparsify dictionary, $D$, is constructed by collecting small duration of the neural spikes generated by the same electrode.

Our previous works have shown that the signal dependent CS framework outperforms other compression approaches (spike detection, wavelet transform, and
Figure 2.6: A schematic of the proposed unsupervised CS approach for neural recordings. The unsupervised structured dictionary learning is our key contribution in this framework.

signal independent CS approach) in terms of hardware efficiency, compression rate and classification performance. In this session, we extend our supervised CS approach to unsupervised dictionary learning. The contributions of the paper are:

a) Unsupervised Discriminative Dictionary learned without label information: It is commonly the case when a single electrode may recorded different spikes with distinct shapes as shown in Figure 2.7. Without prior spike label information, our discriminative dictionary learning algorithm automatically constructs an discriminative dictionary that can be used for spike classification and spike recovery.

b) Improve classification quality jointly using centroid structure and sparse representation: Based on the learned dictionary, we combine sparse representation classifier
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Figure 2.7: A illustration of the different classes (color coded spikes) of “Easy” and “Difficult” neural spikes data base from\(^2\)

\(^{(SRC)}\) and Euclidean distance between spike and spikes centroids to jointly predict the class of neural spike. Introducing Euclidean distance penalty in SRC further improve the classification accuracy in CS framework under high compression ratio.

2.2.2.1 Signal Model

Currently, there are two approaches for sparsifying neural spikes in terms of dictionary learning. One approach is using signal agnostic dictionary such as Wavelet and Gabor\(^{39}\) dictionaries which transform the signal to time-frequency domain. Another approach adopts a signal dependent dictionary trained by prior neural spike information\(^{17,18,47,53}\). Previous works have demonstrated signal dependent dictionary performs signal agnostic dictionary in both reconstruction and classification performance given the same compression ratio. However, previous dictionaries are generated with
label information of spikes in a supervised fashion. However, in real experiments, the spike labels are not available as the training spikes are collected in real-time. Therefore, training a sparsifying dictionary in an unsupervised fashion is extremely valuable for practical purposes for neuroscientists.

It has been shown that signal-dependent dictionary with structures achieved better reconstruction quality and higher classification accuracy. As shown in Figure 2.8, we introduce the group structure in dictionary to discriminate different class of spikes, where each sub-dictionary contains the atoms of corresponding class of neural spikes. However, previous structured dictionary for neural recordings is learned from labelled spikes and concatenated by sub-dictionaries generated by corresponding neural spikes respectively. In this work, we developed an unsupervised structured dictionary learning algorithm without labelled spikes but the number of clusters $g$. The proposed dictionary learning algorithm is able to discriminate different clusters of neural spikes and automatically learn the structured dictionary for sparse recovery and spike sorting in CS framework.

2.2.2.2 Unsupervised Structured Dictionary Learning

Assume there are $G$ clusters of neural spikes $X$ without prior information of class of each spikes. The proposed algorithm falls into initialization stage and learning stage as shown in Algorithm 2. In initialization stage, we firstly implement k-means algorithm to determine the centroids $c_g$ of clusters by optimizing the residual sum.
Figure 2.8: An illustration of discriminative group structures (color coded blocks) for unsupervised structured dictionary learning.
of squares and construct the dictionary $D$ concatenated by sub-dictionaries $D_g$ ($g = 1, 2, ..., G$) in which the atoms are initialized by sum of corresponding centroid $c_g$ and Gaussian noise $\epsilon$. In learning stage, the algorithm iterates between sparse coding stage and dictionary update learning stage. For sparse coding stage, we first calculate sparse vector $a_g$ and the objective function is defined as:

$$\min_{a_g} ||x - D_g a_g||_2 \text{ s.t. } ||a_g||_0 \leq S. \quad (2.10)$$

As shown in Figure 2.9, we combine SRC in sparse coefficient domain and the Euclidean norm distance between spike and the $G$ centroids to jointly predict which cluster the spike most likely belongs to by solving the following problem,

$$\min_{g} \lambda ||x - D_g a_g||_2 + (1 - \lambda)||x - c_g||_2. \quad (2.11)$$

Finally, we recover each signal using Orthogonal Matching Pursuit (OMP) based on the corresponding sub-dictionary $D_g$ and add the index of spike into a trust region set $S$ if the reconstruction error is below $e$. For dictionary update stage, we adopt the same approach as approximate K-SVD based on spikes of which index belongs to the trust region set $S$.

As shown in Figure 2.10 (a), Algorithm 2 finds out the centroids of training data.
Figure 2.9: An intuitive schematic of joint prediction of class of neural spike using SRC and Euclidean norm distance.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Algorithm 2 Unsupervised Structured Dictionary Learning

Require: Training data $X = [x_1 \ x_2 \ ... \ x_T]$, where $x_t$ denotes the signal at $t$-th time stamp. Number of clusters $G$, sparsity $S$, reconstruction error $e$, linear combination coefficient $\lambda \in (0, 1)$ and number of maximum iteration $\text{maxIter}$.

1: Determine centroids $c_g$ of training data $X$ by K-means method, $g = 1, 2, ..., G$.

2: Initialize $D = [D_1 \ D_2 \ ... \ D_G]$, where the atom of $D_g$ is defined as $c_g + \varepsilon$.

3: while $\text{iter} \leq \text{maxIter}$ do

4:   for $t := 1 \ to \ T$ do

5:     for $g := 1 \ to \ G$ do

6:        Solve the representation problem via OMP,

7:         $\min_{a_g} ||x_t - D_g a_g||_2 \ s.t. \ ||a_g||_0 \leq S.$  \hspace{1cm} (2.12)

8:     end for

9:     Determine the cluster $g$ for the $t$-th signal by solving following problem,

10:    $\min_g \lambda ||x_t - D_g a_g||_2 + (1 - \lambda) ||x_t - c_g||_2.$  \hspace{1cm} (2.13)

11:   if $||x_t - D_g a_g||_2 \leq e$ then Add $i$ into $S$.

12:  end if

13: end for

14: Codebook update: we use the same method as in $\text{[33]}$ for updating each column based on the spikes belonging to $S$.

15: Set $\text{iter} = \text{iter} + 1$.

16: end while

17: Return $D$

and constructs the atoms of initialized dictionary by adding Gaussian noise to generate a small ball around each centroids. Then we only update the dictionary based on the signal of the trust region set $S$ in sparse coding stage. As shown in Figure 2.10(b)(c), the members in trust region set cover the entire training spikes while the average error of training data decreases and converges to stable value after several iterations. In Figure 2.10(d), the sparse coefficients matrix of training spikes intu-
2.2.2.3 Reconstruction and Classification Approach

In CS framework, we adopt on-chip random Bernoulli matrix\(^{47}\) to compress signal \(x \in \mathbb{R}^N\) into measurements vector \(y \in \mathbb{R}^M\) where \(y = Sx\) and \(M \ll N\). Given
our unsupervised structured dictionary $D$ and compressed measurements $y$, we reconstruct the signal and determine the class as shown in Algorithm 3.

**Algorithm 3 Reconstruction and Classification Approach**

**Require:** Unsupervised structured dictionary $D = [D_1 \ D_2 \ ... \ D_G]$, measurement vector $y$, random Bernoulli matrix $S$, sparsity $S$, linear combination coefficient $\lambda$ and centroids $c_g, g = 1, 2, ..., G$.

1: **for** $g := 1 \ to \ G$ **do**

2: Solve the representation problem via OMP

   \[
   \min_{a_g} \|y - SD_g a_g\|_2 \ s.t. \ |a_g| \leq S.
   \]  \hspace{1cm} (2.14)

3: **end for**

4: Determine the cluster $g$ for the signal by solving following problem,

   \[
   \min_g \lambda \|y - SD_g a_g\|_2 + (1 - \lambda) \|y - Sc_g\|_2.
   \]  \hspace{1cm} (2.15)

5: **Return** Recovered signal $\hat{x} = D_g a_g$ and class $g$

---

### 2.2.2.4 Experiments

In this section, we compared the reconstruction and classification performance of our proposed dictionary learning algorithm to two prior supervised signal dependent dictionary learning algorithms\textsuperscript{17,48} using the publicly available Leicester neural database\textsuperscript{1} One is the dictionary trained by K-SVD\textsuperscript{48} and the other is the data dictionary\textsuperscript{17} of which atoms are randomly selected from training data set. In Leicester neural database\textsuperscript{1} each data set contains different classes of spikes labelled by “Easy” or “Difficult” which means the difficulty of discrimination of different spikes as shown in Figure 2.7. In each experiment, the data set is randomly divided into two halves where one is for training and the other is for testing. We use the same Bernoulli
matrix for all approaches to compress the neural signals in CS framework and repeat each experiment for 30 times to report the average reconstruction and classification performance.

We compared the reconstruction performance of “Easy” and “Difficult” neural spikes based on our proposed unsupervised structured dictionary, dictionary trained by K-SVD and data dictionary. We use proposed recovery method for our dictionary and OMP for other dictionary in CS framework because this recovery method is implemented in our proposed dictionary learning in order to automatically discriminate unlabeled neural spikes while OMP\textsuperscript{51} has been applied in prior work for supervised dictionary learning. Signal-to-Noise Distortion Ratio (SNDR) in decibel is employed to measure the reconstruction quality, which is defined as Equation 2.8 in\textsuperscript{39} The re-

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.11.png}
\caption{A reconstruction and classification example of “Difficult” spikes using proposed approach at CR of 50 times.}
\end{figure}
Table 2.4: Comparison of reconstruction performance (in SNDR) of “Easy” and “Difficult” spikes between different CS methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Compressed Sensing Approach</th>
<th>CR = 50</th>
<th>CR = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>EASY</td>
<td>Proposed Approach</td>
<td>7.48</td>
<td>10.92</td>
</tr>
<tr>
<td></td>
<td>K-SVD48 + OMP51</td>
<td>4.46</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary17 + OMP51</td>
<td>5.84</td>
<td>7.21</td>
</tr>
<tr>
<td>DIFFICULT</td>
<td>Proposed Approach</td>
<td>7.40</td>
<td>10.25</td>
</tr>
<tr>
<td></td>
<td>K-SVD48 + OMP51</td>
<td>3.10</td>
<td>8.03</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary17 + OMP51</td>
<td>5.73</td>
<td>6.64</td>
</tr>
</tbody>
</table>

Results are demonstrated in Table 2.4. The results show that our approach outperforms other supervised CS approaches using dictionaries generated with prior label information. The improved SNDR is on average around 2 to 3 dB better than K-SVD48 and data dictionary17 based method even at compression ratio (CR) of 50 times.

We also evaluated the classification accuracy of our CS framework against two prior approaches using different three dictionaries. The classification method is based on SRC3 which has been implemented in recovery method. The classification quality is defined in terms of classification accuracy (CA), which is the percentage of correctly classified test spikes. Table 2.5 provides the classification accuracy of “Easy” and “Difficult” spikes, which demonstrate the significant improvement of proposed CS approach. There is no much significant difference of classification accuracy among different CS frameworks given the CR of 10 times. However, our CS framework still achieves more than 90% classification accuracy when the CR goes up to 50 times with
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Table 2.5: Comparison of classification performance (in CA) of “Easy” and “Difficult” spikes between different CS methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Compressed Sensing Approach</th>
<th>CR = 50</th>
<th>CR = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>EASY</td>
<td>Proposed Approach</td>
<td>93.60</td>
<td>98.80</td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>88.67</td>
<td>98.05</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary + OMP</td>
<td>81.10</td>
<td>95.77</td>
</tr>
<tr>
<td>DIFFICULT</td>
<td>Proposed Approach</td>
<td>92.22</td>
<td>94.88</td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>57.90</td>
<td>86.75</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary + OMP</td>
<td>59.87</td>
<td>78.22</td>
</tr>
</tbody>
</table>

only three measurements.

2.2.2.5 Conclusion

To meet the growing demand of wireless and power efficient neural recordings systems, we demonstrate an unsupervised dictionary learning algorithm in Compressed Sensing (CS) framework which can be implemented in VLSI systems. Without prior label information of neural spikes, we extend our previous work to unsupervised learning and construct a dictionary with discriminative structures for spike sorting. To further improve the reconstruction and classification performance, we proposed a joint prediction to determine the class of neural spikes in dictionary learning. When the neural spikes is compressed 50 times, our approach can achieve an average gain of 2 dB and 15 percentage units over state-of-the-art of CS approaches in terms of the reconstruction quality and classification accuracy respectively. This hardware-friendly
framework can also be applied in VLSI systems for in-vivo neural recordings experiment. In the future, we would like to extend our work to real-time unsupervised dictionary learning for multi-channel neural recordings.

2.2.3 An Unsupervised Compressed Sensing Algorithm for Multi-Channel Neural Recording and Spike Sorting

High-density multi-electrode neural recording microsystems have evolved over the years to become essential tools in neural electrophysiology experiments. These microsystems monitor brain activity by collecting extracellular neural action potentials (or spikes) from different areas of the brain. Using high-density multi-electrodes array (MEA) or tetrode drives, the action potential of each neuron can be recorded by multiple electrodes in its proximity. This redundancy of features can greatly improve the spike clustering accuracy. However, the drawback is that the large number of electrodes generate large amount of data. This presents itself as a challenge for the design of the implantable system in terms of chip size and power consumption. Typically, spikes are sampled at around 30 kHz at a resolution of more than 10 bits. A multi-channel neural recording system containing up to thousands of channels generates data at the rate of 300 Mbps. This would cost around 50 mW to transmit wirelessly, which results in significant heat dissipation and impedes large-scale in-
Figure 2.12: Basic block diagram of the proposed CS neural recording system. In the CS approach, multi-channel signals are randomly sampled by an on-chip sensing matrix $S$ and then wirelessly transmitted to an off-chip terminal for reconstruction and sorting. The design of the multi-modal dictionary learning $D$ for sparsifying signals is the major contribution of our proposed CS approach.

Despite their advantages mentioned before, the CS-based systems developed previously for signal compression suffer from several limitations. Most of the CS-based systems are non-adaptive, and use a signal-agnostic dictionary such as the identity matrix or the wavelet matrix to sparsify the signal. It has been shown that the use of a signal-dependent dictionary improves the reconstruction quality and compression ratio ($>10:1$) compared to the signal-agnostic dictionary. The signal-dependent dictionary helps increase the sparsity of the signal significantly in the signal-oriented basis. From the perspective of multi-channel neural recordings, the previous CS-
based systems compress the time varying neural signal on single electrode. They do not consider the signal characteristics and correlation at adjacent recording electrodes in the system design. As a result, the model does not take these useful spatial information into account for signal compression and reconstruction. Additionally, the previous CS-based systems do not incorporate the feature of online analysis, such as spike sorting, in real-time experiments. In order to overcome these limitations in CS-based multi-channel neural recordings and to enable online analysis, we propose a CS-based approach that is more suitable for multi-channel recordings, and combine the post-processing such as spike sorting during the reconstruction process. As shown in Figure 1, the off-chip design of the multi-channel dictionary learning serves as the most important component of our CS framework with the following contributions:

a) Multi-channel dictionary learning using joint-group sparsity

In multi-channel neural recording systems, such as the tetrodes, several close-by electrodes around neurons collect spikes simultaneously. Spikes recorded by these electrodes share similar patterns and features. To take advantage of the correlation among these electrodes in sensing and recovering, we introduce a joint-group sparsity constraint in dictionary learning to enforce spikes recorded at electrodes in close proximity to be recovered using similar items from the dictionary. This method will further increase the compression ratio in multi-channel neural recordings while guaranteeing good reconstruction quality. In other words, higher compression ratio will further reduce the transmission bandwidth and promote the design of power
efficiency of the large-scale integration of neural recording systems.

b) Online spike sorting using spectral clustering and group sparsity

An electrode can detect spikes from a group of neurons in its proximity. As shown in Figure 2.13, these spikes have particular shapes and can be clustered, corresponding to different neurons. Conventional spike sorting, which was supervised and for offline post-processing, used prior information to train a classifier. However, the large amounts of spikes generated are not labeled in real-time experiments. Neuroscientists have to manually sort and cluster these spikes using offline sorting software (e.g., Plexon). This process is neither time-efficient, nor integrated with the neural recording systems to realize online spike sorting. In our work, we combine spectral clustering and group sparsity in dictionary learning to maximize the discriminative property of the dictionary and enable online sorting. Based on these distinct group structures, spikes can be represented sparsely and sorted based on only a few items associated with the corresponding sub-dictionary.

2.2.3.1 Signal Model

We assume there are $C$ multi-channels and $G$ groups (clusters) of neural spikes in the data samples $X \in \mathbb{R}^{N \times T}$. We incorporate a key feature in the proposed signal model: joint-group sparsity. As shown in Figure 2.14, dictionary $D_c$ is a concatenation of sub dictionary $D_{c,g}$, where $c$ and $g$ separately indicate the indices of channels and groups. Intuitively, if neural signal $x_c$ associated with channel $c$ belongs to group $g$, it
Figure 2.13: An illustration of the different clusters (color-coded spikes) of neural signals from the Leicester and MGH databases. In spike sorting, spikes are grouped into different clusters based on their distinct shapes. Normally, one cluster corresponds to a single neuron.
should be ideally represented by the corresponding dictionary $D_{c,g}$, in which only the elements of its sparse vector $a_{c,g}$ are possibly non-zero. For example, $D_{2,3}$ contains atoms that can sparsely represent the neural signals of group 3 collected from channel 2. $x_{1,t}, x_{2,t}, ..., x_{C,t}$, the windowed segments from raw data, indicate the neural signals collected by multi-channels at timestamp $t$. Because they are recorded by electrodes such as tetrodes simultaneously in the close proximity, $x_{1,t}, x_{2,t}, ..., x_{C,t}$ share a similar pattern in terms of particular shapes. Taking the correlation among channels into account, we implement a joint-group sparsity in our signal model to further improve the recovery quality, spike sorting accuracy and compression ratio. $a_c$ indicates the sparse coefficient vector of channel $c$, while $a_{c,g}$ indicates the sub-vector of $a_c$. Among $C$ channels, $A$ is defined as $[a_1, a_2, ..., a_C] = [A_1, A_2, ..., A_G]^T$. $A_g$ ($g = 1, 2, ..., G$) is the sub-matrix associated with group $G$. Based on these definitions, joint-group sparsity is defined as:

\[
||A||_{\text{group,0}} = \sum_{l=1}^{G} I(||A_g||_F > 0) = 1, \tag{2.16}
\]

\[
||a_c||_0 \leq S, \ \forall c. \tag{2.17}
\]

In our formulation, $I$ is the indicator function and $S$ denotes the sparsity. $||A||_{\text{group,0}}$ is constrained as 1 to determine the corresponding group $g$ of the spike. $||A_g||$ denotes the Frobenius norm. Therefore, the mathematical definition of the proposed signal
Figure 2.14: Intuitive illustration of the proposed signal model with discriminative group structures (color-coded blocks) and joint-group sparsity (red filled) for multi-mode structured dictionary learning.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

model is:

\[ x_c = [D_{c,1} \, D_{c,2} \ldots D_{c,G}] [a_{c,1}^\top \, a_{c,2}^\top \ldots a_{c,G}^\top]^\top, \]  \hspace{1cm} (2.18)

\[ ||A||_{\text{group,0}} = 1, ||a_c||_0 \leq S, \forall c. \]  \hspace{1cm} (2.19)

Intuitively, a spike should be represented by atoms from the corresponding group, and also be constrained by the information given by neighboring electrodes. Taking neighboring spikes into account, the compression ratio \( \frac{M}{N} \) can be further improved, which also promotes the performance of the neural recording systems in terms of power efficiency.

2.2.3.2 Unsupervised Dictionary Learning for Multi-channel Neural Recordings and Spike Sorting

2.2.3.2.1 Dictionary Initialization

The task of finding clusters of spikes fired by different neurons has been a focus of considerable research in the field of neuroscience. Given prior label information, previous supervised work initialized the dictionary with discriminate group structure (sub-dictionaries), consisting of atoms from the training samples. However, it is still a challenge to reconstruct and sort spikes in real-time CS-based neural recording systems. Normally, neural spikes are manually sorted after \textit{in vivo} experiments using
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

an offline sorter such as Plexon. In order to extend our work to online, unsupervised spike sorting, we implement the clustering approach. Previously, the K-Means method has been shown to help initialize the dictionary and successfully improve the spike sorting accuracy in unsupervised CS-based neural recording systems.

In this framework, we adopt spectral clustering to initialize the dictionary with a discriminate group structure. The initialization is divided into two stages: (i) initialization of the similarity matrix, and (ii) spectral clustering. As shown in Algorithm 4, the similarity matrix $E$ represents the quantitative assessment of similarity between spikes. The similarity matrix is generated based on the nearest-neighbour method and the similarity of neural signals in the multi-channel is defined as:

$$e(t, t') = \sum_{c=1}^{C} ||x_{c,t} - x_{c,t'}||_2,$$

(2.20)

$$t, t' \in \{1, 2, ..., T\}, \ t \neq t'.$$  

(2.21)

$x_{c,t}$ denotes the spike from $c$th channel and timestamp $t$ and $e(t, t')$ denotes the summation of squared Euclidean distance between $t$th and $t'$th spikes from channels 1 to $C$ (i.e., $C$=1 indicates single channel). The smaller the $e(t, t')$ is, the closer correlation the neural signals share with each other. As shown in Algorithm 4, $K$ is defined as the nearest neighbour search number. The elements of $t$-th row of similarity matrix $E$ are set to 1 if the corresponding indexes belong to the $K$ smallest set while
the others are set to 0. Then, we update the similarity matrix by $E = E + E^T$.

**Algorithm 4 Similarity Matrix Initialization**

**Require:** Training data $X_c = [x_{c,1} \ x_{c,2} \ ... \ x_{c,T}]$, where $c = 1,2,...,C$ (C = 1 indicates the single channel). $x_{c,n}$ denotes the signal at timestamp $t$ of $c$-th channel. The number of nearest neighbour search $K$.

1: Determine the $K$ nearest neighbour vector $v_t \in \mathbb{Z}^K$ for $t$-th spike $x_{c,t}$ based on the definition of Euclidean distance $e(t,t')$. Thereby, $v_{t,k}$ denotes the index of $K$ nearest neighbour of $t$-th neural signal.

2: Initialize similarity matrix $E \in \mathbb{R}^{T \times T}$, where $E(t,t') = 0$, $\forall t,t'$.

3: Set $E(t,v_{t,k}) = 1$, $\forall t,k$.

4: Symmetrize the similarity matrix $E = E + E^T$.

5: Set $E(t,t) = 1$, $\forall t$.

6: **Return** similarity matrix $E$.

After the similarity matrix $E$ is generated, as shown in Algorithm 4, we pre-define the group number $E$ and then adopt spectral clustering to group neural signals into $E$ different clusters, providing prior information to help initialize the dictionary with group structures. The details of spectral clustering can be found in.

Given the clustering information $g$, the dictionary $D_c$ of $c$-th channel is defined as:

$$D_c = [D_{c,1} \ D_{c,2} \ ... \ D_{c,G}]. \quad (2.22)$$

$D_{c,g}$ indicates the sub-dictionary of $D_c$, in which its atoms are randomly picked up from the group of cluster $g$. We also obtain the mean shape, defined as centroids $c_{c,g}$ associated with a distinct cluster, which is used for template matching in the sparse
Figure 2.15: Illustration of different groups of spikes with distinct shapes. The red color-coded spikes indicate the centroids (mean shape) associated with the corresponding groups. The mean shape matching provides another perspective of similarity in the sparse coding stage.

coding stage. Centroids $c_{c,g}$, representing the template and a particular pattern of groups, $g$ are found by:

$$c_{c,g} = \frac{1}{|S_g|} \sum_{t \in S_g} x_{c,g}, \quad S_g = \{t|g_t = g\}. \quad (2.23)$$

$$\quad (2.24)$$

2.2.3.2.2 Dictionary Learning

After initializing the dictionary $D_c$ ($c = 1$ indicates single channel, $c > 1$ indicates the multi-channel), as shown in Algorithm 6, the unsupervised multi-mode structured dictionary learning is basically divided into two stages in each iteration: the sparse
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Algorithm 5 Spectral Clustering based on the Similarity Matrix

Require: The similarity matrix \( E \in \mathbb{R}^{T \times T} \) and the number of clusters \( G \).

1: Construct diagonal matrix \( W \), where \( W(t, t) \) is defined as the sum of \( t \)-th row of similarity matrix \( E \).
2: Construct the matrix \( H = W^{-\frac{1}{2}} E W^{-\frac{1}{2}} \).
3: Calculate \( v_1, v_2, ..., v_G \), the \( G \) largest eigenvectors of \( H \).
4: Construct the matrix \( V = [v_1 \ v_2 \ ... \ v_G] \in \mathbb{R}^{T \times G} \) and normalize each row of the matrix \( V \).
5: Apply \textit{k}-means algorithm to the rows of the matrix \( V \) and assign the cluster \( g_t \) to the original signal \( x_{c,t} \).
6: Return Clusters vector \( g \)

coding stage and the dictionary update stage.

In the sparse coding stage, we introduce joint-group sparsity and then solve the sparse representation problem below, using Orthogonal Matching Pursuit (OMP)\(^{51}\)

\[
\min_{a_{c,g}} \sum_{c=1}^{C} ||x_{c,t} - D_{c,g} a_{c,g}||_2 \quad \text{s.t.} \quad \text{\( \|A\|_{\text{group,0}} = 1, \|a_{c,g}\|_0 \leq S. \)}
\]  (2.25)

Here, we find out the best sparse representation \( a_{c,g} \) of each spike \( x_{c,t} \) in the training samples based on each sub-dictionary \( D_{c,g} \). Then, we use the linear combination coefficient \( \lambda \in (0, 1) \) to balance the residual of the sparse representation and the squared Euclidean distance between the spike and its centroids. Thereby, the cluster \( g \) of the spike is determined by solving the problem below:
\[
\min_g \sum_{c=1}^C \{ \lambda \|x_{c,t} - D_{c,g}a_{c,g}\|_2^2 + (1 - \lambda)\|D_{c,g}a_{c,g} - c_{c,g}\|_2^2 \}. \tag{2.27}
\]

As shown in Figure 2.15, the squared Euclidean distance for mean shape matching provides another evaluation of the similarity of spikes in the sparse representation stage. Previous work has demonstrated that using the centroid significantly improves the accuracy of spike sorting in the CS-based neural recordings. Given group \( g \) of each spike, we define a trust region set \( S_g \) associated with group \( g \). To construct the trust region set \( S_g \), we add the index of spike \( t \) into it if the spike is represented perfectly in the sparse coding stage, which indicates the reconstruction error is smaller than the pre-defined error. Intuitively, the trust region set \( S_g \) only contains spikes with high reconstruction quality in each learning iteration.

In the dictionary update stage, we simply fix the sparse coefficients matrix \( A_c \) and update each atom of the dictionary using the same approach as in the K-SVD. While the K-SVD updates the dictionary based on the whole training samples, our approach only updates it based on the current trust region set \( S \), which is the union of set \( S_g \). Iteratively, the trust region covers the entire training samples. Figure 2.16 and 2.17 illustrate that the trust region set \( S \) approaches the entire training samples after several learning iterations. Furthermore, we dynamically update the centroid \( c_{c,g} \) depending on the clustering result obtained from the sparse representation stage.
Figure 2.16: An illustration of how the trust region $S$ performs in principal component analysis (PCA). As Algorithm 6 iterates from 1 to 10, the percentage of spikes in the trust region increases from 29.33 % to 92.00 %, indicating that most spikes in the training samples satisfy the pre-defined reconstruction quality after 10 iterations.
As shown in Figure 2.17, the average recovery error converges as the trust region $S$ covers the entire training samples.

**Algorithm 6** Unsupervised Multi-Mode Structured Dictionary Learning

**Require:** Initialized dictionary $D_c$, training data $X_c = [x_{c,1}, x_{c,2}, ..., x_{c,T}]$, where $c = 1, 2, ..., C$ ($C = 1$ indicates single channel). Clusters vector $g$, number of clusters $G$, sparsity $S$, reconstruction error, linear combination coefficient $\lambda \in (0, 1)$ and number of maximum iteration $maxIter$.

1: while $iter \leq maxIter$ do
2: Set $S_l = \emptyset$, $\forall l$.
3: Solve the representation problem via Orthogonal Matching Pursuit,
\[
\min_{a_{c,g}} \sum_{c=1}^{C} ||x_{c,t} - D_{c,g}a_{c,g}||_2 \text{ s.t.} \]
\[
||A||_{group,0} = 1, ||a_{c,g}||_0 \leq S, \forall g, n. \tag{2.28}
\]
4: Determine the cluster $g$ for the $n$-th signal by solving following problem,
\[
\min_{g} \sum_{c=1}^{C} \{\lambda||x_{c,t} - D_{c,g}a_{c,g}||_2 \}
+(1 - \lambda)||x_{c,g} - c_{c,g}||_2 \}.
\tag{2.29}
\]
5: If $\sum_{c=1}^{C} ||x_{c,n} - D_{c,g}a_{c,g}||_2 \leq error$, then add $n$ into $S_g$.
6: Codebook update: we use the same method of approximation K-SVD for updating each atom based on spikes belonging to $S = \bigcup_{1}^{G} S_g$.
7: Centroids update:
\[
\begin{align*}
    c_{c,g} &= \frac{1}{|S_g|} \sum_{t \in S_g} x_{c,g} \\
    &\forall g = 1, 2, ..., G, c = 1, 2, ..., C, t = 1, 2, ..., T.
\end{align*}
\]
8: Set $iter = iter + 1$.
9: end while
10: Return $D_c = [D_{c,1}D_{c,2}...D_{c,G}]$ and updated the centroids $c_{c,1}c_{c,2}...c_{c,G}$

Taking advantage of iterative refinement in the dictionary learning, Algorithm 6 is able to correct the spike sorting error generated by the dictionary initialization,
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Figure 2.17: Illustration of the trust region and training error during the iterative training. (a) indicates the percentage change in trust region $S$ and (b) indicates average recovery error as dictionary learning iterates from 1 to 20.

as shown in Figure 2.18 Figure 2.18(a) shows that Algorithm 4 mistakenly clusters some spikes, which are denoted as blue dots and distributed in the cluster of green dots. But as shown in Figure 2.18(b), after the dictionary learning, the PCA result illustrates that the spike sorting performance is refined and intuitively demonstrates the robustness of the proposed approach.

2.2.3.2.3 Reconstruction and Spike Sorting Approach

In our CS-based neural recording systems, we adopt the on-chip random Bernoulli matrix $S \in \mathbb{R}^{M \times N}$ to compress the signal $x \in \mathbb{R}^N$ into the measurement $y \in \mathbb{R}^M$. Mathematically, $y = Sx$ and $M << N$. The Bernoulli matrix, of which the element
For each channel $c$, we adopt the same Bernoulli matrix $S$ to sense the neural signal $x_c$ into the measurements $y_c$. Given the trained dictionary $D_c$, the sensing matrix $S$, the centroids $c_{c,g}$ and the measurements $y_c$, we reconstruct the signal $\hat{x}_c$ and determine the cluster $g$ as shown in Algorithm 7.

### 2.2.3.2.4 Dictionary Update

In real-time neural recording experiments, it is impractical to observe the original signal $x$ because the CS-based system only transmits the compressed information $y$. Therefore, the reconstruction quality cannot be quantitatively evaluated by $x$ and the recovered signal $\hat{x}$. Normally, the trained dictionary is fixed during the recording.

**Figure 2.18:** An example of robustness of spike sorting from the perspective of sparse coding, visualized in the PCA domain. The iterative refinement helps correct the mistakenly sorted spikes generated from the initialization.
Algorithm 7 Reconstruction and Spike Sorting Approach

Require: The initialized dictionaries $D_c$, the centroids $c_{c,g}$, measurements $y_c$, where $c = 1, 2, ..., C$ ($C = 1$ indicates single channel) and random Bernoulli matrix $S$. Number of clusters $G$, sparsity $S$ and linear combination coefficient $\lambda \in (0, 1)$.

1: Solve the representation problem via Orthogonal Matching Pursuit,

$$
\min_{a_{c,g}} \sum_{c=1}^{C} ||y_c - SD_{c,g}a_{c,g}||_2 \text{ s.t. } ||a_{c,g}||_0 \leq S, \forall g.
$$

(2.33)

2: Determine the cluster $g$ of spikes by solving following problem,

$$
\min_g \sum_{c=1}^{C} \{\lambda||y_c - SD_{c,g}a_{c,g}||_2^2 + (1 - \lambda)||y_{c,g} - Sc_{c,g}||_2^2\}.
$$

(2.34)

3: Return The recovered signal $\hat{x}_c = D_{c,g}a_{c,g}$ and cluster $g$.

If the CS-based neural recording system encounters a new spike that is dramatically different from spikes from the training samples, it might be difficult to reconstruct the spike sparsely using the dictionary, which significantly degrades the quality of systems. To address this challenge, the dictionary update has to be adaptive to change in spikes. Fortunately, the strong correlation between the reconstruction quality of the original signal $x$ and the reconstruction quality of the measurement $y$ has been found to quantify the recording performance as shown in Figure 2.19. It helps in adapting the trade-off between the reconstruction quality and compression ratio. Instead of evaluating the reconstruction quality of $x$, which cannot be observed in a CS-based neural recording system, the signal-to-noise distortion ratio ($SNDR$) of $y$ is adopted to efficiently quantify the online performance of the CS-based neural recording systems. When $SNDR_y$ drops below the pre-defined threshold, the CS-based system
Figure 2.19: An example of $SNDR_x$ and $SNDR_y$ values for all the recording electrode over 15 weeks of recordings.

will automatically switch to the non-CS mode for collecting more samples at the full bandwidth for the dictionary update.

2.2.3.3 Experiments

In this section, we compare the reconstruction and spike sorting performance of our proposed approach with the other CS-based approaches on both single-channel and multi-channel databases. In each training, the database was randomly divided into
two halves: one for training, and the other for testing. The quality of reconstruction is measured in terms of the $SNDR$ which is defined as:

$$SNDR = 20 \times \log \frac{||x||_2}{||x - \hat{x}||_2},$$  \hspace{1cm} (2.35) \nonumber

where $x$ and $\hat{x}$ indicate the original and recovered signals, respectively. The spike sorting performance is measured in terms of classification accuracy (CA), which is defined as:

$$CA = \frac{\text{# of Correctly Sorted Spikes}}{\text{Total Number of Spikes}} \times 100\%. \hspace{1cm} (2.36) \nonumber$$

All neural spikes are extracted from the raw data using a window of pre-defined length, and aligned properly before training and testing. In each experiment, the same Bernoulli matrix is adopted to compress the neural signal. We construct the K-SVD dictionary and the data dictionary with group structures based on the training samples. Then, we adopt OMP and sparse representation classifier (SRC) for recovery and spike sorting. For the proposed approach, we assume that the number of clusters $G$ is pre-defined and the dictionary is learned by Algorithm in an unsupervised manner. Additionally, we assign the same number of spikes to each group in the training and testing samples to eliminate the clustering bias.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Our experiments were implemented in MATLAB on a PC with Intel Core i7 and 16 GB RAM. The average computational time for the dictionary training was 20 seconds on average. The number of iterations was 10 and each training database consisted of 1000 samples of pre-defined length. It took only 6 ms to reconstruct and sort a spike.

2.2.3.3.1 Single Channel

We first compared the reconstruction performance between the proposed CS-based approach and the other dictionary learning approaches using the K-SVD, data dictionary and Wavelet dictionary on the synthetic Leicester database and the Massachusetts General Hospital (MGH) database. The Leicester database consists of neural signals of length 128, and the MGH database consists of neural signals of length 32, recorded from primates (monkeys), “Pogo” and “Romeo”. The MGH database was collected at the MGH at a sampling rate of 40 kHz.

Furthermore, we compared the spike sorting accuracy of our CS-based approach to the other CS-based approaches using the signal-dependent dictionaries. The Leicester database consists of three classes of neural spikes grouped into two categories: “Easy” and “Difficult”, which indicates the difficulty level of discriminating spikes. Generally, “Difficult” indicates a lot of noise in spikes. The MGH database contains two or three classes of spikes that have been manually sorted at the MGH.

Tables I and II, and Figure 2.20 demonstrate the reconstruction and spike sorting...
Figure 2.20: Examples of reconstruction performance of single-channel neural recordings. For (a)-(d), recovered signals (red) still preserve the major features of original signals (blue) at CR of 20:1 and 10:1, respectively. (a) and (b) demonstrate synthetic spikes from the Leicester database\(^\text{1}\) while (c) and (d) demonstrate real spikes from the MGH database\(^\text{4}\).
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

performance on the Leicester database at compression ratios of 20:1 and 10:1. The proposed approach outperforms the other CS-based approaches, and achieves an average gain of 2 dB and 4% in terms of SNDR and classification accuracy on the “Easy” database at the CR of 20:1. For the “Difficult” database, the approach attains more than 90% spike sorting success rate, while achieving a CR of 10:1 to 20:1. Tables III and IV show the reconstruction and spike sorting performance of the MGH “Pogo” and “Romeo” databases, respectively. Here too, the proposed approach outperforms the other CS-based approaches. Especially, the proposed approach shows more than 90% spike sorting success rate at the CR of 10:1, and achieves an average gain of 30% over other methods. Figure 2.21 intuitively illustrates the spike sorting result at the CR of 20:1 and 10:1 in the PCA domain. The pink, green and blue dots indicate distinct groups of spikes in the testing samples, while red dots indicate spikes that are incorrectly sorted. As shown in Figure 2.21, most of the spikes are correctly sorted and the spike sorting success rate can still achieve more than 90% accuracy, even after the CR increases to 20:1, which means we only use 5% information to reconstruct and sort the spike. The performance on the MGH database achieves more gains in terms of the recovery quality and spike sorting success rate compared to the performance on the Leicester database, which indicates the proposed approach is more robust to highly noisy signals.
Table 2.6: Comparison of reconstruction performance (in SNDR) of different CS methods on “Leicester”.

<table>
<thead>
<tr>
<th>Database</th>
<th>CS Approach</th>
<th>CR = 20:1</th>
<th>CR = 10:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EASY</td>
<td>Proposed Approach</td>
<td>10.44</td>
<td>11.60</td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>9.23</td>
<td>10.40</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary + OMP</td>
<td>7.11</td>
<td>7.76</td>
</tr>
<tr>
<td></td>
<td>Wavelet + OMP</td>
<td>-1.81</td>
<td>-0.82</td>
</tr>
<tr>
<td>DIFFICULT</td>
<td>Proposed Approach</td>
<td>8.64</td>
<td>10.21</td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>6.40</td>
<td>8.03</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary + OMP</td>
<td>6.38</td>
<td>6.64</td>
</tr>
<tr>
<td></td>
<td>Wavelet + OMP</td>
<td>-2.71</td>
<td>-1.78</td>
</tr>
</tbody>
</table>

Table 2.7: Comparison of classification performance (in CA) of different CS methods on “Leicester”.

<table>
<thead>
<tr>
<th>Database</th>
<th>CS Approach</th>
<th>CR = 20:1</th>
<th>CR = 10:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EASY</td>
<td>Proposed Approach</td>
<td>97.62</td>
<td>98.08</td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>94.04</td>
<td>98.05</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary + OMP</td>
<td>93.88</td>
<td>95.77</td>
</tr>
<tr>
<td>DIFFICULT</td>
<td>Proposed Approach</td>
<td>90.24</td>
<td>95.87</td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>74.03</td>
<td>86.84</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary + OMP</td>
<td>73.24</td>
<td>78.22</td>
</tr>
</tbody>
</table>
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Table 2.8: Comparison of reconstruction and classification performance of different CS methods on “Pogo”.

<table>
<thead>
<tr>
<th>Database</th>
<th>CS Approach</th>
<th>CR = 10:1</th>
<th>CR = 5:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNDR (dB)</td>
<td>Proposed Approach</td>
<td>7.46</td>
<td>10.30</td>
</tr>
<tr>
<td></td>
<td>K-SVT(^{54}) + OMP(^{51})</td>
<td>2.73</td>
<td>7.34</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary(^{17}) + OMP(^{51})</td>
<td>6.96</td>
<td>8.18</td>
</tr>
<tr>
<td></td>
<td>Wavelet(^{27}) + OMP(^{51})</td>
<td>-1.51</td>
<td>-1.17</td>
</tr>
<tr>
<td>CA (%)</td>
<td>Proposed Approach</td>
<td>93.63</td>
<td>95.11</td>
</tr>
<tr>
<td></td>
<td>K-SVT(^{54}) + OMP(^{51})</td>
<td>51.07</td>
<td>64.07</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary(^{17}) + OMP(^{51})</td>
<td>64.13</td>
<td>82.59</td>
</tr>
</tbody>
</table>

Table 2.9: Comparison of reconstruction and classification performance of different CS methods on “Romeo”.

<table>
<thead>
<tr>
<th>Database</th>
<th>CS Approach</th>
<th>CR = 10:1</th>
<th>CR = 5:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNDR (dB)</td>
<td>Proposed Approach</td>
<td>8.20</td>
<td>11.00</td>
</tr>
<tr>
<td></td>
<td>K-SVT(^{54}) + OMP(^{51})</td>
<td>3.14</td>
<td>7.27</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary(^{17}) + OMP(^{51})</td>
<td>6.69</td>
<td>8.00</td>
</tr>
<tr>
<td></td>
<td>Wavelet(^{27}) + OMP(^{51})</td>
<td>-1.53</td>
<td>-1.27</td>
</tr>
<tr>
<td>CA (%)</td>
<td>Proposed Approach</td>
<td>94.54</td>
<td>96.88</td>
</tr>
<tr>
<td></td>
<td>K-SVT(^{54}) + OMP(^{51})</td>
<td>44.32</td>
<td>57.98</td>
</tr>
<tr>
<td></td>
<td>Data Dictionary(^{17}) + OMP(^{51})</td>
<td>62.65</td>
<td>88.66</td>
</tr>
</tbody>
</table>
Figure 2.21: Examples of spike sorting performance shown in the PCA domain. (a) and (b) illustrate the spike sorting result of Leicester’s “Easy” and “Difficult” databases at a CR of 20:1. (c) and (d) illustrate the spike sorting result of MGH’s “Pogo” and “Romeo” databases at a CR of 10:1.
Figure 2.22: An example of reconstruction performance of multi-channel neural recordings on the hc-1 database at a CR of 8:1. Blue and red spikes indicate the original neural spikes and the recovered neural spikes, respectively.
2.2.3.3.2 Multi-Channel

In multi-channel experiments, we also compared the reconstruction quality and spike sorting success rate between the proposed approach and other methods. We evaluated performance on the hc-1 database, whose neural signals were recorded by the tetrodes setup. The hc-1 database was recorded from the hippocampus of mice in in vivo experiments. The tetrodes setup consists of four electrodes and one reference that indicates the firing of neurons. Based on the reference, we extracted neural spikes of length 64 from the raw data. The reconstruction quality is measured in terms of the SNDR. However, the database has no prior labels, which means there was no benchmark for us to evaluate the spike sorting success rate quantitatively. Thereby, in this session, we intuitively demonstrate the spike sorting performance using the PCA. Taking advantage of the PCA, we map the spike sorting result into the PCA domain, where different colors intuitively indicate different clusters.

Table V indicates that the proposed approach achieves an average gain of 4 to 5 dB over the other CS-based approaches in terms of the SNDR in multi-channel reconstruction. Figure 2.22 illustrates the multi-channel reconstruction example on the hc-1 database at a CR of 8:1. The blue signals denote the original spikes recorded from the tetrodes setup, which show similar pattern and correlation among the four channels as shown in Figure 2.22. The red signals denote the spikes recovered by the proposed CS-based approach. As shown in Figure 2.22, the recovered signals
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Table 2.10: Comparison of reconstruction performance (in SNDR) of different CS methods on “hc-1”.

<table>
<thead>
<tr>
<th>Database</th>
<th>Compressed Sensing Approach</th>
<th>CR = 4:1</th>
<th>CR = 8:1</th>
</tr>
</thead>
<tbody>
<tr>
<td>hc-1-14521</td>
<td>Proposed Approach</td>
<td><strong>13.93</strong></td>
<td><strong>8.85</strong></td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>6.03</td>
<td>3.18</td>
</tr>
<tr>
<td></td>
<td>Wavelet + OMP</td>
<td>-0.05</td>
<td>-3.64</td>
</tr>
<tr>
<td>hc-1-14531</td>
<td>Proposed Approach</td>
<td><strong>12.18</strong></td>
<td><strong>9.55</strong></td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>7.17</td>
<td>3.43</td>
</tr>
<tr>
<td></td>
<td>Wavelet + OMP</td>
<td>-0.38</td>
<td>-1.97</td>
</tr>
<tr>
<td>hc-1-14921</td>
<td>Proposed Approach</td>
<td><strong>12.64</strong></td>
<td><strong>8.54</strong></td>
</tr>
<tr>
<td></td>
<td>K-SVD + OMP</td>
<td>7.73</td>
<td>4.12</td>
</tr>
<tr>
<td></td>
<td>Wavelet + OMP</td>
<td>-0.24</td>
<td>-2.02</td>
</tr>
</tbody>
</table>

still preserve most of the features, even though only 12.5% of the information of the original signals is used for the reconstruction. The proposed approach is also able to sense and reconstruct neural signals in the continuous time domain, including the low activity region between spikes.

Figure 2.23 shows the multi-channel spike sorting performance at a CR of 16:1 in the PCA domain. Although only 5% of the information is collected for the spike sorting, the clustering results (color coded dots) are consistent with the distinct feature of the original spikes in the PCA domain. As shown in Figure 2.23, the distribution of the principal components among different channels illustrates that neural spikes share similar pattern, which implies correlation in the multi-channel neural recordings.
Figure 2.23: An example of spike sorting performance of multi-channel neural recordings based on the hc-1 database. Figure 11 (a)-(d) indicate the clustering results of channels 1-4, respectively. Different colors represent different clusters.
2.2.3.4 Conclusion

In this section, we presented an unsupervised multi-mode CS approach for neural recording systems. We incorporate the joint-group sparsity in the dictionary learning to extend previous works to multi-channel neural recordings. Additionally, we take advantage of spectral clustering, group structure and template matching to enable spike sorting in real-time experiments in an unsupervised manner.

The approach was evaluated on both synthetic and real databases. The experimental results demonstrated that our approach significantly improved both the reconstruction quality (>8 dB) of neural signals and the spike sorting success rates (>90%) at a high compression ratio (8:1 to 20:1). Our proposed framework, which is hardware friendly, can be integrated in CS-based implantable microsystems for \textit{in vivo} neural recordings. From the perspective of hardware design, the proposed approach further enables energy-efficient CMOS implementations in terms of power consumption. In addition, it also enables online spike sorting in real-time neural recordings, which provides more feasibility for neuroscientists compared to conventional offline spike sorting techniques.

In order to realize a large-scale integration of neural recording systems, we plan to study the quantitative details of the correlation between spikes. By incorporating more structures in the CS framework, we will be able to further improve the performance in terms of reconstruction quality and spike sorting accuracy. Additionally,
a sophisticated online dictionary update approach will also be introduced in the CS framework to enable a more adaptive real-time neural recording system in the future.

2.2.4 Hardware Live Demonstration

In this section, a complete neural recording ASIC based on Compressive Sensing (CS) is demonstrated. Implemented using efficient digital circuit, this CS technique is able to achieve 10 times data compression while consuming only 0.83uW(\text@0.53VDD) additional digital power per electrode. Since CS performance is strongly signal dependent, the ASIC has been tested with standard public database. The characteristic spikes and inter-spike data can be recovered while guarantee >95% classification success rate. The complete signal processing circuit consumes <16\mu W per electrode.

Multi-Electrode Arrays (MEA) are widely used to record neural activity from freely moving animals. These micro-systems must occupy small volumes to fit within the cranium or body of small animals, and must have low heat signatures to avoid tissue damage. As a consequence, signal processing ASICs used for these micro systems often face contradictory bandwidth and power requirements. Compressed Sensing (CS) offers a promising method to condense the entire neural signal waveform using efficient on-chip circuits. According to the CS theory, if the signal can be represented using only a few coefficients in a transform domain (a dictionary e.g. Wavelet or Gabor), then the number of compressed samples required to recover the signal can be much smaller than the Nyquist rate. However, the previous CS systems, using a
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

signal-agnostic CS approach, only demonstrated limited performance in terms of compression rate (CR) and recovery quality when tested using neural signals in a standard database. Furthermore, these systems with predetermine CR operate in open loop configuration where the users have no means to measure the signal reconstruction performance in real time and adjust the system’s compression rate accordingly.

2.2.4.1 Proposed Demonstration

In our previous work,[17][18][46][47] we demonstrated that due to unique characteristic of each neuron’s action potential’s shape, a signal dependent dictionary can be constructed and utilized to increase compression rate and recovery quality in the CS framework.[16][19][22] In this demonstration, we present a complete neural recording system with an integrated CS circuit. To fully utilize the signal-dependent framework, this CS circuit is highly configurable and capable of operating in closed-loop with an off-chip recovery algorithm. Proposed demonstration setup is shown in Figure 2.24. Spike generator, implemented using a PC and a DAC is capable of generating spikes of different shapes and magnitudes. The output of the spike generator is then attenuated to the magnitudes of typical extracellular spikes (50 - 500 uV). The analog output of the attenuator is recorded by the Compressed Sensing ASIC (CS-ASIC) as shown in Figure 2.25 containing analog pre-processors, ADCs, and the core compressed sensing block. The CS-ASIC can be configured to operate in either Dictionary Learning Mode (DLM) or Compression Mode (CM). During DLM, the CS circuit is
bypassed and the raw waveforms are transmitted to allow the off-chip algorithm to construct a dictionary. The chip is then switched back to CM, where the raw data is condensed by the CS circuit. The output of the CS block is transmitted through a wireless link to a quality evaluation (QE) algorithm which continuously computes how close a dictionary item resembles the recovered signal and provides feedback to the on-chip ASIC. Through this feedback, the CR can be adjusted to achieve optimal trade-off between transmission bandwidth and signal recovery quality. The QE results could also detect persistent occurrence of new types of spikes which were not included in the original dictionary. In this case, the ASIC can be switched back to DLM to learn a new dictionary. The performance of the ASIC is tested in an in-vivo experiment as shown in Figure 2.26.
2.2.4.2 Audience Interaction

After the signal is recovered, it is displayed in real-time on a laptop, together with the computed recovery quality and spike cluster performances. The original spike is also displayed along side with the recovered spike. A graphic user interface (GUI) has also been provided for audiences to interact with the system by changing the input spike shapes and viewing how the close loop system can adapt to the newly detect spikes. This system has been successfully demonstrated on IEEE 2014 Biomedical Circuits and Systems (BioCAS) Conference in Lausanne, Switzerland as shown in Figure 2.27.
2.3 Discussion and Conclusion

In Chapter 2, we have proposed the CS-based neural recording systems and developed several supervised and unsupervised dictionary learning algorithms for multi-channel neural recordings and spike sorting. First of all, we described a supervised multi-modal structure dictionary learning for multi-channel neural recordings given the scenario of high-density neural recording electrodes. Taking advantage of group structure and joint sparsity, we boosted both reconstruction and classification performances of multi-channel recordings. Then, in order to overcome the limitations in real-time neural recordings and enable unsupervised spike sorting, we extended the previous work into an unsupervised fashion. We proposed an unsupervised dic-
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

Figure 2.27: Live demonstration using CS-ASIC on IEEE 2014 Biomedical Circuits and Systems Conference in Lausanne, Switzerland.
CHAPTER 2. COMPRESSED SENSING FOR NEURAL RECORDING

tionary learning algorithm with discriminative structures and template matching to enable accurate online spike sorting for in vivo experiments. Finally, we married the previous works to present an unsupervised multi-mode CS approach for neural recordings systems. We incorporated the joint-group sparsity in the dictionary learning compared to individual group structure and joint sparsity. Additionally, the spectral clustering and template matching are also adopted to enable unsupervised spike sorting in real-time experiments. All frameworks have been comprehensively evaluated on both synthetic and real databases. The experiments results have demonstrated that our approaches significantly improved both the reconstruction quality and the spike sorting accuracy by a margin at a high compression ratio. To show its real-time performance, we also presented the hardware live demonstration of proposed frameworks to show the power-efficiency and high accuracy in CMOS implementation.
Chapter 3

Compressed Sensing Pixel-Wise Exposure Control Imaging System

3.1 Introduction

Modern imaging sensors face some fundamental trade-offs: power consumption vs. frame rate, pixel resolution vs. frame rate, signal-to-noise-ratio (SNR) vs. motion blur. High frame rate imaging leads to fast pixel readout with increased power consumption. Given the same power budget, a camera has to sacrifice pixel resolution to provide more frames rate. In addition, high frame rates also limits pixel exposure time which degrades the scene contrast and SNR. In order to meet the low power requirement and leverage the trade-offs, the compression technique for video must be utilized. Inspired by the theory of Compressed Sensing, we developed a low power...
all-CMOS implementation of temporal compressed sensing with pixel-wise coded exposure (PCE). This implementation can increase video pixel resolution and frame rate simultaneously which reducing data readout speed.

As shown in Figure 3.1, a conventional global exposure camera exposes all pixels for fixed amount of time \( T_v \) to readout one image at frame rate of \( \frac{1}{T_v} \). However, in pixel-wise coded exposure framework, pixels are only exposed through a random short “single-on” exposure of fixed amount of time \( T_e \) which is less than \( T_v \). The spatial-temporal video is essentially compressed into one coded image \( I \). Intuitively, the pixel-wise code exposure can be formulated as a sensing matrix \( S \), which takes random samples in the spatial temporal domain of a video scene \( X \). Therefore, the pixel-wise coded exposure can be mathematically defined as a Compressed Sensing optimization problem. By solving the optimization below, the sparse reconstruction can reconstruct the video scene \( \hat{X} \) from the coded image \( I \) using an over-completed
 CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

70 Reconstructed frames. @100 fps Reconstruction (Rec)

\[ \hat{X} = \text{argmin}_a ||a||_0 \text{ s.t. } ||Y - SDa||_2 \leq \epsilon. \] (3.1)

The proposed pixel-wise coded exposure image sensor has been fabricated using a 180 nm CMOS process. Benefiting from the pixel-wise coded exposure, the proposed system has successfully reconstructed the video scene at a high frame rate of 100 frames per second as shown in Figure 3.2. At a compression ratio of 20, the chip consumption only 14 µW compared to 13 mW at full rate. However, this framework can only expose pixels at fixed amount of time \( T_e \) and raises the limitation of sensing.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

high-motion video and high dynamic range scene. Therefore, we are going to exploit the pixel-wise exposure control behind the basic spatiotemporal CS framework and further improve the recovery quality as well as compression ratio. Additionally, we also evaluate this novel technique as a video compression compared to conventional MPEG. The details will be discussed in the following section 3.2.

3.2 Methods

3.2.1 Compressed Sensing Block-Wise Exposure Control Algorithm using Optical Flow Estimation

Converting insects to carrier for low-power CMOS image sensors are becoming popular in the field of machine vision applications such as navigation, monitoring or search-and-rescue operation. To take the advantages of small size and advantageous flight mechanisms of insects, CMOS image sensors have to be fast, low power and adaptable to the environment.

In order to meet the low power requirement for wireless transmission, the compression technique for video must be utilized. In recent years, the pixel-wise coded exposure imaging technique based on the theory of compressed sensing has demon-
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Stratified promising results to increase spatial resolution as well as temporal resolution of image sensors. Within this framework, we are able to compress the video volume into single frame through pixel-wise coded exposure and recover the video volume with high reconstruction quality using the sparsity of nature scenes.

Moreover, insects are able to detect fast motion by visual scene to avoid collisions during flying in nature, hence leading to the additional requirement of high flight speed and adaptivity for insects-based CMOS image sensors design. Optical flow estimation, successfully implemented on CMOS chip, has been widely used in motion detection, objective segmentation and video compression. Therefore, we introduce optical flow estimation in pixel-wise coded exposure framework to detect the pattern of motion and control the different blocks of the sensing cube, changing the exposure duration depending on the speed of motion. We further demonstrate that the reconstruction quality can be further improved through block-wise exposure control.

In this section, we describe a compressed sensing block-wise exposure control framework as shown in Figure 3.3 using optical flow estimation with two specific contributions:

a) **Block-wise exposure control using optical flow estimation**: It is well known that short exposure time preserves high-frequency details and avoids motion blurring caused by fast motion. At the same time, long exposure has denoising effect at low illumination scenes. Therefore, our approach is to control the exposure du-
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

ration of different blocks in the sensing video cube based on the estimated motion information of a sequence of time-varying images.

b) Low power consumption and hardware-friendly for CMOS architecture: Optical flow algorithm has been successfully implemented in CMOS for low power motion detection and velocity estimation. The compressed sensing framework has also been implemented on CMOS image sensor for multiple biological applications. The combination of optical flow and compressed sensing could further improve reconstruction quality of image sensor, while also increase compression ratio and reduce power.

3.2.1.1 Compressed Sensing Block-Wise Exposure Control

Assume we have a video volume $X \in \mathbb{R}^{M \times N \times T}$ containing a sequence of time-varying images, where $M$ and $N$ indicate the height and width of each frame while $T$ and $X(i, j, t)$ indicate the frame number of volume and the intensity of pixel at the position $(i, j)$ of frame $t$, respectively. Our approach is organized as follows.

a) Optical flow estimation: In optical flow estimation, we compute the optical flow velocity between every two consecutive images $X_i$ and $X_{i+1}$ based on the approach given by Lucas et al. The average of optical flow velocity is saved as the matrix $V \in \mathbb{R}^{M \times N \times \frac{1}{T}}$. As shown in Figure 3.4, the red arrows indicate the region with high-motion velocity which is supposed to reduce exposure time in block-wise control step. On the contrary, the background without optical velocity indicates the static region of the image, which means we should increase the exposure time appropriately.
Figure 3.3: Block diagram of the proposed compressed sensing CMOS image sensors using optical flow estimation. Video frames are sampled by the exposure-coded pixels. Optical flow pixels are placed within a block of exposure coded pixels. Frames are then compressed into one coded image through exposure controlled pixel. The optical flow pixel measures the scene motion and adjust the exposure time for optimal image quality. One coded image is transmitted wirelessly. The spatial temporal video scene is then reconstructed off-chip.
Figure 3.4: An intuitive demonstration of optical flow estimation using real dataset. Red arrows indicate the region with high-motion.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

b) Block-wise exposure control: Based on the motion information $\mathbf{V}$ computed from optical flow estimation, we design the block-wise sensing cube as shown in Algorithm 8. The algorithm generates sub-sensing cube for different blocks with size $p \times p$. Figure 3 intuitively demonstrates the block-wise coded exposure control. If the velocity of the block is larger than the pre-set threshold, then the exposure duration of the corresponding sub-sensing cube is designed to be short and vice versa. As shown in Figure 3, red blocks associated with high-motion velocity generate short exposure duration while blue blocks associated with low-motion velocity generate long exposure duration with long exposure time. Specifically, each point belonging to the same block has the same exposure time but the starting point of the exposure duration is random. Currently Algorithm 1 only presents two exposure durations based on a single threshold. Multiple level threshold design has to be considered to trade-off between complexity of circuits and reconstruction quality, which will be discussed in the future.

c) Compressed sensing and sparse recovery: Algorithm 9 demonstrates the compressed sensing and sparse recovery algorithm. Based on the theory of compressed sensing, we are able to compress the video volume $\mathbf{X}$ into single frame measurement $\mathbf{y}$ using block-wise sensing cube $\mathbf{S}$ with compression ratio of $T$. To maximize the level of sparsity, we rely on an overcomplete dictionary $\mathbf{D}$ for sparsifying the signal $\mathbf{X}$, which is learned by K-SVD algorithm using training dataset. For sparse recovery, we implement LARS algorithm to solve the optimization problem.
Figure 3.5: An intuitive example of block-wise sensing cube control using optical flow estimation. Red and blue arrows indicate the area with high-motion velocity and low-motion velocity respectively, which leads to short exposure duration in red block sensing cube and long exposure duration in blue block sensing cube.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Algorithm 8 Block-Wise Exposure Control Algorithm

Require: Video volume $\mathbf{X} \in \mathbb{R}^{M \times N \times T}$, $M \times N$ indicates the size of each video frame and $T$ indicates the frame number of the video volume. Patch size $p$ indicates the dimension of sub-sensing cube. Velocity matrix $\mathbf{V} \in M \times N$ and velocity threshold. Short exposure time $t_1 \in \mathbb{Z}$ and long exposure time $t_2 \in \mathbb{Z}$ and $0 < t_1 < t_2 < T$.

1: Generate Sensing Cube $\mathbf{S} \in \mathbb{R}^{M \times N \times T}$ and set $\mathbf{S}(i,j,t) = 0$ for all $i \in \{1,2,...,M\}$, $j \in \{1,2,...,N\}$ and $t \in \{1,2,...,T\}$.
2: for $i := 1$ to $M - p + 1$ do
3:     for $j := 1$ to $N - p + 1$ do
4:         if $\max(\mathbf{V}(i,j)) > \text{threshold}$ then
5:             Set $\mathbf{S}(i : i + p - 1, j : j + p - 1, t : t + t_1) = 1$ and $t$ is a number randomly picked from the set $\{1,2,...,T - t_1\}$ for each pixel.
6:         else
7:             Set $\mathbf{S}(i : i + p - 1, j : j + p - 1, t : t + t_2) = 1$ and $t$ is a number randomly picked from the set $\{1,2,...,T - t_2\}$ for each pixel.
8:     end if
10:    end for
11:   $i = i + p$.
12: end for
13: Return Sensing Cube $\mathbf{S}$

3.2.1.2 Experiments

In this section, we compare the recovery performance of video frames of proposed approach with the global shutter, flutter shutter, rolling shutter, random shutter with short exposure time and random shutter with long exposure time. The dataset that we use is Caltech Pedestrian Dataset where the size of each image is $480 \times 640$ and the frame number is fixed at 10 and 30 for experiments. For all experiments, we randomly divide the data into two halves with one part for dictionary training and the other for testing. We implement the same recovery algorithm in each experiment for video recovery.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Algorithm 9 Compressed Sensing and Sparse Recovery

Require: Video volume \( X \in \mathbb{R}^{M \times N \times T} \). Sensing Cube \( S \in \mathbb{R}^{M \times N \times T} \) where \( X_i \) and \( S_i \) indicate \( i \)th frame of video volume \( X \) and sensing cube \( S \), \( i = 1, 2, ..., T \). The overcomplete dictionary \( D \) learned by K-SVD algorithm using training dataset.\(^\text{13}\) Reconstruction error \( \epsilon \).

1: Compressed Sensing: Compute measurement \( y(i, j) = \sum_{t=1}^{T} S(i, j, t)X(i, j, t) \) where \( y \in \mathbb{R}^{M \times N} \), \( i \in \{1, 2, ..., M\} \) and \( j \in \{1, 2, ..., N\} \).
2: Sparse Recovery: Solve the optimization problem via LARS algorithm,\(^\text{13}\) where \( y \) and atoms of \( SD \) are vectorized.
\[
\min_{a} \|a\|_0 \ \text{s.t.} \ \|y - SDa\|_2 \leq \epsilon.
\]
3: Return Recovered video volume \( \hat{X} = Da \).

The recovery performance is measured in the average of peak signal-to-noise ratio (Avg. PSNR) and root-mean-square error (Avg. RMSE) of all frames between original images and recovered images, which are defined as:

\[
\text{Avg. RMSE} = \frac{1}{T} \sum_{t=1}^{T} \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (X(i, j, t) - \hat{X}(i, j, t))^2} \tag{3.2}
\]

\[
\text{Avg. PSNR} = \frac{1}{T} \sum_{i=1}^{T} 20 \log \frac{\text{max}(X_i)}{\text{RMSE}}. \tag{3.3}
\]

\( X \) and \( \hat{X} \) indicate respectively original video volume and recovered video volume whose size is \( m \times n \times T \) while \( \text{max}(X_i) \) is the maximum pixel value of the \( i \)th frame of video volume \( X \).
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Table 3.1: Comparison of reconstruction performance (in PSNR and RMSE) of different CS methods with compression ratio of 10.

<table>
<thead>
<tr>
<th>CS Approach</th>
<th>Global Shutter</th>
<th>Flutter Shutter</th>
<th>Rolling Shutter</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (in dB)</td>
<td>20.70</td>
<td>17.49</td>
<td>11.36</td>
</tr>
<tr>
<td>RMSE</td>
<td>34.80</td>
<td>50.79</td>
<td>80.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CS Approach</th>
<th>Random Short</th>
<th>Random Long</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (in dB)</td>
<td>22.86</td>
<td>23.64</td>
<td><strong>25.60</strong></td>
</tr>
<tr>
<td>RMSE</td>
<td>23.24</td>
<td>22.75</td>
<td><strong>17.71</strong></td>
</tr>
</tbody>
</table>

Table 3.1 and Table 3.2 demonstrate the reconstruction performance of different approaches with two different compression ratio of 10 : 1 and 30 : 1. Random shutter with fixed exposure duration and proposed approach outperform global shutter, flutter shutter and rolling shutter in terms of reconstruction performance. Moreover, proposed approach with block-wise exposure control further improve the recovery quality compared with the random shutter with fixed exposure time. As shown in Figure 3.6, flutter shutter and rolling shutter are not able to recover the frame globally while global shutter leads to blurring in high-motion region. At the same time, block-wise approach performs much better than the three approaches and also further improve the reconstruction details especially in the region of walking pedestrians.

3.2.1.3 Conclusion

In this section, we described a compressed sensing block-wise exposure control algorithm using optical flow estimation. This framework is designed to be implemented
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Table 3.2: Comparison of reconstruction performance (in PSNR and RMSE) of different CS methods with compression ratio of 30.

<table>
<thead>
<tr>
<th>CS Approach</th>
<th>Global Shutter</th>
<th>Flutter Shutter</th>
<th>Rolling Shutter</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (in dB)</td>
<td>19.12</td>
<td>18.18</td>
<td>9.99</td>
</tr>
<tr>
<td>RMSE</td>
<td>39.29</td>
<td>43.18</td>
<td>90.53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CS Approach</th>
<th>Random Short</th>
<th>Random Long</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (in dB)</td>
<td>22.67</td>
<td>21.32</td>
<td>23.33</td>
</tr>
<tr>
<td>RMSE</td>
<td>25.00</td>
<td>32.20</td>
<td>23.75</td>
</tr>
</tbody>
</table>

Figure 3.6: The reconstruction example of one frame using different approaches with a compression ratio of 10 : 1.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

for CMOS image sensors mounted on insects. To meet the requirement of high mobility and power efficiency of insect-based image sensors, we combine the compressed sensing theory with optical flow estimation. The block-wise exposure control brings dynamic exposure duration on space domain, which achieves better reconstruction quality ($> 25 \, dB$) at high compression ratio ($\geq 10 : 1$) than the approach with fixed exposure time. In the future, we also would like to extend this work to real implementation of CMOS image sensors mounted on insects and rigorously study the fundamental limits as well as various engineering trade-offs of the proposed framework.

### 3.2.2 Spatiotemporal Compressed Sensing for Video Compression

As advancements in integrated circuit fabrication steadily continue yielding smaller process sizes and higher circuit densities, new low power methods of data compression are needed in applications where high data rates are ubiquitous. One such application is mobile video transmission where raw data rates, which can range from 2 Mb/s to 2 Gb/s, are too large for conventional transmission methods. Past compression methods that demonstrate high compression factors require complex circuitry and are computationally demanding, neither of which are suitable for low power, resource limited IC design. To satisfy the power and computational requirements for these mobile ap-
Applications, our approach combines the processes of data acquisition and compression while reducing computational complexity and processing time as compared to methods that treat them as disjoint. Biomedical applications such as functional imaging\textsuperscript{66} and calcium imaging\textsuperscript{67} can benefit from such an energy efficient and real-time imaging acquisition system.

MPEG standard is one of the most heavily used methods for video compression, which serves as the core of many DVD formats and digital television broadcasts. This lossy form of compression achieves high compression rate (CR) that vary depending on the version of encoding. The high level of compression achieved by MPEG is a result of multiple operations (i.e. motion estimation) taken on the video data in both the spatial and temporal domain, however, the most effective steps in the process have the caveat of also being the most computationally intensive and time dependent, which adds the difficulty of efficient hardware implementation.

One of the computational intensive operations is information reduction through video motion estimation. This operation first computes motion vectors for blocks of pixels between frames. But computing the motion vectors can require an entire frame search for similar blocks which is a time intensive process. Thus the search is typically reduced to a percentage of the frame to increase processing speed. But this may lead to unsuccessful searches. State of the art solutions to computing these motion vectors without an entire frame search have arisen but at the cost of computational complexity. An integral step in compressing the data is taking the discrete cosine
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Transform (DCT) of pixel blocks. Taking advantage of the DCT by removing the insignificant high frequency components of individual frames yields high amounts of compression, however, it is computationally expensive and increases quadratically with the pixel block size. Reducing this size of the pixel blocks has the effect of reducing the time and processing power needed for the DCT but also increases the amount of motion vectors needed to encode the data and increases the compression scheme’s susceptibility to noise.

Inspired by the theory of Compressed Sensing (CS), a number of temporal CS systems has been proposed to alleviate the intensive computation and enable hardware-friendly implementation for video compression. For example, Llull demonstrated a coded aperture compressive temporal imaging system, which is able to reconstruct 148 frames per coded image. Koller et al. also showed a prototype compressive video camera at 740 fps using CMOS sensors and silicon-dioxide optical coded mask. Tsai extended this technique to compress a multi-spectral, high-speed scene into a monochrome scene using objective lens, coded aperture, piezoelectric stage and monochrome CCD camera. Finally, Liu proposed an efficient space-time sampling approach with pixel-wise coded exposure, which uses a prototype liquid-crystal-on-silicon device to modulate light prior to the image sensor.

In order to meet the requirements of high compression rate and power efficient implementation, we proposed a hardware-friendly spatiotemporal compressed sensing framework for video compression with following contributions.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Figure 3.7: The block diagram of spatiotemporal compressed sensing framework

a) Spatiotemporal Compressed Sampling: The spatiotemporal compressed sensing is the key component of our proposed framework, as shown in Figure 3.7. Unlike conventional video compression technique, the spatiotemporal compressed sensing requires no motion estimation, compensation and DCT to reduce bits by identifying and eliminating statistical redundancy. In spatiotemporal compressed sensing, the pixels are simply exposed through a random short “single on” exposure of fixed duration, which essentially compresses a spatiotemporal video into a single coded image. Additionally, as shown in Figure 3.8, the spatiotemporal compressed sensing is more optimal for video compression because it samples both the spatial and temporal information simultaneously, which sets it apart from the conventional spatial compressed sampling.

b) Hardware-friendly On-Chip Encoding for CMOS Architecture: In spatiotemporal compressed sensing, a sensing cube is adopted to encode the spatiotemporal
video into a single coded image. In details, the sensing cube is composed of either 1 or 0, where 1 intuitively indicates exposure is turned on and vice versa as shown in Fig. 3.8. Therefore, the encoding for video compression can be formulated as a simple addition operation compared to conventional technique, which suffers from the intensive computation. Taking advantage of the simple arithmetic computation, the spatiotemporal compressed sensing is hardware-friendly and enables the real-time and power-efficient implementation on CMOS architecture.\footnote{This section is organized in the following structure: In section II, we introduce compressed sensing theory and demonstrate the spatiotemporal compressed sensing approach with corresponding hardware architecture. In section III, we demonstrate the experiments results and discuss the advantages and limitations of proposed approach for video compression compared to other video compression techniques such as MPEG. Finally, we conclude this section in section IV.

3.2.2.1 Compressed Sensing

Compressed Sensing (CS) theory\cite{Donoho06, Candes06} demonstrates that a $S$-sparse signal $x \in \mathbb{R}^N$ is essentially compressed into a measurement $y \in \mathbb{R}^M$ by a sensing matrix $S \in \mathbb{R}^{M \times N}$, where normally $S \ll M < N$. Given the Restricted Isometry Property (RIP) and $M \sim S \log \frac{N}{S}$ satisfied, the signal $x$ can be exactly reconstructed by solving the optimization problem below.
min_x ||x||_1 s.t. y = Sx. \quad (3.4)

However, the signal of image or video scene x is not sparse with respect to time or frequency domain. An over-complete dictionary \( D \in \mathbb{R}^{N \times L} \) needs to be adopted to sparsify the signal x, where \( x = Da \) and \( a \in \mathbb{R}^L \) is sparse. Therefore, the reconstruction problem can be formulated as:

\[
\min_a ||a||_1 \text{ s.t. } y = SDa. \quad (3.5)
\]

The reconstruction problem can be solved by using \( \ell - 1 \) norm optimization and then the signal is recovered as \( \hat{x} = Da \) at compression rate \( \frac{M}{N} : 1 \).

### 3.2.2.2 Spatiotemporal Compressed Sensing

As shown in Figure 3.7, the block diagram illustrates the basic framework of the spatiotemporal compressed sensing. In our proposed framework, the spatiotemporal compressed sensing essentially compresses the video scene into a single coded image and wirelessly transmits the coded image to off-chip terminal for reconstruction. Basically, the framework is composed of two stages: encoder and decoder. The encoder, which has been successfully implemented on chip\(^m\), encodes the video scene into the
To illustrate how spatiotemporal compressed sensing works, we assume there is a spatiotemporal video signal \( \mathbf{X} \in \mathbb{R}^{W \times H \times T} \), where \( W \times H \) denotes the size of each frame, \( T \) denotes the total number of frames in the video and \( \mathbf{X}(w,h,t) \) denotes the intensity value associated with the frame \( t \) at position \( (w,h) \). A sensing cube \( \mathbf{S} \in \mathbb{R}^{W \times H \times T} \) is also given, which stores the spatiotemporal exposure control values for pixel at \( (w,h,t) \). In details, the sensing cube value for each pixel is defined as:

\[
\mathbf{S}(w,h,t) = \begin{cases} 
1 & t \in [t_1, t_2] \\
0 & \text{otherwise}
\end{cases}
\]  

(3.6)

where \( 0 \leq t_1 < t_2 \leq T \) and \( t_2 - t_1 \) intuitively indicates the exposure duration. \( t_1 \)
is randomly chosen for each pixel while the exposure duration is fixed.

Given the video scene \( X \) and sensing cube \( S \), the coded image \( Y \in \mathbb{R}^{M \times N} \) is computed as:

\[
Y(w, h) = \sum_{t=1}^{T} S(w, h, t) \cdot X(w, h, t) \quad \forall w, h.
\]

Therefore, the spatiotemporal compressed sensing encodes the video \( X \) into a coded image \( Y \) at compression rate \( T : 1 \).

During the reconstruction, we recover the spatiotemporal video, \( \hat{X} \in \mathbb{R}^{W \times H \times T} \), by solving the optimization problem,

\[
\min_{a} \frac{1}{2} ||Y - SDa||^2 + \lambda ||a||_1, \tag{3.7}
\]

where \( D \in \mathbb{R}^{N \times L} \) is an over-complete dictionary learned from the training sample and \( a \in \mathbb{R}^L \) is the sparse coefficient vector. \( \lambda \) is the linear combination coefficient for controlling the sparsity in the recovery. Finally, the reconstructed video is computed as \( \hat{X} = Da \).

### 3.2.2.3 Experiments

In this section, we evaluate the spatiotemporal CS approach on the database compared to standard video compression MPEG in terms of reconstruction quality and
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

compression rate (CR). The reconstruction quality is measured in peak signal-to-noise ratio (PSNR). The spatiotemporal video scenes are composed of two categories: “car” with slow motion and “pedestrian” with fast motion. The resolution of each video frame is $480 \times 640$ while the number of frame of video scenes is 10, 20 and 30 respectively. We also demonstrate the performance on the noisy video scenes to demonstrate the robustness. Additionally, we compare the spatiotemporal CS with other video compression techniques from the perspective of hardware implementation.

Table 3.3 and 3.4 demonstrate the performance on the videos without noise. In this experiment, MPEG technique dominates the spatiotemporal CS approach in terms of PSNR because MPEG can take advantage of precise motion detection and estimation to preserve the details of video scene without noise. The spatiotemporal CS approach loses some information in the encoding but the compression rate is more flexible compared to MPEG. Table 3.5 and 3.6 demonstrate the performance on the videos with low noise. The proposed approach achieves better performance on the “car” videos and comparable performance on the “pedestrian” videos in terms of reconstruction quality. Furthermore, the proposed approach still achieves high compression rate while MPEG is reduced to around 5 : 1. Table 3.7 and 3.8 show the performance on the videos with high noise, which demonstrates the robustness of proposed approach on the videos with high noise. The spatiotemporal CS approach outperforms MPEG in terms of reconstruction quality and compression rate. The robustness of spatiotemporal CS on noisy data benefits from the pixel-wise exposure,
which averages the noise during the exposure. Besides, sparse reconstruction using the dictionary that learned from the training sample also further helps improve the robustness. Figure 3.9 demonstrates the example of reconstruction video frames on different noisy level.

**Table 3.3:** Comparison of reconstruction performance (in PSNR and CR) of “Car” without noise.

<table>
<thead>
<tr>
<th>Approach</th>
<th>10 Frames</th>
<th>20 Frames</th>
<th>30 Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
</tr>
<tr>
<td>Spatiotemporal</td>
<td>25.08</td>
<td>10:1</td>
<td>25.17</td>
</tr>
<tr>
<td>MPEG</td>
<td>31.45</td>
<td>16:1</td>
<td>31.30</td>
</tr>
</tbody>
</table>

**Table 3.4:** Comparison of reconstruction performance (in PSNR and CR) of “Pedestrian” without noise.

<table>
<thead>
<tr>
<th>Approach</th>
<th>10 Frames</th>
<th>20 Frames</th>
<th>30 Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
</tr>
<tr>
<td>Spatiotemporal</td>
<td>24.59</td>
<td>10:1</td>
<td>23.74</td>
</tr>
<tr>
<td>MPEG</td>
<td>30.03</td>
<td>15:1</td>
<td>30.46</td>
</tr>
</tbody>
</table>

**Table 3.5:** Comparison of reconstruction performance (in PSNR and CR) of “Car” with low noise.

<table>
<thead>
<tr>
<th>Approach</th>
<th>10 Frames</th>
<th>20 Frames</th>
<th>30 Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
</tr>
<tr>
<td>Spatiotemporal</td>
<td>23.69</td>
<td>10:1</td>
<td>23.73</td>
</tr>
<tr>
<td>MPEG</td>
<td>22.62</td>
<td>5:1</td>
<td>22.55</td>
</tr>
</tbody>
</table>

Additionally, we also compare the spatiotemporal compressed sensing with other video compression techniques from the perspective of hardware implementation in Table 3.9. In order to enable MPEG technique for mobile applications, the encoders
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Table 3.6: Comparison of reconstruction performance (in PSNR and CR) of “Pedestrian” with low noise.

<table>
<thead>
<tr>
<th>Approach</th>
<th>10 Frames</th>
<th></th>
<th>20 Frames</th>
<th></th>
<th>30 Frames</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
<td>CR</td>
</tr>
<tr>
<td>Spatiotemporal</td>
<td>23.00</td>
<td>10:1</td>
<td>21.45</td>
<td>20:1</td>
<td>22.48</td>
<td>30:1</td>
</tr>
<tr>
<td>MPEG</td>
<td>22.50</td>
<td>4:1</td>
<td>22.51</td>
<td>4:1</td>
<td>22.53</td>
<td>4:1</td>
</tr>
</tbody>
</table>

Table 3.7: Comparison of reconstruction performance (in PSNR and CR) of “Car” with high noise.

<table>
<thead>
<tr>
<th>Approach</th>
<th>10 Frames</th>
<th></th>
<th>20 Frames</th>
<th></th>
<th>30 Frames</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
<td>CR</td>
</tr>
<tr>
<td>Spatiotemporal</td>
<td>25.08</td>
<td>10:1</td>
<td>25.17</td>
<td>20:1</td>
<td>23.43</td>
<td>30:1</td>
</tr>
<tr>
<td>MPEG</td>
<td>19.59</td>
<td>3:1</td>
<td>19.57</td>
<td>3:1</td>
<td>19.55</td>
<td>3:1</td>
</tr>
</tbody>
</table>

have optimized the processing steps to reduce power and realize real-time encoding. Mochizuki\cite{Mochizuki2009} demonstrates a fully developed H.264/MPEG-4 codec on chip in the 90 nm process capable of encoding 30 fps HD-sized video in real-time. This implementation achieves more than a 50% increase in power efficiency as compared to other implementations of other MPEG on chip codecs. Comparably, our work implemented on chip demonstrates a 69% increase in power efficiency over this leading implementation. Other work\cite{OtherWork2020} in MPEG-like compression for mobile applications demonstrate increased power efficiency as compared to our work. These new methods of encoding greatly reduce the computational intensity by approximating the DCT coefficients and reducing the area of search for motion estimation. The work presented by employing a H.265/HVEC encoder in the 28nm process compresses 30 fps HD-sized video with a power efficiency rating of .5 nJ/pixel. This is compared to our
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Table 3.8: Comparison of reconstruction performance (in PSNR and CR) of “Pedestrian” with high noise.

<table>
<thead>
<tr>
<th>Approach</th>
<th>10 Frames</th>
<th></th>
<th>20 Frames</th>
<th></th>
<th>30 Frames</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
<td>CR</td>
<td>PSNR</td>
<td>CR</td>
</tr>
<tr>
<td>Spatiotemporal</td>
<td>22.36</td>
<td>10:1</td>
<td>22.05</td>
<td>20:1</td>
<td>22.04</td>
<td>30:1</td>
</tr>
<tr>
<td>MPEG</td>
<td>19.68</td>
<td>3:1</td>
<td>19.69</td>
<td>3:1</td>
<td>19.68</td>
<td>3:1</td>
</tr>
</tbody>
</table>

Figure 3.9: The demonstration of coded images and reconstructed video frames at compression rate 20:1. A (red): without noise; B (green): low noise; C (blue): high noise.

work which demonstrates a power efficiency rating of .7174 nJ/pixel and incorporates both video acquisition as well as compression in a process size 84% larger. Our work demonstrates the ability of spatial temporal compressive sampling to outperform conventional compression methods in terms of power efficiency and shows promise for further performance increase in smaller process sizes.
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Table 3.9: Comparison of On-Chip Implementation (in Technology (nm) and Power (nJ/Pixel))

<table>
<thead>
<tr>
<th>Technology</th>
<th>Power</th>
<th>Technology</th>
<th>Power</th>
<th>Technology</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>0.7</td>
<td>90</td>
<td>2.3</td>
<td>28</td>
<td>0.5</td>
</tr>
</tbody>
</table>

3.2.2.4 Conclusion

In this section, we presented a hardware-friendly spatiotemporal compressed sensing framework for video compression. The spatiotemporal compressed sensing incorporates random sampling in both spatial and temporal domain to encode the video scene into a single coded image. During decoding, the video is reconstructed using dictionary learning and sparse recovery. The evaluation results demonstrate the proposed approach can achieve high compression rate ($10:1 - 30:1$) and robustness reconstruction quality ($>20dB$) on noisy database. Additionally, it also enables power efficient and real-time CMOS implementation (0.7 nJ/pixel).

Taking advantage of spatiotemporal compressed sampling, the video scene can be encoded efficiently at high compression rate and decoded using sparse recovery. The evaluation results also demonstrate its performance in terms of reconstruction quality and compression rate. Compared to conventional MPEG standard, proposed approach achieves better robustness on noisy database and realizes flexible compression rates. Furthermore, the simple arithmetic computation of spatiotemporal compressed sensing is suitable for the power-efficient and real-time biomedical CMOS
3.2.3 Hardware Live Demonstration

In this section, a compact all-CMOS spatiotemporal compressed sensing (CS) video camera is demonstrated. This CS-based framework implemented on integrated circuits, is able to achieve 20-fold reduction in the readout speed and consumes only 14µW to provide 100 fps videos. Taking advantage of dictionary learning and sparse recovery, this prototype image sensor (127×90 pixels) can reconstruct 100 fps videos from the coded images sampled at 5 fps.

Modern video cameras face some fundamental trade-offs: power consumption vs. frame rate, pixel resolution vs. frame rate, and signal-to-noise ratio (SNR) vs. motion blur. A high frame rate enables fast pixel readout speed, but increases power consumption. Given their limited power budget, the cameras suffer from the trade-off between high frame rate and high pixel resolution. Additionally, a high frame rate also adds to the difficulty of flexible control of exposure time, which is able to denoise and improve the SNR.

Inspired by the theory of CS, a number of CS-based imaging techniques have been proposed to improve the spatial and temporal resolution of image sensors. However, all these works adopt optical apparatus to pre-modulate the video scene before the image sensor. The size and power consumption of such an image sensor cannot be significantly reduced from the perspective of hardware implementation.
3.2.3.1 Proposed Demonstration

In order to overcome the limitation of previous optical-based CS systems, we propose an all-CMOS spatiotemporal CS video camera with pixel-wise code exposure (PCE). In this demonstration, we present the complete CS video camera with integrated CS circuits. We fabricated the image sensor using a 180nm CMOS process, as shown in Figure 3.10 (a). The CS circuit is highly configurable and capable of interacting with an off-chip reconstruction algorithm.

The proposed demonstration prototype is shown in Figure 3.10 (b). An in-house spatiotemporal video is compressed by on-chip sensing into a single coded image. Upon receiving the coded image, the sparse recovery algorithm together with the trained dictionary reconstructs the video from the coded image. The sensing is operated on-chip, while the reconstruction is operated off-chip.

3.2.3.2 Audience Interaction

After the spatiotemporal video is compressed and recovered, the coded image and recovery video will be displayed on a PC, together with the compression ratio (CR). Since the original video is compressed before Analog to Digital conversion, we cannot demonstrate the SNR of the recovered video in the real-time experiment. However, a simulation demonstration will be presented to show the performance of the proposed system in terms of SNR and CR. The original video will also be displayed along with
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

![Figure 3.10: Demonstration of (a) chip micrograph and (b) prototype](image)

Figure 3.10: Demonstration of (a) chip micrograph and (b) prototype

the recovered video. A GUI as shown in Figure 3.11 is also be designed for audiences to interact with the system by changing the CR, exposure time and readout speed. This system has been successfully demonstrated on IEEE 2017 International Symposium on Circuits and Systems (ISCAS) in Baltimore, Maryland, USA as shown in Figure 3.12.

3.3 Discussion and Conclusion

In this chapter, we introduced a novel CS-based spatiotemporal pixel-wise exposure (PCE) control framework for imaging systems. The proposed approach can increase video pixel resolution and frame rate simultaneously which reduces data readout speed. Given the same power budget, such a design is significantly energy-efficient
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

Figure 3.11: Demonstration of GUI in ISCAS 2017

Figure 3.12: Demonstration of CS spatiotemoral image sensor in ISCAS 2017
CHAPTER 3. COMPRESSED SENSING PIXEL-WISE EXPOSURE CONTROL IMAGING SYSTEM

from the perspective of hardware implementation using an over-complete dictionary and sparse representation. We also investigated the relationship between motions and the pixel-wise exposure control feature in this framework. Conventionally, the duration of pixel-wise exposure in PCE control is fixed at an amount of time. Compared to the traditional scheme, we studied the dynamic block-wise exposure control using optical flow to further improve the reconstruction quality of compressed video with motions and satisfy the demand of high mobility sensors. In addition, we also have studied the compression function behind the spatiotemporal CS scheme. Compared to the conventional video compression technique like MPEG, the spatiotemporal CS requires no motion estimation, compensation and discrete cosine transformation (DCT) to reduce bits. The pixels in videos are simply and randomly integrated into a coded image for post reconstruction without any handcrafted feature extraction. We experimentally showed that the proposed spatiotemporal CS approach could achieve high compression rate and robustness reconstruction in noisy database. We also demonstrated its corresponding CMOS implementation and enabled the compression as well as reconstruction in real-time experiment.
Chapter 4

Deep Learning for Medical Image Analysis

4.1 Automatic Localization and Identification of Human Vertebrae using Deep Neural Network

4.1.1 Introduction

Automatic and accurate landmark localization and identification such as human vertebrae detection and labeling has been developed to a crucial tool in 2D or 3D medical imaging, e.g. computed tomography (CT), magnetic resonance imaging (MRI) and
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

X-ray. General clinical tasks such as pathological diagnosis, surgical planning and post-operative assessment can be beneficial from this locate-and-name tool. Specific applications in human vertebrae detection and labeling include vertebrae segmentation, fracture detection, tumor detection, registration and statistical shape analysis. However, designing such an automatic and accurate vertebrae detection and labeling framework meets several challenges such as pathological cases, image artifacts and limited field-of-view (FOV) as shown in Figure 4.1. Pathological cases are arise from spine curvature, fractures, deformity and degeneration, of which spinal shapes are significantly different compared to normal anatomy. Image artifacts such as surgical metal implants greatly alter the appearance of vertebrae and change the intensity of image greatly. Additionally, FOVs given by spine-focused scans also add difficulty to the localization and identification of each vertebra due to the repetitive nature of these vertebrae and lack of global spatial and contextual information. In order to address these challenges, an accurate and efficient spine detection algorithm is required for clinical tasks.

To meet both the requirements of accuracy and efficiency, many approaches have been presented in recent decade. Generally, they are broken into two approaches: conventional machine learning approach and deep neural network approach. Schmidt et al. proposed an efficient method for part-based localization of spine detection which incorporates contextual shape information in a probabilistic graphic model. Features for detecting parts are learned from the training database and detected by a
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

Figure 4.1: Demonstration of pathological cases. (a) Surgical Metal Implants (b) Spine Curvature (c) Limited FOV

multi-class classifier followed by a graphical model. It is evaluated on a MRI database and demonstrates robust detection even when some of vertebrae are missing in the image. Glocker et al.\textsuperscript{76} designed an algorithm based on regression forests and probabilistic graphical models. This two-stage approach is quantitatively evaluated on 200 CT scans, which has achieved an identification rate of 81 %. Furthermore, Glocker et al.\textsuperscript{77} extend this vertebrae localization approach to address the challenge in pathological spine CT. This approach is built on supervised classification forests and evaluated on a challenging database of 224 pathological spine CT scans. It obtains an overall mean localization error of less than 9 mm with an identification rate of 70 %, which outperforms state-of-the-art on pathological cases at that moment. Recently, deep neural networks (DNN) have been introduced to solve computer vision tasks such as image classification, scene segmentation and object detection. DNN have been
highlighted in the research of landmark detection in medical imaging and demonstrated its outstanding performance compared to conventional approaches. Chen et al.\cite{78} proposed a joint learning model with convolutional neural networks (J-CNN) to effectively localize and identify the vertebrae. This approach, which is composed of a random forest classifier, a J-CNN and a shape regression model, has improved the identification rate (85%) with a large margin with smaller localization errors in the same challenging database.\cite{127} \cite{77} Suzani et al.\cite{79} presented a fast automatic vertebrae detection and localization approach using deep learning. The approach first extracts intensity-based features from the voxels in the CT scans and then performs a deep neural network on these features to regress the distance between the center of vertebrae and the reference voxel. It has achieved high detection rate with fast evaluation time but suffers from the large mean error compared to other approaches.\cite{77,78}

In order to take advantage of deep neural networks and overcome the limitations in vertebrae detection, we proposed an approach with the following contributions.

a) Deep Image-to-Image Network (DI2IN) for Voxel-Wise Regression

Compared to approaches which require handcrafted features from input images, the proposed deep image-to-image network (DI2IN) directly performs on the 2D X-ray images or 3D CT volumes and generates the multi-channel probability maps which are associated with different vertebrae. The probability map itself intuitively indicates the location and type of vertebra. Additionally, the proposed DI2IN does not adopt any classifiers to coarsely remove outliers in pre-processing for speed-up.
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

By building the DI2IN in a fully convolutional manner, it is significantly efficient in terms of computation time, which sets it apart from the sliding window approaches.

\( b) \) Response Enhancement with Message Passing

Although the proposed DI2IN usually provides high confident probability maps, sometimes it produces few false positives due to the similar appearance of vertebrae. The anatomical structure of spine provides a strong geometric prior for vertebral centroids. In order to fully explore such prior, we introduce a message-passing scheme which can communicate information of the neighborhood in space. At first, the chain-structured graph is constructed based on the prior on vertebra structure. The graph connection directly defines the neighborhood of each vertebra. Second, for the neighboring centroids, we learn the convolutional kernels between the probability maps. At inference, the probability maps from previous step are further convoluted with the learned kernels to help refine the prediction of neighbors’ probability maps. In this chapter, the messages are passed via the convolution operations and recurrent neural networks between neighbors. After a few iterations of message passing, the probability maps converge to a stable state. Finally, the probability maps of vertebrae are enhanced, and the issues, such as missing response or false positive response, are well compensated.

\( c) \) Joint Refinement using Shape-Based Sparse Representation

Given the coordinates of vertebrae, which are the outputs of DI2IN and message passing, we present a joint refinement approach using dictionary learning and sparse
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

representation. In details, we first incorporate a shape-based dictionary in the refinement which embeds the holistic structure of the spine. Instead of learning a shape regression model or Hidden Markov Model to fit the spinal shape, the shape-based dictionary is simply built by the coordinates of spines in the training samples. Then the refinement can be formulated as an $\ell_1$ norm optimization problem and solved by sparse coding in a pre-defined subspace. The task of this optimization is to find the best sparse representation of the coordinates with respect to the dictionary. Taking the regularity of the spine shape into account, ambiguous coordinates and false positives are removed. Finally, the coordinates from all directions are jointly refined, which leads to further improvement in performance.

The rest of this chapter is organized as follow. In Section 4.1.2 we introduce the details of the proposed approach for human vertebrae localization and identification, which is composed of three subsections. In Section 4.1.3 we compare the proposed approach to other state-of-the-art works on both 2D X-ray database and 3D CT database. Finally, we end this chapter with a conclusion and discussion.

### 4.1.2 Methods

In this section, we introduce the methodology and details of proposed frame, which is composed of three sections: deep image-to-image network, response enhancement using different neural networks and refinement approach for post-processing. In details, we introduce two response enhancement approaches using message passing network
and recurrent neural network, respectively. In post-processing refinement, we also discuss two kinds of refinement methods. One is based on dictionary learning and another is derived from shape-based neural network.

4.1.2.1 Deep Image-to-Image Neural Network (DI2IN)

In this section, we present the architecture and details of the proposed deep image-to-image network, as shown in Figure 4.2. The basic architecture is designed as a convolutional encoder-decoder network.\[80\] Compared to sliding-window approach, the DI2IN is implemented in a voxel-wise fully convolutional end-to-end learning. It performs the network on input images such as 2D X-ray images or 3D CT volumes directly. Basically, the DI2IN takes the image or volume as input and simply generates the multichannel probability maps simultaneously as output.

In order to train such a neural network, the ground truth probability maps are defined and generated by Gaussian distribution as following equation:

$$I_{gt} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}. \quad (4.1)$$

The ground truth probability maps are generated by Gaussian distribution $I_{gt} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}$, where $x \in \mathbb{R}^3$ and $\mu$ denote the voxel coordinates and ground truth location, respectively. $\sigma$ is predefined to control the scale of the Gaussian distribution.
Each channel’s prediction $I_{\text{prediction}}$ is associated with the centroid location and type of vertebra. The loss function is defined as $|I_{\text{prediction}} - I_{\text{gt}}|^2$ for each voxel. Therefore, the whole learning problem is formulated as a multichannel voxel-wise regression. Instead of using classification formulation for detection, regression is tremendously helpful for determining predicted coordinates and it relieves the issue of imbalanced training samples, which is very common in semantic segmentation.

The encoder is composed of convolution, max-pooling and rectified linear unit (ReLU) layers while the decoder is composed of convolution, ReLU and upsampling layers. Max-pooling layers are of great importance to increase receptive field and extract large contextual information. It helps to improve the sensory area of neurons and alleviate the pressure of GPU memory. Upsampling layers are designed with the bi-linear interpolation to enlarge and densify the activation, which also further enables the end-to-end voxel-wise training without losing resolution details. The convolutional filter size is $1 \times 1 \times 1$ in the output layer and $3 \times 3 \times 3$ in other layers. The max-pooling filter size is $2 \times 2 \times 2$ for down-sampling by half in each dimension. In upsampling layers, the input features are upsampled by a factor of 2 in each dimension. The stride is set as 1 in order to maintain the same size in each channel.

Additionally, we incorporate the feature concatenation and deep supervision in the DI2IN. In feature concatenation, a bridge is built directly from the encoder layer to the decoder layer, which passes forward the feature information from the encoder and then concatenates it with the decoder layer\[81\] As a result, the DI2IN benefits from
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

Figure 4.2: Proposed deep image-to-image network (DI2IN). The front part is a convolutional encoder-decoder network with feature concatenation, and the backend is multi-level deep supervision network. Numbers next to convolutional layers are channel numbers. Extra 26-channel convolution layers are implicitly used in deep supervision.

both local and global contextual information. Deep supervision has been adopted in \cite{82,83} to achieve good boundary detection and organ segmentation. In the DI2IN, we incorporated a more complex deep supervision approach to further improve the performance. Several branches are diverged from the middle layers of the decoder network. With the appropriate upsampling and convolutional operations, the output size of all branches matches the size of multi-channel ground truth. In order to take advantage of deep supervision, the total loss function \( \text{loss}_{\text{total}} \) of DI2IN is defined as the combination of \( \text{loss}_i \) for all output branches as follows:
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

\[ \text{loss}_{\text{total}} = \sum_i \text{loss}_i + \text{loss}_{\text{final}}, \]

(4.2)

where \( i \) indicates the indexes of branches except the \textit{final} output.

Therefore, the DI2IN is able to perform on various input images such as 3D CT volumes or 2D X-ray images and generate the feature maps such as heat maps or probability maps to indicate localization and identification information. Training such a network can be formulated as a multi-channel voxel-wise regression task. In our case, the DI2IN takes 3D CT volumes or X-ray scans as inputs and directly outputs multi-channel probability maps, with each map associated with one vertebra landmark. Specifically, each probability map illustrates the location of corresponding vertebra, where high-response region indicates the location of the vertebra and vice versa as shown in Figure 4.3. Taking advantage of fully convolution, this framework is more efficient at computing the probability maps as well as the centroid locations than the patch-wise classification or regression approaches\(^{78,79}\).

4.1.2.2 Message Passing and Structure Learning

Given the image \( I \), the DI2IN generates one probability map \( P(v_i|I) \) for the center of each individual vertebra \( i \) with high confidence. Intuitively, the vertebrae will be located at the peak positions \( v_i \) of probability maps. However, these probability
maps are not perfect yet: some probability maps don’t have response or have very low response at the ground truth locations because of similar image appearances of several vertebrae (e.g. $T1 \sim T12$), low-resolution scans or FOVs. In order to address the problem of missing response, we propose two message passing schemes using deep neural network to effectively enhance the probability maps by utilizing the prior knowledge of the spine structure.

4.1.2.2.1 Response Enhancement using Convolutional Neural Network

The concept of message passing was first introduced in the context of probabilistic graphical models. It is used in the sum-product or max-product algorithms for exact inference of the marginal probabilities of nodes or the distribution mode in a tree-structured graph. Messages are passed iteratively between neighboring nodes to exchange information and optimize the overall probability distribution. Similarly, we
introduce an Markov Random Field-like model, a chain-structure graph shown in Figure 4.4, to express the spatial relationship among vertebrae, where each node in the graph represents one vertebra center \( v_i \). Then we propose the following formulation to update the \( P(v_i|I) \) during the iteration \( t \) of the message passing.

\[
P_{t+1}(v_i|I) = \frac{1}{Z} \left[ \alpha \cdot \sum_{j \in S_i} \frac{m_{j \rightarrow i}}{|S_i|} + P_t(v_i|I) \right] \\
= \frac{1}{Z} \left[ \alpha \cdot \sum_{j \in S_i} P_t(v_j|I) \cdot k(v_i|v_j) \right] + P_t(v_i|I).
\]

In details, \( S_i \) is the neighbourhood set of vertebra \( i \) in the graph, \( Z \) is a normalization constant, and \( \alpha \in (0, 1) \) is a discounted factor. The messages \( m_{j \rightarrow i} \), defined as \( P_t(v_j|I) \cdot k(v_i|v_j) \), are passed along the chain shown in Figure 4.4.* indicates the
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

convolution operation. \( k(v_i|v_j) \) is a single convolution kernel which is learned from the ground truth distribution of vertebra pair \( i, j \). Multi-dimensional convolution itself is capable to shift the mass of the probability map \( P_t(v_i|I) \) to its neighbourhood with a fixed orientation (kernel). If \( P_t(v_i|I) \) is confident at its correct location, then the message \( m_{j \rightarrow i} \) would be a strong prior for \( P_{t+1}(v_j|I) \) at the correct location of the vertebra \( j \). After several iterations of message passing, the vertebra with missing response can be compensated with the aggregated messages from its neighbouring vertebrae. The underlying assumption of this scheme is that majority of the vertebra probability maps are confident and well distributed around the true locations, which is guaranteed by the powerful DI2IN in our method. The advantage of the proposed scheme is that it can be concatenated into the DI2IN for further end-to-end training (fine-tuning) when the iteration number is fixed. The location of vertebra \( i \) centroid can simply be determined by the location of the maximum value in the corresponding probability map \( P_t(v_i|I) \).

Several recent works have deployed the message-passing concept for different landmark detection tasks. Chu et al.\(^{85}\) proposed the passing scheme between the feature maps instead of landmark probability maps. Yang et al.\(^{86}\) introduced a fully connected graphical model for message passing between probability maps. The hand-crafted features were adopted in the pair-wise terms of the messages. Payer et al.\(^{87}\) also brought up the fully connected graphical model, applying one-time passing with pixel-wise dot-product for noise cancelling. In our proposed method, the passing is
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

directly among the response maps along the chain-structure model. The response maps are gradually enhanced within several passing iterations, since one passing is not enough to make necessary adjustment for probability maps. Compared to the hand-craft features, the single convolutional kernel is eligible to generate messages between neighbours because the designed neighbourhood is compact. In our framework, the missing response is the major issue instead of the noisy output, so the dot-product operation is not applicable and may hurt the output probabilities.

4.1.2.2.2 Response Enhancement using Recurrent Neural Network

Recurrent neural network (RNN) has been widely developed and used in many applications, such as natural language processing, video analysis and speech processing. It is capable to handle arbitrary sequences of input, and performs the same processing on every element of the sequence with memory of the previous computation. In our case, the spatial relation of vertebrae naturally forms a chain structure from top to bottom. The element of the chain is a response (probability) map \( P(v_i|I) \) of a vertebra centroid \( v_i \). The proposed RNN model treats the chain as a sequence and enables vertebra responses of DI2IN to communicate with each other. In order to adjust the 2D or 3D response map \( P(v_i|I) \) of vertebrae \( i \), we apply the convolutional long short-term memory (ConvLSTM) as our RNN model shown in Figure 4.5. Because the \( z \) direction is the most informative dimension in the spinal structure, the \( x, y \) dimensions are set to 1 for all the convolution kernels.
During inference, we pass information forward and backward to regularize the output of DI2IN. The passing process can be conducted \( k \) iterations (\( k = 2 \) in our experiments). All input-to-hidden and hidden-to-hidden operations are convolutions. Therefore, the response distributions can be adjusted with necessary displacement or enhanced by the neighbours’ responses.

Equations 4.5 describes how the LSTM unit is updated at each time step. In details, \( X_1, X_2, \ldots \) and \( X_t \) are input states for vertebrae, cell states are \( C_1, C_2, \ldots \) and \( C_t \), and the hidden states are \( H_1, H_2, \ldots \) and \( H_t \). \( i_t, f_t \) and \( o_t \) indicate the gates of ConvLSTM. We use several sub-networks \( G \) to update \( X_t \) and \( H_t \), which differs from
the original ConvLSTM setting (original work only uses single kernel). Each $G$ is consist of three convolutional layers with $1 \times 1 \times 9$ kernels, and filter numbers are 9, 1 and 1. The sub-networks are more flexible and have a larger receptive field compared to that uses a single kernel. Therefore, it is helpful to capture the spatial relationship of all vertebrae.

\[
\begin{align*}
    i_t &= \sigma(G_{xi}(X_t) + G_{hi}(H_{t-1}) + W_{ci} \odot C_{t-1} + b_i) \\
    f_t &= \sigma(G_{xf}(X_t) + G_{hf}(H_{t-1}) + W_{cf} \odot C_{t-1} + b_f) \\
    C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(G_{xc}(X_t) + G_{hc}(H_{t-1}) + b_c) \\
    o_t &= \sigma(G_{xo}(X_t) + G_{ho}(H_{t-1}) + W_{co} \odot C_t + b_o) \\
    H_t &= o_t \odot \tanh(C_t)
\end{align*}
\]

(4.5)

4.1.2.3 Post Refinement Schemes

Given the probability maps generated by DI2IN and message passing enhancement, it is not guaranteed that there are no outliers or false positives. For example, even though the DI2IN followed by message passing enhancement outputs a quite clear and reasonable probability map, there is still false positive as shown in Figure 4.3. This might arise from reasons such as low resolution scans, image artifacts or lack of global contextual information. In order to overcome this limitation, localization refinement has been introduced in many works. In a HMM with hidden states
is defined for vertebrae location, appearance likelihoods and inter-vertebra shape priors, which can yield a refined localization based on several thousand candidate locations from the forest prediction. In a quadratic polynomial curve is proposed to refine the coordinate in the vertical axis. By optimizing an energy function, the parameters for the shape regression model are learned to refine the coordinates of vertebrae. However, this model assumes the shape of the spine can be represented by a quadratic form. In addition, only coordinates in the vertical axis are refined. In our case, we have designed two post refinement methods to further improve the quality of performance. The first one takes advantage of sparse representation and dictionary learning. Another one is designed in an end-to-end fashion from the perspective of neural network.

4.1.2.3.1 Shape-Based Refinement using Sparse Representation

Inspired by dictionary learning and sparse representation, we design a joint refinement using shape-based dictionary. Given a pre-defined shape-based dictionary, the coordinates are refined jointly in all $x$, $y$ and $z$ axes. The refinement itself can be formulated as an $\ell_1$ norm optimization and solved by sparse coding approach. In details, given the shape-based dictionary $D \in \mathbb{R}^{M \times N}$ and the coordinate prediction $v \in \mathbb{R}^N$, we propose a joint refinement algorithm as shown in to solve the sparse coefficient vector $a \in \mathbb{R}^M$. Then the refined coordinate vector is defined as $\hat{v} = Da$. Specifically, the shape-based dictionary $D$ is simply built by the coordinates of ver-
tebrae in training samples. For example, the notation $D_z$ indicates the shape-based
dictionary associated with vertical axis or $z$ direction. $d_{z,i} \in \mathbb{R}^M$, which is a column
of $D_z$, is defined as $[z_{i,1} \ z_{i,2} \ ... \ z_{i,26}]^T$. For instance, $z_{i,1}$ denotes the vertical ground
truth coordinate of $i$th sample corresponding to vertebrae $C_1$. The $D_x$ and $D_y$ denote
the dictionaries associated with $x$ and $y$ directions, respectively. They are both build
in the same manner as $D_z$. Similarly, $v_z$, defined as $[v_{z,1} \ v_{z,2} \ ... \ v_{z,26}]$, is the vertical
coordinate of prediction. $v_x$ and $v_y$ are defined in the same manner.

In order to address the challenges such as outliers and limited FOV in spinal scans,
we define the original space $S_0$ and a subspace $S_1$ in proposed refinement approach.
The original space denotes a set which contains all indexes of 26 vertebrae. In our case,
$S_0$ contains the indexes from 1 to 26 which are corresponding to $C_1$ to $S_2$. Compared
to the original space $S_0$, the subspace $S_1$ denotes a subset which only contains the
partial indexes of $S_1$. Based on the subspace $S_1$, we define sub-dictionary $D_{S_1}$ and
sub-coordinate vector $v_{S_1}$. Intuitively, $D_{z,S_1}$ indicates the sub-dictionary associated
with axis $z$, which is also simply a sub-matrix of $D_{z,S_0}$. Basically, the optimization
problem is solved based on the subspace $S_1$ instead of the original space $S_0$.

The details are demonstrated in Algorithm [10]. Taking the shape regularity into
account, we firstly find the maximum descending subsequence in the coordinate pre-
diction $v_z$ via dynamic programming. The reason we choose the vertical axis $z$ to
determine the maximum subsequence instead of $v_x$ and $v_y$ is the vertical axis of the
human spine naturally demonstrates the most robust geometric shape compared to
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

$x$ and $y$ axes. Based on the subspace $\mathcal{S}_1$ generated by Step 1, we further remove the indexes of neighboring vertebrae of which distance is too large or too small. Given the subspace $\mathcal{S}_1$, we define the sub-dictionary and sub-coordinate vector for each axis, respectively. Then, the $\ell_1$ norm problem in Step 5 is optimized for $x$, $y$ and $z$ individually based on the same subspace $\mathcal{S}_1$. Finally, all coordinates are refined based on the original space $\mathcal{S}_0$ (i.e. $\mathbf{D}_{z,\mathcal{S}_0}$ and $\mathbf{v}_{z,\mathcal{S}_0}$). Intuitively, we remove the ambiguous outliers from the preliminary prediction and then jointly refine the coordinates without these outliers. Based on the subspace, we optimize the refinement problem to find the best sparse combination in the shape-based sub-dictionary. By taking advantage of the original shape-based dictionary, all coordinates are refined jointly as shown in Figure 4.6.

![Figure 4.6: Maximum errors of vertebra localization before and after the shape-based refinement](image)

**Figure 4.6:** Maximum errors of vertebra localization before and after the shape-based refinement.
Algorithm 10 Joint Refinement using Shape-Based Dictionary

Require: The dictionary \( D_x, S_0 \), \( D_y, S_0 \) and \( D_z, S_0 \) ∈ \( \mathbb{R}^{M \times N} \), the predicted coordinates vector \( v_x, v_y \) and \( v_z \), the error threshold \( \epsilon_1 \) and \( \epsilon_2 \), and the coefficient \( \lambda \). \( M \) and \( N \) indicate the number of landmarks and size of items in dictionary, respectively.

1: Given the predicted coordinates \( v_z \) from the DI2IN and message passing, the maximum descending subsequence is found via dynamic programming.
2: Add the indexes associated with the maximum descending subsequence into the set \( S_1 \).
3: Remove the pair of neighbouring indexes if \( |v_{z,i} - v_{z,j}| \leq \epsilon_1 \) or \( |v_{z,i} - v_{z,j}| \geq \epsilon_2 \), where \( i, j \in S_1 \) and \( |i - j| = 1 \).
4: Based on the subspace \( S_1 \), define the sub-dictionary \( D_x, S_1 \), \( D_y, S_1 \), and \( D_z, S_1 \) and the sub-coordinate predictions \( v_x, S_1 \), \( v_y, S_1 \) and \( v_z, S_1 \).
5: Solve the optimization problem below by \( \ell_1 \) norm recovery for the vertical axis \( z \):

\[
\min_{a_z} \frac{1}{2} ||v_{z, S_1} - D_{z, S_1} a_z||_2^2 + \lambda ||a_z||_1 .
\]
6: Solve the same optimization problem in Step 3 for \( v_x, S_1 \) and \( v_y, S_1 \), respectively.
7: Return the refined coordinate vectors \( \hat{v}_x = D_{x, S_0} a_x \), \( \hat{v}_y = D_{y, S_0} a_y \) and \( \hat{v}_z = D_{z, S_0} a_z \).

4.1.2.3.2 Shape-Based Refinement using Neural Network

As shown in Figure 4.7, the ConvLSTM generates clear probability maps, where the high response in the map indicates the potential location of the landmark (centroid of the vertebrae). However, sometimes due to image artifacts and low image resolution, it is difficult to guarantee there is no false positive. Therefore, we present a shape basis network to help refine the coordinates inspired by.

Given a pre-defined shape-based dictionary \( D \in \mathbb{R}^{N \times M} \) and coordinate vector \( y \in \mathbb{R}^N \) generated by ConvLSTM, the proposed shape basis network takes \( y \) as input and outputs the coefficient vector \( x \in \mathbb{R}^M \) associated with dictionary \( D \). Therefore, the refined coordinate vector \( \hat{y} \) is defined as \( Dx \). In practice, the shape-based dic-
Figure 4.7: Probability map examples from DI2IN (left in each case) and ConvLSTM (right in each case). The prediction in “Good Cases” is close to ground truth location. In “Bad Cases”, some false positives exist remotely besides the response at the ground truth location.

The dictionary $D$ is simply learned from the training samples. For example, the dictionary $D_z$ associated with the vertical axis is constructed by the $z$ coordinate of vertebrae centroids in the training sample. $N$ and $M$ indicate the number of vertebrae and number of atoms in dictionary, respectively.

The proposed shape basis network consists of several fully connected layers. Instead of regressing the refined coordinates directly, the network is trained to regress the coefficients $x$ associated with the shape-based dictionary $D$. The learning problem is formulated as a regression model and the loss function is defined as:
\begin{equation}
loss_{shape} = \sum_i ||Dx_i - y_i||_2^2 + \lambda||x_i||_1
\end{equation}

$x_i$ and $y_i$ denote the coefficient vector and ground truth coordinate vector of $i$th training sample. $\lambda$ is the $\ell_1$ norm coefficient to leverage the sparsity and residual. Intuitively, the shape-based neural network is learned to find out the best linear combination in the dictionary to refine the coordinates. In our case, we focus on the refinement of vertical coordinates $z$. The input of shape basis network is obtained directly from the output of ConvLSTM using a non-trainable fully connected layer. The layer has uniform weights and no bias term, and it generates the correct coordinates when the response is clear. Such setting enables the end-to-end scheme for fast inference instead of solving the loss function directly in a conventional way. Figure 4.8 illustrates the examples of shape-based neural network. Taking the vertebral shape regularity into account, the shape-based neural network jointly refines the coordinates. The maximum errors of localization have been significantly reduced compared to the output from the DI2IN and convLSTM.

### 4.1.3 Experiments

In this section, we evaluate the performance of the proposed two approaches on two different, large bases. The first one has been introduced in which contains 302 spine-
focused 3D CT scans with various pathologies. These unusual appearance includes abnormal curvature, fractures and bright visual artifacts such as surgical implants in post-operative cases. In addition, the FOV of each 3D CT scans varies greatly in terms of vertical cropping. The whole spine is visible only in a few examples. Generally, the 3D CT scans contains 5-15 vertebrae in each case. In particular, in order to boost the performance of our approaches, we further introduce extra 1000+ 3D CT scans in our experiments for training the models.

In order to extend our frameworks to 2D X-ray Scans, we introduce another large scale database in this session. This second database consists of 1000+ 2D X-ray scans described in. The ground truth is marked on the center of gravity of each vertebra. The location and label of each ground truth is manually annotated by
4.1.3.1 Deep Image-to-Image Network with Message Passing and Sparse Regularization

Table 4.1 and 4.2 summarizes the quantitative results in terms of localization mean error and identification rate defined by\cite{76} on Set 1 and Set 2. We compare our approach to other results reported in\cite{77-79} on the 3D CT scans. In previous works,\cite{76,78,79} there are two different settings on the 3D CT scans: the first one uses 112 of the images as training and another 112 images as testing; the second one takes all images (242) in setting one with extra 18 images as training data and an additional 60 images as testing data. For a fair comparison, we follow the same database settings in our experiments. They are denoted as “Set 1” and “Set 2” respectively. We also follow the evaluation metrics described in\cite{77} in terms of the Euclidean distance error (in mm) and identification rates (Id.Rates) defined in\cite{76}. In details, “DI2IN”, “MP” and “S” denote the deep image-to-image network, convolutional message passing and shape-based refinement, respectively. “1000” indicates this model is trained with additional 1000 scans and evaluated on the same testing samples. In order to show the improvement of the performance, we list the results after each step for comparison.

Overall, our approach outperforms the state-of-art approaches\cite{77,78} by 13 % and 6 % on the same evaluation settings respectively. For Set 1, the DI2IN itself improves the Id. Rates by a margin of 6 % compared to the approach in\cite{77} Message passing and
shape-based refinement further increase the Id. Rates to 77 % and 80 %, respectively. In addition, we have demonstrated that extra 1000 samples boost the performance to 83 %. Similarly, the proposed approach also demonstrates better performance in Set 2 compared to.\textsuperscript{77–79} Our approach has achieved a Id. Rates of 85 % and a localization mean error of 8.6 mm, which is better than the state-of-art work.\textsuperscript{78} Taking advantage of extra 1000 samples, the Id. Rates has achieved 90 %. Furthermore, other metrics such as stand deviation (Std), median (Med) and maximum (Max) also intuitively demonstrate the efficiency of our approach. For example, the maximum errors in both sets are significantly reduced to 42.3 mm and 37.9 mm. Figure 4.6 intuitively illustrates the refinement of proposed shape-based refinement in vertical direction. As shown in Figure 4.6, the shape-based refinement takes the shape regularity of spine into account and remove the false positive coordinates. Specifically, the max error is significantly reduced.

Additionally, in order to demonstrate the robustness of our approach, we extend our experiments into a 2D X-ray database for training and evaluation. For 2D X-ray scans, the database\textsuperscript{91–93} is randomly divided into two parts: 1170 scans as training samples and 50 scans for testing samples. It is the first time by our knowledge to evaluate such an approach on 2D X-ray scan for human vertebrae localization and identification task. Table 4.3 and 4.4 demonstrate the performance of each step of our approach in terms localization error and identification rates on databases with 0.7 mm and 0.35 mm resolution. Because most of vertebrae in X-ray scans belong
Table 4.1: Comparison of localization errors in mm and identification rates among different methods for Set 1.

<table>
<thead>
<tr>
<th>Region</th>
<th>Method</th>
<th>Set 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>All</td>
<td>Glocker et al [74]</td>
<td>12.4</td>
</tr>
<tr>
<td></td>
<td>Suzani et al [74]</td>
<td>18.2</td>
</tr>
<tr>
<td></td>
<td>Chen et al [75]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DI2IN</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP</td>
<td>11.7</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>9.1</td>
</tr>
<tr>
<td></td>
<td>DI2IN+1000</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+1000</td>
<td>9.4</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td><strong>8.5</strong></td>
</tr>
<tr>
<td>Cervical</td>
<td>Glocker et al [74]</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>Suzani et al [74]</td>
<td>17.1</td>
</tr>
<tr>
<td></td>
<td>Chen et al [75]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>6.6</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td><strong>5.8</strong></td>
</tr>
<tr>
<td>Thoracic</td>
<td>Glocker et al [74]</td>
<td>13.8</td>
</tr>
<tr>
<td></td>
<td>Suzani et al [74]</td>
<td>17.2</td>
</tr>
<tr>
<td></td>
<td>Chen et al [75]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>9.9</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td><strong>9.5</strong></td>
</tr>
<tr>
<td>Lumbar</td>
<td>Glocker et al [74]</td>
<td>14.3</td>
</tr>
<tr>
<td></td>
<td>Suzani et al [74]</td>
<td>20.3</td>
</tr>
<tr>
<td></td>
<td>Chen et al [75]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>10.9</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td><strong>9.9</strong></td>
</tr>
</tbody>
</table>
### Table 4.2: Comparison of localization errors in mm and identification rates among different methods for Set 2.

<table>
<thead>
<tr>
<th>Region</th>
<th>Method</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>All</td>
<td>Glocker et al[27]</td>
<td>13.2</td>
</tr>
<tr>
<td></td>
<td>Suzani et al[27]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Chen et al[27]</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>DI2IN</td>
<td>13.6</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP</td>
<td>10.2</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>8.6</td>
</tr>
<tr>
<td></td>
<td>DI2IN+1000</td>
<td>7.1</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+1000</td>
<td>6.9</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td>6.4</td>
</tr>
<tr>
<td>Cervical</td>
<td>Glocker et al[27]</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>Suzani et al[27]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Chen et al[27]</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td>5.2</td>
</tr>
<tr>
<td>Thoracic</td>
<td>Glocker et al[27]</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>Suzani et al[27]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Chen et al[27]</td>
<td>11.4</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>9.2</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td>6.7</td>
</tr>
<tr>
<td>Lumbar</td>
<td>Glocker et al[27]</td>
<td>13.0</td>
</tr>
<tr>
<td></td>
<td>Suzani et al[27]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Chen et al[27]</td>
<td>8.4</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>11.0</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S+1000</td>
<td>7.1</td>
</tr>
</tbody>
</table>
to the thoracic region ($T_1 - T_{12}$), we only present the overall results instead of showing results in individual region. In details, the DI2IN itself achieves a localization error of 8.4 mm and 7.8 mm and an identification rate of 80% and 82% on 0.7 mm and 0.35 mm resolution, respectively. We also introduce message passing scheme and shape-based refinement to evaluate the performance. The quality of performance is further improved compared to the DI2IN itself. The identification rate is also greatly improved after the introduction of message passing and shape-based refinement. Overall, the identification rate has been significantly increased by the message passing and refinement and finally reached 91% on higher resolution scans. Our experiment demonstrates the proposed approach is able to achieve better performance on higher resolution database. Given more memory allocation and model capacity, our approach could further improve the quality of landmark detection.

Table 4.3: Comparison of localization errors in $mm$ and identification rates among different methods for 0.7 mm X-ray Set.

<table>
<thead>
<tr>
<th>Region</th>
<th>Method</th>
<th>0.7 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>All</td>
<td>DI2IN</td>
<td>8.4</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>7.1</td>
</tr>
</tbody>
</table>
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

Table 4.4: Comparison of localization errors in \( \text{mm} \) and identification rates among different methods for 0.35 mmX-ray Set.

<table>
<thead>
<tr>
<th>Region</th>
<th>Method</th>
<th>0.35 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>All</td>
<td>DI2IN</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>DI2IN+MP+S</td>
<td>6.4</td>
</tr>
</tbody>
</table>

4.1.3.2 Deep Image-to-Image Recurrent Neural Network with Shape Basis Learning

In the section, we also evaluate the proposed method on database introduced in which consists of 302 CT scans with various types of lesions. The dataset has some cases with unusual appearance, such as abnormal spinal structure and bright visual artifacts due to metal implants by post-operative procedures. Furthermore, the FOV of each CT image varies greatly in terms of vertical cropping, image noise and physical resolution. Most cases contain only part of the entire spine. The overall spinal structure can be seen only in a few examples. Large changes in lesions and limited FOV increase the complexity of the appearance of the vertebrae. It is difficult to accurately localize and identify the spinal column. The ground truth is marked on the center of gravity of each vertebra and annotated by the clinical experts. In previous works, two different settings have been conducted on this database: the first one uses 112 images as training and other 112 images as testing. The second
one takes all data in first setting plus extra 18 images as the training data (overall 242 training images), and 60 unseen images are used as the testing data. For fair comparison, we follow the same configuration, which are referred as Set 1 and Set 2, respectively, in the experiments. Table 4.5 compares our result with the numerical results reported in previous methods \cite{77,78,79} in terms of the Euclidean distance error (mm) and identification rate (Id.Rates) defined by. The average mean errors of these two databases are 10.6 mm and 8.7 mm, respectively, and the identification rates are 78% and 85%, respectively. Overall, the proposed method is superior to the state-of-the-art methods on the same database with respect to mean error and identification rate. In addition, the extra 1000+ 3D CT volumes help our model significantly boost the performance. Finally, the performance has achieved 83 % and 89 % respectively.

Table 4.5: Comparison of localization errors in \textit{mm} and identification rates among different methods. Our method is trained and tested using default data setting in ”Set 1” and ”Set 2”, while ”+1000” indicates training with additional 1000 labeled spine data and evaluated on the same testing data.
All experiments are conducted on a cluster equipped with an Intel 3.5 GHz CPU as well as a 12 GB Nvidia Titan X GPU. In order to alleviate the pressure of memory, experiments on 3D CT scans and X-rays scans are conducted on a resolution of 4 mm, 0.7 mm and 0.35 mm, respectively. The evaluation time of our approach is around three seconds per case on average using GPU. In order to extract valid information from noisy probability maps, the response maps of DI2IN are compared to a heuristic threshold in an element-wise. Only channels with strong response are considered as valid outputs. Then vertebra centroids associated with these channels are identified to be present in the image. The other vertebrae associated with other probability maps are identified as non-presented in the same image. Therefore, we are able to localize and identify all vertebrae simultaneously in an efficient way.

4.1.4 Discussion and Conclusion

Although our approach has achieved high identification rates on various pathological cases in both 3D CT scans and 2D X-ray scans, there are still some challenging cases. As shown in Figure 4.9 and Figure 4.10, the proposed approach occasionally fails to refine the coordinates which are jointly offset. This limitation might arise from special pathological cases, limited FOV and low resolution input images. In our approach, the underlying assumption is that majority of the vertebra probability maps are confident and well distributed around the true locations, which is guaranteed by the powerful DI2IN. In order to address this limitation, more sophisticated network will be further
studied in the future. From Figure 4.11, we can see that vertebrae in thoracic region are comparatively hard to locate because those vertebrae share the similar imaging appearance.

In conclusion, we proposed an effective and fast automatic method for human vertebrae localization and identification task using deep learning techniques. Basically, our framework is composed of three stages: a deep image-to-image network (DI2IN), message passing scheme and shape-based refinement. The DI2IN performs directly on a input image such as a 3D CT volume or 2D X-ray scan in a fully convolutional manner and generates probability maps associated with landmarks. Taking advantage of the human vertebral structure, message passing schemes, using convolutional neural networks or recurrent neural networks, have been deployed to further iterat-
Figure 4.10: Maximum errors of vertebra localization in challenging cases before and after the message passing and shape-based network refinement.

Figure 4.11: Localization errors (in mm) for the vertebral centroids. The statistical results are for the testing datasets of set 1 and set 2 using the proposed methods with extra 1000 training volumes. "C" is for cervical vertebrae, "T" is for thoracic vertebrae, "L" is for thoracic vertebrae, and "S" is for sacral vertebrae.
CHAPTER 4. DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

tively refine the probability maps and remove false positives. Finally, shape-based post processing approaches refine the vertebral coordinates from the perspective of both sparse representation and neural network.

In order to demonstrate the quality of the performance, we evaluated our approach comprehensively on a challenging pathological 3D CT database and a 2D X-ray database. Overall, our performance outperforms the state-of-the-art works in terms of localization mean error and identification rate. The proposed approach is also time-efficient in inference compared to other sliding window approaches. The running time is around 3 seconds in average using GPU. Furthermore, the performance on both 3D CT and 2D X-ray databases demonstrates the robustness of such an approach. To further boost the performance, we also introduce additional 1000+ 3D CT volumes and 2D X-ray scans in both training processes. To the best of our knowledge, this is the first time more than 1000 3D CT volumes with expert annotation are adopted in research for anatomic detection tasks.
Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we have demonstrated and evaluated three major frameworks from the perspectives of compressed sensing, sparse representation and deep learning with applications on neural recording, video compression and human vertebrae detection. Specifically, using compressed sensing and sparse representation, we developed several compressed sensing (CS) based frameworks for multi-channel neural recording and spike sorting in both supervised and unsupervised manner. Furthermore, we extend such a CS-based neural recording system to imaging system. In this imaging system, we took advantage of spatiotemporal structure in both spatial and temporal domains and achieved better and comparable performance compared to other related approaches. Finally, after exploiting hidden structure in data using CS and sparse
representation, we move forward to the field of deep neural networks to exploit more hidden and hierarchy structure and representation in medical images such as computed tomography (CT) and X-ray scans. In this framework, we basically focused on the design of architecture of deep neural networks with message passing schemes that takes the structure and regularity of human spine into account. The future work will be mainly focusing on the exploration of the potential combination of sparse representation and deep neural networks. The details will be discussed in Section 5.2.

In Chapter 2, we have exploited various sparse representation structures such as joint-group structure and developed several supervised and unsupervised dictionary learning algorithms for efficient representation in neural recordings. First, we proposed a supervised multi-modal structure dictionary learning for multi-channel neural recordings. Taking advantage of group structure and joint sparsity, we further promoted both reconstruction and classification performances of multi-channel recordings. Then, in order to address the challenge in real-time neural recordings and spike sorting, we extended the previous work into an unsupervised fashion. We developed an unsupervised dictionary learning algorithm with discriminative structures to enable online spike sorting for in vivo experiments. Finally, we integrated the previous works to present an unsupervised multi-mode CS approach for neural recordings systems. We incorporated the joint-group sparsity in the dictionary learning compared to individual group structure and joint sparsity. Furthermore, the spectral clustering and template matching are also adopted to enable spike sort-
CHAPTER 5. CONCLUSION AND FUTURE WORK

ing in real-time experiments in an unsupervised manner. All frameworks have been comprehensively evaluated on both synthetic and real databases. The experiments results have demonstrated that our approaches significantly improved both the reconstruction quality and the spike sorting accuracy by a margin at a high compression ratio. In addition, we also presented the hardware live demonstration of proposed frameworks to show the power-efficient implementation in CMOS.

We presented a novel CS-based spatiotemporal pixel-wise exposure (PCE) control for imaging systems in Chapter 3. This framework can increase video pixel resolution and frame rate simultaneously which reduces data readout speed. Given the same power budget, such a design is significantly energy-efficient from the perspective of hardware implementation using over-complete dictionary and sparse representation. We also exploited the pixel-wise exposure control feature in this framework. Conventionally, the duration of pixel-wise exposure in PCE control is fixed at an amount of time. Compared to the traditional scheme, we studied the dynamic block-wise exposure control using optical flow to further improve the reconstruction quality of compressed video with motions and satisfy the demand of high mobility sensors. In addition, we also evaluated the novel compression technique behind the spatiotemporal CS scheme. Unlike the conventional video compression technique, the spatiotemporal CS requires no motion estimation, compensation and discrete cosine transformation (DCT) to reduce bits. The pixels in videos are simply and randomly encoded into a coded image for reconstruction without any handcrafted
CHAPTER 5. CONCLUSION AND FUTURE WORK

feature extraction. We experimentally showed that the proposed spatiotemporal CS approach could achieve high compression rate and robustness reconstruction in noisy database. We also presented its corresponding CMOS implementation and demonstrated real-time compression as well as reconstruction\cite{20}\cite{21}.

After the study of sparse representation in CS framework, we started to exploit the hidden and hierarchy representation for computer vision tasks from the perspective of deep learning in Chapter 4. In this work, we proposed an automatic and accurate framework for human vertebrae localization and identification task using deep learning techniques\cite{22}\cite{27}. In order to overcome the limitations raised by pathological cases and limited field of view (FOV), the framework is essentially composed of three major stages: a deep image-to-image network (DI2IN), a message passing neural network and a post-refinement scheme. Specifically, the DI2IN takes advantage of the fashion of deep learning techniques such as fully convolutional neural networks, feature concatenation and deep supervision. Compared to sliding window approaches, it is able directly performed on input images and generate the multi-channel probability maps that explicitly indicate the location and label information of corresponding vertebrae. Then, we also incorporated the message passing schemes to enable structure learning and reduce false positives. Two message passing schemes have been implemented, which are based on convolutional neural networks and recurrent neural networks individually. Intuitively, the message passing schemes iteratively refine the probability map using its neighbouring information based on a graphic model. Finally, the loca-
tions are further improved by two different post-refinement schemes that are based on shape-based sparse representation and neural networks separately. We also evaluated proposed approach comprehensively on both large-scale 3D CT database\textsuperscript{77} and 2D X-ray database\textsuperscript{91–93} to demonstrate its performance. The experiments have demonstrated its better performance in terms of localization mean error and identification rate compared to other state-of-the-art works\textsuperscript{40,75–79}. Additionally, we experimentally showed that the proposed approach could be greatly boosted by large-scale database and outperformed previous works by a large margin (> 5 mm and >5%) in both localization mean error and identification rates.

5.2 Future Work

In this thesis, we have described and evaluated several frameworks from the perspectives of compressed sensing, sparse representation and deep neural networks individually. Compressed sensing and sparse representation are good at finding the intrinsic signal information coded in the data\textsuperscript{10,11}. It leads to better signal compression for bandwidth and storage efficiency as well as more effective signal separation for detection, classification and recognition tasks. For instance, multi-channel neural recording systems benefit from the CS framework and achieve high power efficiency given the same sampling rate\textsuperscript{5,18,19}. In addition, sparse representation classifier (SRC), which has demonstrated its robustness in face recognition\textsuperscript{3,94,95} also greatly improves the
accuracy of spiking sorting and enables *in vivo* application in such a CS neural recording framework. We also showed spatiotemporal CS successfully increased the frame rate and reduced the readout speed in image sensors \cite{7,24,25} using sparse representation in an over-complete dictionary \cite{18} On the other hand, recent advances in deep learning have demonstrated its exceptional performances on various recognition purposes such as object classification \cite{13,15,16}, scene segmentation \cite{80}, landmark detection and even strategy decision. In our work, we have developed a deep image-to-image networks with message passing scheme and sparse regularization in an end-to-end manner to achieve the state-of-the-art performance on public challenging database for human vertebrae localization and labeling task \cite{23,27}. Furthermore, we have also experimentally showed deep neural networks thrived on large-scale databases.

Therefore, inspired by the theory of sparse representation and advances of deep neural networks, we would like to investigate the combination of conventional sparse representation and deep multi-layer architecture together expand their learning capacity in future work. Conventionally, sparse representation has shown promising results on many computer vision tasks \cite{3}. Given even a few amount of training samples, it can still demonstrate the robustness against corrupted and noisy data. However, the sparse representation suffers from the large-scale databases if the intrinsic variability is broad and anomalies are common. Compared to the sparse representation, deep neural networks are boosted by the large-scale data and augment the learning capacity of models. In \cite{96} Sun et al. presented a novel deep sparse coding network to
CHAPTER 5. CONCLUSION AND FUTURE WORK

combine the best of sparse representation and deep neural networks. The experimental results have showed the combination of both strategies achieved comparable performance with deep neural networks and adopted significantly fewer parameters (0.69 million) and layers (15). From the perspective of hardware implementation, the combination of these two techniques might provide an efficient implementation solution for deep architectures. However, endeavors to bridge the sparse representation and deep neural network architecture have not been well developed over their individual counterparts. The related topics remain open and the research is still in progress.

Besides, we will also be working on the deep neural networks and explore more novel and efficient architectures for solving machine learning and pattern recognition tasks in applications such as medical imaging analysis and self-driving. In our work, the limitations are raised by challenging cases as shown in Figure 4.10 and memory pressure in the training scheme. In order to address the challenging cases and further improve the performance, novel deep neural networks such as residual networks or densely connected convolutional networks will be investigated in the future. On the other hand, another bottleneck in the development of deep neural networks is how to alleviate the memory pressure during the training as well as inference. For example, to enable training in previous work we had to down-sample the 3D CT volumes at the resolution of 4 mm instead of raw resolution, which might contribute to the degradation of the performance. Therefore, we will also exploit
CHAPTER 5. CONCLUSION AND FUTURE WORK

more light-weight architectures to reduce computational memory and enable efficient training and inference on higher resolution databases.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


[37] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vanderheyndt, “Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor
nodes,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.


BIBLIOGRAPHY


[61] S. Mehta and R. Etienne-Cummings, “Normal optical flow measurement on a
BIBLIOGRAPHY


BIBLIOGRAPHY


Vita

Tao Xiong received B.S. degree in Electrical Engineering and Computer Science as well as B.S. degree in Economics as minor from Peking University in 2012, and then directly enrolled in the Electrical and Computer Engineering Ph.D. program at Johns Hopkins University. He received M.S.E. degree in Electrical and Computer Engineering and M.S.E. degree in Applied Mathematics and Statistics from Johns Hopkins University in 2014 and 2016, respectively. From 2009 to 2012, he was a research assistant in National Key Laboratory of Science and Technology on Micro/Nano Fabrication, Beijing, China. Currently, he is pursuing the Ph.D. degree in Electrical and Computer Engineering at Johns Hopkins University. Since 2013, he has been working in Digital Signal Processing Lab and Computational and Sensory Motor Lab in Johns Hopkins University where his research is focusing on solving signal processing and computer vision tasks using
VITA

deep learning and compressed sensing techniques. He also worked as a deep learning research intern at Siemens Corporate Research in Princeton, New Jersey, USA from March to August in 2016 and at LG Silicon Valley Lab in Santa Clara, California, USA from June to September in 2017. He has published more than 17 papers on Journals such as Applied Physics Letters, Optics Express, IEEE Transactions on Biomedical Circuits and Systems, and IEEE Transactions on Circuits and Systems I, and conferences such as International Conference on Information Processing in Medical Imaging, International Conference on Medical Image Computing and Computer Assisted Intervention, AAAI Conference on Artificial Intelligence, IEEE International Symposium on Circuits and Systems, IEEE Biomedical Circuits and Systems Conference and IEEE International Midwest Symposium on Circuits and Systems.