Title
From Image to Video, Depth Data Reconstruction from a Subset of Samples: Representations, Algorithms, and Sampling Strategies.

Author
Liu, Lee-Kang

Publication Date
2015

UNIVERSITY OF CALIFORNIA, SAN DIEGO

From Image to Video, Depth Data Reconstruction from a Subset of Samples: Representations, Algorithms, and Sampling Strategies.

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Lee-Kang Liu

Committee in charge:

Professor Truong Q. Nguyen, Chair
Professor Pamela Cosman
Professor William Hodgkiss
Professor David Kriegman
Professor Bhaskar Rao

2015

The dissertation of Lee-Kang Liu is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2015

DEDICATION

I dedicate my dissertation work to my family and my friends. I have a special grateful feeling to my parents, Chiao-Tang Liu and Yi-Chih Chung, who support me as I make it my mind to broaden my knowledge and visions across The Pacific, encourage me whenever I face challenges and encounter difficulties, and remind me that you will always stand by me whatever happens, and to my older brother, Li-Wei Liu, who always stays by my side, and lights up a path for me when I get lost in the dark. You are always my mental mentors of my life.

I also dedicate this dissertation to my friends in the United States, Min-Chih Kuo, Wu-Ting Wu, Tsung-Feng Wu, Chih-Wei Shin, and Chun-Lun Yen, and friends in Taiwan, Tzu-Fan Chen, Chih-Huei Huang, Shang-Ming Tai, and Kuo-Hsieh Hsu. With the accompany of you, my will is heartened at the moments that I am depressed, anxious, satisfied, or fulfilled at any stage on the road map to this dissertation.

EPIGRAPH

As we light a path for others,
we naturally light our own way.
—Mary Anne Radmacher

TABLE OF CONTENTS

Signature Page . . . iii

Dedication . . . iv

Epigraph . . . iv

Table of Contents . . . vi

List of Figures . . . ix

List of Tables . . . xi

Acknowledgements . . . xii

Vita . . . xiii

Abstract of the Dissertation . . . xiv

Chapter 1 Introduction . . . 1
- 1.1 Scope and Contributions . . . 2
- 1.2 Related Work . . . 3
- 1.3 Notations and Problem Formulation . . . 4
  - 1.3.1 Depth and Disparity . . . 4
  - 1.3.2 Sampling Model . . . 5
  - 1.3.3 Representation Model . . . 6
  - 1.3.4 Penalty Functions . . . 7

Chapter 2 Sparse Representation of Depth Data . . . 9
- 2.1 Natural Images vs Depth Data . . . 9
- 2.2 Wavelet vs Contourlet . . . 10
  - 2.2.1 Representations of Wavelet and Contourlet Bases . . . 10
  - 2.2.2 Sparse Representation of Disparity Maps . . . 14
  - 2.2.3 Evaluation Metric . . . 16
  - 2.2.4 Comparison Results . . . 16

Chapter 3 Algorithms for Depth Data Reconstruction . . . 18
- 3.1 Symbols and Problem Formulation . . . 18
- 3.2 Algorithm: Conjugate Sub-Gradient . . . 20
  - 3.2.1 Derivation . . . 20
  - 3.2.2 Experimental Setup . . . 23
  - 3.2.3 Discussions . . . 24
- 3.3 Algorithm: Alternating Direction Method of Multipliers . . . 26
  - 3.3.1 ADMM and Operator Splitting . . . 26

3.3.2 Subproblems 28
3.3.3 Parameters 30
3.3.4 Convergence Comparison 30
3.3.5 Multiscale ADMM 32

Chapter 4 Sparse Sampling for Depth Data 36
4.1 Motivating Example 36
4.2 Oracle Random Sampling Scheme 37
4.2.1 Practical Random Sampling Scheme 39
4.3 Further Improvement by PCA 41
4.4 Comparisons 42
4.5 Experimental Results 44
4.5.1 Synthetic Data 44
4.5.2 Real Data 48

Chapter 5 Depth Reconstruction Algorithm For Spatio-Temporal Depth Data 51
5.1 Introduction 51
5.1.1 Related Works 52
5.2 Depth Video Reconstruction Framework and Notations 53
5.2.1 Systematic Configurations and Depth Data Descriptions 53
5.2.2 Notations for Spatio-Temporal Volume 54
5.2.3 Sampling Patterns 55
5.3 Spatio-Temporal Depth Reconstruction 58
5.3.1 Problem Formulations 58
5.3.2 Spatio-Temporal Depth Reconstruction Algorithm . . 60
5.3.3 Parameter Tuning and Temporal Volume Scalability . 61
5.3.4 Initialization using Temporal Information 65
5.3.5 Preliminary Comparisons 66

Chapter 6 Efficient Sampling Strategy for Spatio-Temporal Depth Data 69
6.1 Motion Compensation Assisted Sampling 69
6.1.1 Oracle Random Sampling Assisted by a Linear Combination of Gradient Maps 70
6.1.2 Synthesis of Gradient Maps 71
6.1.3 Framework for Depth Video Reconstruction 74
6.2 Experimental Results and Discussions 75
6.2.1 Depth Video Reconstruction with Ground Truth Depth 75
6.3 Depth Video Reconstruction from Uniform-Grid Subsampled Data 79
6.4 Dense Disparity Video Estimation 80

Appendix A Derivation on Reconstruction Algorithm 90
A.1 Derivation of Subgradient Algorithm 91
A.2 Derivation of Alternating Direction Method of Multipliers . . 95
A.2.1 **$\boldsymbol{x}$-subproblem:** 95
A.2.2 **$\boldsymbol{u}_\ell$-subproblem:** 96
A.3 Parameter Tuning for ADMM 97
A.3.1 Experimental Configurations 97
A.3.2 Regularization Parameters $(\lambda_1, \lambda_2, \beta)$ 97
A.3.3 Internal Parameters $(\mu, \rho_1, \rho_2, \gamma)$ 98
A.3.4 Summary 98

Bibliography 103

LIST OF FIGURES

Figure 2.1: PSNR values of approximating a disparity patch and a image patch using the leading 5% of the wavelet coefficients. 10

Figure 2.2: Representation schemes. For representing a curve, wavelet has square compact supports and Contourlet has directional rectangular compact support. (Left) Wavelet and (Right) Contourlet [1]. 11

Figure 2.3: Representations of Wavelet and Contourlet. 12

Figure 2.4: MSE curves for Triangle, Square, Circle and Ellipse. 13

Figure 2.5: MSE curves and Representations for Dots. 14

Figure 2.6: Log scale mean square error curves. 15

Figure 2.7: ADMM reconstruction result as a function of sampling ratio ξ 17

Figure 2.8: Snapshot of the comparison between wavelet dictionary and a combined wavelet-contourlet dictionary at $\xi = 0.1$. 17

Figure 3.1: Reconstructed disparity maps from 10% random samples (Aloe). . . 24

Figure 3.2: Reconstructed disparity maps from 10% random samples (Art). . . . 25

Figure 3.3: Comparison of the rate of convergence between ADMM (proposed) and subgradient algorithms [2] for single wavelet dictionary. 31

Figure 3.4: Schematic diagram showing the operations of $\boldsymbol{A}_q$ and $\boldsymbol{B}_q$ 33

Figure 3.5: Runtime comparison of original ADMM algorithm, multiscale ADMM algorithm and subgradient algorithm. 34

Figure 3.6: Multilevel Scheme for the proposed ADMM algorithm. 35

Figure 4.1: Three sampling patterns and the corresponding reconstruction results using the proposed ADMM algorithm. 37

Figure 4.2: Comparison between a deterministic sampling pattern by selecting samples greedily according to the magnitude of $\{a_j\}$, and a randomized sampling pattern using the proposed scheme. 39

Figure 4.3: Snapshot of eigenvectors. 42

Figure 4.4: Comparison between four sampling patterns. 43

Figure 4.5: Comparison of reconstruction performance with noisy samples. We use “Art” disparity map as a test image, and set $\xi = 0.2$. 45

Figure 4.6: Examples of reconstructed results from 10% measured samples using real data using the “Newspaper” dataset, and real captured disparity maps. 50

Figure 5.1: Systematic Overview of the proposed depth video reconstruction framework. 55

Figure 5.2: Flowchart for depth video reconstruction from sparse samples. 57

Figure 5.3: Histogram of magnitude of 3D-DWT transform coefficients. 59

Figure 5.4: Experiments on regularization parameter selection for the proposed spatio-temporal depth reconstruction. 63

Figure 5.5: Experiments on internal parameter selection. 63
Figure 5.6: Reconstruction performance verses number of frames T. 64
Figure 5.7: Convergence rate comparisons. We feed frames 20-21 ($T = 2$) of "tanks" sequence to STDR algorithm. 66

Figure 6.1: Example of reference images, sampled depth data with varying θ_1 and reconstructed disparity map. 72
Figure 6.2: Mean absolute curves with varying θ_1 values. 73
Figure 6.3: Examples of sampling maps and reconstructed results of 8th frame of temple sequence with $\xi = 0.05$. 76
Figure 6.4: Snapshots of reconstructing high resolution depth from downsampled depth data, $\boldsymbol{x}_{\downarrow 8}$, and PSNR comparisons. 81
Figure 6.5: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$ (tanks). 83
Figure 6.6: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$ (books). 85
Figure 6.7: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$ (temples). 87

Figure 1.1: Example disparity maps from Middlebury dataset. 98
Figure 1.2: Comparison of reconstruction performance with varying regularization parameters and depth images. 99
Figure 1.3: MSE for $\xi = 0.2$. 100
Figure 1.4: MSE for $\xi = 0.1$. 101

LIST OF TABLES

Table 3.1: Summary of Parameters. 31

Table 4.1: Comparisons of reconstruction algorithms in terms of PSNR. We put N/A when the algorithm does not converge in 1000 iterations. 46

Table 4.2: Comparisons of reconstruction algorithms in terms of % Bad Pixel. . 47

Table 5.1: Summarize of parameters for STDR. 65

Table 5.2: Comparison of reconstruction algorithms from 5%, 10%, 15% and 20% uniformly random samples. 68

Table 6.1: Comparisons for depth video reconstruction algorithms. 78

Table 6.2: MAE and Bad pixel % comparisons for the depth video estimations. . 89

Table A.1: Summary of Parameters and typical values. 102

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my adviser Prof. Truong Q. Nguyen for the continuous support of my Ph.D study and related research, for his countless hours of reading, encouraging, and most of all patience. His generous guidance helped me in all the time of research and writing of this thesis.

Besides my adviser, I would like to thank my committee members, Prof. Pamela Cosman, Prof. William Hodgkiss, Prof. David Kriegman and Prof. Bhaskar Rao, for agreeing to serve as my committee and for their generous expertise, insightful comments and encouragements.

My sincere thanks also goes to Dr. Stanley Chan, who provides me with selfless guidance and considerable help on the research discussions on reconstruction algorithms and efficient sampling strategies. He is currently a professor in Purdue University, and his vast knowledge and suggestions are invaluable.

Finally, I would like to thank our Video Processing Lab members, Dr. Can Bal, Dr. Ankit Jain, Dr. Kyoung Rok Lee, Dr. Zucheul Lee, Dr. Yujia Wang, Enming Luo, Jason Juang, and Yung-Huan Hsieh, for the stimulating discussions.

Chapter 1, 2, 3, and 4 include materials that have been published in IEEE Transaction on Image Processing 2015, titled “Depth Reconstruction from Sparse Samples: Representation, Algorithm, and Sampling,” with Truong Q. Nguyen and Stanley H. Chan.

Chapter 2 and 3 include materials that have been published in IEEE International Conference on Acoustics, Speech and Signal Processing 2014, titled “Sparse Reconstruction for Disparity Maps using Combined Wavelet and Contourlet Transforms,” with Truong Q. Nguyen.

Chapter 5 includes materials that have been published in IEEE Global Conference on Signal and Information Processing 2015, titled “Spatio-Temporal Depth Data Reconstruction from a Subset of Samples,” with Truong Q. Nguyen.

Chapter 5 and 6 include materials that have been submitted to IEEE Transaction on Image Processing, titled “A Framework for Depth Video Reconstruction from a subset of Samples and its Applications,” with Truong Q. Nguyen.

VITA

2003-2007 B. S. in Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan.

2009-2011 M. S, in Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla.

2011-2015 Ph. D. in Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla.

ABSTRACT OF THE DISSERTATION

From Image to Video, Depth Data Reconstruction from a Subset of Samples: Representations, Algorithms, and Sampling Strategies.

by

Lee-Kang Liu

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2015

Professor Truong Q. Nguyen, Chair

Depth data acquisition has drawn considerable interest in recent years as a result of the rapid development of 3D technology. A large number of acquisition techniques are based on hardware devices, e.g., infra-red sensors, time-of-flight camera, and LiDAR, etc, whereas they have limited performance due to poor depth precision and low resolution. In some situations computational methods are preferred due to its flexibility and low cost. These computational techniques, typically known as depth estimation algorithms, estimate depth maps (in terms of disparities) from a pair of stereo images. However, existing computational techniques are sensitive to various factors such as noise, camera alignment, and illumination, resulting that a few samples are reliable. Therefore, dense depth data reconstruction from sparse samples is a significant technological challenge.

In this thesis, we mainly consider the problem of dense depth data reconstruction from a subset of samples. We present computationally efficient methods to estimate dense depth maps from sparse measurements, and we further extend the work to dense depth video estimation. Working on single depth image, we have three main contributions: First, we provide empirical evidence that depth maps can be encoded much more sparsely than natural images by using common dictionaries such as wavelets and contourlets, and show that disparity maps can be sparsely represented by a combined wavelet and contourlet dictionary. Second, we propose a subgradient algorithm for dense depth image reconstruction, and propose an alternating direction methods of multipliers (ADMM) algorithm with a multi-scale warm start procedure to further speed up the convergence. Third, we propose a two-stage randomized sampling scheme to optimally choose the sampling locations, thus maximizing the reconstruction performance for a given sampling budget. Experimental results show that the proposed methods produce high quality dense depth estimates, and are robust to noisy measurements.

For dealing with depth video sequences, a framework for depth video reconstruction from a subset of samples is proposed. By redefining classical dense depth estimation into two individual problems, sensing and synthesis, we propose a motion compensation assisted sampling (MCAS) scheme and a spatio-temporal depth reconstruction (STDR) algorithm for reconstructing depth video sequences from a subset of samples. Using the 3-dimensional extensible dictionary, 3D-DWT, and applying alternating direction method of multiplier technique, the proposed STDR algorithm possesses scability for temporal volume and efficiency for processing large scale depth data. Exploiting the temporal information and corresponding RGB images, the proposed MCAS scheme achieves an efficient 1-Stage sampling. Experimental results show that the proposed depth reconstruction framework outperforms the existing methods and is competitive comparing to our previous work on sampling single depth image, which requires a pilot signal in the 2-Stage sampling scheme. Finally, to estimate missing reliable depth samples from varying input sources, we present an inference approach using geometrical and color similarities. Applications for depth video super resolution from uniform-grid subsampled data and dense disparity video estimation from a subset of reliable samples are presented.

Chapter 1

Introduction

The rapid development of 3D technology has created a new wave of visualization and sensing impacts to the digital signal processing community. From remote sensing [3] to preserving historical heritages [4], and from rescue [5] to 3D laparoscopic surgery [6, 7], the footprints of 3D have been influencing a broad spectrum of the technological frontiers.

The successful development of 3D signal processing is fundamentally linked to a system's ability to acquire depth. To date, there are two major classes of depth acquisition techniques: hardware solutions and computational procedures. Hardware devices are usually equipped with active sensors such as time-of-flight (ToF) camera [8] and LiDAR [9]. While being able to produce high quality depth maps, these hardware systems have high instrumentation cost. Moreover, the data acquisition time of the devices is long (*e.g.*, a recently proposed ToF cameras can only achieve 10fps [10], whereas standard cameras nowadays can easily achieve 60fps.) Although speeding up is possible, spatial resolution has to be traded off in return.

An alternative solution to acquiring depth is to estimate depth using a set of computational procedures. This class of computational methods, broadly referred to as disparity estimation algorithms [11, 12, 13, 14], estimates the depth by computing the disparities between a pair of stereo images through their corresponding matching points [15, 16]. Disparity estimation algorithms usually work well under well conditioned environments, but they could be sensitive to illumination, noise, stereo alignments, and other camera factors. Thus, the effective number of matching points that one can use for disparity estimation is actually much fewer than the number of pixels of the depth map [17, 18].

1.1 Scope and Contributions

The objective of this thesis is to present a sampling and reconstruction framework for improving the depth acquisition process. The key idea is to carefully select a sparse subset of spatial samples and use an optimization algorithm to reconstruct the final dense depth map.

In this thesis, we first consider the framework for single depth image and then extend it to depth video sequences. The six major contributions are

1. Representation (Chapter 2): In order to reconstruct the depth map, we must first define an appropriate representation. We show that, as opposed to natural images, depth maps can be well approximated using a sparse subset of wavelet bases. Moreover, we show that a combined dictionary of wavelets and contourlets can further improve the reconstruction quality.

2. Algorithm (Chapter 3): We first discuss a subgradient algorithm for dense depth image reconstruction using combined wavelet-contourlet dictionary, and we further propose a fast numerical algorithm based on the alternating direction method of multipliers (ADMM). We derive novel splitting strategies that allow one to solve a sequence of parallelizable subproblems. We also present a multiscale implementation that utilizes the depth structures for efficient warm starts.

3. Sampling (Chapter 4): We propose an efficient spatial sampling strategy that maximizes the reconstruction performance. In particular, we show that for a fixed sampling budget, a high quality sampling pattern can be obtained by allocating random samples with probabilities in proportional to the magnitudes of the depth gradients.

4. Reconstruction for Spatio-Temporal Depth Data (Chapter 5): To deal with spatio-temporal depth data, we propose a spatio-temporal depth reconstruction (STDR) algorithm using the technique of alternating direction method of multipliers. We

formulate a mathematical model that achieves temporal scalability using 3D-DWT, leading to robust reconstruction performance to varying sizes of temporal volumes. Moreover, using temporal information we present a speed-up scheme for the proposed STDR algorithm.

5. Sampling for Spatio-Temporal Depth Data (Chapter 6): With a fixed sampling budget, we propose a motion compensation assisted sampling (MCAS) scheme that predicts and determines locations of reliable samples using a combined gradient information from RGB and motion compensated depth images, achieving an efficient 1-Stage sampling strategy without the requirement of pilot signal proposed in our previous work [19]. The resulting method is more suitable for reconstructing depth video sequences.

6. Applications (Chapter 6): Using geometrical and color similarities, we propose an internal reliable depth data estimation for missing samples between MCAS predictions and input sources. With the integration of the proposed depth video reconstruction framework to practical systems, we demonstrate the following applications: (1) depth video SR from uniformly-grid subsampled depth data, and (2) dense disparity video from a subset of estimated and reliable disparities.

1.2 Related Work

The focus of this work lies in the intersection of two closely related subjects: depth enhancement and compressed sensing. Both subjects have a rich collection of prior works but there are also limitations which we will now discuss.

The goal of depth enhancement is to improve the resolution of a depth map. Some classical examples include Markov Random Field (MRF) [20], bilateral filter [21], and other approaches [22, 23]. One limitation of these methods is that the low-resolution depth maps are sampled uniformly. Also, it is usually assumed that a color image of the scene is available. In contrast, our proposed method is applicable to *any* non-uniformly sampled low-resolution depth map and does not require color images. Thus, the new method allows for a greater flexibility for the enhancement.

Compressed sensing (CS) is a popular mathematical framework for sampling and recovery [24]. In many cases, CS methods assume that *natural images* exhibit sparse structures in certain domains, *e.g.*, wavelet. However, as will be discussed in Chapter 2 of this thesis, natural images are indeed *not* sparse. If we compare natural images to depth maps, the latter would show a much sparser structure than the former. Furthermore, the theory of combined bases [25, 26] shows that a pair of incoherent bases are typically more effective for signal recovery. Yet, the application of these theories to depth maps is not fully explored.

The most relevant paper to our work is perhaps [2]. However, our work has two advantages. First, we propose a new ADMM algorithm for the reconstruction task (Chapter 3). We show that the ADMM algorithm is significantly more efficient than the subgradient method proposed in [2]. Second, we present a sampling scheme to choose optimal sampling patterns to improve the depth reconstruction (Chapter 4), which was not discussed in [2].

We should also mention a saliency-guided CS method proposed in [27, 28]. In these two papers, the spatial sampling is achieved by a mixing-plus-sampling process, meaning that the unknown pixels are filtered and then sub-sampled. The filtering coefficients are constructed using a pre-defined saliency map and certain density functions (*e.g.*, Gaussian-Bernoulli). In our work, the mixing process is *not* required so that depth values are sampled without filtering. This makes our proposed method applicable to disparity estimation where mixing cannot be used (otherwise it will defeat the purpose of reconstructing dense depth maps from a few estimated values.)

Finally, advanced computational photography techniques are recently proposed for fast depth acquisition, *e.g.*, [29, 30]. However, the problem settings of these works involve hardware designs and are thus different from this work.

1.3 Notations and Problem Formulation

1.3.1 Depth and Disparity

The type of data that we are interested in studying is the depth map. Depth can be directly measured using active sensors, or inferred from the disparity of a pair of stereo images. Since the correspondence between depth and disparity is unique by simple geometry [31], in the rest of this thesis we shall use depth and disparity interchangeably. In the following, we note symbols and describe the problem formulations for the framework of single depth image sampling, representation and reconstruction.

1.3.2 Sampling Model

Let $\boldsymbol{x} \in \mathbb{R}^N$ be an $N \times 1$ vector representing a disparity map. For simplicity we assume that $\boldsymbol{x}$ is normalized so that $0 \leq x_j \leq 1$ for $j = 1, \ldots, N$.

To acquire a set of spatial samples, we define a diagonal matrix $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ with the (j, j)th entry being

$$S_{jj} \overset{\text{def}}{=} \begin{cases} 1, & \text{with probability } p_j, \\ 0, & \text{with probability } 1 - p_j, \end{cases} \tag{1.1}$$

where $\{p_j\}_{j=1}^N$ is a sequence of pre-defined probabilities. Specific examples of $\{p_j\}_{j=1}^N$ will be discussed below. For now, we only require $\{p_j\}_{j=1}^N$ to satisfy two criteria: (1) for each $j = 1, \ldots, N$, p_j must be bounded so that $0 \leq p_j \leq 1$; (2) the average of the probabilities must achieve a target *sampling ratio* ξ:

$$\frac{1}{N} \sum_{j=1}^{N} p_j = \xi, \tag{1.2}$$

where $0 < \xi < 1$.

Example 1. *If $p_j = \xi$ for all j, then the sampling pattern $\boldsymbol{S}$ is a diagonal matrix with uniformly random entries. This sampling pattern corresponds to a uniform sampling without filtering in the classical compressed sensing, e.g., [24].*

Example 2. *If $p_j = 1$ for $j \in \Omega_1$ and $p_j = 0$ for $j \in \Omega_0$, where Ω_1 and Ω_0 are two predefined sets such that $|\Omega_1| = \xi N$ and $|\Omega_0| = (1-\xi)N$, then $\boldsymbol{S}$ is a deterministic sampling pattern. In particular, if Ω_1 and Ω_0 are designed so that the indices are uniformly gridded, then $\boldsymbol{S}$ will become the usual down-sampling operator.*

With $\boldsymbol{S}$, we define the sampled disparity map as

$$\boldsymbol{b} = \boldsymbol{S}\boldsymbol{x}. \tag{1.3}$$

Note that in (1.3), the sampled disparity $\boldsymbol{b} \in \mathbb{R}^{N\times 1}$ will contain zeros, *i.e.*, $b_j = 0$ if $S_{jj} = 0$. Physically, this corresponds to the situation where the unsampled pixels are marked with a value of zero.

Remark 1. *Since $\boldsymbol{S}$ is a random diagonal matrix, readers at this point may have concerns about the overall number of samples which is also random. However, we argue that such randomness has negligible effects for the following reason. For large N, standard concentration inequality guarantees that the average number of ones in $\boldsymbol{S}$ stays closely to ξN. In particular, by Bernstein's inequality [32] we can show that for $\varepsilon > 0$,*

$$\Pr\left(\left|\frac{1}{N}\sum_{j=1}^{N} S_{jj} - \xi\right| > \varepsilon\right) \leq 2\exp\left\{-\frac{N\varepsilon^2}{1/2 + 2\varepsilon/3}\right\}. \tag{1.4}$$

Therefore, although the sampling pattern in our framework is randomized, the average number of samples is concentrated around ξN for large N.

1.3.3 Representation Model

To properly formulate the reconstruction problem, we assume that the disparity map can be efficiently represented as a linear combination of basis vectors $\{\boldsymbol{\varphi}_i\}_{i=1}^{M}$:

$$\boldsymbol{x} = \sum_{i=1}^{M} \langle \boldsymbol{x}, \boldsymbol{\varphi}_i \rangle \boldsymbol{\varphi}_i, \tag{1.5}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product. Defining $\alpha_i \overset{\text{def}}{=} \langle \boldsymbol{x}, \boldsymbol{\varphi}_i \rangle$ as the ith basis coefficient, $\boldsymbol{\alpha} \overset{\text{def}}{=} [\alpha_1, \ldots, \alpha_M]^T$, and $\boldsymbol{\Phi} \overset{\text{def}}{=} [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M]$, the relationship in (1.5) can be equivalently written as $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\alpha}$.

The reconstruction problem can be posed as an optimization problem in which the goal is to seek a sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^M$ such that the observed samples $\boldsymbol{b}$ are best approximated. Mathematically, we consider the problem

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{S}\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{1.6}$$

where $\lambda > 0$ is a regularization parameter, and $\|\cdot\|_1$ is the ℓ_1-norm of a vector.

In this paper, we are mainly interested in two types of $\boldsymbol{\Phi}$ — the wavelet frame and the contourlet frame [1]. Frames are generalizations of the standard bases in which M, the number of bases, can be more than N, the dimension of $\boldsymbol{x}$. Moreover, for any frame $\boldsymbol{\Phi}$, it holds that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{I}$. Therefore, $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\alpha}$ if and only if $\boldsymbol{\alpha} = \boldsymbol{\Phi}^T\boldsymbol{x}$. Using this result, we can equivalently express (1.6) as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{S}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{\Phi}^T\boldsymbol{x}\|_1. \tag{1.7}$$

Remark 2. *In compressed sensing literature,* (1.6) *is known as the synthesis problem and* (1.7) *is known as the analysis problem [33]. Furthermore, the overall measurement matrix* $\boldsymbol{S}\boldsymbol{\Phi}$ *in* (1.6) *suggests that if* $p_j = \xi$ *for all* j*, then* $\boldsymbol{S}\boldsymbol{\Phi}$ *corresponds to the partial orthogonal system as discussed in [34]. In this case, the restricted isometry property (RIP) holds [35] and exact recovery can be guaranteed under appropriate assumptions of sparsity and number of measurements. For general* $\{p_j\}_{j=1}^N$*, establishing RIP is more challenging, but empirically we observe that the optimization produces reasonable solutions.*

1.3.4 Penalty Functions

As discussed in [2], (1.7) is not an effective formulation because the ℓ_1 norm penalizes *both* the approximation (lowpass) and the detailed (highpass) coefficients. In reality, since disparity maps are mostly piecewise linear functions, the lowpass coefficients should be maintained whereas the highpass coefficients are desirable to be sparse. To this end, we introduce a binary diagonal matrix $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ where the (j,j)th entry is 0 if j is an index in the lowest passband, and is 1 otherwise. Consequently, we modify the optimization problem as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{S}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{W}\boldsymbol{\Phi}^T\boldsymbol{x}\|_1. \tag{1.8}$$

Finally, it is desirable to further enforce smoothness of the reconstructed disparity map. Therefore, we introduce a total variation penalty so that the problem becomes

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{S}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{W}\boldsymbol{\Phi}^T\boldsymbol{x}\|_1 + \beta\|\boldsymbol{x}\|_{TV}. \tag{1.9}$$

Here, the total variation norm is defined as

$$\|\boldsymbol{x}\|_{TV} \overset{\text{def}}{=} \|\boldsymbol{D}_x\boldsymbol{x}\|_1 + \|\boldsymbol{D}_y\boldsymbol{x}\|_1, \tag{1.10}$$

where $\boldsymbol{D} = [\boldsymbol{D}_x;\, \boldsymbol{D}_y]$ is the first-order finite difference operator in the horizontal and vertical directions. The above definition of total variation is known as the anisotropic total variation. The same formulation holds for isotropic total variation, in which $\|\boldsymbol{x}\|_{TV} = \sum_{j=1}^{N}\sqrt{[\boldsymbol{D}_x\boldsymbol{x}]_j^2 + [\boldsymbol{D}_y\boldsymbol{x}]_j^2}$.

The problem in (1.9) is generalizable to take into account of a combination of L dictionaries. In this case, one can consider a sum of L penalty terms as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{S}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \sum_{\ell=1}^{L}\lambda_\ell\|\boldsymbol{W}_\ell\boldsymbol{\Phi}_\ell^T\boldsymbol{x}\|_1 + \beta\|\boldsymbol{x}\|_{TV}. \tag{1.11}$$

For example, in the case of combined wavelet and contourlet dictionaries, we let $L = 2$.

This Chapter includes materials that have been published in IEEE Transaction on Image Processing 2015, titled “Depth Reconstruction from Sparse Samples: Representation, Algorithm, and Sampling,” with Truong Q. Nguyen and Stanley H. Chan.

Chapter 2

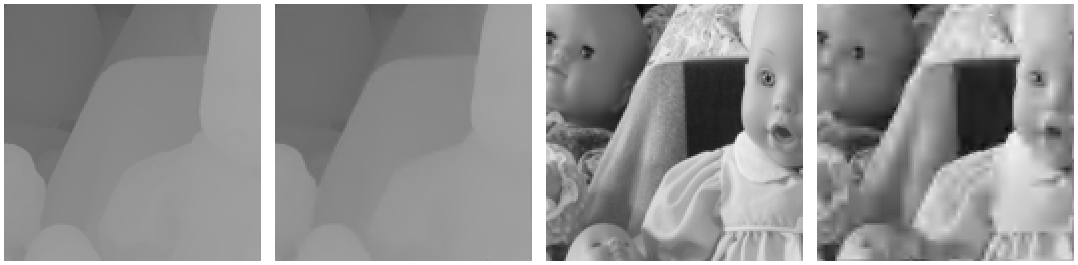
Sparse Representation of Depth Data

The choice of the dictionary $\mathbf{\Phi}$ in (1.11) is an important factor for the reconstruction performance. In this chapter we discuss the general representation problem of disparity maps. We show that disparity maps can be represented more sparsely than natural images. We also show that a combined wavelet-contourlet dictionary is more effective in representing disparity maps than using the wavelet dictionary alone.

2.1 Natural Images vs Depth Data

Seeking effective representations for *natural images* is a well-studied subject in image processing [36, 37, 38, 39, 1, 40, 41, 42]. However, representations of *disparity maps* seems to be less studied. For example, how efficient can a pre-defined dictionary (i.e., wavelets) represent disparity maps as compared to natural images captured by RGB sensors from the same scene. To address this question, we consider a 128×128 cropped patch from a gray-scaled image and the corresponding patch in the disparity map. For each of the image and the disparity, we apply the wavelet transform with Daubechies 5/3 filter and 5 decomposition levels. Then, we truncate the wavelet coefficients to the leading 5% coefficients with the largest magnitudes. The reconstructed patches are compared and the results are shown in Figure 2.1.

The result indicates that for the same number of wavelet coefficients, the disparity



(a) Original disparity

(b) Approx. disparity (50.25 dB)

(c) Original view

(d) Approx. view (29.29 dB)

Figure 2.1: PSNR values of approximating a disparity patch and a image patch using the leading 5% of the wavelet coefficients.

map can be synthesized with significantly lower approximation error than the image. While such result is not surprising, the big difference in the PSNRs provides evidence that reconstruction of disparity maps from sparse samples should achieve better results than that of natural images.

2.2 Wavelet vs Contourlet

The above results indicate that wavelets are efficient representations for disparity maps. Our next question is to ask whether some of the dictionaries would perform better than other dictionaries.

2.2.1 Representations of Wavelet and Contourlet Bases

As wavelet transform has sparse representation for images [38], it is widely used in several image applications. For example, JPEG 2000 uses the 5/3 wavelet function for loseless compression and 9/7 wavelet function for lossy compression. Since wavelet functions are designed in square shape compact supports, local compact group of points can be efficiently compressed. In addition to points, contours (lines) are also a major components in images. For describing a depth map, lines and points are two fundamental bases. Contourlet has been proposed and claimed that it has better representation for piecewise smooth contours. Since directional filter banks are applied, directional line bases are generated, and by selecting the frequency partitions for fitting parabolic scaling functions, contourlet has better representation for image contours [1, 39]. Aside from wavelet transform, contourlet has directional rectangular compact supports since directional filter bank is applied. Figure 2.2 denotes the basis function used to represent a curve by wavelet basis (left) and contourlet basis (right) [1].

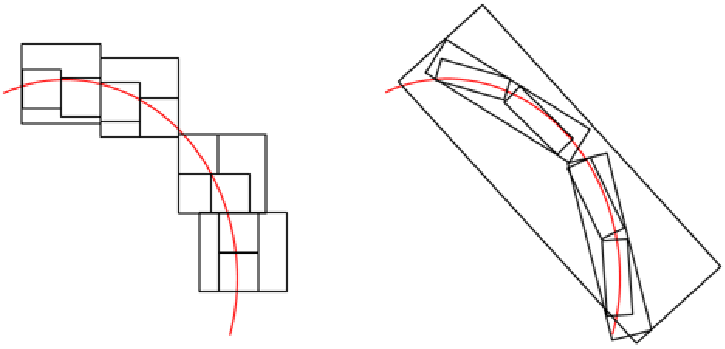


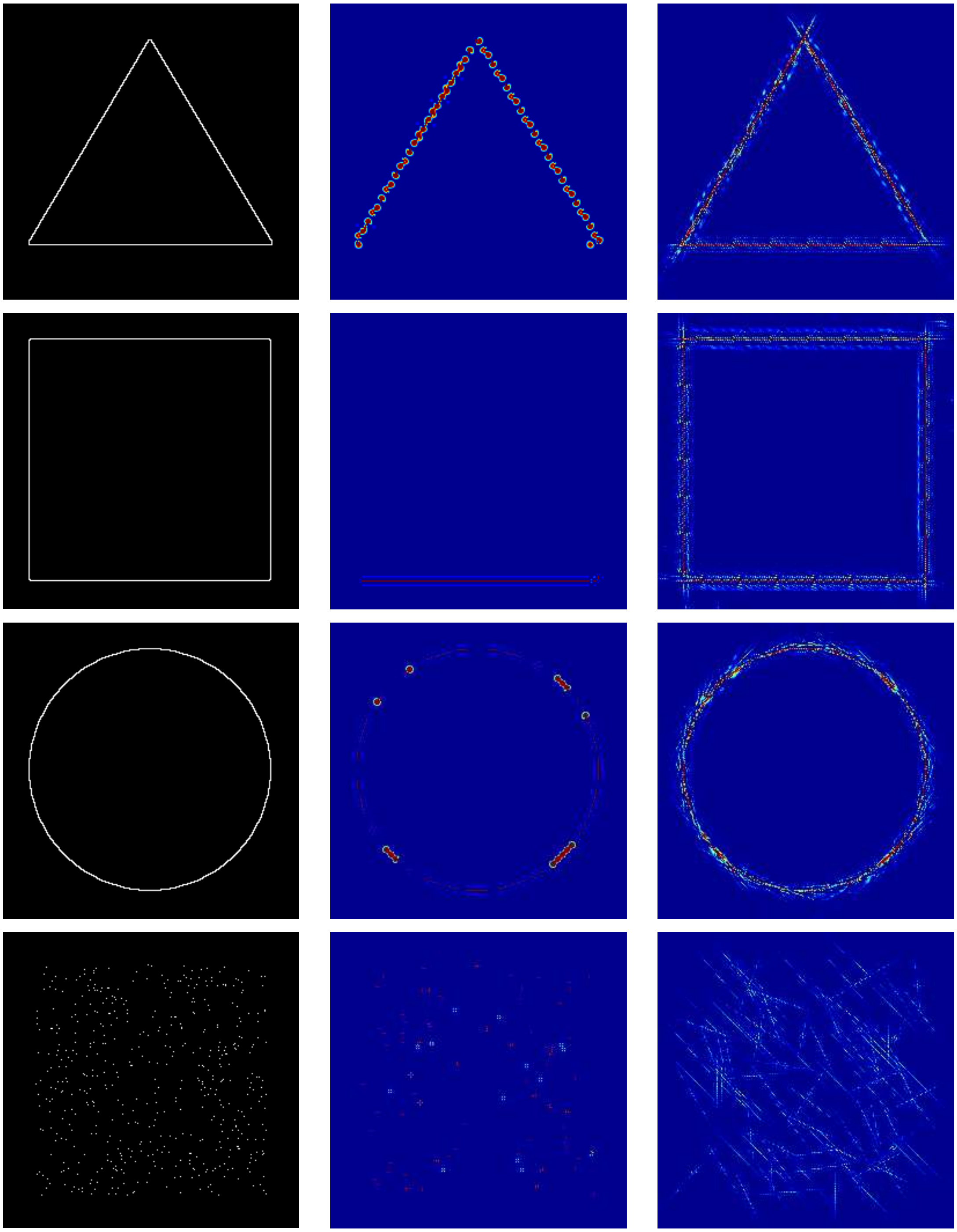
Figure 2.2: Representation schemes. For representing a curve, wavelet has square compact supports and Contourlet has directional rectangular compact support. (Left) Wavelet and (Right) Contourlet [1].

In this section, we are presenting the bases' difference between wavelet and contourlet. Since triangle, square, ellipse, circle and points are fundamental structures that commonly exist in disparity maps, we use them as test images. In addition, 'bior9.7' filter is applied in contourlet, and for fair comparison, 'bior9.7' is selected in wavelet transform. As Do and Vetterli [1] suggest that for fitting parabolic scaling function, the frequency partitioning should be doubled than the previous scale, hence we select nlevel = [5 6], which means that there are $2^5 = 32$ wedge-shaped frequency bands in the first finer scale and $2^6 = 64$ wedge-shaped frequency bands in the finest scale.

For evaluation, we use mean square error as evaluation metric. Given a test image $\boldsymbol{f} \in \mathbb{R}^n$, and a synthesized image from the most M significant coefficients, $\hat{\boldsymbol{f}}^{((M)} \in \mathbb{R}^n$, the mean square error is defined as

$$\frac{1}{n}\|\boldsymbol{f} - \hat{\boldsymbol{f}}^{(M)}\|_2^2. \tag{2.1}$$

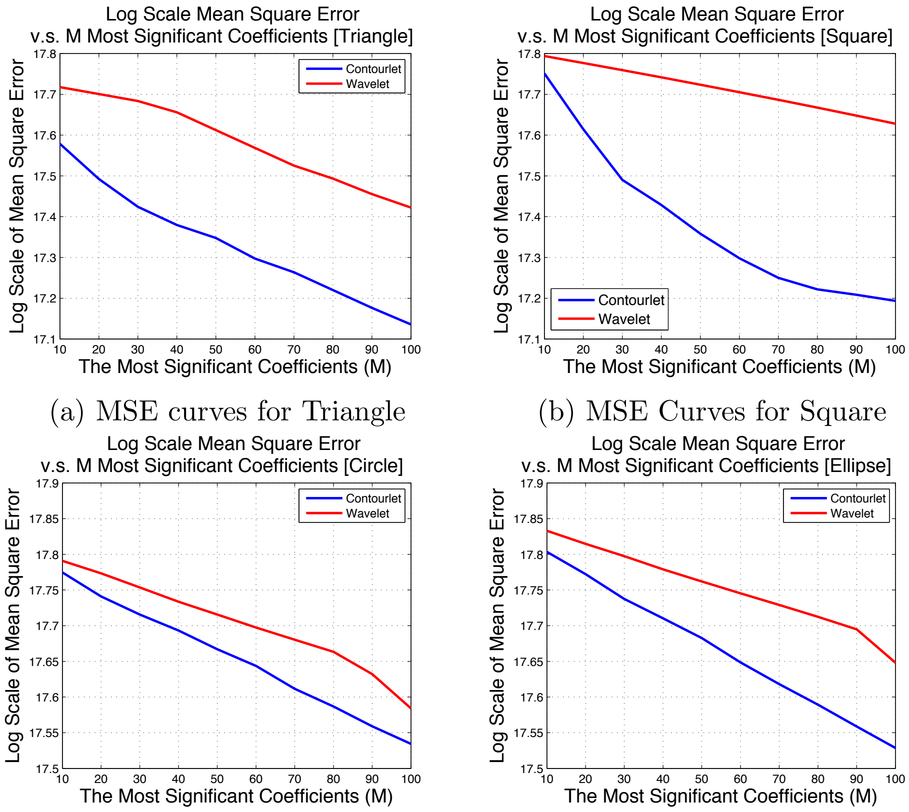
As the ground truth image is given, the synthesized image, $\hat{\boldsymbol{f}}^{(M)}$, is estimated by keeping the most significant M coefficients in transform domain. Using (2.1), the log scale mean square is shown in Figure 2.4 and the bottom Figure 2.5. If the trans-



(a) Ground Truth (b) Wavelet (c) Contourlet

Figure 2.3: Representations of Wavelet and Contourlet.

form analyses the selected image more efficiently, the error will be less. As the bases have better representation for the image structures, fewer number of coefficients will be



(a) MSE curves for Triangle

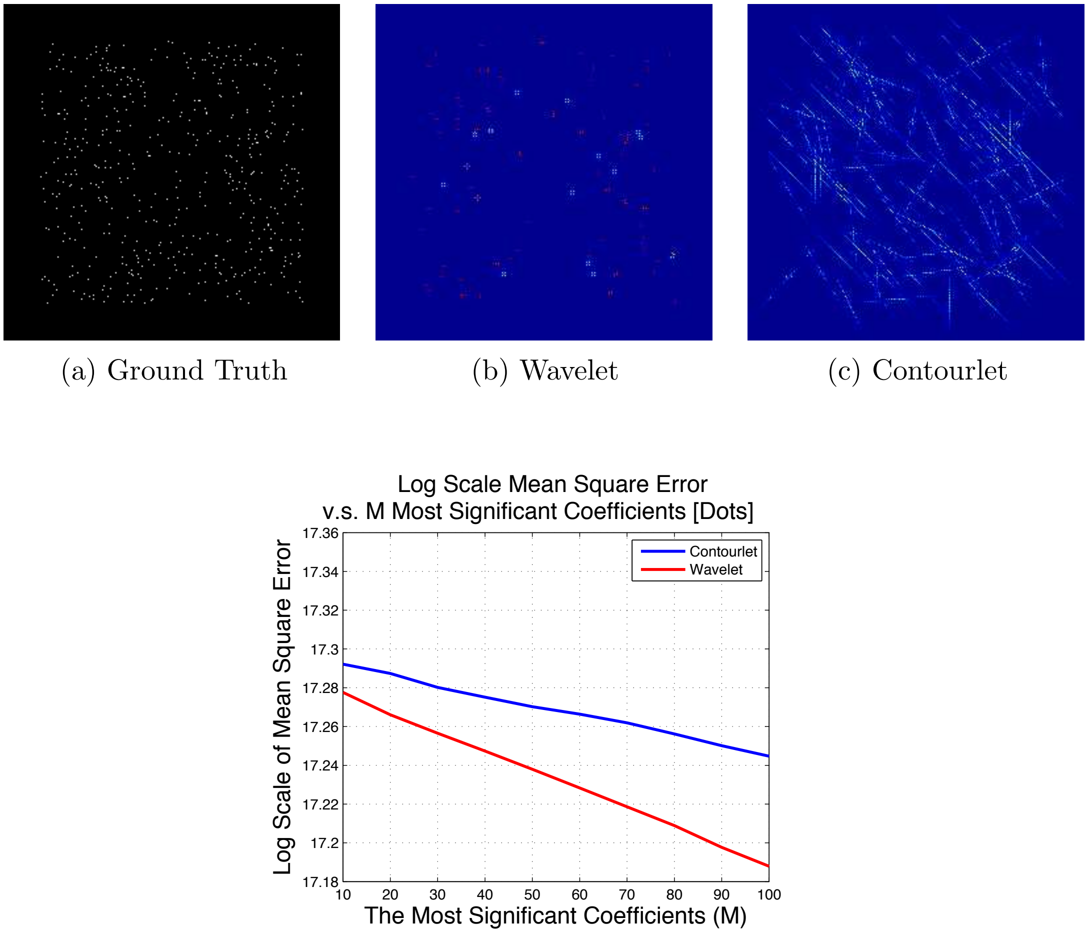
(b) MSE Curves for Square

(c) MSE curves for Circle

(d) MSE curves for Ellipse

Figure 2.4: MSE curves for Triangle, Square, Circle and Ellipse.

needed for synthesizing the selected image. Figure 2.3 and Figure 2.5 summarize our studies on representations. The ground truth images are shown in the first column, the synthesized images by keeping M=100 of the most significant transform coefficients in contourlet are shown in the second column, and images in the third column are synthesized images by keeping M=100 of the most significant wavelet coefficients. According to the results, contourlet transform has less mean square errors than wavelet as the test images are triangle, square, circle and ellipse. However, wavelet has less mean square error than contourlet as the input is points image. Observing the synthesized images in Figure 2.3(c) and Figure 2.5(c), since contourlet bases are directional lines, contourlet transform has better representation for image contours. Similarly, observing the synthesized images in Figure 2.3(b) and Figure 2.5(b), since wavelet bases have square shape compact supports, wavelet transform has better representation for local compact group of points. Additionally, wrong selection of transforms can yield erroneous synthesized



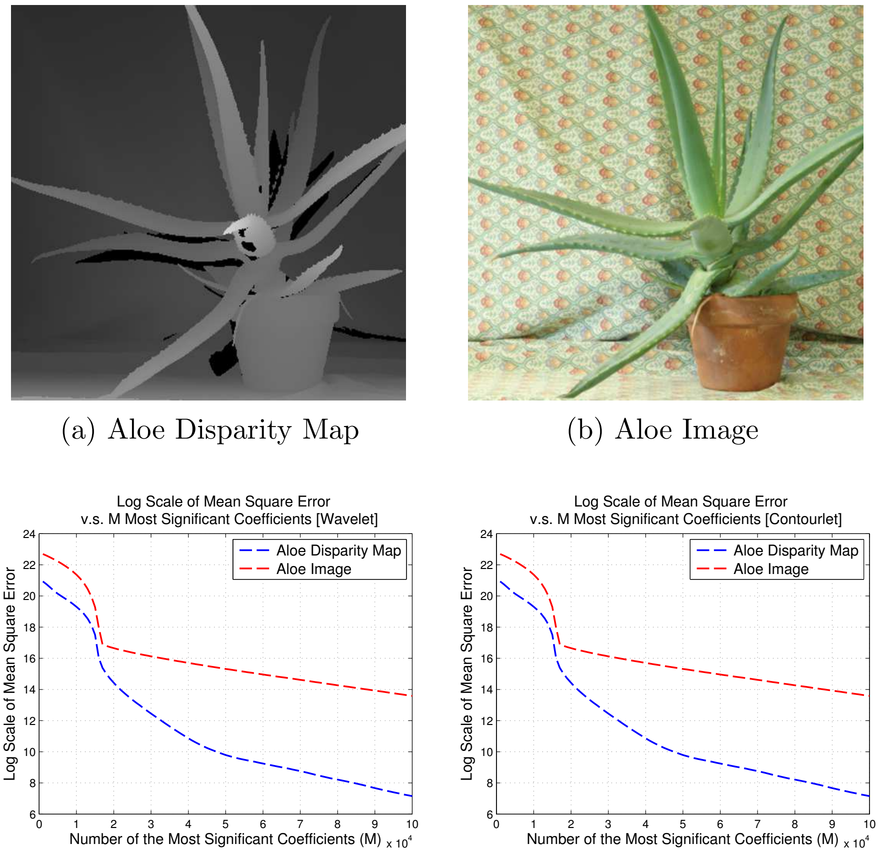
(a) Ground Truth (b) Wavelet (c) Contourlet

Figure 2.5: MSE curves and Representations for Dots.

images. Observing synthesized points image, as contourlet transform is selected, the synthesized image has errors resulting from directional bases. In contrast, as wavelet transform is selected, the synthesized image has matched results. In summary, wavelet has better representation for local compact group of points since it has square compact support, and contourlet has better representation for image contours since it has directional rectangular compact supports.

2.2.2 Sparse Representation of Disparity Maps

Lines and points are two fundamental structures that mainly describe objects or scenes in images. Since images are composed of points and lines, these components are relatively informative. As these information are presented, the semantic of images can always be understood by human beings. Disparity maps, which encode depth information, especially satisfy these conditions. Since disparity maps contain mainly points and contours, and possess piecewise smooth regions, points and lines are critical components for describing objects in depth maps. Moreover, the transform coefficients of disparity maps are more sparse than those of the corresponding images. As shown in Figure 2.6(c)(d), the blue curve represents the errors of Aloe disparity map and the red curve represents the errors of Aloe image. In both wavelet and contourlet transforms, disparity map has less errors than its corresponding image, seeing Figure 2.6(a)(b). Therefore, in terms of sparse representation, disparity maps have higher sparsity in both wavelet and contourlet transforms than images.



(a) Aloe Disparity Map

(b) Aloe Image

(c) Wavelet

(d)Contourlet

Figure 2.6: (a)(b): Test Data. (c)(d): Log scale mean square error curves of aloe disparity map and aloe image.

2.2.3 Evaluation Metric

To compare the performance of two dictionaries, it is necessary to first specify what metric to use. For the purpose of reconstruction, we compare the mean squared error (MSE) of the reconstructed disparity maps obtained by choosing different dictionaries in (1.11). For any fixed sampling pattern $\boldsymbol{S}$, we say that a dictionary $\mathbf{\Phi}_1$ is better than another dictionary $\mathbf{\Phi}_2$ if the reconstruction result using $\mathbf{\Phi}_1$ has a lower MSE than using $\mathbf{\Phi}_2$, for the best choice of parameters λ_1, λ_2 and β. Note that in this evaluation we do not compare the sparsity of the signal using different dictionaries. In fact, sparsity is not an appropriate metric because contourlets typically require 33% more coefficients than wavelets [43]. However, it is known that contourlets have better representations of lines and curves than wavelets.

2.2.4 Comparison Results

We synthetically create a gray-scaled image consisting of a triangle overlapping with an ellipse to simulate a disparity map. We choose the uniformly random sampling pattern $\boldsymbol{S}$ so that there is no bias caused by a particular sampling pattern. As parameters are concerned, we set $\lambda_1 = 4 \times 10^{-5}$ and $\beta = 2 \times 10^{-3}$ for the single wavelet dictionary model ($L = 1$), and $\lambda_1 = 4 \times 10^{-5}$, $\lambda_2 = 2 \times 10^{-4}$ and $\beta = 2 \times 10^{-3}$ for the combined dictionary model ($L = 2$). The choices of these parameters are discussed in Appendix A.3.

Using the proposed ADMM algorithm (See Chapter 3.3), we plot the performance of the reconstruction result as a function of the sampling ratio. For each point of the sampling ratio, we perform a Monte-Carlo simulation over 20 independent trials to reduce the fluctuation caused by the randomness in the sampling pattern. The result in Figure 2.7 indicates that the combined dictionary is consistently better than the wavelet dictionary alone. A snapshot of the result at $\xi = 0.1$ is shown in Figure 2.8. As observed, the reconstruction along the edges of the ellipse is better in the combined dictionary than using wavelet alone.

This Chapter is published in IEEE Transactions on Image Processing with Stanley H. Chan and Truong Q. Nguyen, and in IEEE international Conference on Acoustics, Speech and Signal Processing with Truong Q. Nguyen.

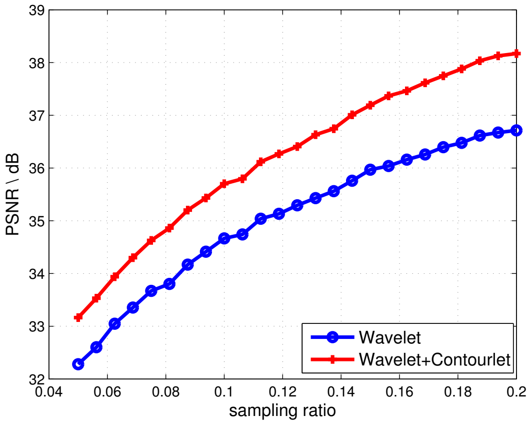
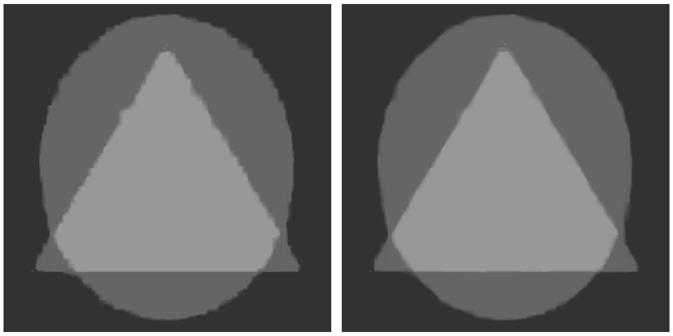


Figure 2.7: ADMM reconstruction result as a function of sampling ratio ξ. Each point on the curves is averaged over 20 independent Monte-Carlo trials. The PSNR evaluates the performance of solving (1.11) using wavelet and wavelet+contourlet.



(a) Wavelet, 34.77 dB (b) Combined, 35.86 dB

Figure 2.8: Snapshot of the comparison between wavelet dictionary and a combined wavelet-contourlet dictionary at $\xi = 0.1$.

This Chapter includes materials that have been published in IEEE Transaction on Image Processing 2015, titled “Depth Reconstruction from Sparse Samples: Representation, Algorithm, and Sampling,” with Truong Q. Nguyen and Stanley H. Chan, and in IEEE International Conference on Acoustics, Speech and Signal Processing 2014, titled “Sparse Reconstruction for Disparity Maps using Combined Wavelet and Contourlet Transforms,” with Truong Q. Nguyen.

Chapter 3

Algorithms for Depth Data Reconstruction

3.1 Symbols and Problem Formulation

Based on the reasons that disparity maps have sparsity in both wavelet and contourlet transform domains, and the point and line representations of wavelet and contourlet bases, we propose a convex model for reconstructing dense disparity maps from sparse samples by using wavelet and contourlet transforms. Based on the compressed sensing theory, to reconstruct an image, sparsity in transform domain corresponds to the needed number of samples in spatial domain. The fewer transform coefficients are needed for representing an image in transform domain, the fewer spatial measurements are needed for image reconstruction [44]. Moreover, disparity maps have no texture information and possess piecewise smooth regions, hence the main structures are points and lines. Since wavelet has better representation for points (dots), and contourlet has better representation for contours (lines), wavelet and contourlet bases are suitable for representing disparity maps. Therefore, we propose a convex model for disparity map reconstruction by utilizing both wavelet and contourlet transforms.

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{W}_1\boldsymbol{\Phi}_1^T\boldsymbol{x}\|_1 + \|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{x}\|_1 \qquad \text{subject to} \quad \|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{x}\|_2^2 < \epsilon. \tag{3.1}$$

Matrices $\boldsymbol{\Phi}_1 \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\Phi}_2 \in \mathbb{R}^{N\times N}$ represent wavelet and contourlet bases. $\boldsymbol{W}_1 \in [0,1]^{N\times N}$ and $\boldsymbol{W}_2 \in [0,1]^{N\times N}$ are two diagonal matrices with zeros at the locations of approximation coefficients and ones at the locations of detail coefficients. $\boldsymbol{S} \in [0,1]^{N\times N}$ is a sampling matrix. Vectors $\boldsymbol{b} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{x} \in \mathbb{R}^{N\times 1}$ denote the observations (sparse samples) and disparity map, respectively. The parameter ϵ is the tolerance of errors between measurements and the reconstructed disparity map. Since disparity maps have piecewise smooth regions, to preserve the discontinuity, we introduce smoothness prior, *total variation.* For implementing simplicity, we focus on anisotropic total variation. Given a disparity map, $\boldsymbol{X} \in \mathbb{R}^{r\times c}$, the total variation norm is described as,

$$\|\boldsymbol{X}\|_{\mathrm{TV}} = \sum_{i=0}^{i=r-1}\sum_{j=0}^{j=c-1} \sqrt{(\boldsymbol{X}_{i,j} - \boldsymbol{X}_{i,j+1})^2} + \sqrt{(\boldsymbol{X}_{i,j} - \boldsymbol{X}_{i+1,j})^2}. \tag{3.2}$$

According to (3.2), the difference operators can be represented by matrices. Therefore, given a canonical form of disparity map, $\boldsymbol{x}$, the total variation is

$$\|\boldsymbol{x}\|_{\mathrm{TV}} = \sum_{k=0}^{k=n-1} \sqrt{(\boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{x})^2} + \sqrt{(\boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{x})^2}. \tag{3.3}$$

$\boldsymbol{D}_x$ and $\boldsymbol{D}_y$, represent gradient in horizontal and vertical directions, respectively. Vector $\mathbf{e}_k$ denotes the basis with 1 at location k and 0's otherwise. Moreover, since the approximation of total variation results in bias in low frequency components and the sparsity exists in detail coefficients, we further apply weight matrices, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, for discarding approximation coefficients, hence our proposed model for dense disparity reconstruction is

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{W}_1\boldsymbol{\Phi}_1^T\boldsymbol{x}\|_1 + \|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{x}\|_1 + \beta\|\boldsymbol{x}\|_{\mathrm{TV}} \qquad \text{subject to} \quad \|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{x}\|_2^2 < \epsilon. \tag{3.4}$$

The weight matrices, $\boldsymbol{W}_1 \in [0,1]^{N\times N}$ and $\boldsymbol{W}_2 \in [0,1]^{N\times N}$, are diagonal matrices with zeros at locations of approximation coefficients and ones at locations of detail coefficients. According to the equivalent form, $\boldsymbol{c} = \boldsymbol{\Phi}_1^T\boldsymbol{x}$ and $\boldsymbol{\Phi}_1\boldsymbol{\Phi}_1^T = \boldsymbol{I}$, we replace (3.4) by

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \|\boldsymbol{W}_1\boldsymbol{c}\|_1 + \|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1 + \beta\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\mathrm{TV}} \qquad \text{subject to} \quad \|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{\Phi}_1\boldsymbol{c}\|_2^2 < \epsilon. \tag{3.5}$$

Based on the Lagrangian duality, the (3.5) can be reformulated as an unconstrained problem.

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{b}-\boldsymbol{S}\boldsymbol{\Phi}_1\boldsymbol{c}\|_2^2+\lambda\left(\|\boldsymbol{W}_1\boldsymbol{c}\|_1+\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1+\beta\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\mathrm{TV}}\right). \tag{3.6}$$

Since the Lagrangian multiplier, λ, gives a weight between constraints and objective function, in order to make this problem to be more general, we introduce another parameter to the regularization term, $\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1$, and reformulate the equation as follows:

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{b}-\boldsymbol{S}\boldsymbol{\Phi}_1\boldsymbol{c}\|_2^2+\lambda\|\boldsymbol{W}_1\boldsymbol{c}\|_1+\gamma\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1+\beta\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\mathrm{TV}}. \tag{3.7}$$

3.2 Algorithm: Conjugate Sub-Gradient

3.2.1 Derivation

For solving the unconstrained minimization problem in (3.7), we propose a method that utilizes conjugate subgradients to minimize the gradient of cost function. First of all, finding gradients of each terms is the first step, and the gradient of $\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1$ at location k is

$$\partial_{\boldsymbol{c}}\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1(k)=\left\{\boldsymbol{\Phi}_1^T\boldsymbol{\Phi}_2\mathrm{sign}\left[\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\right]\right\}(k), \tag{3.8}$$

where the operator $\partial_{\boldsymbol{c}}$ denotes element-wise derivative of vector $\boldsymbol{c}$. The definition of function *sign* is

$$\mathrm{sign}\,(v)=\begin{cases}1, & \text{if} \quad v>0,\\ 0, & \text{if} \quad v=0,\\ -1, & \text{if} \quad v<0.\end{cases} \tag{3.9}$$

Since the anisotropic total variation is equivalent to ℓ_1 norm, and by introducing difference operator $\boldsymbol{D}=[\boldsymbol{D}_x,\boldsymbol{D}_y]^T$, we can rewrite $\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\mathrm{TV}}$ as,

$$\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\mathrm{TV}}=\|\boldsymbol{D}\left(\boldsymbol{\Phi}_1\boldsymbol{c}\right)\|_1. \tag{3.10}$$

Moreover, as subgradients at zero value are ambiguous, we introduce Huber functional to approximate the ℓ_1 norm.

$$H_\delta(x) = \begin{cases} |x| - \dfrac{\delta}{2}, & \text{if} \quad |x| \geq \delta, \\ \dfrac{x^2}{2\delta}, & \text{otherwise.} \end{cases} \tag{3.11}$$

Therefore, the gradient of $\|\boldsymbol{D}\left(\boldsymbol{\Phi}_1 \boldsymbol{c}\right)\|_1$ is,

$$\partial_{\boldsymbol{c}} \|\boldsymbol{\Phi}_1 \boldsymbol{c}\|_{\mathrm{TV}}(k) = \partial_{\boldsymbol{c}} H_\delta \left(\sqrt{(\boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c})^2} \right) + \partial_{\boldsymbol{c}} H_\delta \left(\sqrt{(\boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c})^2} \right). \tag{3.12}$$

where

$$\partial_{\boldsymbol{c}} H_\delta \left(\sqrt{\boldsymbol{s}_x^2} \right) = \begin{cases} \boldsymbol{\Phi}_1^T \boldsymbol{D}_x^T \boldsymbol{e}_k^T \operatorname{sign}(\boldsymbol{s}_x), & \text{if} \quad |\boldsymbol{s}_x| \geq \delta, \\ \dfrac{\boldsymbol{\Phi}_1^T \boldsymbol{D}_x^T \boldsymbol{e}_k^T \boldsymbol{s}_x}{\delta}, & \text{otherwise.} \end{cases} \tag{3.13}$$

and

$$\partial_{\boldsymbol{c}} H_\delta \left(\sqrt{\boldsymbol{s}_y^2} \right) = \begin{cases} \boldsymbol{\Phi}_1^T \boldsymbol{D}_y^T \boldsymbol{e}_k^T \operatorname{sign}(\boldsymbol{s}_y), & \text{if} \quad |\boldsymbol{s}_y| \geq \delta, \\ \dfrac{\boldsymbol{\Phi}_1^T \boldsymbol{D}_y^T \boldsymbol{e}_k^T \boldsymbol{s}_y}{\delta}, & \text{otherwise.} \end{cases} \tag{3.14}$$

The variables $\boldsymbol{s}_x = \boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c}$ and $\boldsymbol{s}_y = \boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c}$. Finally, the conjugate subgradient of $\|\boldsymbol{W}_1 \boldsymbol{c}\|_1$ is as follows

$$\triangledown \|\boldsymbol{W}_1 \boldsymbol{c}\|_1 = \sum_k \begin{cases} \operatorname{sign}(\left[\boldsymbol{W}_1 \boldsymbol{c}\right](k)), & \text{if} \quad |\boldsymbol{W}_1 \boldsymbol{c}|(k) \neq 0, \\ -\operatorname{sign}(\boldsymbol{q}(k)) \cdot \min\left\{|\boldsymbol{q}(k)|, 1\right\}, & \text{otherwise.} \end{cases} \tag{3.15}$$

where the variable $\boldsymbol{q}$ is,

$$\boldsymbol{q} = \frac{1}{\lambda} \{ -\boldsymbol{\Phi}_1^T \boldsymbol{S}^T \left(\boldsymbol{b} - \boldsymbol{S} \boldsymbol{\Phi}_1 \boldsymbol{c} \right) + \gamma \triangledown \|\boldsymbol{W}_2 \boldsymbol{\Phi}_2^T \boldsymbol{\Phi}_1 \boldsymbol{c}\|_1 + \beta \triangledown \|\boldsymbol{\Phi}_1 \boldsymbol{c}\|_{TV} \}. \tag{3.16}$$

After calculating gradient of (3.7), the next step is to update variable $\boldsymbol{c}$. At the i-th iteration, the updating equation is

$$\boldsymbol{c}_{i+1} = \boldsymbol{c}_i + \alpha_i \boldsymbol{h}_i. \tag{3.17}$$

The descent direction, $\boldsymbol{h}_i$, is

$$\boldsymbol{h}_{i+1} = -\mathbf{d}_{i+1} + \frac{\mathbf{d}_{i+1}^T \left(\mathbf{d}_{i+1} - \mathbf{d}_i\right)}{\boldsymbol{h}_i^T \left(\mathbf{d}_{i+1} - \mathbf{d}_i\right)}. \tag{3.18}$$

The update method is called, Hestenes-Stiefel(HS) [45], where $\mathbf{d}_i$ is the gradient of (3.7).

$$\mathbf{d}_i = -\boldsymbol{\Phi}_1^T \boldsymbol{S}^T \left(\boldsymbol{b} - \boldsymbol{S}\boldsymbol{\Phi}_1 \boldsymbol{c}_i\right) + \lambda \bigtriangledown \|\boldsymbol{W}_1 \boldsymbol{c}_i\|_1 + \gamma \bigtriangledown \|\boldsymbol{W}_2 \boldsymbol{\Phi}_2^T \boldsymbol{\Phi}_1 \boldsymbol{c}_i\|_1 + \beta \bigtriangledown \|\boldsymbol{\Phi}_1 \boldsymbol{c}_i\|_{\mathrm{TV}}. \tag{3.19}$$

Selecting the step size, α_i, is an important issue, and line search methods are commonly used for solving gradient descent problems. Defining $f(\boldsymbol{c}_i)$ as (3.7) and $d(\boldsymbol{c}_i) = -f'(\boldsymbol{c}_i)$ as (3.19) , the line search algorithm is shown in as Algorithm 1. For detailed derivations, readers can refer to Appendix A.1. For the initial process, variables settings are

$$\boldsymbol{h}_0 = -\mathbf{d}_0,$$

$$\boldsymbol{c}_0 = \boldsymbol{\Phi}_1^T \boldsymbol{S}^T \boldsymbol{b},$$

$$\alpha_0 = 1.$$

As we use gradient descent method to search for the optimal point and run the iteration

Algorithm 1 Backtracking Line Search Algorithm

```
Require: ($\boldsymbol{c}_i$, $\alpha_{i-1}$)
  $c_1 \leftarrow 1e-5, c_2 \leftarrow 0.8$
  $f_c \leftarrow f(\boldsymbol{c}_i)$
  $\delta f \leftarrow c_1 \langle f'(\boldsymbol{c}_i), d(\boldsymbol{c}_i) \rangle$
  $t \leftarrow \alpha^{i-1}$
  $f_{new} \leftarrow f(\boldsymbol{c}_i + t d(\boldsymbol{c}_i))$
  $itr \leftarrow 0$
  while $(f_{new} > f_c + t\delta f) \quad \| \quad (itr == 0)$ do
    $itr \leftarrow itr + 1$
    $t \leftarrow c_2 t$
    $f_{new} \leftarrow f(\boldsymbol{c}_i + t d(\boldsymbol{c}_i))$
  end while
  if $(itr == 1)$ then
    $t \leftarrow t/c_2$
  end if
  return $\alpha_i \leftarrow t$
```

for infinite number of times, the optimal point, $\boldsymbol{c}^*$, should be found. Practically, running the algorithm foran infinite number of time is impossible, hence it is necessary to define the stop criteria. Therefore, given the current iteration, i, we define the stop criteria as follows,

$$\frac{\left|\left(\frac{1}{N}\sum_{j=i-N}^{j=i-1}\boldsymbol{g}_j\right)-\boldsymbol{g}_i\right|}{\left(\frac{1}{N}\sum_{j=i-N}^{j=i-1}\boldsymbol{g}_j\right)} \leq tol. \tag{3.20}$$

where ϵ is a positive scalar value, and $\boldsymbol{g}_i = \boldsymbol{g}(\boldsymbol{c}_i)$ is the cost function of regularization terms,

$$\boldsymbol{g}(\boldsymbol{c}_i) = \lambda\|\boldsymbol{W}_1\boldsymbol{c}_i\|_1 + \gamma\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}_i\|_1 + \beta\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\text{TV}}. \tag{3.21}$$

Finally, the overall dense disparity algorithm is summarized in Algorithm 2.

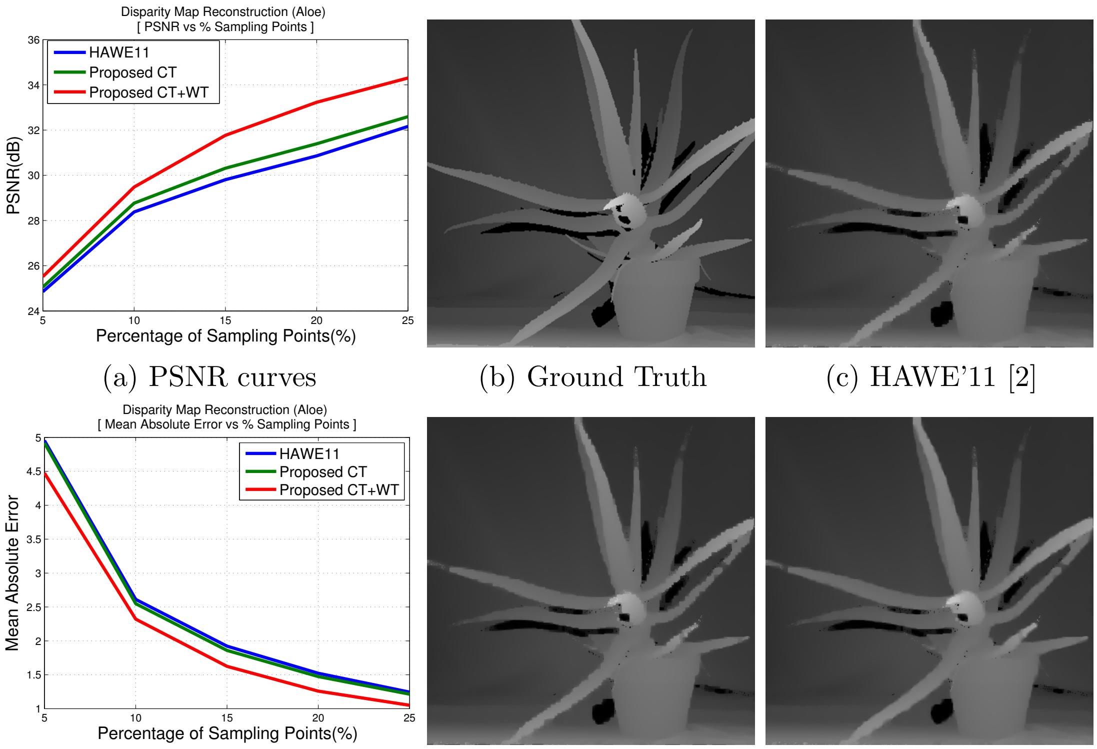
```
Algorithm 2 Dense Disparity Reconstruction Algorithm (Conjugate Subgradient)
Require: $(\boldsymbol{b}, \boldsymbol{S})$
  $\boldsymbol{c}_0 \leftarrow \boldsymbol{\Phi}_1^T\boldsymbol{S}^T\boldsymbol{b}$
  $\boldsymbol{h}_0 \leftarrow -\mathbf{d}_0$
  $\alpha_{-1} \leftarrow 1$
  $i \leftarrow 0$
  while not converge do
    $\mathbf{d}_{i+1}$ = Subgradient of (3.7) by using (3.8), (3.12) and (3.15).
    $\alpha_i$ = BacktrackingLineSearch($\boldsymbol{x}_i, \alpha_{i-1}$).
    $\boldsymbol{c}_{i+1} = \boldsymbol{c}_i + \alpha_i\boldsymbol{h}_i$.
    $\boldsymbol{h}_{i+1} = -\mathbf{d}_{i+1} + \frac{\mathbf{d}_{i+1}^T(\mathbf{d}_{i+1}-\mathbf{d}_i)}{\boldsymbol{h}_i^T(\mathbf{d}_{i+1}-\mathbf{d}_i)}$.
  end while
```

3.2.2 Experimental Setup

In our experiment, we test the algorithm using the disparity maps from Middlebury dataset [46]. The ground truth disparities and reconstructed results are shown in Figure 3.1 and Figure 3.2. The size of each disparity maps is 512×512, and the range of disparity values is [0-255]. Regarding the parameters, we use *"db2"* and *level*=2 in wavelet transform, and choose *nLevel* = [5, 6] for contourlet transform. In addition, the regularization parameters are $\lambda = 0.01$, $\gamma = 0.01$ and $\beta = 0.5$. We examine reconstruction performance by randomly selecting 5%, 10%, 15%, 20% and 25% sampling points. Two comparisons are presented in our experiment. Besides the model proposed in [2], we also consider the case that only uses contourlet transform. The PSNR and mean absolute error (MAE) are presented to evaluate the performance of the algorithms.

Given an estimated image $\hat{\boldsymbol{x}}$, ground truth image $\boldsymbol{x}$ and total number of pixels, N, the definition of MAE is:

$$\text{Mean Absolute Error} = \frac{1}{N}\sum_{i=1}^{N} |\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i|. \tag{3.22}$$



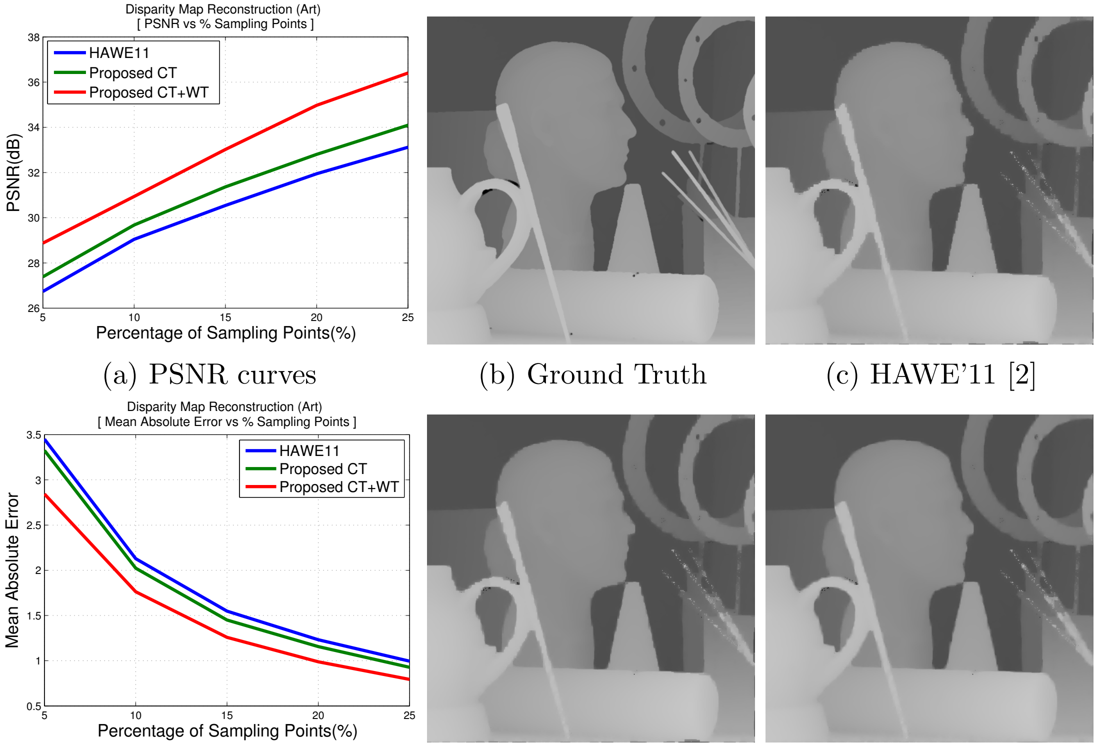
(a) PSNR curves (b) Ground Truth (c) HAWE'11 [2]

(d) MAE curves (e) Proposed CT (f) Proposed WT+CT

Figure 3.1: Reconstructed disparity maps from 10% random samples. MAE and PSNR curves, and snapshots of "Aloe" depth image.

3.2.3 Discussions

As shown in the Figure 3.1(a) and Figure 3.2(a), the proposed CT+WT method has the highest PSNR. Since the proposed CT+WT method utilizes both contourlet



(a) PSNR curves (b) Ground Truth (c) HAWE'11 [2]

(d) MAE curves (e) Proposed CT (f) Proposed WT+CT

Figure 3.2: Reconstructed disparity maps from 10% random samples. MAE and PSNR curves, and snapshots of "Art" depth image.

and wavelet transforms, local features and contours are reconstructed with high quality. Additionally, the proposed CT method has higher PSNR than HAWE'11 because the major structures of disparity maps are contours. Referring to MAE curves, the proposed CT+WT method outperforms the proposed CT and HAWE'11. As disparity maps correspond to depth information, the lower MAE infers better dense depth estimation performance. Thus, the proposed CT+WT method not only reconstructs disparity structures but also has less depth errors. While visually comparing object boundaries, the proposed CT+WT and proposed CT have smooth boundaries, whereas the staircase artifact along object boundaries exists in HAWE'11 method. In summary, the experiment shows that using combined wavelet and contourlet transforms yields better reconstruction performance than utilizing either wavelet or contourlet transform.

3.3 Algorithm: Alternating Direction Method of Multipliers

In this section we present an alternating direction method of multipliers (ADMM) algorithm to solve (1.11). ADMM algorithms can be traced back to the proximal operators proposed by Moreau in the 60's [47], and later studied by Eckstein and Bertsekas [48] in the 90's. The application of ADMM to image deconvolution was first mentioned in [49]. For brevity we skip the introduction of the ADMM algorithm because comprehensive tutorials are easily accessible [50, 51]. Instead, we highlight the unique contributions of this thesis, which includes a particular operator splitting strategy and a multiscale implementation.

For notational simplicity we consider a single dictionary so that $L = 1$. Generalization to $L > 1$ is straight forward. Also, in our derivation we focus on the anisotropic total variation so that $\|\boldsymbol{x}\|_{TV} = \|\boldsymbol{D}_x\boldsymbol{x}\|_1 + \|\boldsymbol{D}_y\boldsymbol{x}\|_1$. Extension to isotropic total variation follows the same idea as presented in [7].

3.3.1 ADMM and Operator Splitting

A central question about ADMM algorithms is which of the variables should be splitted so that the subsequent subproblems can be efficiently solved. Inspecting (1.11), we observe that there are many possible choices. For example, we could split the quadratic term in (1.11) by defining an auxiliary variable $\boldsymbol{u} = \boldsymbol{S}\boldsymbol{x}$, or we could keep the quadratic term without a split. In what follows, we present an overview of our proposed splitting method and discuss the steps in subsequent subsections.

We start the ADMM algorithm by introducing three auxiliary variables $\boldsymbol{r} = \boldsymbol{x}$, $\boldsymbol{u}_\ell = \boldsymbol{\Phi}_\ell \boldsymbol{x}$, and $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{x}$. Consequently, we rewrite the optimization problem as

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{r},\boldsymbol{u}_\ell,\boldsymbol{v}}{\text{minimize}} \quad & \tfrac{1}{2}\|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{r}\|^2 + \lambda_\ell \|\boldsymbol{W}_\ell \boldsymbol{u}_\ell\|_1 + \beta\|\boldsymbol{v}\|_1 \\ \text{subject to} \quad & \boldsymbol{r} = \boldsymbol{x}, \quad \boldsymbol{u}_\ell = \boldsymbol{\Phi}_\ell^T \boldsymbol{x}, \quad \boldsymbol{v} = \boldsymbol{D}\boldsymbol{x}. \end{aligned} \tag{3.23}$$

The ADMM algorithm is a computational procedure to find a stationary point of (3.23). The idea is to consider the augmented Lagrangian function defined as

$$\begin{aligned}
&\mathcal{L}\left(\boldsymbol{x}, \boldsymbol{u}_\ell, \boldsymbol{r}, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{y}_\ell, \boldsymbol{z}\right) \\
&\quad = \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{r}\|^2 + \lambda_\ell \|\boldsymbol{W}_\ell \boldsymbol{u}_\ell\|_1 + \beta \|\boldsymbol{v}\|_1 \qquad (3.24) \\
&\qquad - \boldsymbol{w}^T\left(\boldsymbol{r} - \boldsymbol{x}\right) - \boldsymbol{y}_\ell^T\left(\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{x}\right) - \boldsymbol{z}^T\left(\boldsymbol{v} - \boldsymbol{D}\boldsymbol{x}\right) \\
&\qquad + \frac{\mu}{2}\|\boldsymbol{r} - \boldsymbol{x}\|^2 + \frac{\rho_\ell}{2}\|\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{x}\|^2 + \frac{\gamma}{2}\|\boldsymbol{v} - \boldsymbol{D}\boldsymbol{x}\|^2.
\end{aligned}$$

In (3.24), the vectors $\boldsymbol{w}$, $\boldsymbol{y}_\ell$ and $\boldsymbol{z}$ are the Lagrange multipliers; λ_ℓ and β are the regularization parameters, and μ, ρ_ℓ and γ are the internal half quadratic penalty parameters. The stationary point of the augmented Lagrangian function can be determined by solving the following sequence of subproblems

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{u}_\ell^{(k)}, \boldsymbol{r}^{(k)}, \boldsymbol{v}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{y}_\ell^{(k)}, \boldsymbol{z}^{(k)}\right), \\
\boldsymbol{u}_\ell^{(k+1)} &= \underset{\boldsymbol{u}_\ell}{\operatorname{argmin}}\, \mathcal{L}\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}_\ell, \boldsymbol{r}^{(k)}, \boldsymbol{v}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{y}_\ell^{(k)}, \boldsymbol{z}^{(k)}\right), \\
\boldsymbol{r}^{(k+1)} &= \underset{\boldsymbol{r}}{\operatorname{argmin}}\, \mathcal{L}\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}_\ell^{(k+1)}, \boldsymbol{r}, \boldsymbol{v}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{y}_\ell^{(k)}, \boldsymbol{z}^{(k)}\right), \\
\boldsymbol{v}^{(k+1)} &= \underset{\boldsymbol{v}}{\operatorname{argmin}}\, \mathcal{L}\left(\boldsymbol{x}^{(k+1)}, \boldsymbol{u}_\ell^{(k+1)}, \boldsymbol{r}^{(k+1)}, \boldsymbol{v}, \boldsymbol{w}^{(k)}, \boldsymbol{y}_\ell^{(k)}, \boldsymbol{z}^{(k)}\right),
\end{aligned}$$

and the Lagrange multipliers are updated as

$$\boldsymbol{y}_\ell^{(k+1)} = \boldsymbol{y}_\ell^{(k)} - \rho_\ell\left(\boldsymbol{u}_\ell^{(k+1)} - \boldsymbol{\Phi}_\ell^T \boldsymbol{x}^{(k+1)}\right), \qquad (3.25a)$$

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \mu\left(\boldsymbol{r}^{(k+1)} - \boldsymbol{x}^{(k+1)}\right), \qquad (3.25b)$$

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - \gamma\left(\boldsymbol{v}^{(k+1)} - \boldsymbol{D}\boldsymbol{x}^{(k+1)}\right). \qquad (3.25c)$$

We now discuss how each subproblem is solved.

3.3.2 Subproblems

$\boldsymbol{x}$-subproblem

The $\boldsymbol{x}$-subproblem is obtained by dropping terms that do not involve $\boldsymbol{x}$ in (3.24). This yields

$$\begin{aligned}\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} & -\boldsymbol{w}^T(\boldsymbol{r}-\boldsymbol{x}) - \boldsymbol{y}_\ell^T\left(\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T\boldsymbol{x}\right) - \boldsymbol{z}^T(\boldsymbol{v}-\boldsymbol{D}\boldsymbol{x}) \\ & + \frac{\mu}{2}\|\boldsymbol{r}-\boldsymbol{x}\|^2 + \frac{\rho_\ell}{2}\|\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T\boldsymbol{x}\|^2 + \frac{\gamma}{2}\|\boldsymbol{v}-\boldsymbol{D}\boldsymbol{x}\|^2.\end{aligned} \tag{3.26}$$

Problem (3.26) can be solved by considering the first-order optimality condition, which yields a normal equation

$$\begin{aligned}&\left(\rho_\ell\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \mu\boldsymbol{I} + \gamma\boldsymbol{D}^T\boldsymbol{D}\right)\boldsymbol{x}^{(k+1)} \\ &\quad = \boldsymbol{\Phi}_\ell\left(\rho_\ell\boldsymbol{u}_\ell - \boldsymbol{y}_\ell\right) + (\mu\boldsymbol{r}-\boldsymbol{w}) + \boldsymbol{D}^T(\gamma\boldsymbol{v}-\boldsymbol{z}).\end{aligned} \tag{3.27}$$

The matrix in (3.27) can be simplified as $(\rho_\ell+\mu)\boldsymbol{I}+\gamma\boldsymbol{D}^T\boldsymbol{D}$, because for any frame $\boldsymbol{\Phi}_\ell$, it holds that $\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T = \boldsymbol{I}$. Now, since the matrix $\boldsymbol{D}^T\boldsymbol{D}$ is a circulant matrix, the matrix $(\rho_\ell+\mu)\boldsymbol{I}+\gamma\boldsymbol{D}^T\boldsymbol{D}$ is diagonalizable by the Fourier transform. This leads to a closed form solution as

$$\boldsymbol{x}^{(k+1)} = \mathcal{F}^{-1}\left[\frac{\mathcal{F}(\text{RHS})}{(\rho_\ell+\mu)\boldsymbol{I}+\gamma|\mathcal{F}(\boldsymbol{D})|^2}\right], \tag{3.28}$$

where RHS denotes the right hand side of (3.27), $\mathcal{F}(\cdot)$ denotes the 2D Fourier transform, $\mathcal{F}^{-1}(\cdot)$ denotes the 2D inverse Fourier transform, and $|\mathcal{F}(\boldsymbol{D})|^2$ denotes the magnitude square of the eigenvalues of the differential operator $\boldsymbol{D}$.

Remark 3. *If we do not split the quadratic function $\|\boldsymbol{b}-\boldsymbol{S}\boldsymbol{x}\|^2$ using $\boldsymbol{r}=\boldsymbol{x}$, then the identity matrix $\mu\boldsymbol{I}$ in* (3.27) *would become $\mu\boldsymbol{S}^T\boldsymbol{S}$. Since $\boldsymbol{S}$ is a diagonal matrix containing 1's and 0's, the matrix $\rho_\ell\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T+\mu\boldsymbol{S}^T\boldsymbol{S}+\gamma\boldsymbol{D}^T\boldsymbol{D}$ is not diagonalizable using the Fourier transform.*

$\boldsymbol{u}_\ell$-subproblem

The $\boldsymbol{u}_\ell$-subproblem is given by

$$\min_{\boldsymbol{u}_\ell} \quad \lambda_\ell \|\boldsymbol{W}_\ell \boldsymbol{u}_\ell\|_1 - \boldsymbol{y}_\ell^T \left(\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{x} \right) + \frac{\rho_\ell}{2} \|\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{x}\|^2. \tag{3.29}$$

Since $\boldsymbol{W}_\ell$ is a diagonal matrix, (3.29) is a separable optimization consisting of a sum of scalar problems. By using the standard shrinkage formula [7], one can show that the closed-form solution of (3.29) exists and is given by

$$\boldsymbol{u}_\ell^{(k+1)} = \max\left(\left| \boldsymbol{\alpha}_\ell + \frac{\boldsymbol{y}_\ell}{\rho_\ell} \right| - \frac{\lambda_\ell \tilde{\boldsymbol{w}}_\ell}{\rho_\ell}, 0 \right) \cdot \text{sign}\left(\boldsymbol{\alpha}_\ell + \frac{\boldsymbol{y}_\ell}{\rho_\ell} \right), \tag{3.30}$$

where $\tilde{\boldsymbol{w}}_\ell = \text{diag}(\boldsymbol{W}_\ell)$ and $\boldsymbol{\alpha}_\ell = \boldsymbol{\Phi}_\ell^T \boldsymbol{x}$.

Remark 4. *If we do not split using $\boldsymbol{u}_\ell = \boldsymbol{\Phi}_\ell^T \boldsymbol{x}$, then the $\boldsymbol{u}_\ell$-subproblem is not separable and hence the shrinkage formula cannot be applied. Moreover, if we split $\boldsymbol{u}_\ell = \boldsymbol{W}_\ell \boldsymbol{\Phi}_\ell^T \boldsymbol{x}$,* i.e., *include $\boldsymbol{W}_\ell$, then the $\boldsymbol{x}$-subproblem will contain $\boldsymbol{\Phi}_\ell \boldsymbol{W}_\ell \boldsymbol{\Phi}_\ell^T$, which is not diagonalizable using the Fourier transform.*

$\boldsymbol{r}$-subproblem

The $\boldsymbol{r}$-subproblem is the standard quadratic minimization problem:

$$\min_{\boldsymbol{r}} \ \frac{1}{2} \|\boldsymbol{S}\boldsymbol{r} - \boldsymbol{b}\|^2 - \boldsymbol{w}^T (\boldsymbol{r} - \boldsymbol{x}) + \frac{\mu}{2} \|\boldsymbol{r} - \boldsymbol{x}\|^2. \tag{3.31}$$

Taking the first-order optimality yields a normal equation

$$\left(\boldsymbol{S}^T \boldsymbol{S} + \mu \boldsymbol{I} \right) \boldsymbol{r} = \left(\boldsymbol{S}^T \boldsymbol{b} + \boldsymbol{w} + \mu \boldsymbol{x} \right). \tag{3.32}$$

Since $\boldsymbol{S}$ is a diagonal binary matrix, (3.32) can be evaluated via an element-wise computation.

Remark 5. (3.32) *shows that our splitting strategy of using $\boldsymbol{r} = \boldsymbol{x}$ is particularly efficient because $\boldsymbol{S}$ is a diagonal matrix. If $\boldsymbol{S}$ is a general matrix,* e.g., *i.i.d. Gaussian matrix in [52], then solving* (3.32) *will be less efficient.*

$\boldsymbol{v}$-subproblem

The $\boldsymbol{v}$-subproblem is the standard total variation problem:

$$\min_{\boldsymbol{v}} \ \beta\|\boldsymbol{v}\|_1 - \boldsymbol{z}^T\left(\boldsymbol{v} - \boldsymbol{D}\boldsymbol{x}\right) + \frac{\gamma}{2}\|\boldsymbol{v} - \boldsymbol{D}\boldsymbol{x}\|^2. \tag{3.33}$$

The solution is given by

$$\boldsymbol{v}^{(k+1)} = \max\left(\left|\boldsymbol{D}\boldsymbol{x} + \frac{\boldsymbol{z}}{\gamma}\right| - \frac{\beta}{\gamma}, 0\right) \cdot \operatorname{sign}\left(\boldsymbol{D}\boldsymbol{x} + \frac{\boldsymbol{z}}{\gamma}\right). \tag{3.34}$$

The overall ADMM algorithm is shown in Algorithm 3. For detailed derivations of solutions to subproblems, readers can refer to Appendix A.2.

Algorithm 3 ADMM Algorithm

```
Require: $\boldsymbol{b}$,$\boldsymbol{S}$
$\boldsymbol{x}^{(0)} = \boldsymbol{S}\boldsymbol{b}$, $\boldsymbol{u}_\ell^{(0)} = \boldsymbol{\Phi}_\ell^T\boldsymbol{x}^{(0)}$, $\boldsymbol{r}^{(0)} = \boldsymbol{x}^{(0)}$, $\boldsymbol{v}^{(0)} = \boldsymbol{D}\boldsymbol{x}^{(0)}$
while $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2/\|\boldsymbol{x}^{(k)}\|_2 \geq \text{tol}$ do
    Solve $\boldsymbol{x}$-subproblem by (3.28).
    Solve $\boldsymbol{u}_\ell$, $\boldsymbol{r}$ and $\boldsymbol{v}$ subproblems by (3.30), (3.32) and (3.34).
    Update multipliers by (3.25a), (3.25b) and (3.25c).
end while
return $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{(k+1)}$
```

3.3.3 Parameters

The regularization parameters (λ_ℓ, β) and internal half quadratic penalty parameters (ρ_ℓ, μ, γ) are chosen empirically. Table 3.1 provides a summary of the parameters we use in this paper. These values are the typical values we found over a wide range of images and testing conditions. For detailed experiments of the parameter selection process, we refer the readers to our supplementary technical report in [53] or Appendix A.3.

3.3.4 Convergence Comparison

Since (1.11) is convex, standard convergence proof of ADMM applies (c.f. [51]). Thus, instead of repeating the convergence theory, we compare our proposed algorithm

Table 3.1: Summary of Parameters.

Parameter	Functionality	Values
λ_1	Wavelet sparsity	4×10^{-5}
λ_2	Contourlet sparsity	2×10^{-4}
β	Total variation	2×10^{-3}
ρ_1	Half quad. penalty for Wavelet	0.001
ρ_2	Half quad. penalty for Contourlet	0.001
μ	Half quad. penalty for $\boldsymbol{r} = \boldsymbol{x}$	0.01
γ	Half quad. penalty for $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{x}$	0.1

with a subgradient algorithm proposed by Hawe et al. [2].

To set up the experiment, we consider the uniformly random sampling pattern $\boldsymbol{S}$ with sampling ratios $\xi = 0.1, 0.15, 0.2$. For both our algorithm and the subgradient algorithm proposed in [2], we consider a single wavelet dictionary using Daubechies wavelet "db2" with 2 decomposition levels. Other choices of wavelets are possible, but we observe that the difference is not significant.

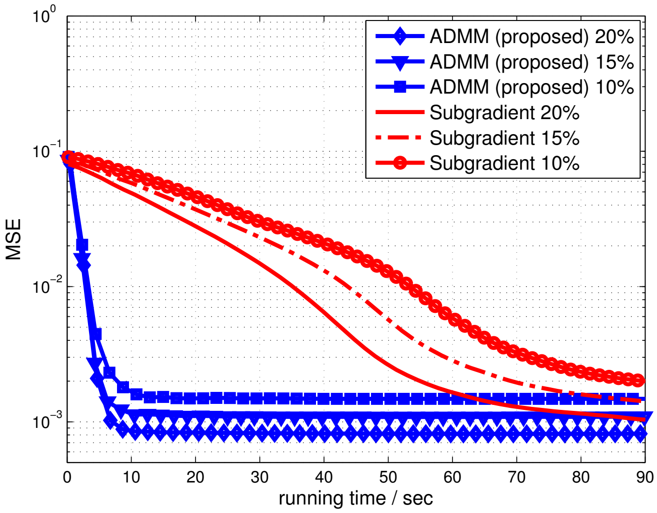


Figure 3.3: Comparison of the rate of convergence between ADMM (proposed) and subgradient algorithms [2] for single wavelet dictionary. We used "Aloe" as a test image. The ADMM algorithm requires approximately 10 seconds to reach steady state. The subgradient algorithm requires more than 9× running time than the ADMM algorithm to reach steady state.

Figure 3.3 shows the convergence results of our proposed algorithm and the subgradient algorithm. It is evident from the figure that the ADMM algorithm converges at a significantly faster rate than the subgradient algorithm. In particular, we see that the

ADMM algorithm reaches a steady state in around 10 seconds, whereas the subgradient algorithm requires more than 90 seconds.

3.3.5 Multiscale ADMM

The ADMM algorithm shown in Algorithm 3 can be modified to incorporate a multiscale warm start. The idea works as follows.

First, given the observed data $\boldsymbol{b}$, we construct a multiscale pyramid $\{\boldsymbol{b}_q \mid q = 0, \ldots, Q-1\}$ of Q levels, with a scale factor of 2 across adjacent levels. Mathematically, by assuming without loss of generality that N is a power of 2, we define a downsampling matrix $\boldsymbol{A}_q$ at the qth level as

$$\boldsymbol{A}_q = [\boldsymbol{e}_1, \boldsymbol{0}, \boldsymbol{e}_2, \boldsymbol{0}, \ldots, \boldsymbol{0}, \boldsymbol{e}_{N/2^q}],$$

where $\boldsymbol{e}_k$ is the kth standard basis. Then, we define $\boldsymbol{b}_q$ as

$$\boldsymbol{b}_q = \boldsymbol{A}_q \boldsymbol{b}_{q-1}, \tag{3.35}$$

for $q = 1, \ldots, Q-1$, and $\boldsymbol{b}_0 = \boldsymbol{b}$. Correspondingly, we define a pyramid of sampling matrices $\{\boldsymbol{S}_q \mid q = 0, \ldots, Q-1\}$, where

$$\boldsymbol{S}_q = \boldsymbol{A}_q \boldsymbol{S}_{q-1}, \tag{3.36}$$

with the initial sampling matrix $\boldsymbol{S}_0 = \boldsymbol{S}$.

The above downsampling operation allows us to solve (1.11) at different resolution levels. That is, for each $q = 0, \ldots, Q-1$, we solve the problem

$$\boldsymbol{x}_q = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{S}_q \boldsymbol{x} - \boldsymbol{b}_q\|_2^2 + \lambda_\ell \|\boldsymbol{W}_\ell \boldsymbol{\Phi}_\ell^T \boldsymbol{x}\|_1 + \beta \|\boldsymbol{x}\|_{TV}, \tag{3.37}$$

where $\boldsymbol{\Phi}_\ell$ and $\boldsymbol{W}_\ell$ are understood to have appropriate dimensions.

Once $\boldsymbol{x}_q$ is computed, we feed an upsampled version of $\boldsymbol{x}_q$ as the initial point to the $(q-1)$th level's optimization. More specifically, we define an upsampling and averaging operation:

$$\boldsymbol{B}_q = \left[\boldsymbol{e}_1^T;\, \boldsymbol{e}_1^T;\, \boldsymbol{e}_2^T;\, \boldsymbol{e}_2^T;\, \ldots;\, \boldsymbol{e}_{N/2^q}^T;\, \boldsymbol{e}_{N/2^q}^T\right], \tag{3.38}$$

and we feed $\boldsymbol{x}_q$, the solution at the qth level, as the initial guess to the problem at the $(q-1)$th level:

$$\boldsymbol{x}_{q-1}^{(0)} = \boldsymbol{B}_q \boldsymbol{x}_q. \tag{3.39}$$

A pictorial illustration of the operations of $\boldsymbol{A}_q$ and $\boldsymbol{B}_q$ is shown in Figure 3.4. The algorithm is shown in Algorithm 4.

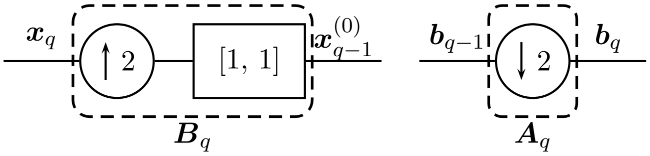


Figure 3.4: Schematic diagram showing the operations of $\boldsymbol{A}_q$ and $\boldsymbol{B}_q$: $\boldsymbol{A}_q$ downsamples the observed data $\boldsymbol{b}_q$ by a factor of 2; $\boldsymbol{B}_q$ upsamples the solution $\boldsymbol{x}_q$ by a factor of 2, followed by a two-tap filter of impulse response [1, 1].

Algorithm 4 Multiscale ADMM Algorithm

```
Require: $\boldsymbol{S}_0, \ldots, \boldsymbol{S}_{Q-1}$ and $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{Q-1}$
for $q = Q-1$ to $0$ do
    $\boldsymbol{x}_q$ = ADMM($\boldsymbol{b}_q, \boldsymbol{S}_q$) with initial guess $\boldsymbol{x}_q^{(0)}$
    Let $\boldsymbol{x}_{q-1}^{(0)} = \boldsymbol{B}_q \boldsymbol{x}_q$, if $q \geq 1$.
end for
Output $\boldsymbol{x} = \boldsymbol{x}_0$.
```

To validate the effectiveness of the proposed multiscale warm start, we compare the convergence rate against the original ADMM algorithm for a combined dictionary case. In Figure 3.5, we observe that the multiscale ADMM converges at a significantly faster rate than the original ADMM algorithm. More specifically, at a sampling ratio of 20%, the multiscale ADMM algorithm converges in 20 seconds whereas the original ADMM algorithm converges in 50 seconds which corresponds to a factor of 2.5 in runtime reduction. For fairness, both algorithms are tested under the same platform of MATLAB 2012b / 64-bit Windows 7 / Intel Core i7 / CPU 3.2GHz (single thread) / 12 GB RAM.

Remark 6. *When propagating the qth solution, $\boldsymbol{x}_q$, to the $(q-1)$th level, we should also propagate the corresponding auxiliary variables $\boldsymbol{u}_\ell$, $\boldsymbol{r}$, $\boldsymbol{v}$ and the Lagrange multipliers $\boldsymbol{y}_\ell$,*

$\boldsymbol{w}$ and $\boldsymbol{z}$. The auxiliary variables can be updated according to $\boldsymbol{x}_{q-1}^{(0)}$ as $\boldsymbol{u}_{\ell,q-1}^{(0)} = \boldsymbol{\Phi}_\ell \boldsymbol{x}_{q-1}^{(0)}$, $\boldsymbol{r}_{q-1}^{(0)} = \boldsymbol{x}_{q-1}^{(0)}$, and $\boldsymbol{v}_{q-1}^{(0)} = \boldsymbol{D}\boldsymbol{x}_{q-1}^{(0)}$. For the Lagrange multipliers, we let $\boldsymbol{y}_{\ell,q-1}^{(0)} = \boldsymbol{B}_q \boldsymbol{y}_{\ell,q}$, $\boldsymbol{w}_{q-1}^{(0)} = \boldsymbol{B}_q \boldsymbol{w}_q$, and $\boldsymbol{z}_{q-1}^{(0)} = \boldsymbol{B}_q \boldsymbol{z}_q$.

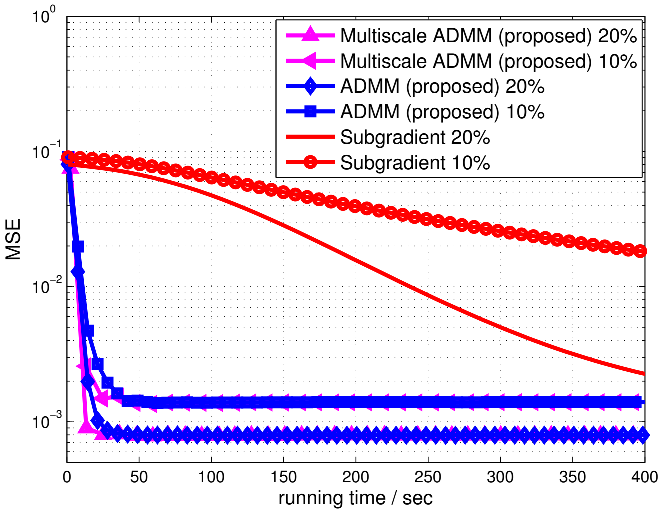


Figure 3.5: Runtime comparison of original ADMM algorithm, multiscale ADMM algorithm and subgradient algorithm. All algorithms use the combined wavelet-contourlet dictionary. The testing image is "Aloe" and two sampling ratios (10% and 20%) are tested. $Q = 3$ multiscale levels are implemented in this experiment.

Remark 7. *The choice of the up/down sampling factor is not important. In our experiment, we choose a factor of 2 for simplicity in implementation. Other sampling factors such as $\sqrt{2}$ are equally applicable. Furthermore, the two-tap average filter $[1, 1]$ in Figure 3.4 can be replaced by any valid averaging filter. However, experimentally we find that other choices of filters do not make a significant difference comparing to $[1, 1]$.*

This Chapter includes materials that have been published in IEEE Transaction on Image Processing 2015, titled "Depth Reconstruction from Sparse Samples: Representation, Algorithm, and Sampling," with Truong Q. Nguyen and Stanley H. Chan, and in IEEE International Conference on Acoustics, Speech and Signal Processing 2014, titled "Sparse Reconstruction for Disparity Maps using Combined Wavelet and Contourlet Transforms," with Truong Q. Nguyen.

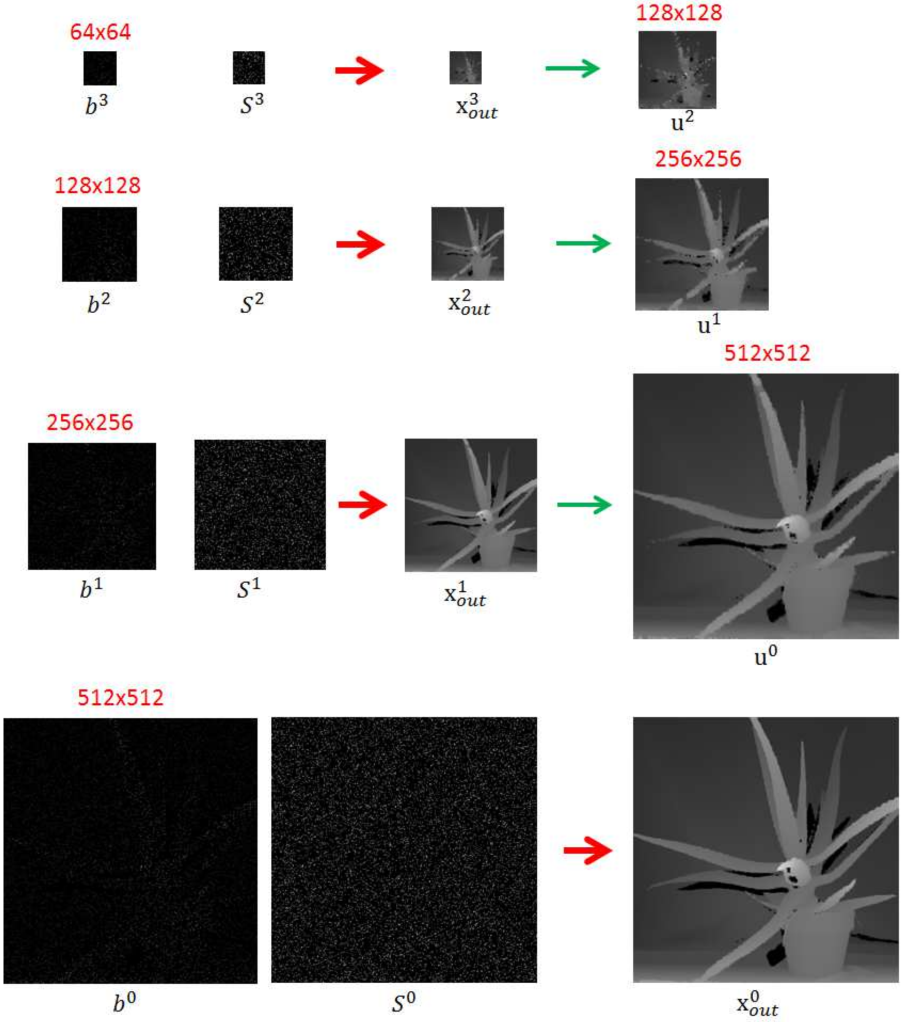


Figure 3.6: Multilevel Scheme. The red arrow represents ADMM dense disparity reconstruction process, and the green arrow stands for the upsampling process with nearest neighbor interpolation.

Chapter 4

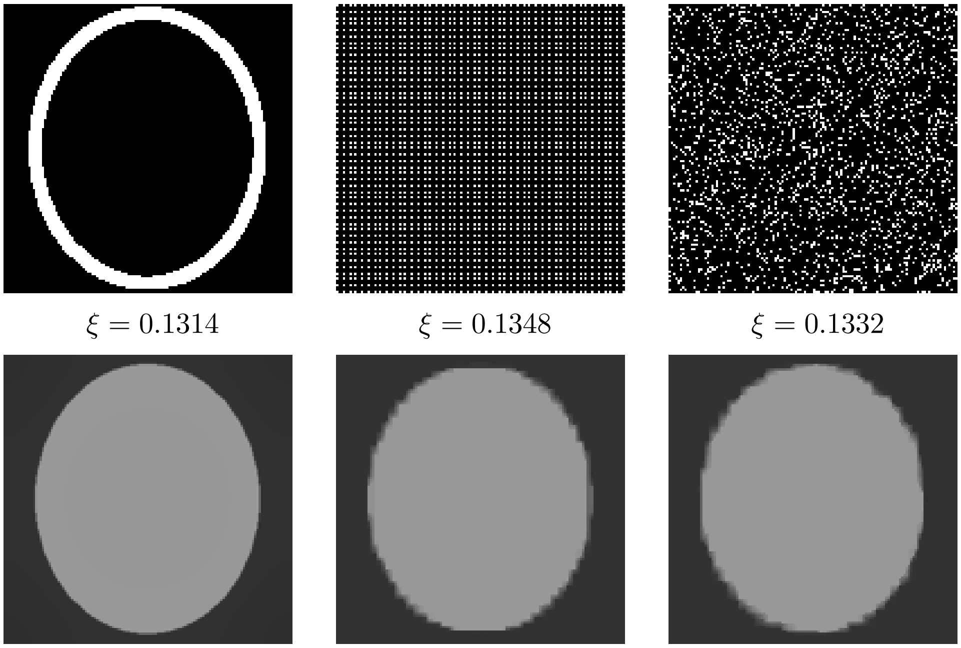
Sparse Sampling for Depth Data

In the above Chapters, we assume that the sampling matrix $\boldsymbol{S}$ is given and is fixed. However, we have not yet discussed the design of the sampling probability $\{p_j\}_{j=1}^N$. The purpose of this section is to present an efficient design procedure.

4.1 Motivating Example

Before our discussion, perhaps we should first ask about what kind of sampling matrix $\boldsymbol{S}$ would work (or would not work). To answer this question, we consider an example shown in Figure 4.1. In Figure 4.1 we try to recover a simple disparity map consisting of an ellipse of constant intensity and a plain background. We consider three sampling patterns of approximately equal sampling ratios ξ: (a) a sampling pattern defined according to the magnitude of the disparity gradient; (b) an uniform grid with specified sampling ratio $\sqrt{\xi}$ along both directions; (c) a random sampling pattern drawn from an uniform distribution with probability ξ. The three sampling patterns correspondingly generate three sampled disparity maps. For each sampled disparity map, we run the proposed ADMM algorithm and record the reconstructed disparity map. In all experiments, we use a wavelet dictionary for demonstration.

The results in Figure 4.1 suggest a strong message: For a fixed sampling budget ξ, one should pick samples along gradients. However, the pitfall is that this approach is not practical for two reasons. First, the gradient of the disparity map is not available



(a) 45.527dB (b) 29.488dB (c) 30.857dB

Figure 4.1: Three sampling patterns and the corresponding reconstruction results using the proposed ADMM algorithm. Here, ξ denotes the actual sampling ratio. (a) Sampling along the gradient; (b) Sampling from a grid; (c) Sampling from an uniformly random pattern.

prior to reconstructing the disparity. Therefore, all gradient information can only be inferred from the color image. Second, the gradients of a color image could be very different from the gradients of the corresponding disparity map. Thus, inferring the disparity gradient from the color image gradient is a challenging task. In the followings, we present a randomized sampling scheme to address these two issues.

4.2 Oracle Random Sampling Scheme

We first consider an oracle situation where the gradients are assumed *known.* The goal is to see how much improvement one should expect to see.

Let $\boldsymbol{a} = [a_1, \ldots, a_N]^T$ be a vector denoting the magnitude of the ground truth disparity map's gradient. Given this oracle information about the disparity gradients, we consider a soft decision rule where a pixel is sampled with probability defined according to some function of $\{a_j\}_{j=1}^N$. Such a function is chosen based on the intuition that the sampled subset of gradients should carry the maximum amount of information compared to the full set of gradients. One way to capture this intuition is to require that the average gradient computed from *all* N samples is similar to the average gradient computed from a *subset* of ξN samples.

To be more precise, we define the average gradient computed from all N samples as

$$\mu \overset{\text{def}}{=} \frac{1}{N}\sum_{j=1}^{N} a_j. \tag{4.1}$$

Similarly, we define the average gradient computed from a random subset of ξN samples as

$$Y \overset{\text{def}}{=} \frac{1}{N}\sum_{j=1}^{N} \frac{a_j}{p_j} I_j, \tag{4.2}$$

where $\{I_j\}_{j=1}^{N}$ is a sequence of Bernoulli random variables with probability $\Pr[I_j = 1] = p_j$. Here, the division of a_j by p_j is to ensure that Y is unbiased, *i.e.*, $\mathbb{E}[Y] = \mu$.

From (4.1) and (4.2), minimizing the difference between Y and μ can be achieved by minimizing the variance $\mathbb{E}[(Y-\mu)^2]$. Moreover, we observe that

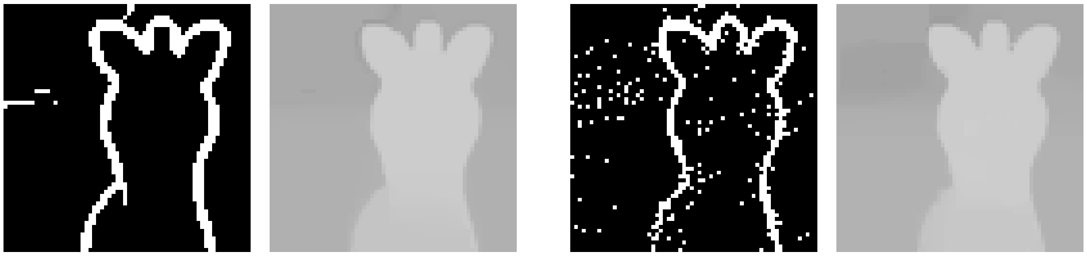
$$\mathbb{E}\left[(Y-\mu)^2\right] = \frac{1}{N}\sum_{j=1}^{N} \frac{a_j^2}{p_j^2}\mathrm{Var}\left[I_j\right] = \frac{1}{N}\sum_{j=1}^{N} a_j^2\left(\frac{1-p_j}{p_j}\right),$$

where the last equality holds because $\mathrm{Var}[I_j] = p_j(1-p_j)$. Therefore, the optimal sampling probability $\{p_j\}_{j=1}^{N}$ can be found by solving the optimization problem

$$\begin{aligned} (P): \qquad & \underset{p_1,\ldots,p_N}{\text{minimize}} \quad && \frac{1}{N}\sum_{j=1}^{N}\frac{a_j^2}{p_j} \\ & \text{subject to} && \frac{1}{N}\sum_{j=1}^{N} p_j = \xi, \text{ and } 0 \le p_j \le 1, \end{aligned}$$

of which the solution is given by [54, Lemma 2]

$$p_j = \min(\tau a_j, 1), \tag{4.3}$$



(a) Greedy sampling
35.5201 dB, $\xi = 0.1157$

(b) Random sampling
39.8976 dB, $\xi = 0.1167$

Figure 4.2: Comparison between a deterministic sampling pattern by selecting samples greedily according to the magnitude of $\{a_j\}$, and a randomized sampling pattern using the proposed scheme.

where τ is the root of the equation

$$g(\tau) \overset{\text{def}}{=} \sum_{j=1}^{N} \min(\tau a_j, 1) - \xi N. \tag{4.4}$$

It is interesting to compare this new random sampling scheme versus a greedy sampling scheme by picking the ξN pixels with the largest gradients. Figure 4.2 shows the result. For the greedy sampling scheme, we first compute the gradient of the disparity map $\nabla \boldsymbol{x} \overset{\text{def}}{=} \sqrt{(\boldsymbol{D}_x \boldsymbol{x})^2 + (\boldsymbol{D}_y \boldsymbol{x})^2}$ and threshold it to obtain a set of samples $\Omega \overset{\text{def}}{=} \{j \mid [\nabla \boldsymbol{x}]_j > \alpha \|\nabla \boldsymbol{x}\|_\infty\}$, where $\alpha = 0.1$ is the threshold. The actual sampling ratio is then $|\Omega|/N$. For the randomized scheme, we let $\boldsymbol{a} = \nabla \boldsymbol{x}$ and we compute p_j according to (4.3). In this particular example, we observe that the randomized sampling scheme achieves a PSNR improvement of more than 4 dB.

4.2.1 Practical Random Sampling Scheme

We now present a practically implementable sampling scheme. The challenge that we have to overcome is that the gradient information of the disparity is not available. Therefore, we propose the following two-stage sampling process.

Our proposed sampling scheme consists of two stages - a pilot stage to obtain a rough estimate of the disparity, and a refinement stage to improve the disparity estimate. In the first step pilot stage, we pick $\xi N/2$ samples (*i.e.*, half of the desired number of samples) using an uniformly random sampling pattern. This gives a sampling pattern $\{I_j^{(1)}\}_{j=1}^N$, where the superscript denotes the first stage. Correspondingly, we have a sampling matrix $\boldsymbol{S}^{(1)}$ and the sampled data $\boldsymbol{b}^{(1)}$. Given $\boldsymbol{S}^{(1)}$ and $\boldsymbol{b}^{(1)}$, we apply the ADMM algorithm to obtain a pilot estimate $\boldsymbol{x}^{(1)}$.

In the second stage, we use the pilot estimate $\boldsymbol{x}^{(1)}$ as a guide to compute the gradient $\nabla\boldsymbol{x}^{(1)}$. By (4.3), this suggests that the optimal sampling probability is $p_j = \min(\tau[\nabla\boldsymbol{x}^{(1)}]_j, 1)$. However, in order to ensure that the $\xi N/2$ samples picked at the second stage *do not overlap* with those picked in the first stage, instead of letting $p_j = \min(\tau[\nabla\boldsymbol{x}^{(1)}]_j, 1)$, we let $p_j = \min(\tau a_j, 1)$, where

$$a_j = \begin{cases} [\nabla\boldsymbol{x}^{(1)}]_j, & \text{if} \quad I_j^{(1)} = 0, \\ 0, & \text{if} \quad I_j^{(1)} = 1. \end{cases} \tag{4.5}$$

In other words, a_j defined by (4.5) forces $p_j = 0$ when the jth pixel is picked in the first step. Thus, the non-zero entries of $\{I_j^{(1)}\}$ and $\{I_j^{(2)}\}$ are mutually exclusive, and hence we can now apply the ADMM algorithm to recover $\boldsymbol{x}^{(2)}$ from $\boldsymbol{S}_1 + \boldsymbol{S}_2$ and $\boldsymbol{b}_1 + \boldsymbol{b}_2$. The overall method is summarized in Algorithm 5.

Algorithm 5 Two-Stage Algorithm

```
Input: $N$, $\xi$, $\boldsymbol{b}$
Output: $\boldsymbol{x}^{(2)}$

Stage 1:
    Let $I_j^{(1)} = 1$ with probability $\xi/2$, for $j = 1, \ldots, N$.
    Define $\boldsymbol{S}^{(1)}$ and $\boldsymbol{b}^{(1)}$ according to $\{I_j^{(1)}\}$.
    Compute $\boldsymbol{x}^{(1)} = \text{ADMM}\ (\boldsymbol{S}^{(1)}, \boldsymbol{b}^{(1)})$.

Stage 2:
    Compute $\nabla\boldsymbol{x}^{(1)}$.
    For $j = 1, \ldots, N$, define $a_j = \begin{cases} [\nabla\boldsymbol{x}^{(1)}]_j, & \text{if } I_j^{(1)} = 0, \\ 0, & \text{if } I_j^{(1)} = 1. \end{cases}$.
    Compute $\tau$ such that $\sum_{j=1}^N \min\{\tau a_j,\ 1\} = N\xi/2$.
    Let $p_j = \min\{\tau a_j,\ 1\}$, for $j = 1, \ldots, N$.
    Let $I_j^{(2)} = 1$ with probability $p_j$, for $j = 1, \ldots, N$.
    Define $\boldsymbol{S}^{(2)}$ and $\boldsymbol{b}^{(2)}$ according to $\{I_j^{(2)}\}$.
    Compute $\boldsymbol{x}^{(2)} = \text{ADMM}\ (\boldsymbol{S}^{(1)} + \boldsymbol{S}^{(2)}, \boldsymbol{b}^{(1)} + \boldsymbol{b}^{(2)})$.
```

4.3 Further Improvement by PCA

The two-stage sampling procedure can be further improved by utilizing the prior information of the color image. The intuition is that since both color image and disparity map are captured from the same scene, strong gradients in the disparity map should align with those in the color image. However, since a color image typically contains complex gradients which are irrelevant to the disparity reconstruction, it is important to filter out these unwanted gradients while preserving the important ones. To this end, we consider the following patch-based principal component analysis.

Given a color image $\boldsymbol{y} \in \mathbb{R}^N$, we construct a collection of patches $\{\boldsymbol{y}_j\}_{j=1}^N$ where $\boldsymbol{y}_j \in \mathbb{R}^d$ denotes a vectorization of the jth patch of size $\sqrt{d} \times \sqrt{d}$ centered at pixel j of the image. For patches centered at the corners or boundaries of the image, we apply a symmetrical padding to make sure that their sizes are $\sqrt{d} \times \sqrt{d}$. This will give us a total of N patches.

Next, we form a data matrix $\boldsymbol{Y} \overset{\text{def}}{=} [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$. This data matrix leads to a principal component decomposition as

$$\boldsymbol{Y}\boldsymbol{Y}^T = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T, \tag{4.6}$$

where $\boldsymbol{U}$ is the eigenvector matrix, and $\boldsymbol{\Lambda}$ is the eigenvalue matrix. Given $\boldsymbol{U}$, we can compute $\boldsymbol{u}_i^T \boldsymbol{y}_j$, *i.e.*, the projection of a patch $\boldsymbol{y}_j$ onto the subspace spanned by an eigenvector $\boldsymbol{u}_i$. If we view $\boldsymbol{u}_i$ as a finite impulse response filter, then the projection is equivalent to a filtering.

The structure of the filters deserves a closer look. In Figure 4.3 we show the 6 leading eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_6$. It can be seen that except for the first eigenvector $\boldsymbol{u}_1$ (which is a constant vector), all remaining eigenvectors $\boldsymbol{u}_2, \ldots, \boldsymbol{u}_d$ are in the form of differential operators, with different orders and orientations. Moreover, these filters are *bandpass* filters, which suggests that various gradients of the color image can be extracted by the filtering. Consequently, if one would like to extract major gradients while rejecting gradients of the textures, the following filtered signal can be considered:

$$a_j = \sum_{i=2}^{d'} |\boldsymbol{u}_i^T \boldsymbol{y}_j|, \quad j = 1, \ldots, N, \tag{4.7}$$

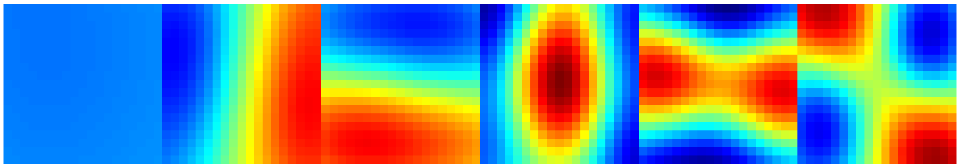


Figure 4.3: The first 6 eigenvectors of the data matrix $\boldsymbol{Y}\boldsymbol{Y}^T$, where $\boldsymbol{Y}$ is obtained from the color image corresponding to Figure 4.2. In this example we set the patch size as 19×19 so that $d = 361$. The range of the color index of this figure is $[-0.1,\ 0.1]$.

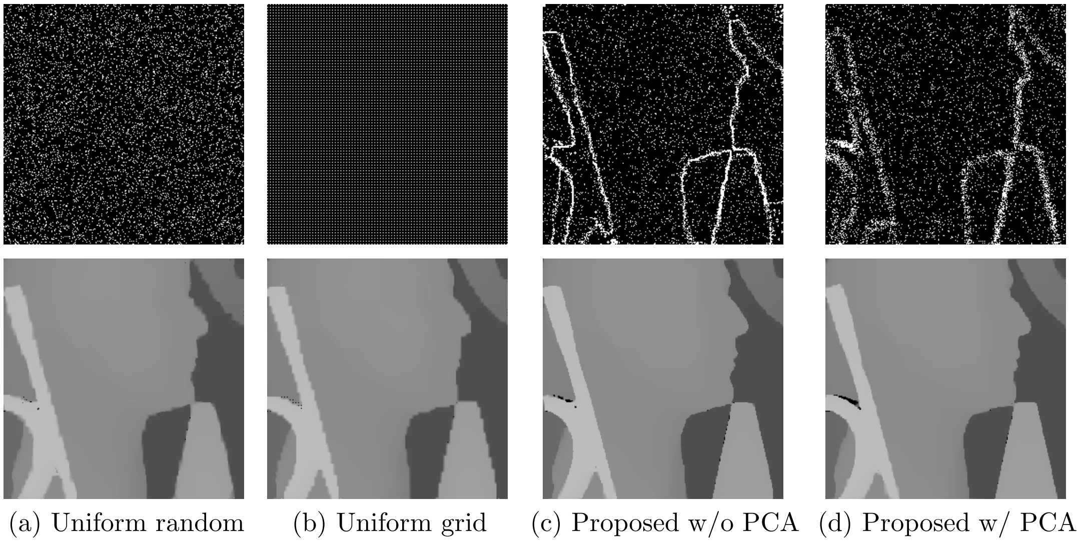
where $d' < d$ is a tunable parameter (which was set to $d' = 16$ for $d = 49$ in this thesis). Here, the absolute value in (4.7) is used to get the magnitude of $\langle \boldsymbol{u}_i, \boldsymbol{y}_j \rangle$, as a_j must be a non-negative number.

To see how this PCA concept can be incorporated into our two-stage sampling scheme, we make the following observations. First, the uniform sampling in Stage-1 can well be replaced by the PCA approach. In particular, instead of setting $I_j^{(1)} = 1$ with probability $\xi/2$, we can define a_j according to (4.7), and let $p_j = \min(\tau a_j, 1)$ for τ being the root of (4.4). Consequently, we let $I_j^{(1)} = 1$ with probability p_j.

In Stage-2, since we have already had a pilot estimate of the disparity map, it is now possible to replace $\boldsymbol{Y}$ in (4.6) by a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_N^{(1)}]$, where each $\boldsymbol{x}_j^{(1)}$ is a d-dimensional patch centered at the jth pixel of $\boldsymbol{x}^{(1)}$. Thus, instead of setting $a_j = [\nabla \boldsymbol{x}^{(1)}]_j$ in (4.5), we can set $a_j = \sum_{i=2}^{d'} |\langle \boldsymbol{u}_i, \boldsymbol{x}_j^{(1)} \rangle|$ using (4.7). The advantage of this new a_j is that it softens the sampling probability at the object boundaries to a neighborhood surrounding the boundary. This reduces the risk of selecting irrelevant samples because of a bad pilot estimate.

4.4 Comparisons

As a comparison between sampling patterns, we consider a disparity map shown in Figure 4.4. Setting $\xi = 0.1$ (*i.e.*, 10%), we study four sampling patterns including two versions of our proposed two-stage method. We conduct a Monte-Carlo simulation by repeating 32 independent trials, and average the PSNRs. The results shown in Figure 4.4(c) are generated using the original two-stage sampling scheme without PCA improvement, whereas the results shown in Figure 4.4(d) are generated using an improved two-stage sampling scheme where the first stage is uniform and the second stage is PCA. These results indicate that for the same sampling ratio ξ, the choice of the sampling pattern has some strong influence to the reconstruction quality. For example, as compared to both uniform random sampling and grid sampling, the original two-stage sampling has about 2.44 dB improvement, and can be further improved by almost 3.76 dB using the PCA idea.


(a) Uniform random (b) Uniform grid (c) Proposed w/o PCA (d) Proposed w/ PCA

Method	Actual Sampling Ratio	Average PSNR / dB	Standard Deviation
Uniform random	0.1001	29.7495	0.3768
Uniform grid	0.1128	30.2726	0.0000
Proposed w/o PCA	0.1000	32.4532	0.8962
Proposed w/ PCA	0.1002	**33.7707**	1.0435

Figure 4.4: Comparison between four sampling patterns. (a) Uniformly random sampling pattern; (b) Uniform grid; (c) Proposed two-stage sampling without PCA improvement; (d) Proposed two-stage sampling with PCA improvement. For the two-stage sampling in (c)-(d), we pick $\xi N/2$ uniformly random samples in stage 1, and pick the remaining $\xi N/2$ samples according to the pilot estimate from Stage 1. We conduct a Monte-Carlo simulation with 32 independent trials. The averages of PSNRs are presented in the Table.

4.5 Experimental Results

In this section we present additional results to illustrate the performance of the proposed method.

4.5.1 Synthetic Data

We first compare the proposed algorithm with existing methods on the Middlebury dataset[46], where ground truth disparities are available. We consider two versions of the proposed algorithm: "Proposed WT+CT Grid" and "Proposed WT+CT 2-Stage". "Proposed WT+CT Grid" is the ADMM algorithm presented in Section IV using both wavelet and contourlet bases. Here, "Grid" refers to using a deterministic uniform grid sampling pattern and "2-stage" refers to using the 2-stage randomized sampling scheme presented in Section V. We use Daubechies wavelet "db2" with 2 decomposition levels for wavelet dictionary, and we set "bior9-7" wavelet function with [5 6] directional decompositions for contourlet dictionary.

We also compare our method with [2], which has three differences from ours: (1) [2] uses a subgradient descent algorithm whereas we use an ADMM algorithm; (2) [2] considers only a wavelet basis whereas we consider a combined wavelet-contourlet basis; (3) [2] uses a combination of canny edges and uniformly random samples whereas we use a principled design process to determine samples.

In this experiment we do not compare with depth super resolution algorithms, *e.g.*, [21, 55, 56]. These methods require a color image to guide the actual reconstruction process, which is different from what is presented here because we only use the color image for designing the sampling pattern. As a reference of these methods, we show the results of a bicubic interpolation using uniform grid sampling pattern.

Table 4.1 shows the PSNR values of various methods at different sampling ratios and sampling methods. It is clear that "Proposed WT+CT 2-Stage" outperforms the other methods by a significant margin. Moreover, as the sampling ratio increases, the PSNR gain of "Proposed WT+CT 2-Stage" is more prominent than that of other methods. For example, increasing from 5% to 25% for "Art", "Proposed WT+CT 2-Stage" demonstrates an 18 dB PSNR improvement whereas bicubic only demonstrates 3 dB improvement.

It is also instructive to compare the percentage of bad pixels (% Bad Pixel), which is a popular metric to measure the quality of disparity estimates [57]. Given a threshold $\tau > 0$, the percentage of bad pixels is defined as

$$\% \text{ Bad Pixel} \overset{\text{def}}{=} \frac{1}{N}\sum_{j=1}^{N}\left(|\widehat{x}_j - x_j^*| > \tau\right), \tag{4.8}$$

where $\widehat{\boldsymbol{x}}$ is the reconstructed disparity and $\boldsymbol{x}^*$ is the ground truth disparity. Percentage of bad pixels can be considered as an absolute difference metric as compared to the mean squared metric of PSNR.

Table 4.2 shows the percentage of bad pixels of various methods at different sampling ratios and sampling methods. The results indicate that "Proposed WT+CT 2-Stage" has a relatively higher % Bad Pixel at $\tau = 1$ than other methods, but has a lower % Bad Pixel at $\tau = 2$ and $\tau = 3$. This result suggests that most of the errors of "Proposed WT+CT 2-Stage" are *small* and there are very few outliers. In contrast, bicubic grid (for example) has a low % Bad Pixel at $\tau = 1$ but high % Bad Pixel at $\tau = 2$ and $\tau = 3$. This implies that a significant portion of the bicubic results has large error. Intuitively, the results suggest that in the bicubic case, some strong edges and corners are completely missed, whereas these information are kept in "Proposed WT+CT 2-Stage".

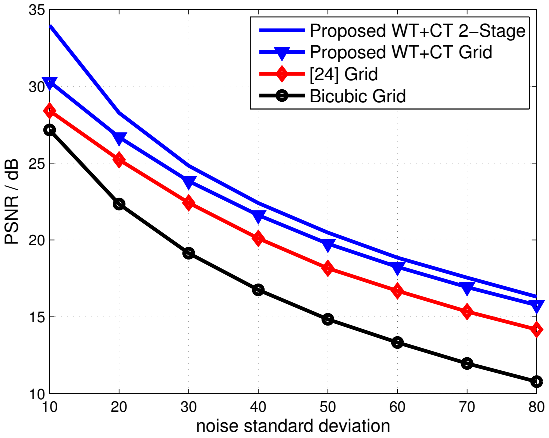


Figure 4.5: Comparison of reconstruction performance with noisy samples. We use "Art" disparity map as a test image, and set $\xi = 0.2$.

Finally, we show the performance of the proposed algorithm towards additive i.i.d. Gaussian noise. The purpose of this experiment is to demonstrate the sensitivity and robustness of the algorithm in the presence of noise. While in reality the noise in disparity estimates is not i.i.d. Gaussian, the result presented here serves as a reference for the algorithm's performance. A more realistic experiment on real data will be illustrated in the next subsection.

The results are shown in Figure 4.5. Using "Bicubic Grid" as the baseline, we observe that "Proposed WT+CT 2-Stage" on average has 5.79 dB improvement, "Proposed WT+CT Grid" has 3.60 dB improvement, whereas "[2] Grid" has only 3.02 dB improvement. This provides a good indicator of the robustness of the proposed methods.

Table 4.1: Comparisons of reconstruction algorithms in terms of PSNR. We put N/A when the algorithm does not converge in 1000 iterations.

Disparity	Method	Percentage of Samples / PSNR (dB)				
Name	Algorithm / Sampling Strategy	5%	10%	15%	20%	25%
Aloe	Proposed WT+CT 2-Stage	27.5998	**31.3877**	**33.3693**	**36.4102**	**38.6265**
	Proposed WT+CT Grid	25.3236	28.9052	30.0940	31.2956	32.3548
	[2] Grid	25.1248	27.8941	28.9504	30.2371	31.6646
	Bicubic Grid	**27.8899**	29.3532	30.1019	31.0031	31.8908
Art	Proposed WT+CT 2-Stage	**30.8669**	**34.1495**	**37.2801**	**42.9706**	**48.0002**
	Proposed WT+CT Grid	27.5176	28.9528	30.8371	32.5150	33.7126
	[2] Grid	27.0300	N/A	N/A	N/A	N/A
	Bicubic Grid	29.1550	30.3536	31.1098	31.9473	32.8366
Baby	Proposed WT+CT 2-Stage	**39.6978**	**44.8958**	**48.6631**	**52.5000**	**52.0031**
	Proposed WT+CT Grid	34.4421	36.7965	37.6708	39.0504	40.0689
	[2] Grid	33.6627	35.3166	36.2522	37.4513	38.7670
	Bicubic Grid	34.8368	36.2385	37.1749	37.5973	38.3961
Dolls	Proposed WT+CT 2-Stage	**29.5087**	**32.5336**	**33.9974**	**36.2741**	**37.6527**
	Proposed WT+CT Grid	28.4858	29.0453	30.0949	30.8123	31.6725
	[2] Grid	28.4959	N/A	N/A	N/A	32.0521
	Bicubic Grid	29.0612	30.0475	30.4374	31.0053	31.8800
Moebius	Proposed WT+CT 2-Stage	**31.0663**	**35.1060**	**37.7626**	**39.9225**	**41.8933**
	Proposed WT+CT Grid	27.6882	28.7245	29.8527	31.1663	32.2399
	[2] Grid	27.6851	28.7973	N/A	N/A	32.0990
	Bicubic Grid	28.3987	29.9338	30.6607	30.9427	32.0143
Rocks	Proposed WT+CT 2-Stage	**30.7662**	**35.3975**	**37.5056**	**40.4494**	**42.5089**
	Proposed WT+CT Grid	25.5924	29.0848	30.4766	31.2311	32.9218
	[2] Grid	25.4444	28.7973	29.5364	30.2058	32.1672
	Bicubic Grid	28.7241	30.4212	30.7552	31.6722	32.6706

Table 4.2: Comparisons of reconstruction algorithms in terms of % Bad Pixel.

	Method	% of Bad Pixels [$\tau = 1$]					% of Bad Pixels [$\tau = 2$]					% of Bad Pixels [$\tau = 3$]				
Disparity	Algorithm	Percentage of Samples					Percentage of Samples					Percentage of Samples				
Name	Sampling Strategy	5%	10%	15%	20%	25%	5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
Aloe	Prop. WT+CT 2-Stage	41.47	21.37	14.00	8.85	5.81	**20.03**	**7.15**	**3.70**	**1.99**	**1.11**	**13.42**	**4.80**	**2.52**	**1.43**	**0.79**
	Prop. WT+CT Grid	36.88	22.96	15.61	11.62	8.69	21.16	10.11	6.87	5.17	3.92	15.80	7.62	5.55	4.25	3.27
	[2] Grid	31.44	**17.65**	**11.58**	**8.39**	**5.79**	20.12	8.87	6.08	4.73	3.56	14.73	6.97	5.03	4.01	3.09
	Bicubic Grid	**31.23**	23.37	18.62	15.88	13.39	23.51	17.49	13.78	11.96	10.04	19.40	14.47	11.51	9.96	8.30
Baby	Prop. WT+CT 2-Stage	28.00	12.37	5.72	**2.67**	**1.31**	9.95	**2.31**	**0.39**	**0.13**	**0.07**	**3.69**	**0.64**	**0.16**	**0.03**	**0.01**
	Prop. WT+CT Grid	15.80	8.27	5.80	4.12	3.14	**6.31**	3.01	2.22	1.58	1.22	4.25	2.30	1.77	1.31	1.05
	[2] Grid	12.31	**6.02**	**3.93**	2.71	1.86	6.44	2.73	1.94	1.47	1.10	4.21	2.09	1.55	1.21	0.93
	Bicubic Grid	**12.22**	8.53	6.54	5.59	4.58	7.89	5.63	4.34	3.73	3.10	6.24	4.42	3.51	3.00	2.41
Rocks	Prop. WT+CT 2-Stage	25.90	10.67	6.27	**3.55**	**2.19**	8.26	**2.26**	**0.93**	**0.41**	**0.22**	**4.75**	**1.22**	**0.51**	**0.21**	**0.10**
	Prop. WT+CT Grid	20.67	11.74	8.03	5.79	4.44	**7.64**	4.12	2.93	2.34	1.72	5.16	3.01	2.27	1.88	1.43
	[2] Grid	16.36	**9.09**	**6.10**	4.38	3.00	8.33	4.02	2.86	2.24	1.62	5.52	2.93	2.19	1.76	1.26
	Bicubic Grid	**15.32**	11.51	9.36	7.88	6.59	10.20	7.95	6.46	5.26	4.61	8.28	6.51	5.24	4.42	3.76

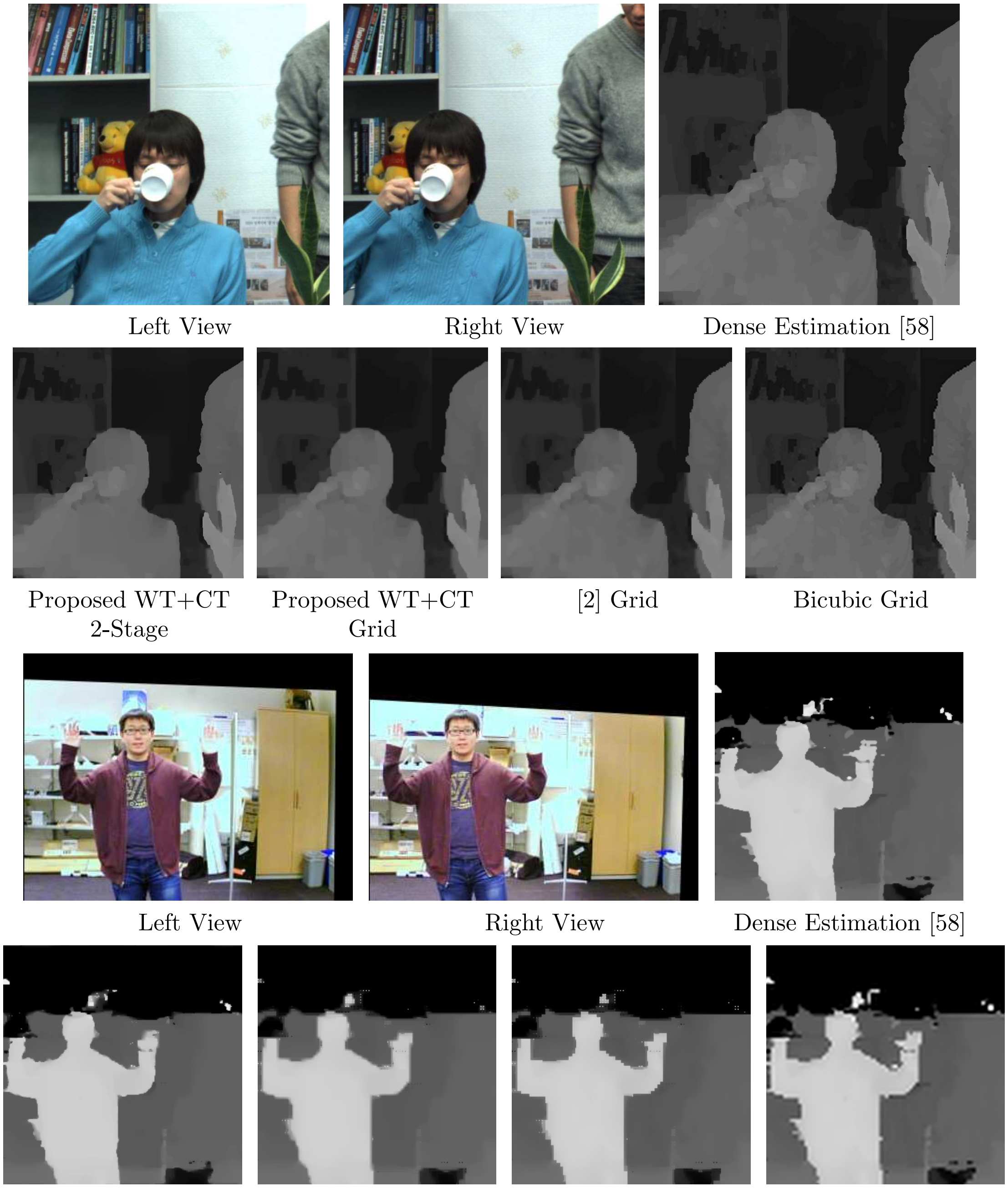
4.5.2 Real Data

In this experiment we study the performance of the proposed algorithm for real data. Figure 4.6 shows snapshots of a stereo video (with resolution 320 × 240, 30 fps). For this video sequence, we apply the block matching algorithm by Lee et al. [58] to obtain the initial disparity estimates. However, instead of computing the *full disparity map*, we only compute 10% of the disparity pixel values and use the proposed reconstruction algorithm to recover the dense disparity map. The 10% samples are selected according to the two stages of "Proposed WT+CT 2-Stage". In the first stage, we select the locations of the 5% samples using our oracle random sampling scheme with PCA improvement applied to the color image. A pilot estimate of the disparity is thus computed and the remaining 5% samples can be located according to the second stage of "Proposed WT+CT 2-Stage". The results shown in Figure 4.6 illustrate that the "Proposed WT+CT 2-Stage" generates the closest disparity maps compared to an ideal dense estimate.

In addition to real video sequences, we also test the proposed algorithm on a stereo system we developed. The system consists of a low cost stereo camera with customized block matching algorithms. In Figure 4.6 shows the results of the reconstructed disparity maps. Referring to the results of "[2] Grid" and "Bicubic Grid", we note that there are serious stair-like artifacts located at object boundaries. In contrast, the two proposed methods in general produce much smoother object boundaries, thanks to the superior modeling and the optimized sampling scheme. More interestingly, we observe that "Proposed WT+CT 2-Stage" indeed removes some undesirable noisy estimates in the recovered disparity maps. This shows that the proposed method could potentially further developed as a depth enhancement method.

Observing the reconstructed disparity maps in Figure 4.6, methods under comparisons include: a dense disparity estimation [58] to acquire initial estimate; "Proposed WT+CT 2-Stage" which applies the 2-Stage randomized scheme to determine sampling locations; "Proposed WT+CT Grid" which picks samples from a uniform grid; "[2] Grid" which applies a subgradient algorithm to samples picked from a uniform grid; "Bicubic Grid" which applies bicubic interpolation to samples picked from a uniform grid.

This Chapter includes materials that have been published in IEEE Transaction on Image Processing 2015, titled “Depth Reconstruction from Sparse Samples: Representation, Algorithm, and Sampling,” with Truong Q. Nguyen and Stanley H. Chan.



Left View

Right View

Dense Estimation [58]

Proposed WT+CT 2-Stage

Proposed WT+CT Grid

[2] Grid

Bicubic Grid

Left View

Right View

Dense Estimation [58]

Proposed WT+CT 2-Stage

Proposed WT+CT Grid

[2] Grid

Bicubic Grid

Figure 4.6: Examples of reconstructed results from 10% measured samples using real data using the “Newspaper” dataset, and real captured disparity maps.

Chapter 5

Depth Reconstruction Algorithm For Spatio-Temporal Depth Data

5.1 Introduction

Depth sensing technologies enable many applications in computer visions. For example, direct sensing techniques, such as Time-of-Flight camera, have been applied to assist gesture recognition [59, 60], 3D object scanning [61], and robot navigation [62]. Indirect sensing techniques, such as estimating disparities from stereo camera, have been applied to aid multiview synthesis [63], object segmentation [64], human pose estimation [65]. Typically, the performance of these applications correlates to the quality of depth data. Therefore, obtaining high quality depth data becomes an important topic in computer vision and image processing societies.

According to the configurations of depth sensing systems, depth data acquisition methods face different image processing problems. For example, given low resolution (LR) depth data and high resolution (HR) RGB images, super-resolution (SR) technique is typically applied for HR depth image synthesis [66]. To fill missing depth pixels, depth image inpainting is usually utilized for occluded region filling [67]. However, both techniques can be jointly worked by searching for the solutions to two problems: (1) finding and estimating reliable sample sets, and (2) reconstructing dense depth data from reliable samples. This leads to a general problem of dense depth reconstruction from a subset of samples. To resolve these problems, studies on efficient representations, sampling and reconstruction algorithm for *single* depth image were first discussed in our previous work [19], whereas in this paper, we mainly focus on extending single image-based to video-based depth reconstruction, which exploits *temporal information* and deals with depth data in the form of *spatio-temporal volume*. To the best of our knowledge, reconstructing depth video sequences has not yet been investigated. Therefore, we herein propose a framework for depth video reconstruction.

5.1.1 Related Works

The proposed depth video reconstruction framework lies in the fields of space-time depth enhancement and sensing. Works on single-frame-based processing are abundant, but research on dealing with spatio-temporal depth volume is limited. In the following, we discuss and highlight related works for both single and spatio-temporal depth images.

Depth enhancement method is used to synthesize high resolution depth data or to improve depth image quality. Recent works, such as iterative bilateral filtering [68], Markov Random Field (MRF) [69], and anisotropic diffusion tensor [66], are proposed to recover high resolution depth images. However, they typically assume that input data are uniformly-grid sampled, yielding less capability of processing input samples with irregular patterns. To deal with scattered depth data, guided filter [70, 67] and bilateral filter [71] are proposed for depth data hole filling, whereas they require the assistance from high quality RGB images during the filtering process. Other techniques, such as learning-based approaches [36, 72] and patch-based methods [73], require large training data and are usually time-consuming during training and synthesizing phases, resulting in less efficiency to deal with spatio-temporal volume. However, neither RGB images nor constraint on uniform-grid samples are required for the proposed spatio-temporal depth reconstruction algorithm (Chapter 5).

Depth sensing is typically associated with reconstruction algorithm, and its purpose is to acquire a sparse amount of meaningful samples, yielding high quality reconstructed depth data. Considering joint work on sensing and dictionary learning, Duarte-Carvajalino and Sapiro proposed a patch-based approach for image recovery [74]. Later, Schwartz et al. proposed to determine sensing matrices using saliency map [27]. These methods define sensing matrix with weights based on Gaussian or Gaussian-Bernoulli distributions. In contrast, our proposed method defines sampling matrix with weights, 1's, as the pixel is sampled, and 0's otherwise.

The most relevant work is probably our previous work on single-frame 2-Stage sampling and ADMM reconstruction algorithm [19], whereas in this paper, we consider both sampling strategy and reconstruction algorithm for spatio-temporal depth data. In 2-Stage sampling, a pilot sampling signal was deployed for the second stage sampling location predictions, yielding less efficiency on conducting reconstruction twice. However, the proposed spatio-temporal sampling scheme achieves 1-Stage sampling prediction and is capable of dealing with multiple frames at a time. Followed by a spatio-temporal reconstruction algorithm (Chapter 6), the proposed framework is thus suitable to depth video reconstruction.

5.2 Depth Video Reconstruction Framework and Notations

In this section, we describe the systematic configuration and notations for our proposed depth video reconstruction framework.

5.2.1 Systematic Configurations and Depth Data Descriptions

In our proposed depth video reconstruction framework, we assume that depth measurement systems contain at least one RGB camera as it is a standard configuration for depth sensing systems. For example, passive depth estimation systems, which indirectly estimate depth information by finding matched points between a pair of RGB images, have two RGB cameras. Active depth estimation systems, which directly measure depth using Time-of-Flight or infrared cameras, typically have at least one RGB camera (i.e., Kinect). Therefore, we assume that RGB video sequences exist in our systematic configuration.

The description of depth data varies as different depth measurement systems are utilized. In practice, depth information can be either directly measured by sensors or indirectly estimated from a pair of images. For the former, depth image is utilized for representing depth information. However, for the later, *disparity* map is used as an alternative because disparity values, which can be converted into depth information by the triangulation geometry [31], are differences between the locations of matched points. Therefore, in the rest of this paper, we use *depth* and *disparity* interchangeably.

5.2.2 Notations for Spatio-Temporal Volume

Assuming that Q is the index of previously reconstructed frame, we let $\boldsymbol{x}_{Q+t}$ to be a $N \times 1$ vector representing a depth image, $\boldsymbol{b}_{Q+t}$ to be a $N \times 1$ vector representing sampled data, and $\boldsymbol{y}_{Q+t}$ to be a $N \times 3$ vector representing the corresponding RGB image. The subscript $(Q+t)$ denotes the frame index and t is an integer with $t \geq 1$. The number of sampled data in $\boldsymbol{b}_{Q+t}$ is m, and the locations of sampled data are defined by a $N \times N$ diagonal sampling matrix with diagonal elements

$$\boldsymbol{S}_{Q+t,ii} = \begin{cases} 1, & i\text{-th element of } \boldsymbol{x}_{Q+t} \text{ is sampled,} \\ 0, & \qquad\qquad \text{otherwise,} \end{cases} \tag{5.1}$$

where the first subscript of $\boldsymbol{S}_{Q+t,ii}$ denotes the frame number and the second subscript denotes the location of the matrix. Noting that the off-diagonal elements of $\boldsymbol{S}_{Q+t}$ are zeros. More specifically, the locations of nonzero elements in $\boldsymbol{b}_{Q+t}$ are the locations of nonzero diagonal entries of $\boldsymbol{S}_{Q+t}$, i.e., $\boldsymbol{b}_{Q+t} = \boldsymbol{S}_{Q+t}\boldsymbol{x}_{Q+t}$. Here, we further define the sampling rate $\xi = \frac{m}{N}$ as an alternative for describing the number of sampled data, and in this work we especially consider $m \ll N$ and $\xi \ll 1$.

Now, considering a spatio-temporal depth volume consisting of T depth images, we let $\boldsymbol{x}$ to be a $TN \times 1$ vector

$$\boldsymbol{x} = [\boldsymbol{x}_{Q+1}; \boldsymbol{x}_{Q+2}; \cdots ; \boldsymbol{x}_{Q+T}] . \tag{5.2}$$

Similarly, the sampled data in the form of spatio-temporal volume $\boldsymbol{b}$ is defined as

$$\boldsymbol{b} = [\boldsymbol{b}_{Q+1}; \boldsymbol{b}_{Q+2}; \cdots ; \boldsymbol{b}_{Q+T}] . \tag{5.3}$$

The spatio-temporal volume of RGB images is

$$\boldsymbol{y} = \left[\boldsymbol{y}_{Q+1}; \boldsymbol{y}_{Q+2}; \cdots ; \boldsymbol{y}_{Q+T}\right] . \tag{5.4}$$

Then, the sampling matrix $\boldsymbol{S}$ is now a $TN \times TN$ matrix with diagonal elements

$$\boldsymbol{S}_{jj} = \begin{cases} 1, & j\text{-th entry of } \boldsymbol{x} \text{ is sampled,} \\ 0, & \text{otherwise,} \end{cases} \tag{5.5}$$

where the subscript of $\boldsymbol{S}_{jj}$ denotes the locations of matrix $\boldsymbol{S}$ and j is equal to $(tN + i)$. Note that given a fixed sampling rate $\xi = \frac{Tm}{TN}$, it is not necessarily to have m samples for each $\boldsymbol{b}_{Q+t}$, whereas only in total Tm number of samples over the whole spatio-temporal volume $\boldsymbol{b}$ is required. With the spatio-temporal depth volume representation, the sampling operation is now simplified as $\boldsymbol{b} = \boldsymbol{Sx}$.

We also exploit the temporal correlation by using motion vector estimation between RGB images, $\boldsymbol{y}_Q$, $\boldsymbol{y}_{Q+1}$, $\boldsymbol{y}_{Q+2}$, $\cdots$, $\boldsymbol{y}_{Q+T}$ using the motion vector estimation and compensation method proposed in [75]. Given the number of images T in a spatio-temporal volume and for $t = 1, 2, ...T$, we first estimate motion vectors $\boldsymbol{v}_{Q,Q+t}$ between RGB images $\boldsymbol{y}_Q$ and $\boldsymbol{y}_{Q+t}$. We then apply the motion compensation on previously reconstructed depth image $\hat{\boldsymbol{x}}_Q$ with estimated motion vector $\boldsymbol{v}_{Q,Q+t}$, and obtain motion compensated depth image $\hat{\boldsymbol{x}}_{Q+t,\text{mc}}$.

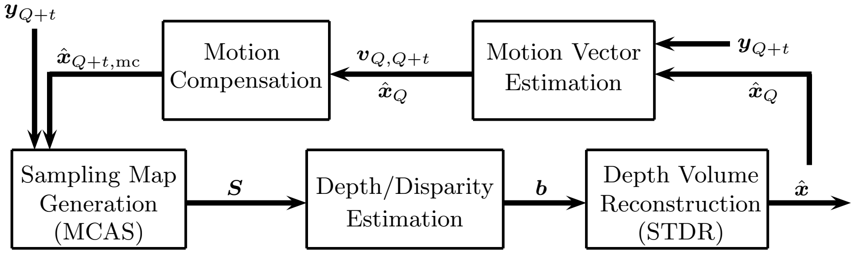


Figure 5.1: Systematic Overview of the proposed depth video reconstruction framework. Variable Q denotes index of previously reconstruction frame number and t denotes the index of the spatio-temporal volume, where $t = 1, 2, ..., T$, and T is the total number of frames of the spatio-temporal volume.

5.2.3 Sampling Patterns

For sampling matrix $\boldsymbol{S}$, the sampling pattern can be either deterministic or random. Given a sequence of probabilities $\{\boldsymbol{p}_j\}_{j=1}^{TN}$, we define the sampling matrix $\boldsymbol{S}_{jj} = 1$ with probability $\boldsymbol{p}_j$, and $\boldsymbol{S}_{jj} = 0$ with probability $\left(1 - \boldsymbol{p}_j\right)$. If $\boldsymbol{p}_j = 1$ for $j \in \Omega$, and the cardinality $|\Omega| = Tm$, the sampling pattern is deterministic. If $\boldsymbol{p}_j = \xi$ for $j = 1...TN$, the sampling pattern is now uniformly random. Apparently, the number of sampled data for both deterministic and uniformly random patterns are mathematically equal to Tm, and

$$\frac{1}{TN}\sum_{j=1}^{TN}\boldsymbol{p}_j = \xi = \frac{Tm}{TN}. \tag{5.6}$$

Remark 8. *As the sampling strategy is random, the number of sampled data may not be exactly the same as Tm. However, as the total number of samples, TN, increases, the number of sampled data, Tm, is asymptotically approaching the exact value, referring to Eq.(4) in [19].*

To compare the performance of existing depth reconstruction algorithms, in Section III, we first assume the sampling pattern is uniformly random, and in Section IV we will discuss the efficient sampling strategy for the proposed depth video reconstruction framework.

Figure 5.2 shows a flowchart representing our main contributions, and Figure 5.1 provides a systematic overview of our proposed depth video reconstruction framework. We will address our works on the “Depth Volume Reconstruction” and “Sampling Map Generation” blocks in Chapter 5.3 and Chapter 6 respectively. Depending on systematic configurations, in Chapter 6.3 and Chapter 6.4 we present the different applications using the proposed framework.

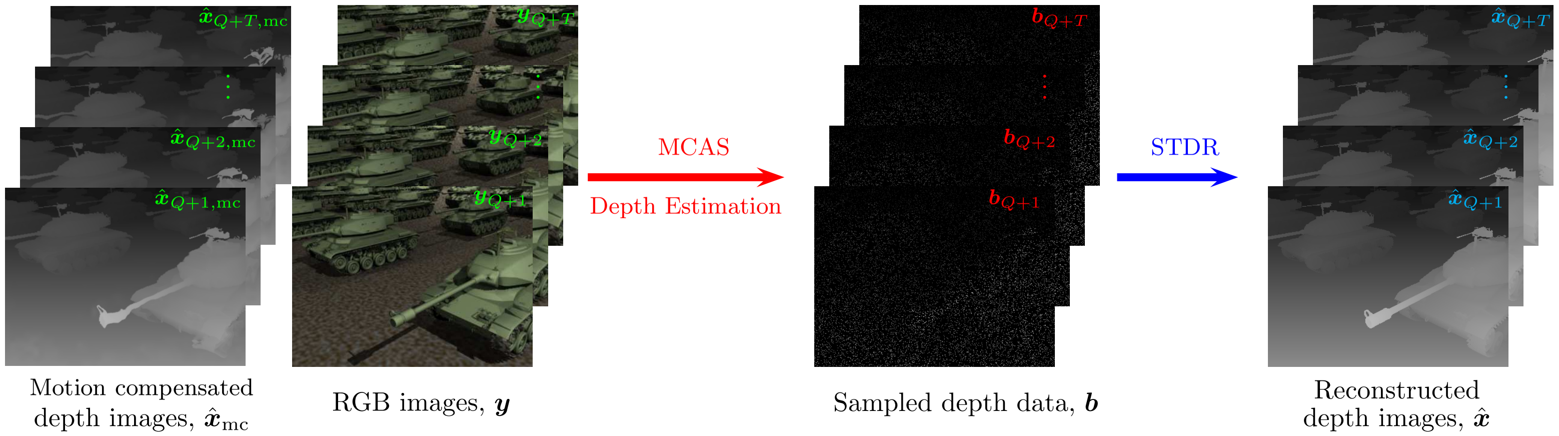

$\hat{x}_{Q+T,\mathrm{mc}}$
$\hat{x}_{Q+2,\mathrm{mc}}$
$\hat{x}_{Q+1,\mathrm{mc}}$
Motion compensated depth images, $\hat{x}_{\mathrm{mc}}$
y_{Q+T}
y_{Q+2}
y_{Q+1}
RGB images, y
MCAS
Depth Estimation
b_{Q+T}
b_{Q+2}
b_{Q+1}
Sampled depth data, b
STDR
$\hat{x}_{Q+T}$
$\hat{x}_{Q+2}$
$\hat{x}_{Q+1}$
Reconstructed depth images, $\hat{x}$

Figure 5.2: Flowchart for depth video reconstruction from sparse samples. Variables $\boldsymbol{y}_{Q+1}, \boldsymbol{y}_{Q+2} \cdots \boldsymbol{y}_{Q+T}$ are RGB images, variables $\boldsymbol{b}_{Q+1}, \boldsymbol{b}_{Q+2}, \cdots \boldsymbol{b}_{Q+T}$ are sparse depth samples, variables $\hat{\boldsymbol{x}}_{Q+1}, \hat{\boldsymbol{x}}_{Q+2}, \cdots \hat{\boldsymbol{x}}_{Q+T}$ are reconstructed depth images. Variables $\hat{\boldsymbol{x}}_{Q+1,\mathrm{mc}}, \hat{\boldsymbol{x}}_{Q+2,\mathrm{mc}}, \cdots, \hat{\boldsymbol{x}}_{Q+T,\mathrm{mc}}$ are motion compensated depth images using estimated motion vectors $\boldsymbol{v}_{Q,Q+1}, \boldsymbol{v}_{Q,Q+2}, \cdots, \boldsymbol{v}_{Q,Q+T}$ and previously reconstruction depth $\hat{\boldsymbol{x}}_{Q}$. For all variables, the first subscript denotes the corresponding frame number.

5.3 Spatio-Temporal Depth Reconstruction

5.3.1 Problem Formulations

In order to reconstruct spatio-temporal depth volume from a subset of samples, we first formulate the problem as

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{W}\boldsymbol{\Psi}\boldsymbol{x}\|_1, \tag{5.7}$$

where $\boldsymbol{\Psi}$ is a $TN \times TN$ orthonormal matrix, and $\boldsymbol{W}$ is a $TN \times TN$ diagonal weighting matrix. Apart from our previous work [19], in this paper we let $\boldsymbol{\Psi}$ to be wavelet transform because it is scalable to spatio-temporal domain, and it is suitable for spatio-temporal data. As $T = 1$, 2D discrete Wavelet transform (2D-DWT) is applied, and we set the weight in $\boldsymbol{W}_{jj}$ to be 1 while location j relates to the detailed coefficients. Otherwise, we set $\boldsymbol{W}_{jj}$ to be 0. As $T > 1$, 3D-DWT is then utilized for dealing with the spatio-temporal data, and the weight in $\boldsymbol{W}_{jj}$, where the location j relates to the detailed coefficients in the bands {LHL, HLL, HHL, LHH, HLH, HHH}, is set to 1. Otherwise, weights are set to 0.

The main advantage of using Wavelet transform is because of its capability of analyzing and synthesizing signals in both spatial and temporal domains. More specifically, 3D-DWT analyzes a signal along 3 dimensions, horizontal (h), vertical (v) and temporal (t), and decomposes a signal in each direction into lowpass (L) and highpass (H) respectively. With the labels of L and H in order, bands of transform coefficients are now defined. For example, transform coefficients in the band LLH are obtained from taking low pass in horizontal and vertical directions, and high pass in temporal direction.

The design of weighting matrix $\boldsymbol{W}$ relates to the characteristic of transform coefficients of dense spatio-temporal depth volume. Figure 5.3 shows a histogram of the magnitude of transform coefficients using 3D-DWT. It is evident that coefficients with large magnitudes locate mainly in {LLL, LLH}. Intuitively, to ensure coefficients in {LLL, LLH} can be recovered, we set the corresponding weights to be 0. Conversely, to limit the magnitude of coefficients in {LHL, HLL, HHL, LHH, HLH, HHH} to be small, we set the relating diagonal elements of $\boldsymbol{W}$ to be 1. Therefore, the ℓ_1 regularization term in (5.7) with designed weighting matrix $\boldsymbol{W}$ fits the characteristic of transformed

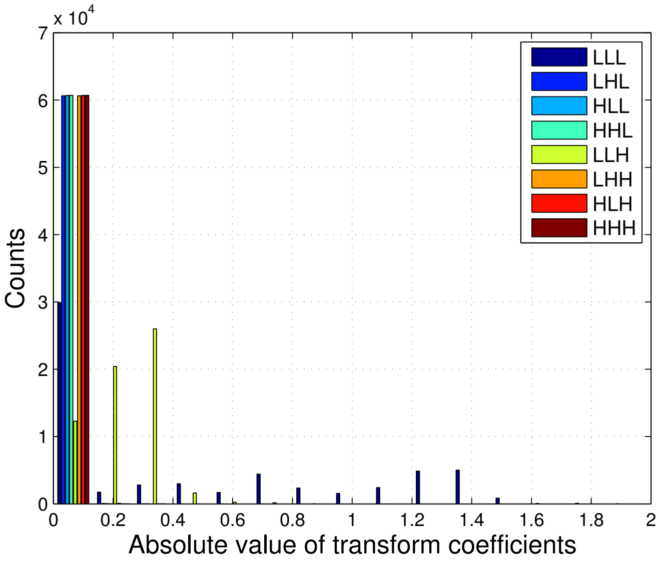


Figure 5.3: Histogram of magnitude of 3D-DWT transform coefficients. We construct the spatio-temporal depth volume using the 20th and 21st frames of tank sequences ($T = 2$), and we then apply 3D-DWT using "db2" and decomposition level=1.

coefficients of dense spatio-temporal depth volume.

Observing that depth data has piecewise smooth property, and the discontinuities locate mainly along the object boundaries. To preserve these properties, we therefore introduce total variation as a regularization term.

$$\min_{x} \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{W}\boldsymbol{\Psi}\boldsymbol{x}\|_1 + \beta\|\boldsymbol{x}\|_{\text{TV}}, \tag{5.8}$$

where $\|\cdot\|_{\text{TV}}$ denotes anisotropic spatial total variation, and $\|\boldsymbol{x}\|_{\text{TV}} = \|\boldsymbol{D}\boldsymbol{x}\|_1$, where $\boldsymbol{D} = [\boldsymbol{D}_h; \boldsymbol{D}_v]$. $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$ are horizontal and vertical difference operators. As $\boldsymbol{x}$ is a spatio-temporal volume, we may consider spatio-temporal total variation, which is defined $\boldsymbol{D} = [\boldsymbol{D}_h; \boldsymbol{D}_v; \boldsymbol{D}_t]$. However, our empirical results indicate that the improvement from spatio-temporal total variation is limited while frames of $\boldsymbol{x}$ are not similar. We thus choose spatial total variation only. In the following, we present our proposed spatio-temporal depth reconstruction using alternating direction method of multipliers (ADMM).

5.3.2 Spatio-Temporal Depth Reconstruction Algorithm

To solve (5.8), we propose to use alternating direction method of multipliers (ADMM) as it is capable of dealing with large scale problem. ADMM was first proposed by [47, 48], and in this decade, it has been widely used for image processing applications, e.g., image deblurring [76], image/video denoising [7]. Recently, ADMM is utilized for single *depth image* reconstruction in our previous work [19], whereas the main difference is that we extend this work to a dense *spatio-temporal depth volume reconstruction* from a subset of samples. In terms of algorithm, the approach is similar to our single frame ADMM. Readers can refer to our previous paper for detailed discussion [19]. We note that the scalability to the sizes of spatio-temporal volume is more important for depth video reconstruction framework. In the following, we briefly present our proposed spatio-temporal depth reconstruction (STDR) algorithm, and then discuss its temporal scability.

First, we introduce new auxiliary variables, $\boldsymbol{r} = \boldsymbol{x}$, $\boldsymbol{u}_1 = \boldsymbol{\Psi x}$ and $\boldsymbol{u}_2 = \boldsymbol{D x}$, and reformulate (5.8) as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{S r}\|^2 + \lambda\|\boldsymbol{W u}_1\|_1 + \beta\|\boldsymbol{u}_2\|_1, \\ & \text{subject to } \boldsymbol{r} = \boldsymbol{x}, \boldsymbol{u}_1 = \boldsymbol{\Psi x}, \text{ and } \boldsymbol{u}_2 = \boldsymbol{D x}. \end{aligned} \tag{5.9}$$

Applying augmented Lagrangian, the constrained minimization problem (5.9) is then reformulated as

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{w}, \boldsymbol{q}_1, \boldsymbol{q}_2\right) &= \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{S r}\|_2^2 + \lambda\|\boldsymbol{W u}_1\|_1 + \beta\|\boldsymbol{u}_2\|_1 \\ & - \boldsymbol{w}^T\left(\boldsymbol{r} - \boldsymbol{x}\right) - \boldsymbol{q}_1^T\left(\boldsymbol{u}_1 - \boldsymbol{\Psi x}\right) - \boldsymbol{q}_2^T\left(\boldsymbol{u}_2 - \boldsymbol{D x}\right) \\ & + \frac{\rho}{2}\|\boldsymbol{r} - \boldsymbol{x}\|_2^2 + \frac{\gamma_1}{2}\|\boldsymbol{u}_1 - \boldsymbol{\Psi x}\|_2^2 + \frac{\gamma_2}{2}\|\boldsymbol{u}_2 - \boldsymbol{D x}\|_2^2. \end{aligned} \tag{5.10}$$

Note that $\boldsymbol{w}$, $\boldsymbol{q}_1$, and $\boldsymbol{q}_2$ are Lagrange multipliers, and scalar variables ρ, γ_1 and γ_2 are internal parameters. These internal parameters are set to be fixed as T varies. Detailed discussions on the parameter selections is presented in Chapter 5.3.3. Equation (5.10) can be split into multiple subproblems, and its optimal solution $\boldsymbol{x}^*$ can be obtained by solving $\boldsymbol{x}$-,$\boldsymbol{r}$-,$\boldsymbol{u}_1$-,$\boldsymbol{u}_2$- subproblems sequentially. Solutions to these subproblems are,

$$\boldsymbol{x}^{(k+1)} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\left[\text{R.H.S.}\right]}{\left(\rho + \gamma_1\right)\mathbf{I} + \gamma_2|\mathcal{F}\left(\boldsymbol{D}\right)|^2}\right\}, \tag{5.11}$$

where R.H.S. $= \rho \boldsymbol{r}^{(k)} + \boldsymbol{\Psi}^T \gamma_1 \boldsymbol{u}_1^{(k)} + \boldsymbol{D}^T \gamma_2 \boldsymbol{u}_2^{(k)} - \boldsymbol{w}^{(k)} - \boldsymbol{\Psi}^T \boldsymbol{q}_1^{(k)} - \boldsymbol{D}^T \boldsymbol{q}_2^{(k)}$.

$$\boldsymbol{r}^{(k+1)} = \left(\boldsymbol{S}^T\boldsymbol{S} + \rho\boldsymbol{I}\right)^{-1} \left(\boldsymbol{S}^T\boldsymbol{b} + \boldsymbol{w}^{(k)} + \rho\boldsymbol{x}^{(k+1)}\right). \tag{5.12}$$

$$\boldsymbol{u}_1^{(k+1)} = \max\left(|\boldsymbol{z}| - \frac{\lambda \mathrm{diag}\left(\boldsymbol{W}\right)}{\gamma_1}, 0\right) \cdot \mathrm{sign}\left(\boldsymbol{z}\right), \tag{5.13}$$

where $\boldsymbol{z} = \boldsymbol{\Psi}\boldsymbol{x}^{(k+1)} + \frac{\boldsymbol{q}_1^{(k)}}{\gamma_1}$.

$$\boldsymbol{u}_2^{(k+1)} = \max\left(|\boldsymbol{z}| - \frac{\beta}{\gamma_2}, 0\right) \cdot \mathrm{sign}\left(\boldsymbol{z}\right), \tag{5.14}$$

where $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{x}^{(k+1)} + \frac{\boldsymbol{q}_2^{(k)}}{\gamma_2}$.

We note that $\boldsymbol{r}$, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ subproblems are independent to each other, parallel processing on these three subproblems is feasible to further speed up the algorithm. Then, the Lagrange multiplier updates are

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \rho\left(\boldsymbol{r}^{(k+1)} - \boldsymbol{x}^{(k+1)}\right), \tag{5.15a}$$

$$\boldsymbol{q}_1^{(k+1)} = \boldsymbol{q}_1^{(k)} - \gamma_1\left(\boldsymbol{u}_1^{(k+1)} - \boldsymbol{\Psi}\boldsymbol{x}^{(k+1)}\right), \tag{5.15b}$$

$$\boldsymbol{q}_2^{(k+1)} = \boldsymbol{q}_2^{(k)} - \gamma_2\left(\boldsymbol{u}_2^{(k+1)} - \boldsymbol{D}\boldsymbol{x}^{(k+1)}\right). \tag{5.15c}$$

The proposed spatio-temporal depth reconstruction algorithm solves subproblems sequentially, and it iterates until stopping criteria is met.

5.3.3 Parameter Tuning and Temporal Volume Scalability

Observing (5.10), the model with augmented Lagrangian has two regularization parameters, λ and β, and three internal parameters, ρ, γ_1 and γ_2. The selection of parameters is typically problem related, and for the proposed STDR algorithm, the robustness to different volumes is especially important. In the following, we discuss our experiments on parameter selection for the spatio-temporal depth reconstruction algorithm.

Evaluation Metrics

Instead of using mean square error (MSE), we consider mean absolute error (MAE) as an evaluation metric for the selection of parameters. The main reasons are that MAE has been shown to be a better metric than mean square error (MSE) [77], and for depth measurements, pixel-wise absolute difference is commonly utilized. Given a reconstructed spatio-temporal disparity volume, $\hat{\boldsymbol{x}}$, the mean absolute error is defined as

$$\text{MAE of } \hat{\boldsymbol{x}} = \frac{1}{TN}\sum_{j=1}^{TN} |\hat{\boldsymbol{x}}_j - \boldsymbol{x}_j|. \tag{5.16}$$

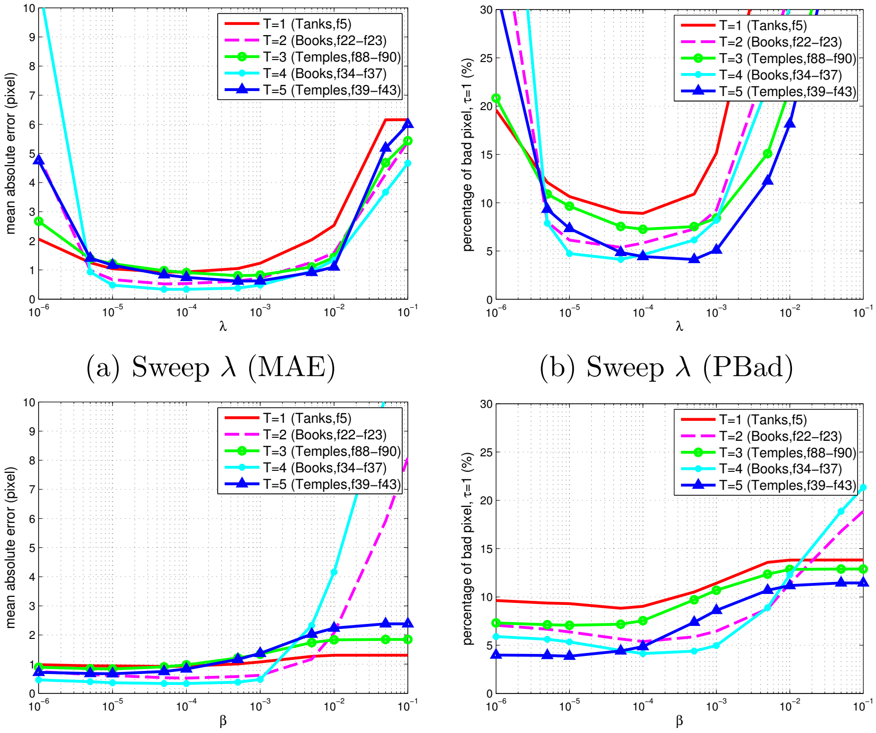
For the simulation, we use the disparity video dataset provided in [78]. During the experiments, we normalize the disparities to $[0, 1]$ for the proposed STDR algorithm, and then rescale disparities back to $[0, 255]$ for the evaluation. Once the parameters are determined, we further validate the reconstructed result using percentage of bad pixel,

$$\text{Bad pixel rate of } \hat{\boldsymbol{x}} = \frac{1}{TN}\sum_{j=1}^{TN} \mathcal{I}\{|\hat{\boldsymbol{x}}_j - \boldsymbol{x}_j| > \tau\}, \tag{5.17}$$

where τ is an integer number, and $\mathcal{I}\{\cdot\}$ is an indicator function. $\mathcal{I}\{\cdot\}$ returns 1 as the statement is true. Otherwise, it returns 0.

Regularization Parameters (λ, β)

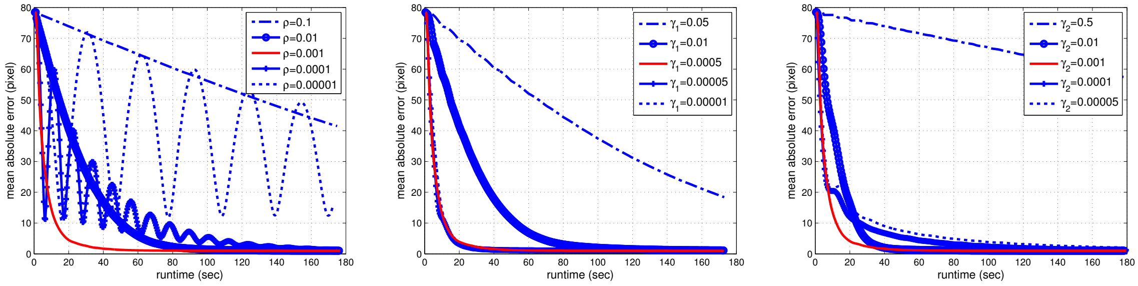
Regularization parameters typically relate to the reconstruction performance. Results on our regularization parameter selection are shown in Figure 5.4. Due to the limited spaces, we show experimental results with randomly selected frames and video sequences. Observing MAE curves shown in (a) and (c), results indicate that lowest mean absolute error locate in the ranges $\left[5 \times 10^{-5} \leq \lambda \leq 5 \times 10^{-4}\right]$ and $\left[10^{-6} \leq \beta \leq 10^{-4}\right]$. We therefore pick $\lambda = 10^{-4}$ and $\beta = 5 \times 10^{-5}$. We further validate our selected parameters using PBad (observing (b) and (d)), justifying that the selected parameters also achieve minimum bad pixel rate.



(a) Sweep λ (MAE) (b) Sweep λ (PBad)

(c) Sweep β (MAE) (d) Sweep β (PBad)

Figure 5.4: Experiments on regularization parameter selection for the proposed spatio-temporal depth reconstruction. We conduct the experiment with varying depth video sequences and number of frames, and set the sampling rate $\xi = 0.1$. While sweeping λ, we set $\beta = 5 \times 10^{-5}$, and while sweep β, we set $\lambda = 10^{-4}$.



(a) Sweep ρ (b) Sweeping γ_1 (c) Sweeping γ_2

Figure 5.5: Experiments on internal parameter selection. We conduct experiment with "tanks" sequence with 5 continual frames (frame no. 72-76, $T = 5$, $\xi = 0.1$). The chosen parameters are shown in red curves. Results indicate that our selected parameters achieves the fastest convergence rate.

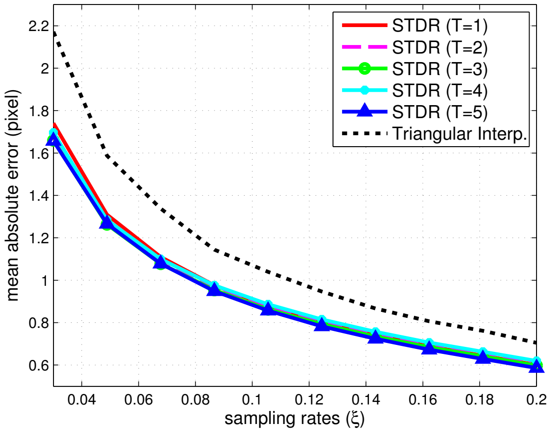


Figure 5.6: Reconstruction performance verses number of frames T. For each sampling rate and T, we average the reconstruction performance over 100 frames of "tank" sequence. Noting that we use 99 frames for the case $T = 3$.

Internal Parameters $(\rho, \lambda_1, \lambda_2)$

Internal parameters typically relate to the convergence rate. Examples on our internal parameter selections are shown in Figure 5.5. The red curves indicate the convergence rates for the selected internal parameters, and we observe that the chosen parameters yield the fastest convergence rate. All the experiments on parameter selection are conducted on computer with Intel 3.2GHz CPU, 12GB RAM, 64-bits Windows 7 and MATLAB R2014a. For more experimental results, readers can refer to our supplementary materials in `http://videoprocessing.ucsd.edu/~leekang/2015JournalPublication.html`.

Based on the selection of parameters, we in addition conduct an experiment on reconstructing dense "tanks" sequence from a subset of random samples with $T = 1, 2, ..., 5$. As shown in Figure 5.6, on average, the reconstruction performance is robust to the sizes of spatio-temporal volume, and we can still observe slight improvement as T increases, especially when sampling rate is around at $\xi = 0.03$ (3%). Additionally, we choose "Triangular Interpolation" method, which utilizes *Delaunary triangulation* for scattered data interpolation [79], as a benchmark. With the evaluation over the whole tank sequence, the proposed STDR algorithm has approximately 0.5 pixels improvement when the sampling rate is at 3%, and 0.2 pixels improvement when the sampling rate is at 20%. Finally, we summarize the selection of parameters in Table 5.1.

Table 5.1: Summarize of parameters for STDR.

Parameter	Functionality	Value
λ	Regularization for Wavelet sparsity	10^{-4}
β	Regularization for Total Variation	5×10^{-5}
ρ	Half quadratic penalty for $\boldsymbol{r} = \boldsymbol{x}$	10^{-3}
γ_1	Half quadratic penalty for $\boldsymbol{u}_1 = \boldsymbol{\Psi x}$	5×10^{-4}
γ_2	Half quadratic penalty for $\boldsymbol{u}_2 = \boldsymbol{Dx}$	10^{-3}

5.3.4 Initialization using Temporal Information

Upon the depth video configuration in Figure 5.1, the convergence rate of the proposed STDR algorithm can be further accelerated by exploiting temporal information. So far, we have discussed the characteristic of convergence rate as the unknown samples are set to be 0. To further speed up the rate of convergence, we propose a spatio-temporal scheme that accommodates the typical configuration of depth video processing, using motion compensation. In the following, we present our proposed spatio-temporal scheme that utilizes motion compensated dense depth images.

Using Motion Compensated Depth Images

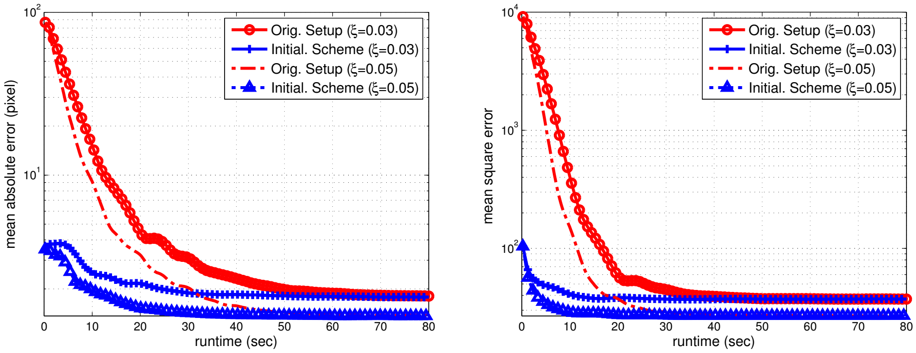
As we discussed in Section II-A, RGB cameras are typical configuration in depth measurement systems, and thus motion vectors estimated by continual RGB images can be further utilized to facilitate the depth video reconstruction process. Now we let $\boldsymbol{v}_{Q,Q+1}$ to be a motion vector estimated between two RGB images $\boldsymbol{y}_Q$ and $\boldsymbol{y}_{Q+1}$, and $\hat{\boldsymbol{x}}_{Q+1,\text{mc},}$ to be the motion compensated depth image of $Q+1$ frame using the motion vector $\boldsymbol{v}_{Q,Q+1}$. Therefore, to obtain motion compensated depth images for the spatio-temporal volume, we repeat the process T times and obtain $\hat{\boldsymbol{x}}_{Q+1,\text{mc},}, \hat{\boldsymbol{x}}_{Q+2,\text{mc},}, \cdots, \hat{\boldsymbol{x}}_{Q+T,\text{mc},}$. Then, we define the spatio-temporal depth video volume for initialization as,

$$\boldsymbol{b}_{\text{init}} = \left[\hat{\boldsymbol{x}}_{Q+1,\text{mc}}; \hat{\boldsymbol{x}}_{Q+2,\text{mc}}; \cdots \hat{\boldsymbol{x}}_{Q+T,\text{mc}}\right]. \tag{5.18}$$

The auxiliary variables are initialized by $\boldsymbol{r}^{(0)} = \boldsymbol{b}_{\text{init}}$, $\boldsymbol{u}_1^{(0)} = \boldsymbol{\Psi b}_{\text{init}}$, and $\boldsymbol{u}_2^{(0)} = \boldsymbol{D b}_{\text{init}}$.

Figure 5.7 shows the convergence rates with configurations of original setup (Orig.), and the proposed initialization scheme. For this experiment, we assume previously reconstructed depth image, $\hat{\boldsymbol{x}}_Q$ is error free, and we apply the motion vector estimation and compensation method proposed in [75]. Motion vectors are estimated using RGB images $\boldsymbol{y}_Q, \boldsymbol{y}_{Q+1}, \cdots \boldsymbol{y}_{Q+T}$. Then, the compensated depth images $\hat{\boldsymbol{x}}_{Q+1,\text{mc},}, \hat{\boldsymbol{x}}_{Q+2,\text{mc},}, \cdots \hat{\boldsymbol{x}}_{Q+T,\text{mc},}$ are obtained using estimated motion vectors and previously reconstructed depth image, $\hat{\boldsymbol{x}}_Q$.

Observing MAE curves with $\xi = 0.03$, the "Orig." configuration reaches steady state in around 65 seconds, whereas "Initial Scheme" configuration requires only 30 seconds to reach steady state. This indicates that the proposed initialization scheme achieves more than 2× faster convergence rate than the original setup. We also observe that MSE curves reach steady state faster than the MAE curves, indicating that MAE is sensitive to subtle variations. This implies that MAE is a better evaluation metric than MSE. The overall Spatio-temporal depth reconstruction is shown in Algorithm 6.



(a) MAE curves

(b) MSE curves

Figure 5.7: Convergence rate comparisons. We feed frames 20-21 ($T = 2$) of "tanks" sequence to STDR algorithm.

5.3.5 Preliminary Comparisons

To the best of our knowledge, the proposed work is the only work that reconstructs depth video without additional information. We therefore justify the comparison to single-frame reconstruction method. For the "Hawe [2]" method, its default configurations are applied. For the proposed STDR algorithm, we apply depth image reconstruction with $T = 1$ to the first frame, and we then conduct the proposed initialization scheme and with $T = 5$. For wavelet transform configurations, we use "db2" as the wavelet function with two decomposition levels.

Algorithm 6 Spatio-Temporal Depth Reconstruction (STDR) Algorithm

```
Require: $\boldsymbol{b}$, $\boldsymbol{S}$, $T$, $\hat{\boldsymbol{x}}_{Q+1,\mathrm{mc}}, \cdots, \hat{\boldsymbol{x}}_{Q+T,\mathrm{mc}}$
Initialization:
if $Q == 0$ then
    $\boldsymbol{x}^{(0)} = \boldsymbol{b}$, $\boldsymbol{r}^{(0)} = \boldsymbol{b}$, $\boldsymbol{u}_1^{(0)} = \boldsymbol{\Psi}\boldsymbol{b}$, $\boldsymbol{u}_2^{(0)} = \boldsymbol{D}\boldsymbol{b}$,
    $\boldsymbol{w}^{(0)} = \boldsymbol{0}$, $\boldsymbol{q}_1^{(0)} = \boldsymbol{0}$, $\boldsymbol{q}_2^{(0)} = \boldsymbol{0}$.
else
    $\boldsymbol{b}_{\mathrm{init}} = [\hat{\boldsymbol{x}}_{Q+1,\mathrm{mc}}; \hat{\boldsymbol{x}}_{Q+1,\mathrm{mc}}; \cdots \hat{\boldsymbol{x}}_{Q+T,\mathrm{mc}}]$.
    $\boldsymbol{x}^{(0)} = \boldsymbol{b}_{\mathrm{init}}$, $\boldsymbol{r}^{(0)} = \boldsymbol{b}_{\mathrm{init}}$, $\boldsymbol{u}_1^{(0)} = \boldsymbol{\Psi}\boldsymbol{b}_{\mathrm{init}}$, $\boldsymbol{u}_2^{(0)} = \boldsymbol{D}\boldsymbol{b}_{\mathrm{init}}$,
    $\boldsymbol{w}^{(0)} = \boldsymbol{0}$, $\boldsymbol{q}_1^{(0)} = \boldsymbol{0}$, $\boldsymbol{q}_2^{(0)} = \boldsymbol{0}$.
end if
while $\frac{\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2}{\|\boldsymbol{x}^{(k)}\|_2} \geq tol$ do
    Solve $\boldsymbol{x}$ subproblem using (5.11).
    Solve $\boldsymbol{r}, \boldsymbol{u}_1, \boldsymbol{u}_2$ subproblems using (5.12), (5.13), (5.14).
    Update Lagrange multipliers using (5.15a), (5.15b), (5.15c).
end while
return $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{(k+1)}$
```

Table 5.2 shows the comparison of reconstruction performance for 5%, 10%, 15%, and 20%. We note that in this experiment, we consider three evaluation metrics, PBad, MAE and PSNR. Referring to the PSNR metric, we can observe that the proposed STDR algorithm outperforms the other existing methods, and the STDR has on average 9 dB improvement than the second best result for the "Book" sequence. Also, for all test sequences, the proposed method mostly achieves the lowest MAE and PBad. So far, we conduct experiments using uniformly random sampling strategy. In the following Chapter, we will discuss an efficient sampling strategy.

This Chapter includes materials that have been published in IEEE Global Conference on Signal and Information Processing 2015, titled "Spatio-Temporal Depth Data Reconstruction from a Subset of Samples," with Truong Q. Nguyen, and materials that have been submitted to IEEE Transaction on Image Processing, titled "A Framework for Depth Video Reconstruction from a subset of Samples and its Applications," with Truong Q. Nguyen.

Table 5.2: Comparison of reconstruction algorithms from 5%, 10%, 15% and 20% uniformly random samples. The results are evaluated over the whole depth video sequence.

Sequence Name		Tanks (100 frames)				Books (41 frames)				Temples (100 frames)			
Sampling Rates	Methods	PSNR (dB)	MAE (pixels)	% Bad Pel.		PSNR (dB)	MAE (pixels)	% Bad Pel.		PSNR (dB)	MAE (pixels)	% Bad Pel.	
				τ=1	τ=2			τ=1	τ=2			τ=1	τ=2
5%	Proposed	**34.5264**	**1.2601**	12.42	**8.66**	**40.4820**	**0.6101**	8.24	4.87	**31.3678**	**1.1994**	**9.45**	**6.05**
	Tri. Interp.	32.1065	1.4015	**12.28**	9.51	29.9535	0.7956	**5.68**	**4.34**	29.7601	1.6862	10.05	7.42
	Hawe [2]	31.1715	1.9065	21.26	13.85	28.3973	1.3913	13.84	8.58	29.5656	1.6928	18.22	11.28
10%	Proposed	**36.3440**	**0.8831**	**8.37**	**5.99**	**42.3340**	**0.3968**	4.44	**2.65**	**32.9085**	**0.8442**	**6.14**	**3.89**
	Tri. Interp.	34.6588	0.9899	9.02	6.98	33.8628	0.4859	**3.85**	2.96	31.5373	1.1595	6.93	5.01
	Hawe [2]	34.0886	1.1358	12.63	7.76	29.9567	0.9242	7.36	4.39	31.5382	1.0048	10.34	5.22
15%	Proposed	**37.3684**	**0.7072**	**6.56**	**4.79**	**43.5286**	**0.3056**	**3.02**	**1.90**	**34.1938**	**0.6459**	**4.52**	**2.91**
	Tri. Interp.	35.9236	0.8079	7.35	5.71	35.8252	0.3857	3.03	2.35	32.6726	0.9115	5.37	3.86
	Hawe [2]	35.5027	0.8340	8.89	5.32	31.3346	0.6926	4.93	2.94	32.8860	0.7231	6.79	3.02
20%	Proposed	**38.1133**	**0.5900**	**5.39**	3.98	**44.3916**	**0.2473**	**2.21**	**1.47**	**35.3459**	**0.5091**	**3.50**	2.26
	Tri. Interp.	36.9418	0.6896	6.23	4.86	37.3015	0.3245	2.49	1.96	33.5109	0.7595	4.42	3.17
	Hawe [2]	36.5424	0.6608	6.69	**3.97**	37.6195	0.3272	3.51	1.99	33.8765	0.5668	4.78	**2.03**

Chapter 6

Efficient Sampling Strategy for Spatio-Temporal Depth Data

6.1 Motion Compensation Assisted Sampling

Assuming that dense ground truth depth videos are not provided and the sampling rate, ξ, is fixed, an efficient sampling strategy that can maximize reconstruction performance becomes especially important. Sampling strategies, such as uniform grid and uniformly random sampling, are commonly utilized. However, by inferring the corresponding view images, sampling locations can be efficiently determined.

An efficient sampling strategy for single depth image reconstruction was discussed in our previous work [19], whereas in this paper, we focus on an efficient sampling scheme for depth video reconstruction. In [19], the 2-Stage sampling strategy uses half of sampling budget for pilot signal estimation and then determines the other half by referring to the reconstructed depth image from stage 1. However, in this paper, we explore another sampling scheme by using the temporal information. In the following, we first show that a linear combination of gradient maps can assist oracle random sampling since more than one gradient information exist. We then discuss the synthesis of gradient maps using gradient of RGB images and PCA responses of motion compensated depth images. Finally, we present a depth video reconstruction framework utilizing the proposed motion compensation assisted sampling (MCAS) strategy and STDR algorithm.

6.1.1 Oracle Random Sampling Assisted by a Linear Combination of Gradient Maps

Let $\boldsymbol{h}_{Q+t}$ be a $N \times 1$ vector representing a linear combination of gradients for the $(Q+t)$th frame, then define $\boldsymbol{h}_{Q+t}$ as

$$\boldsymbol{h}_{Q+t} = \sum_{\ell=1}^{L} \theta_\ell \boldsymbol{h}_{Q+t,\ell}, \tag{6.1}$$

where $\boldsymbol{h}_{Q+t,\ell}$ denotes a gradient map, θ_ℓ is a scalar value, and $\sum_{\ell=1}^{L} \theta_\ell = 1$. In this paper, we consider $L = 2$, and define $\boldsymbol{h}_{Q+t,1}$ to be the gradient of RGB image. Here $\boldsymbol{h}_{Q+t,2}$ is the magnitude of PCA responses of motion compensated depth image. The derivation below is based on the case that $L = 2$ and $\boldsymbol{h}_{Q+t}$ is defined to be

$$\boldsymbol{h}_{Q+t} = \theta_1 \boldsymbol{h}_{Q+t,1} + (1-\theta_1)\, \boldsymbol{h}_{Q+t,2}. \tag{6.2}$$

Now, we let $\{\boldsymbol{I}_{Q+t,j}\}_{j=1}^{N}$ to be independent Bernoulli random variables with a sequence of probability $\{\boldsymbol{p}_{Q+t,j}\}_{j=1}^{N}$. The probability of a pixel to be sampled or not is defined as

$$\begin{aligned} Pr\{\boldsymbol{I}_{Q+t,j} \text{ is sampled}\} &= \boldsymbol{p}_{Q+t,j}, \\ Pr\{\boldsymbol{I}_{Q+t,j} \text{ is not sampled}\} &= 1 - \boldsymbol{p}_{Q+t,j}. \end{aligned} \tag{6.3}$$

Then, we define a new random variable that averages the gradients with aforementioned independent Bernoulli random variables

$$\boldsymbol{Y}_{Q+t} = \frac{1}{N} \sum_{j=1}^{N} \frac{\boldsymbol{I}_{Q+t,j}\left[\theta_1 \boldsymbol{h}_{Q+t,1,j} + (1-\theta_1)\, \boldsymbol{h}_{Q+t,2,j}\right]}{\boldsymbol{p}_{Q+t,j}}. \tag{6.4}$$

According to Section V-B in [19], given a fixed θ_1, it is straight forward to show that the random variable $\boldsymbol{Y}_{Q+t}$ is unbiased. Minimizing the variance of $\boldsymbol{Y}_{Q+t}$ yields

$$\begin{aligned} \boldsymbol{p}_{Q+t,j} &= \min\left\{\left[\theta_1 \boldsymbol{h}_{Q+t,1,j} + (1-\theta_1)\, \boldsymbol{h}_{Q+t,2,j}\right]\tau, 1\right\} \text{ for } 1 \le j \le N, \\ \sum_{j=1}^{N} &\min\left\{\left[\theta_1 \boldsymbol{h}_{Q+t,1,j} + (1-\theta_1)\, \boldsymbol{h}_{Q+t,2,j}\right]\tau, 1\right\} - m = 0. \end{aligned} \tag{6.5}$$

Seeing (6.5), we can observe that given the gradient maps $\boldsymbol{h}_{Q+t,1}$ and $\boldsymbol{h}_{Q+t,2}$, the sampling probability for each point is now a function of θ_1. In the next section, we discuss the generation of gradient maps $\boldsymbol{h}_{Q+t,1}$ and $\boldsymbol{h}_{Q+t,2}$.

6.1.2 Synthesis of Gradient Maps

Based on the depth video reconstruction framework, both the spatial information and temporal correlations to the sequence of RGB images can be further exploited by using motion compensation. With the motion compensated depth image $\hat{\boldsymbol{x}}_{Q+t,\text{mc}}$ and RGB image $\boldsymbol{y}_{Q+t}$, we propose to synthesize $\boldsymbol{h}_{Q+t}$ from the responses of principal components to the motion compensated depth image and the gradient of the corresponding RGB image.

Obtaining $\boldsymbol{h}_{Q+t,1}$

To obtain motion compensated depth image $\hat{\boldsymbol{x}}_{Q+t,\text{mc},}$, we apply motion compensation to the previously reconstructed depth image $\hat{\boldsymbol{x}}_Q$ with the motion vector $\boldsymbol{v}_{Q,Q+t}$ estimated from $\boldsymbol{y}_Q$ and $\boldsymbol{y}_{Q+t}$. As $\hat{\boldsymbol{x}}_{Q+t,\text{mc}}$ is not error free, we apply principal component analysis and obtain PCA responses of motion compensated depth image for the gradient map $\boldsymbol{h}_{Q+t,1}$. We first define a set of $N_p \times 1$ vectors $\{\boldsymbol{a}_j\}_{j=1}^{N}$, where $\boldsymbol{a}_j$ is a canonical representation of a $\sqrt{N_p} \times \sqrt{N_p}$ patch centered at pixel j of $\hat{\boldsymbol{x}}_{Q+t,\text{mc},}$. Noting that patches are obtained from sliding windows with 1 pixel difference, and thus total number of patches is equal to total number of pixels of $\hat{\boldsymbol{x}}_{Q+t,\text{mc}}$. Consequently, we construct a matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N]$, calculate the correlation matrix, and conduct a singular value decomposition (SVD)

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T, \quad \boldsymbol{C} = \boldsymbol{U}\Lambda\boldsymbol{U}^T, \tag{6.6}$$

where each column of $\boldsymbol{U}$ is a $N_p \times 1$ basis, and $\boldsymbol{U} = \left[\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_{N_p}\right]$. Therefore, the gradient map $\boldsymbol{h}_{Q+t,1}$ is defined as the sum of absolute value of PCA responses,

$$\boldsymbol{h}_{Q+t,1,j} = \sum_{k=2}^{M} |\boldsymbol{u}_k^T \boldsymbol{a}_j|, \quad \text{for } j = 1, 2, ...N. \tag{6.7}$$

In this work, we choose $M = 16$ and $N_p = 121$, and normalize $\boldsymbol{h}_{Q+t,1}$ to the range $[0, 1]$.

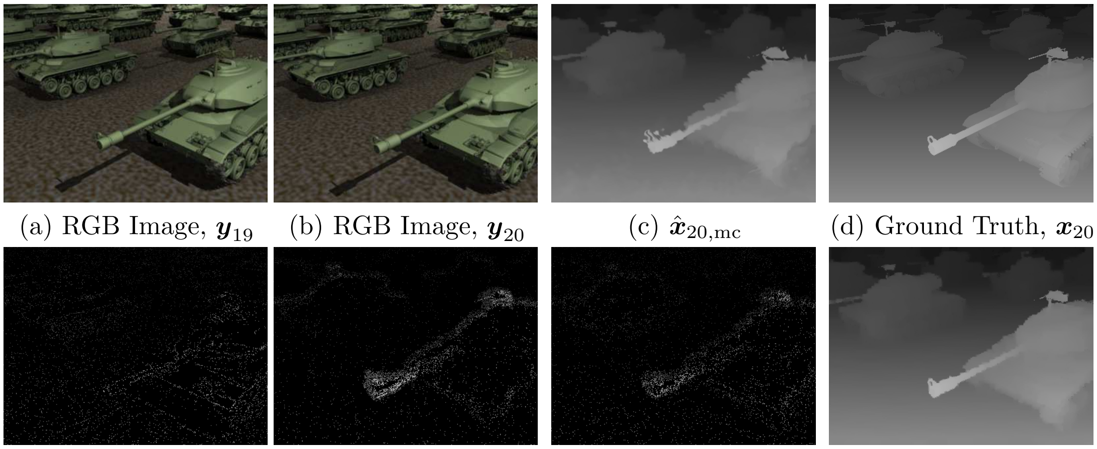
Obtaining $\boldsymbol{h}_{Q+1,2}$

To estimate gradients of corresponding RGB image, we apply

$$\boldsymbol{h}_{Q+t,2} = |\boldsymbol{D}_h \boldsymbol{y}_{Q+t}| + |\boldsymbol{D}_v \boldsymbol{y}_{Q+t}|, \tag{6.8}$$

where $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$ are horizontal and vertical difference operators. As edges in RGB image might be the edges in the depth image, instead of using PCA responses, we propose to use gradients. Similar to $\boldsymbol{h}_{Q+t,1}$, we also normalize the gradient map $\boldsymbol{h}_{Q+t,2}$ to the range $[0, 1]$.

Figure 6.1 shows snapshots of intermediate images. The motion compensated depth image, $\hat{\boldsymbol{x}}_{20,\text{mc}}$ is synthesized from previously reconstructed depth image $\hat{\boldsymbol{x}}_{19}$ with the estimated motion vector between $\boldsymbol{y}_{19}$ and $\boldsymbol{y}_{20}$. Then, $\hat{\boldsymbol{x}}_{20,\text{mc}}$ and $\boldsymbol{y}_{20}$ are utilized for estimating $\boldsymbol{h}_{20,1}$ and $\boldsymbol{h}_{20,2}$. Finally, the sampling locations are determined using the proposed method. Figure 6.1 (e)-(g) are examples of varying θ_1. If we set $\theta_1 = 0$, the samples locate mainly at the edges of the corresponding view images. If we set $\theta_1 = 1$, sampling locations are biased by the errors from the motion compensated depth image.



(a) RGB Image, $\boldsymbol{y}_{19}$ (b) RGB Image, $\boldsymbol{y}_{20}$ (c) $\hat{\boldsymbol{x}}_{20,\text{mc}}$ (d) Ground Truth, $\boldsymbol{x}_{20}$

(e) $\boldsymbol{b}_{20}$ with $\theta_1 = 0$ (f) $\boldsymbol{b}_{20}$ with $\theta_1 = 1$ (g) $\boldsymbol{b}_{20}$ with $\theta_1 = 0.6667$ (h) Rec. Disparity, $\hat{x}_{20}$

Figure 6.1: Example of reference images, sampled depth data with varying θ_1 and reconstructed disparity map. (c) is a motion compensated depth image from $\hat{\boldsymbol{x}}_{19}$ using estimated motion vectors between $\boldsymbol{y}_{19}$ and $\boldsymbol{y}_{20}$. (h) is a reconstructed disparity map from (g), and sampling rates for (e)-(g) are all $\xi = 0.0489$.

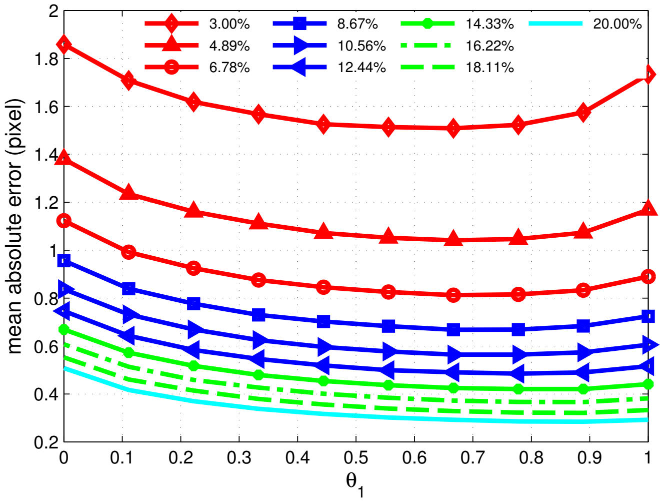


Figure 6.2: Mean absolute curves with varying θ_1 values. We evaluate the reconstruction performance using "tank" sequence (100 frames), we set $T = 1$ for the proposed STDR.

A comparison of MAE curves with varying θ_1 is shown in Figure 6.2. It is visible that the lowest MAE value locates in the range $[0.6, 0.7]$. This justifies that utilizing a linear combination of $\boldsymbol{h}_{Q+t,1}$ and $\boldsymbol{h}_{Q+t,2}$ gives rise to better reconstruction performance than using single gradient map alone. Therefore, we pick $\theta_1 = 0.6667$ as a default. A reconstructed depth image with $\theta = 0.6667$ and $\xi = 0.0489$ is shown in Figure 6.1 (h), and the overall motion compensation assisted sampling (MCAS) strategy is presented in Algorithm 7.

Algorithm 7 Motion Compensation Assisted Sampling (MCAS) Scheme

```
Require: $\xi$, $T$, $\theta_1$, $\hat{\boldsymbol{x}}_{Q+1,\text{mc}} \cdots \hat{\boldsymbol{x}}_{Q+T,\text{mc}}$, $\boldsymbol{y}_{Q+1} \cdots \boldsymbol{y}_{Q+T}$
for $t = 1$ to $T$ do
    Estimate $\boldsymbol{h}_{Q+t,1}$ using (6.6) and (6.7).
    Estimate $\boldsymbol{h}_{Q+t,2}$ using (6.8).
    $\boldsymbol{h}_{Q+t} = \theta_1 \boldsymbol{h}_{Q+t,1} + (1 - \theta_1)\, \boldsymbol{h}_{Q+t,2}$.
end for
Estimate $\boldsymbol{p}_{Q+1}, \cdots, \boldsymbol{p}_{Q+T}$ using (6.5).
Determine $\boldsymbol{S}$ based on $\boldsymbol{p}_{Q+1}, \cdots, \boldsymbol{p}_{Q+T}$ and $\xi$.
return $\boldsymbol{S}$.
```

6.1.3 Framework for Depth Video Reconstruction

Given a fixed size of temporal volume, T, we have presented a motion compensation assisted sampling (MCAS) scheme and a spatio-temporal depth reconstruction algorithm (STDR) for depth video reconstruction framework. However, the determination of temporal volume T relates to the accuracy of motion vector estimation and compensation algorithm. Given a motion vector search limit, r_{max} and the maximum temporal volume size, $T_{\max}$, we determine the size T as (1) (maximum of $\boldsymbol{v}_{Q,Q+t}$) $\leq (0.5 \times r_{max})$ and (2) $T \leq T_{max}$. Finally, the overall depth video reconstruction algorithm is shown in Algorithm 8.

Algorithm 8 Depth Video Reconstruction

```
Require: $\xi$, $T_{max}$, $r_{max}$, $\theta_1$,$\hat{\boldsymbol{x}}_Q$, $\boldsymbol{y}_Q$, $\boldsymbol{y}_{Q+1} \cdots \boldsymbol{y}_{Q+T}$
if $Q == 0$ then
    $T = 1$.
    $\boldsymbol{S}$ = Uniformly Random Sampling($\xi$).
    $\boldsymbol{b}$ = Depth Estimation($\boldsymbol{S}$).
    $\hat{\boldsymbol{x}}$ = STDR($\boldsymbol{S}, \boldsymbol{b}$).
else
    $t = 1$
    $\boldsymbol{v}_{Q,Q+t}$ = Motion Vector Estimation$\big(\boldsymbol{y}_Q, \boldsymbol{y}_{Q+t}\big)$.
    $\hat{\boldsymbol{x}}_{Q+t,\mathrm{mc}}$ = Motion Compensation($\hat{\boldsymbol{x}}_Q, \boldsymbol{v}_{Q,Q+t}$).
    while $t \leq T_{max}$ and $\boldsymbol{v}_{Q,Q+t} \leq (0.5 \times r_{max})$ do
        $t = t + 1$.
        $\boldsymbol{v}_{Q,Q+t}$ = Motion Vector Estimation$\big(\boldsymbol{y}_Q, \boldsymbol{y}_{Q+t}\big)$.
        $\hat{\boldsymbol{x}}_{Q+t,\mathrm{mc}}$ = Motion Compensation($\hat{\boldsymbol{x}}_Q, \boldsymbol{v}_{Q,Q+t}$).
    end while
    $T = t$.
    $\boldsymbol{S}$ = MCAS$\big(\xi, T, \theta_1, \hat{\boldsymbol{x}}_{Q+t,\mathrm{mc}}, \boldsymbol{y}_{Q+t}, t = 1, ...T\big)$.
    $\boldsymbol{b}$ = Depth Estimation($\boldsymbol{S}$).
    $\hat{\boldsymbol{x}}$ = STDR($\boldsymbol{S}, \boldsymbol{b}, \hat{\boldsymbol{x}}_{Q+1,\mathrm{mc}}, \cdots, \hat{\boldsymbol{x}}_{Q+T,\mathrm{mc}}$).
end if
$Q = Q + T$.
return $\hat{\boldsymbol{x}}_{Q+1}$, $\hat{\boldsymbol{x}}_{Q+2}$, $\cdots$, $\hat{\boldsymbol{x}}_{Q+T}$.
```

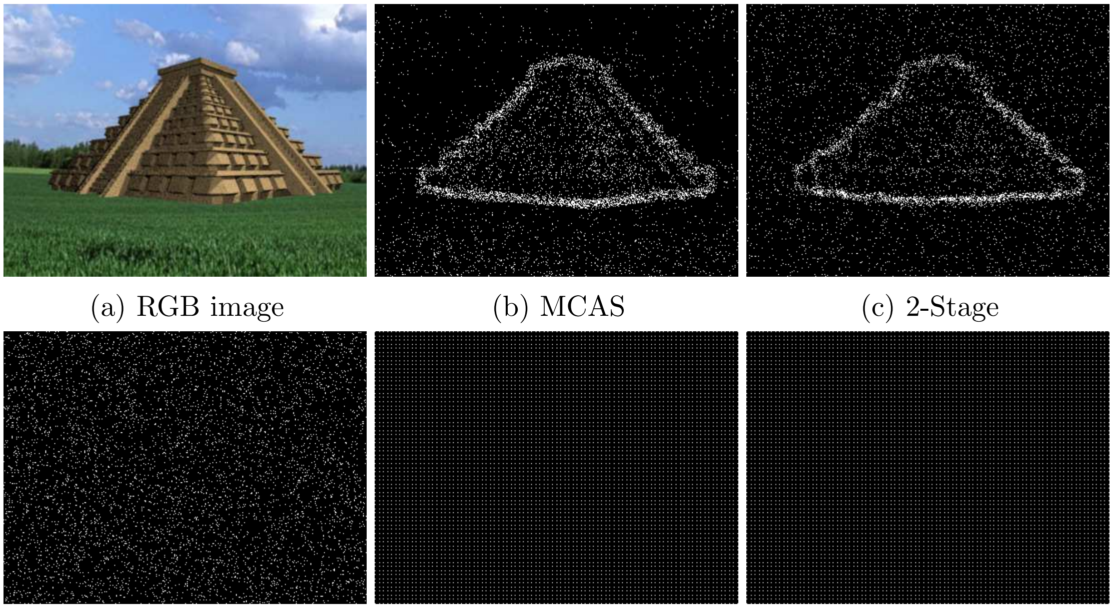
6.2 Experimental Results and Discussions

In this section, we first compare the proposed depth video reconstruction framework to existing depth reconstruction methods given a subset of ground truth depth samples. Then, we present an application that utilizing our proposed framework for depth video reconstruction from low resolution ground truth depth videos. Finally, we show the depth video reconstruction from a subset of real estimated depth data.

6.2.1 Depth Video Reconstruction with Ground Truth Depth

To evaluate the depth video reconstruction framework from a subset of samples, we herein utilize the synthetic depth video dataset provided in [78]. To the best of our knowledge, this work is the first work on depth video reconstruction from a subset of samples, therefore, we justify the reconstruction performance using existing single-frame based sparse reconstruction methods. In the comparisons, we consider two standard interpolation methods, “bicubic” and “guided filter [70]”, as benchmarks. Since RGB images are available in our systematic configuration, we therefore consider guided filter as an additional benchmark. Note that for the “guided filter”, we apply window sizes [7 5 5 5], [9 7 5 5] and [7 5 5 5] for “temples”, “books” and “tanks” sequences respectively. Each window size in the vector relates to $[5\%, 10\%, 15\%, 20\%]$ sampling rates. For our framework we use “db2” as the wavelet function with two decomposition levels and $T_{\max} = 5$ in the proposed STDR algorithm. In the proposed MCAS scheme, we set $\theta_1 = 0.6667$. We additionally compare our framework to our previously work for single depth image reconstruction [19]. We apply the “2-Stage” algorithm for the sampling strategy and use our proposed STDR algorithm with $T = 1$ for reconstruction method. The main differences between this work and [19] is that the proposed MCAS strategy is a “1-Stage” sampling method which utilizes temporal information for estimating optimal sampling locations; therefore, no pilot signal is required in the proposed MCAS scheme.

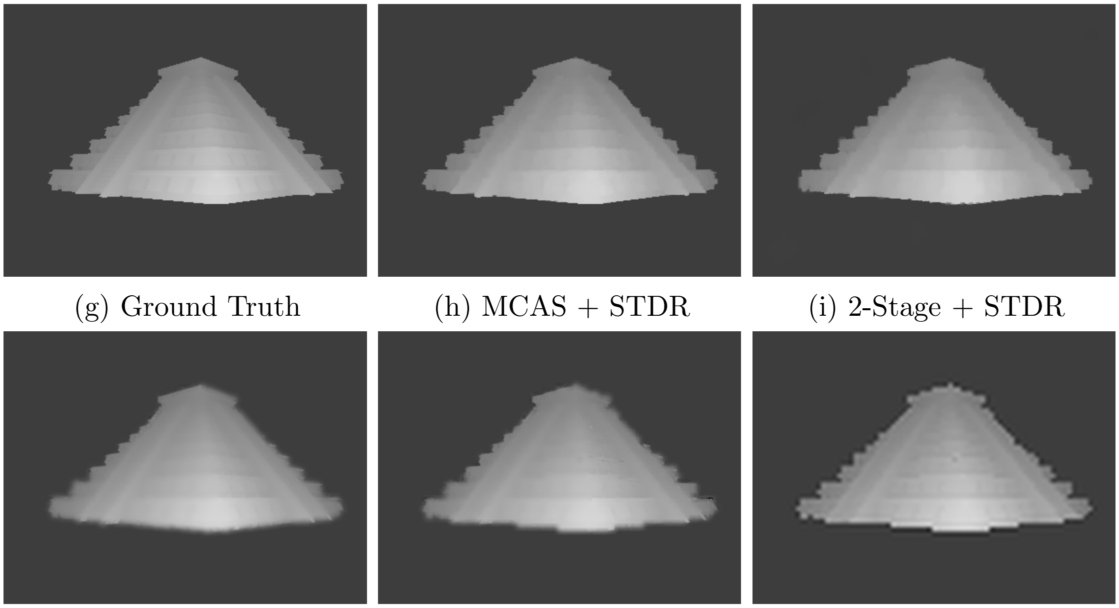
Table 6.1 compares the reconstruction performance. On average, our proposed “MCAS + STDR” approach is very competitive comparing to the “2-Stage [19] + STDR” approach, which conducts reconstruction twice, whereas “MCAS + STDR” utilizes temporal information and is a “1-Stage” sampling approach. Comparing to existing methods, our proposed "MCAS + STDR" mostly outperforms the other methods. Moreover, as sampling rate increases from 5% to 20%, both "MCAS + STDR" and "2-Stage + STDR" have significant performance improvement, whereas the improvement on the other existing methods is limited. Visual comparisons of sampling maps and reconstructed depth images are shown in Figure 6.3. Observing Figure 6.3 (h)-(l), both the "MCAS + STDR" and "2-Stage + STDR" have better visual quality than the others. Observing Figure 6.3 (j)(k), we also can tell that the results with guided filter interpolations have blurry boundaries around the bottom of the temple because of the lack of intensity distinctions in Figure 6.3 (a). From Figure 6.3 (l), the "Uniformly Grid + Bicubic" yields results with erroneous object boundaries. Overall, both "MCAS + STDR" and "2-Stage + STDR" achieve the best visual quality.



(a) RGB image (b) MCAS (c) 2-Stage

(d) Uniformly Random (e) Uniformly Grid (f) Uniformly Grid

Figure 6.3: Examples of sampling maps and reconstructed results of 8th frame of temple sequence with $\xi = 0.05$.



(g) Ground Truth (h) MCAS + STDR (i) 2-Stage + STDR

(j) Uniformly Random + [70] (k) Uniformly Grid + [70] (l) Uniformly Grid + Bicubic

Figure 6.3: Examples of sampling maps and reconstructed results of 8th frame of temple sequence with $\xi = 0.05$. [Cont.]

Table 6.1: Comparisons for depth video reconstruction algorithms. All methods are evaluated over the whole depth video sequence. Noting that “2-Stage [19] + STDR” utilizes STDR algorithm with volume size $T = 1$. We bold the values with the best performance and underline the second best results.

Sequence Name		Tanks				Books				Temples			
Sampling Rates	Methods	PSNR (dB)	MAE (pixels)	% Bad Pel. τ=1	τ=2	PSNR (dB)	MAE (pixels)	% Bad Pel. τ=1	τ=2	PSNR (dB)	MAE (pixels)	% Bad Pel. τ=1	τ=2
5%	2-Stage [19] + STDR	36.7128	**0.9818**	**10.84**	**6.59**	44.1131	0.4683	6.58	3.65	36.6180	0.5837	7.59	4.04
	MCAS + STDR	**37.4331**	1.0029	12.37	7.64	**45.0440**	0.4690	**6.48**	**3.60**	**37.3281**	**0.4807**	**5.76**	**2.97**
	Uni. Random + [70]	36.9283	1.1521	15.01	11.20	43.7558	**0.4141**	6.71	4.08	32.3629	1.4933	12.45	9.02
	Uni. Grid + [70]	36.1690	1.2554	14.60	10.94	43.3343	0.4241	6.65	4.09	30.9911	1.5973	11.91	8.71
	Uni. Grid + Bicubic	35.4886	1.2540	14.47	10.22	41.6948	0.5725	6.50	4.10	30.6302	1.6906	14.79	10.79
10%	2-Stage [19] + STDR	**39.9088**	**0.5626**	**6.01**	**3.78**	**49.4422**	**0.2264**	2.56	1.25	42.5058	0.2671	4.19	1.87
	MCAS + STDR	39.6939	0.6231	6.84	4.21	48.5476	0.2558	**2.39**	**1.10**	**42.7049**	**0.2170**	**2.79**	**1.27**
	Uni. Random + [70]	38.0454	0.8981	11.78	8.71	44.6226	0.3357	5.37	3.31	33.8017	1.0792	8.61	6.25
	Uni. Grid + [70]	37.3508	0.9765	11.45	8.55	44.2932	0.3503	5.44	3.37	32.5952	1.1451	8.40	6.14
	Uni. Grid+ Bicubic	36.8395	0.9562	10.99	7.74	43.2528	0.4187	4.57	2.87	32.2175	1.2022	10.36	7.45
15%	2-Stage [19] + STDR	**42.6989**	**0.3753**	**3.98**	**2.43**	**52.9466**	**0.1400**	1.21	0.58	**50.1001**	0.1456	2.43	0.97
	MCAS + STDR	41.7544	0.4357	4.55	2.76	50.9933	0.1706	**1.08**	**0.49**	48.4641	**0.1158**	**1.52**	**0.62**
	Uni. Random + [70]	38.0594	0.8958	11.78	8.71	45.7139	0.2545	3.99	2.49	33.8191	1.0790	8.62	6.26
	Uni. Grid + [70]	37.6830	0.9521	11.57	8.58	45.2134	0.2965	3.98	2.47	33.1420	1.1116	8.34	6.10
	Uni. Grid + Bicubic	37.5574	0.8277	9.18	6.54	43.9881	0.3783	3.64	2.44	33.0614	0.9888	8.17	5.88

6.3 Depth Video Reconstruction from Uniform-Grid Subsampled Data

The proposed framework can also deal with the problem that input samples are from uniformly subsampled data (e.g., $\downarrow M, M = 2, 4, 8$) by introducing an inference operation. More specifically, the input data are acquired from uniformly sampling grid, whereas the proposed sampling locations from the proposed MCAS scheme are not the same as inputs, observing Figure 6.3 (b) and (e). We need to estimate those missing samples determined by the proposed MCAS scheme. We herein utilize the inference operation presented in [80]. Let $\mathcal{G}$ to be a set of indices (representing in full-resolution) from those downsampled data, $i \in \mathcal{G}$, and let $\mathcal{S}$ to be a set of indices predicted by the proposed MCAS scheme. Our goal is to estimate depth information of indices $j \in \{\mathcal{S}/\mathcal{S} \cap \mathcal{G}\}$ using the RGB image $\boldsymbol{y}$ and the downsampled data $\boldsymbol{x}_{\downarrow D}$. For each pixel j to be estimated, we find the K closest indices $k \in \mathcal{K}$, and estimate the depth by

$$k^* = \underset{k \in \mathcal{K}}{\operatorname{argmin}} \|\boldsymbol{y}_j - \boldsymbol{y}_k\|^2. \tag{6.9}$$

Then, we assign the missing depth pixel by

$$\hat{\boldsymbol{b}}_j = \boldsymbol{x}_{\downarrow M \uparrow M, k^*}. \tag{6.10}$$

Examples of reconstructed depth images are shown in Figure 6.4, in which we input downsampled depth data, $\boldsymbol{x}_{\downarrow 8}$. Figure 6.4 (b) shows the input samples (including sub-sampled and estimated depth data) applied to our proposed depth reconstruction algorithm. Reconstructed results to the proposed method is shown in (c), and to the state-of-the art method [66] is shown in (d). Figure 6.4 (e) shows bicubic interpolated depth image. In terms of visual quality, both the proposed method and [66] are better than bicubic method. Observing the table in Figure 6.4, it is obvious that the proposed method is competitive to [66], and we see that the proposed method achieves the highest PSNR as the sub-sampling factor is at 8, which justifies that the proposed framework is suited for the case that sampling rate is low. We realize that the proposed model is not exactly the same as depth image super-resolution as we do not consider anti-aliasing and anti-imaing filtering during the down/up-sampling operations.

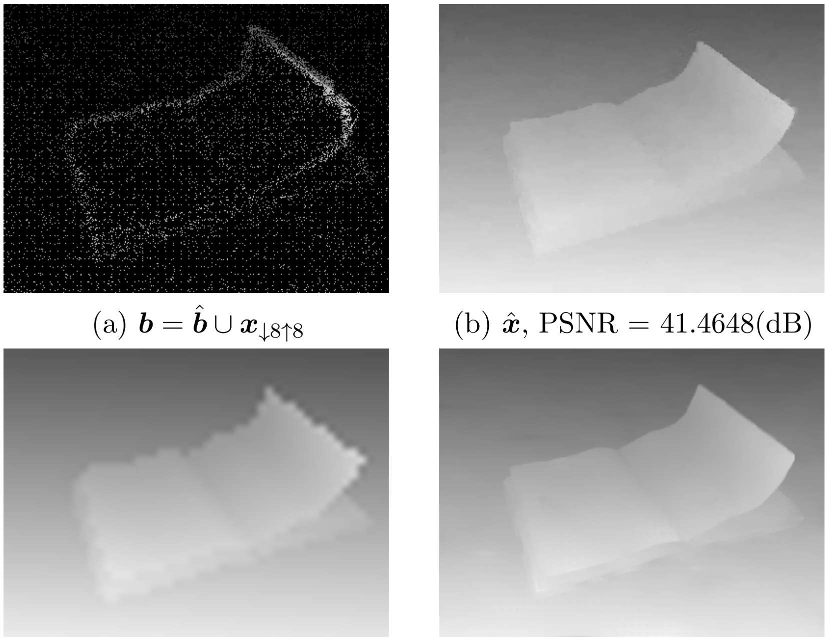
6.4 Dense Disparity Video Estimation

The proposed depth video reconstruction framework is also applicable for disparity video estimation. In practice, a classical problem to dense disparity estimation is the trade-off between the blurry effect on object boundaries and selection of patch window sizes [81]. Both large and small window sizes could lead to erroneous estimated disparity values. Therefore, we propose to estimate reliable disparities using multiple window sizes with mean absolute difference (MAD) as a cost function,

$$\begin{aligned} d^* &= \underset{d}{\operatorname{argmin}} \|\boldsymbol{Y}_{\mathrm{L}}(i,j) - \boldsymbol{Y}_{\mathrm{R}}(i,j+d)\|_1, \\ \boldsymbol{D}\left(i',j'\right) &= d^*, \text{ for } (i,j) \in \mathcal{W}, \end{aligned} \tag{6.11}$$

where $\mathcal{W}$ is a set of indices relating to a given window size W centered at (i',j'). $\boldsymbol{Y}_{\mathrm{L}}$ and $\boldsymbol{Y}_{\mathrm{R}}$ are the left and right view of RGB images. $\boldsymbol{D}$ is the estimated disparity map. In this work, we obtain reliable samples by searching for the majority of disparities while using different window sizes (e.g., 3×3, ...11×11) and thus obtain a subset of estimated reliable disparity samples, $\boldsymbol{x}_{\mathrm{est}}$ (canonical representation of depth image). Then, we conduct the same approach as mentioned in previous subsection, inferring predicted to-be-estimated samples using RGB images and estimated depth data, and finally we reconstruct dense disparity video using the proposed STDR algorithm. Note that only requested samples from the proposed MCAS scheme are used in the method.

A snapshot of densely reconstructed depth video of "tanks" sequence is shown in Figure 6.5. Dense disparity maps with different window sizes are presented in Figure 6.5 (a)-(e), and selected reliable samples are shown in Figure 6.5 (f). According to the proposed MCAS scheme, required sparse samples are shown in Figure 6.5 (g). Equations (6.9) and (6.10) are used to obtain sparse measurements, $\boldsymbol{b}$. Finally, we use proposed STDR algorithm to obtain the dense disparity map, shown in (i). The average mean absolute errors of estimated disparity maps are 9.13984, whereas the proposed scheme with



(a) $\boldsymbol{b} = \hat{\boldsymbol{b}} \cup \boldsymbol{x}_{\downarrow 8 \uparrow 8}$

(b) $\hat{\boldsymbol{x}}$, PSNR = 41.4648(dB)

(c) Bicubic, MSE = 38.5881(dB)

(d) [66], PSNR = 40.8392(dB)

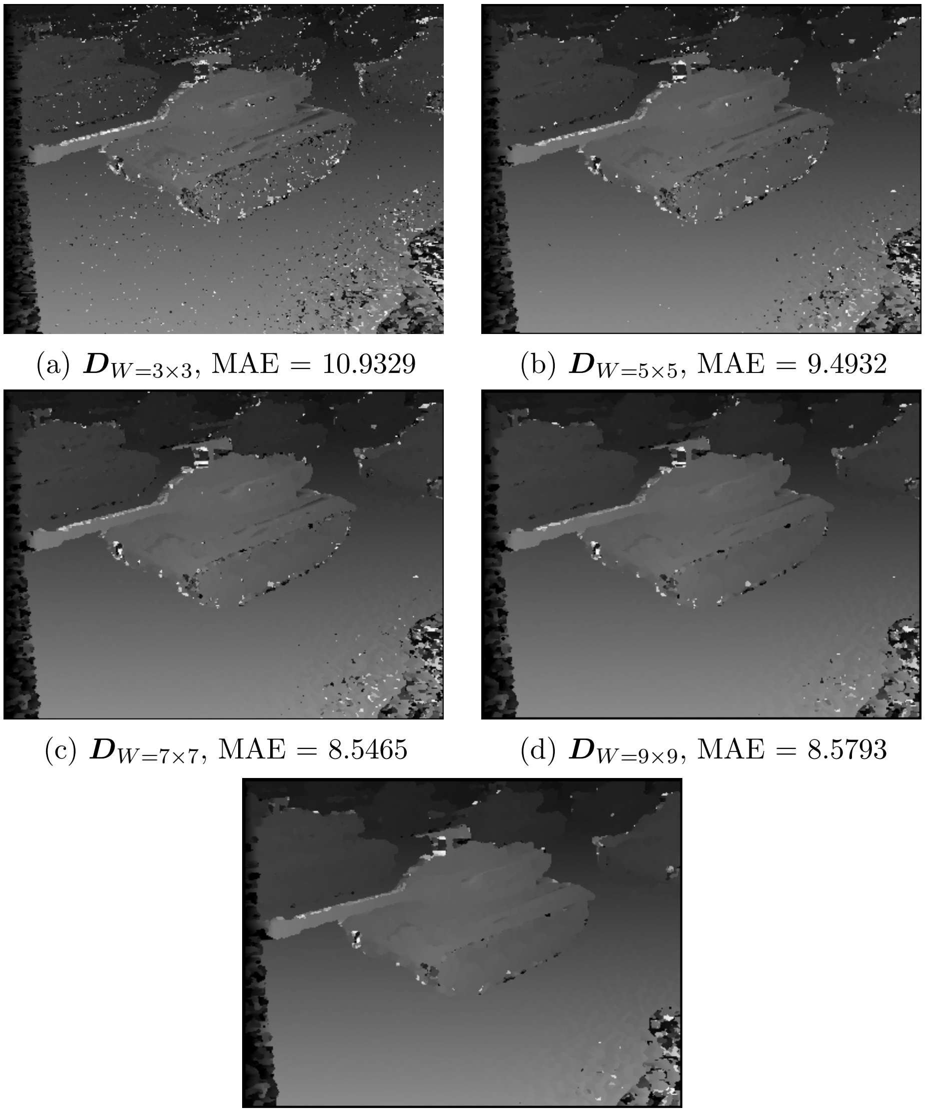
Sequence Name	Tanks			Books		
Methods / Factor	2×	4×	8×	2×	4×	8×
Bicubic	37.8119	33.6391	30.5019	44.6412	40.4048	36.3996
Ferstl [66]	**39.8516**	**36.7817**	32.5354	**46.9691**	**44.7886**	39.2353
Proposed	38.7744	36.0169	**33.6247**	46.1762	43.8714	**40.1676**

Sequence Name	Temples		
Bicubic	33.5664	28.8665	25.6860
Ferstl [66]	35.7973	32.8430	27.2135
Proposed	**37.0280**	**34.2475**	**30.7941**

Figure 6.4: Snapshots of reconstructing high resolution depth from downsampled depth data, $\boldsymbol{x}_{\downarrow 8}$, and PSNR comparisons. Note that we show reconstructed results by applying the 28th frame of "books" sequence.

5% estimated reliable samples as inputs achieves 2.4369. Therefore, the proposed framework can further be utilized for dense disparity estimation for depth video sequences. Additional results and comparisons are shown in Figure 6.6, Figure 6.7, and Table 6.2.

This Chapter includes materials that have been submitted to IEEE Transaction on Image Processing, titled "A Framework for Depth Video Reconstruction from a subset of Samples and its Applications," with Truong Q. Nguyen.



(a) $\boldsymbol{D}_{W=3\times 3}$, MAE = 10.9329

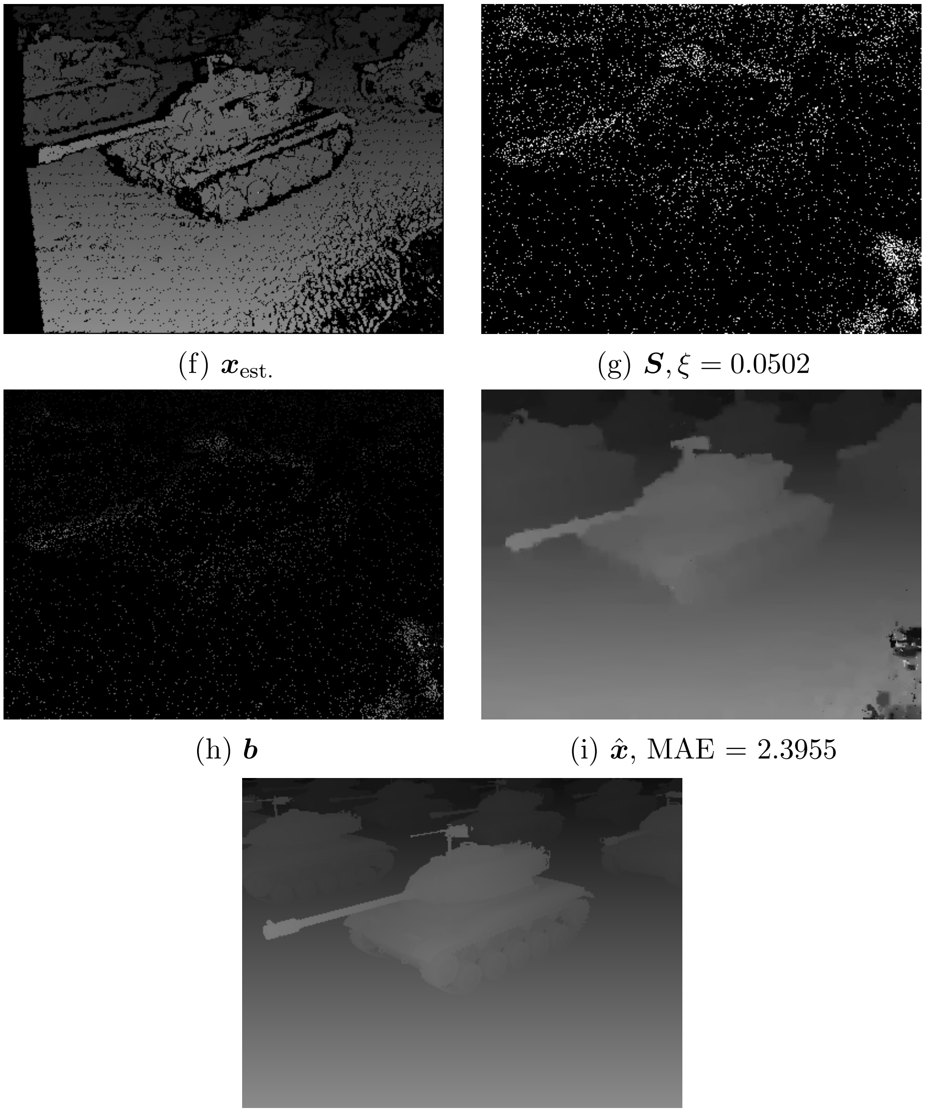
(b) $\boldsymbol{D}_{W=5\times 5}$, MAE = 9.4932

(c) $\boldsymbol{D}_{W=7\times 7}$, MAE = 8.5465

(d) $\boldsymbol{D}_{W=9\times 9}$, MAE = 8.5793

(e) $\boldsymbol{D}_{W=11\times 11}$, MAE = 8.1473

Figure 6.5: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$. We show an example of the 58th frame of “tanks” sequence.



(f) $\boldsymbol{x}_{\text{est.}}$

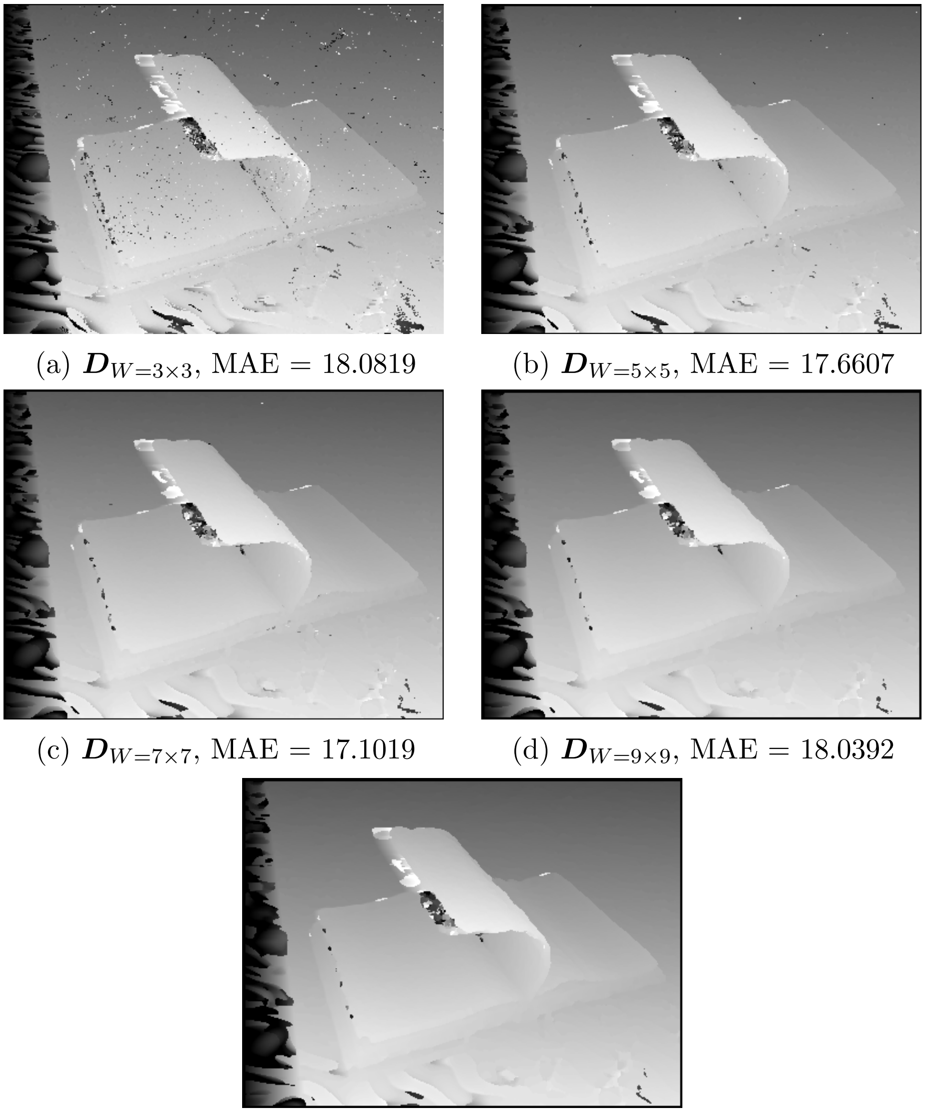
(g) $\boldsymbol{S}, \xi = 0.0502$

(h) $\boldsymbol{b}$

(i) $\hat{\boldsymbol{x}}$, MAE = 2.3955

(j) $\boldsymbol{x}$, Ground Truth

Figure 6.5: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$. We show an example of the 58th frame of "tanks" sequence [Cont.].



(a) $\boldsymbol{D}_{W=3\times3}$, MAE = 18.0819

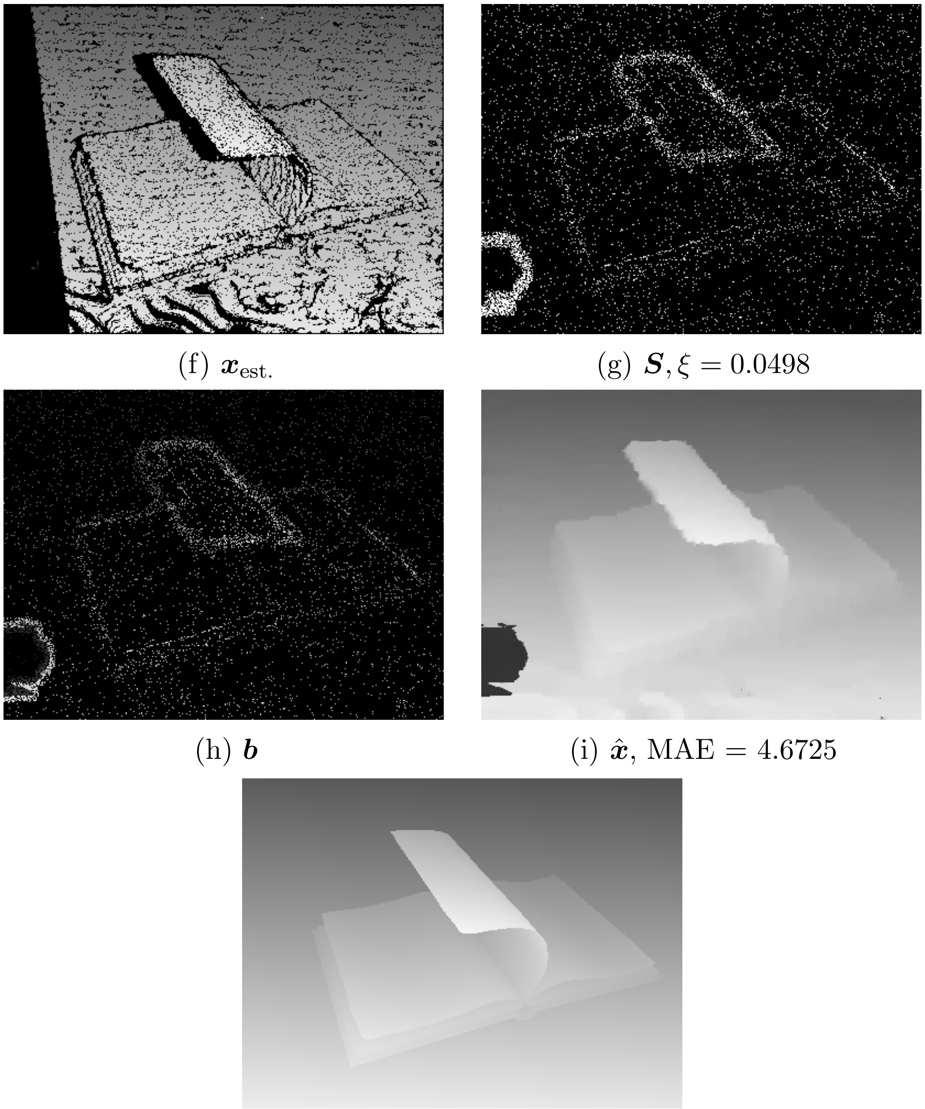
(b) $\boldsymbol{D}_{W=5\times5}$, MAE = 17.6607

(c) $\boldsymbol{D}_{W=7\times7}$, MAE = 17.1019

(d) $\boldsymbol{D}_{W=9\times9}$, MAE = 18.0392

(e) $\boldsymbol{D}_{W=11\times11}$, MAE = 17.7302

Figure 6.6: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$. We show an example of the 17th frame of "books" sequence.



(f) $\boldsymbol{x}_{\text{est.}}$

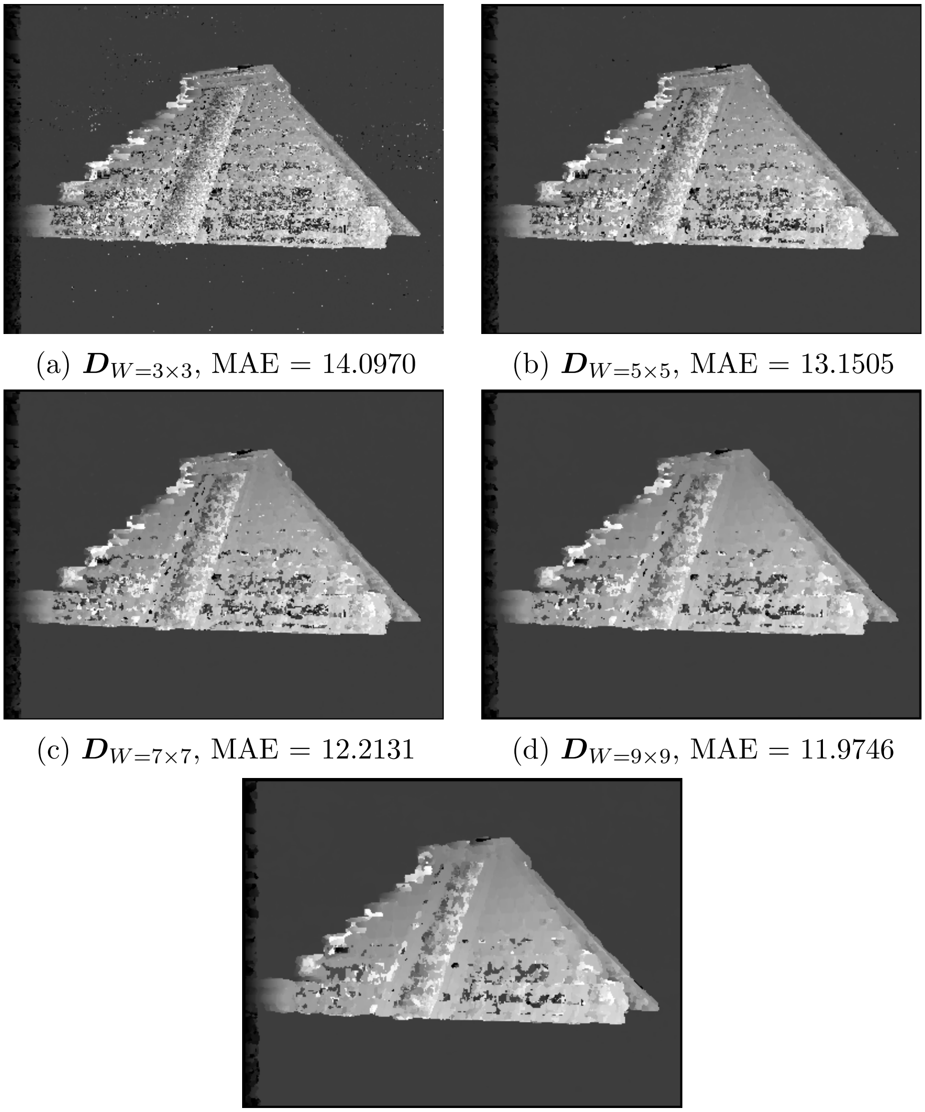
(g) $\boldsymbol{S}, \xi = 0.0498$

(h) $\boldsymbol{b}$

(i) $\hat{\boldsymbol{x}}$, MAE = 4.6725

(j) $\boldsymbol{x}$, Ground Truth

Figure 6.6: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$. We show an example of the 17th frame of “books” sequence [Cont.].



(a) $\boldsymbol{D}_{W=3\times 3}$, MAE = 14.0970

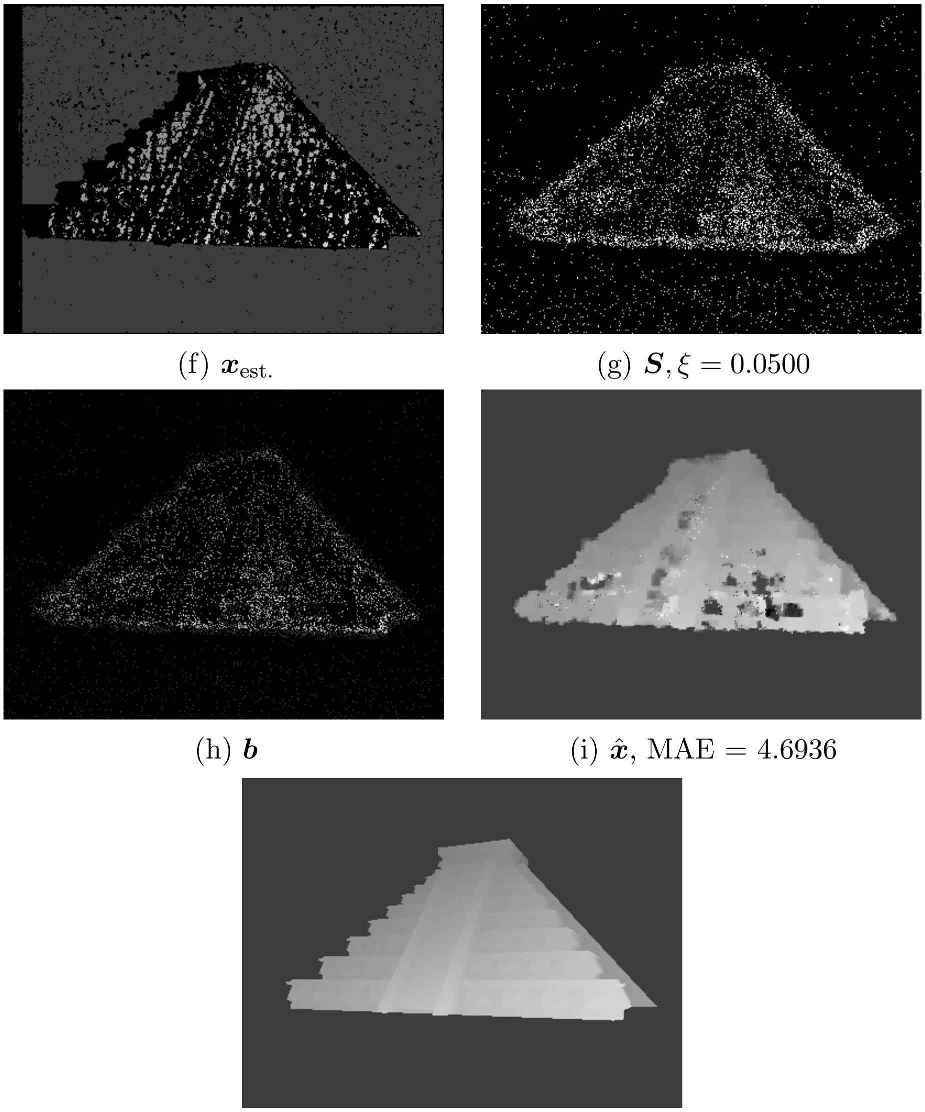
(b) $\boldsymbol{D}_{W=5\times 5}$, MAE = 13.1505

(c) $\boldsymbol{D}_{W=7\times 7}$, MAE = 12.2131

(d) $\boldsymbol{D}_{W=9\times 9}$, MAE = 11.9746

(e) $\boldsymbol{D}_{W=11\times 11}$, MAE = 11.5223

Figure 6.7: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$. We show an example of the 69th frame of "temples" sequence.



(f) $\boldsymbol{x}_{\text{est.}}$

(g) $\boldsymbol{S}, \xi = 0.0500$

(h) $\boldsymbol{b}$

(i) $\hat{\boldsymbol{x}}$, MAE = 4.6936

(j) $\boldsymbol{x}$, Ground Truth

Figure 6.7: Example of reconstructing high resolution depth from estimated depth data, $\boldsymbol{x}_{\text{est.}}$. We show an example of the 69th frame of "temples" sequence [Cont.].

Table 6.2: MAE and Bad pixel % comparisons for the depth video estimations. All methods are evaluated over the whole depth video sequence.

Dataset	Tanks		Books		Temples	
Methods	MAE (pixels)	Bad pixel %	MAE (pixels)	Bad pixel %	MAE (pixels)	Bad pixel %
$\boldsymbol{D}_{W=3\times3}$	10.1042	33.79	17.3273	34.58	14.5321	30.82
$\boldsymbol{D}_{W=5\times5}$	9.0583	29.46	16.9273	30.56	13.5383	29.14
$\boldsymbol{D}_{W=7\times7}$	8.3279	26.90	16.3529	29.36	12.6090	27.59
$\boldsymbol{D}_{W=9\times9}$	8.5258	25.65	17.2708	29.25	12.3026	27.30
$\boldsymbol{D}_{W=11\times11}$	8.1767	24.41	16.9525	29.53	11.7346	26.33
Proposed	**2.0022**	**21.32**	**4.3477**	**24.39**	**4.3517**	**18.20**

Appendix A

Derivation on Reconstruction Algorithm

The following notations are used for all further discussion involving the derivation of dense disparity reconstruction algorithm.

m	:Number of Observations.
N	:Total number of pixels of disparity map.
$\boldsymbol{\Phi}_1 \in \mathbb{R}^{N\times N}$	:Wavelet Bases.
$\boldsymbol{W}_1 \in [0,1]^{N\times N}$	:Weighting Matrices for Wavelet Bases.
$\boldsymbol{\Phi}_2 \in \mathbb{R}^{N\times N}$	:Contourlet Bases.
$\boldsymbol{W}_2 \in [0,1]^{N\times N}$	:Weighting Matrices for Contourlet Bases.
$\boldsymbol{S} \in [0,1]^{n\times n}$	:Sampling Matrix.
$\boldsymbol{b} \in \mathbb{R}^{N\times 1}$	:Observations (sparse samples).
$\boldsymbol{x} \in \mathbb{R}^{N\times 1}$	:Disparity map.
$\boldsymbol{c} \in \mathbb{R}^{N\times 1}$	:Wavelet Transform Coefficients of Disparity map.

A.1 Derivation of Subgradient Algorithm

For reconstructing dense disparity maps, we propose a subgradient algorithm for solving the unconstrained minimization equation.

$$\underset{\boldsymbol{c}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{S}\boldsymbol{\Phi}_1\boldsymbol{c}\|_2^2 + \lambda\|\boldsymbol{W}_1\boldsymbol{c}\|_1 + \gamma\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1 + \beta\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\text{TV}}. \tag{A.1}$$

In order to solve problem in (A.1), we focus on the first order method since the data size is too large while dealing with image processing. For considering the complexity, we address on the first order method. As the first-order method requires the gradient of (A.1), calculating gradient of each terms is our first step. Since the matrix $\boldsymbol{W}_1$ is a diagonal matrix, the subdifferential of $\|\boldsymbol{W}_1\boldsymbol{c}\|_1$ is as follows:

$$\partial_{\boldsymbol{c}}\|\boldsymbol{W}_1\boldsymbol{c}\|_1(i) = \begin{cases} \text{sign}\left[(\boldsymbol{W}_1\boldsymbol{c})(i)\right], & \text{if } (\boldsymbol{W}_1\boldsymbol{c})(i) \neq 0, \\ [-1,1], & \text{otherwise.} \end{cases} \tag{A.2}$$

where the definition of the sign function is defined.

$$\text{sign}(v) = \begin{cases} 1, & \text{if} \quad v > 0, \\ 0, & \text{if} \quad v = 0, \\ -1, & \text{if} \quad v < 0. \end{cases} \tag{A.3}$$

As the operation of matrices $\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1$ is not a diagonal process, the subdifferential of $\boldsymbol{c}$ is defined as follows:

$$\partial_{\boldsymbol{c}}\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1(i) = \left\{\boldsymbol{\Phi}_1^T\boldsymbol{\Phi}_2\text{sign}\left[\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\right]\right\}(i). \tag{A.4}$$

Proof. Instead of using 2D signal decomposition, we consider 1D signal. Given a signal $\boldsymbol{s}$ of length $\mathrm{N} = 2^P$ and transform matrix, $\boldsymbol{\Phi}$, the signal is decomposed by a wavelet function at level L with scaling functions, $\boldsymbol{\phi}_{L,k}$, and wavelet functions, $\boldsymbol{\varphi}_{\ell,k}$.

$$vs = \sum_{k=0}^{2^L-1} a_{L,k}\boldsymbol{\phi}_{L.k} + \sum_{\ell=L+1}^{P}\sum_{k=0}^{2^{\ell-1}-1} a_{\ell,k}\boldsymbol{\varphi}_{\ell,k}. \tag{A.5}$$

The ℓ_1 norm of the function $\|\boldsymbol{\Phi s}\|_1$ is as follows.

$$\|\boldsymbol{\Phi s}\|_1 = \sum_{k=0}^{2^L-1} |a_{L,k}| + \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} |a_{\ell,k}|. \tag{A.6}$$

Given another signal $\boldsymbol{u}$, and it can also be decomposed as

$$\boldsymbol{u} = \sum_{k=0}^{2^L-1} b_{L,k}\boldsymbol{\phi}_{L.k} + \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} b_{\ell,k}\boldsymbol{\varphi}_{\ell,k}. \tag{A.7}$$

Therefore, the subdifferential of the ℓ_1 norm is as follows.

$$\begin{aligned}
&\lim_{\alpha\to 0}\frac{1}{\alpha}\left(\|\boldsymbol{\Phi}(\boldsymbol{s}+\alpha\boldsymbol{u})\|_1 - \|\boldsymbol{\Phi s}\|_1\right)\\
&= \lim_{\alpha\to 0}\frac{1}{\alpha}\left\{\sum_{k=0}^{2^L-1} |a_{L,k}+\alpha b_{L,k}| + \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} |a_{\ell,k}+\alpha b_{\ell,k}|\right.\\
&\left. - \sum_{k=0}^{2^L-1} |a_{L,k}| - \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} |a_{\ell,k}|\right\}\\
&= \sum_{k=0}^{2^L-1} \operatorname{sign}(a_{L,k}) b_{L,k} + \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} \operatorname{sign}(a_{\ell,k}) b_{\ell,k}\\
&= \left\langle \left(\sum_{k=0}^{2^L-1} \operatorname{sign}(a_{L,k})\boldsymbol{\phi}_{L,k}\right), \left(\sum_{k=0}^{2^L-1} b_{L,k}\boldsymbol{\phi}_{L,k}\right)\right\rangle\\
&+ \left\langle \left(\sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} \operatorname{sign}(a_{\ell,k}\boldsymbol{\varphi}_{\ell,k})\right), \left(\sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} b_{\ell,k}\boldsymbol{\varphi}_{\ell,k}\right)\right\rangle\\
&= \left\langle \left(\sum_{k=0}^{2^L-1} \operatorname{sign}(a_{L,k})\boldsymbol{\phi}_{L.k} + \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} \operatorname{sign}(a_{\ell,k})\boldsymbol{\varphi}_{\ell,k}\right), \boldsymbol{u}\right\rangle.
\end{aligned}$$

where the operator $\langle\cdot\rangle$ is inner product. Since the orthogonal property of the scaling functions and wavelet functions, the last two steps are valid. Therefore, the subdifferential of $\|\boldsymbol{\Phi s}\|$ is as follows:

$$\sum_{k=0}^{2^L-1} \mathrm{sign}(a_{L,k})\boldsymbol{\phi}_{L.k} + \sum_{\ell=L+1}^{P} \sum_{k=0}^{2^{\ell-1}-1} \mathrm{sign}(a_{\ell,k})\boldsymbol{\varphi}_{\ell,k}. \tag{A.8}$$

Thus, given an orthogonal matrix $\boldsymbol{\Phi}$ and a signal $\boldsymbol{s}$, the subdifferential of $\boldsymbol{s}$ of the ℓ_1 norm can be written in matrix form.

$$\partial_{\boldsymbol{s}}\|\boldsymbol{\Phi s}\|_1(i) = \boldsymbol{\Phi}^{-1}\mathrm{sign}(\boldsymbol{\Phi s}). \tag{A.9}$$

Therefore, suppose $\boldsymbol{\Phi} = \boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1$, and since $\boldsymbol{\Phi}_2^T = \boldsymbol{\Phi}_2^{-1}$ and $\boldsymbol{\Phi}_1^T = \boldsymbol{\Phi}_1^{-1}$, the subdifferential of $\|\mathbf{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1$ is as follows.

$$\begin{aligned}
\partial_{\boldsymbol{c}}\|\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\|_1(i) &= \left\{(\boldsymbol{\Phi}_1)^{-1}\boldsymbol{\Phi}_2^{-T}\boldsymbol{W}_2\mathrm{sign}\left[\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\right]\right\}(i) \\
&= \left\{\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2\boldsymbol{W}_2\mathrm{sign}\left[\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\right]\right\}(i) \\
&= \left\{\boldsymbol{\Phi}_1^T\boldsymbol{\Phi}_2\mathrm{sign}\left[\boldsymbol{W}_2\boldsymbol{\Phi}_2^T\boldsymbol{\Phi}_1\boldsymbol{c}\right]\right\}(i).
\end{aligned}$$

where the variable $\boldsymbol{W}_2$ is a matrix with coefficients 1 and 0 in diagonal entries. Here, we let $\boldsymbol{W}_2^{-1} = \boldsymbol{W}_2^T = \boldsymbol{W}_2$. □

The anisotropic total variation is defined as,

$$\begin{aligned}
\|\boldsymbol{u}\|_{\mathrm{TV}} &= \sum_{k=0}^{k=n-1} \sqrt{(\boldsymbol{e}_k\boldsymbol{D}_x\boldsymbol{u})^2} + \sqrt{(\boldsymbol{e}_k\boldsymbol{D}_y\boldsymbol{u})^2} \\
&= \sum_{k=0}^{k=n-1} |\boldsymbol{e}_k\boldsymbol{D}_x\boldsymbol{u}|_1 + |\boldsymbol{e}_k\boldsymbol{D}_y\boldsymbol{u}|_1.
\end{aligned} \tag{A.10}$$

The gradient of $\|\boldsymbol{u}\|_{\mathrm{TV}}$ is,

$$\partial_{\boldsymbol{c}}\|\boldsymbol{\Phi}_1\boldsymbol{c}\|_{\mathrm{TV}}(k) = \partial_{\boldsymbol{c}}H_\delta\left(\sqrt{(\boldsymbol{e}_k\boldsymbol{D}_x\boldsymbol{\Phi}_1\boldsymbol{c})^2}\right) + \partial_{\boldsymbol{c}}H_\delta\left(\sqrt{(\boldsymbol{e}_k\boldsymbol{D}_y\boldsymbol{\Phi}_1\boldsymbol{c})^2}\right), \tag{A.11}$$

where

$$\partial_{\boldsymbol{c}}H_\delta\left(\sqrt{\boldsymbol{s}_x^2}\right) = \begin{cases} \boldsymbol{\Phi}_1^T\boldsymbol{D}_x^T\boldsymbol{e}_k^T\mathrm{sign}(\boldsymbol{s}_x), & \text{if} \quad |\boldsymbol{s}_x| \geq \delta, \\ \dfrac{\boldsymbol{\Phi}_1^T\boldsymbol{D}_x^T\boldsymbol{e}_k^T\boldsymbol{s}_x}{\delta}, & \text{otherwise.} \end{cases} \tag{A.12}$$

and

$$\partial_{\boldsymbol{c}} H_\delta\left(\sqrt{\boldsymbol{s}_y^2}\right) = \begin{cases} \boldsymbol{\Phi}_1^T \boldsymbol{D}_y^T \boldsymbol{e}_k^T \operatorname{sign}(\boldsymbol{s}_y), & \text{if} \quad |\boldsymbol{s}_y| \geq \delta, \\ \dfrac{\boldsymbol{\Phi}_1^T \boldsymbol{D}_y^T \boldsymbol{e}_k^T \boldsymbol{s}_y}{\delta}, & \text{otherwise.} \end{cases} \tag{A.13}$$

The variables $\boldsymbol{s}_x = \boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c}$ and $\boldsymbol{s}_y = \boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c}$.

Proof. For implementing total variation, we introduce the Huber functional for the ℓ_1 norm approximation.

$$H_\delta(x) = \begin{cases} |x| - \dfrac{\delta}{2}, & \text{if} \quad |x| \geq \delta, \\ \dfrac{x^2}{2\delta}, & \text{otherwise.} \end{cases} \tag{A.14}$$

As total variation equation has two terms, $\sqrt{(\boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c})^2}$ and $\sqrt{(\boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2}$, we apply Huber functional to each term separately.

$$H_\delta(\sqrt{(\boldsymbol{D}_x \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2}) = \begin{cases} \sqrt{(\boldsymbol{D}_x \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2} - \dfrac{\delta}{2}, & \text{if} \quad \sqrt{(\boldsymbol{D}_x \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2} \geq \delta, \\ \dfrac{(\boldsymbol{D}_x \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2}{2\delta}, & \text{otherwise.} \end{cases} \tag{A.15}$$

and

$$H_\delta(\sqrt{(\boldsymbol{D}_y \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2}) = \begin{cases} \sqrt{(\boldsymbol{D}_x \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2} - \dfrac{\delta}{2}, & \text{if} \quad \sqrt{(\boldsymbol{D}_y \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2} \geq \delta, \\ \dfrac{(\boldsymbol{D}_y \boldsymbol{e}_k \boldsymbol{\Phi}_1 \boldsymbol{c})^2}{2\delta}, & \text{otherwise.} \end{cases} \tag{A.16}$$

After taking derivative of each terms, we can get the gradient of total variation.

$$\partial_{\boldsymbol{c}} H_\delta\left(\sqrt{(\boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c})^2}\right) = \begin{cases} \boldsymbol{\Phi}_1^T \boldsymbol{D}_x^T \boldsymbol{e}_k^T \operatorname{sign}(\boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c}), & \text{if} \quad |\boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c}| \geq \delta, \\ \dfrac{\boldsymbol{\Phi}_1^T \boldsymbol{D}_x^T \boldsymbol{e}_k^T \boldsymbol{e}_k \boldsymbol{D}_x \boldsymbol{\Phi}_1 \boldsymbol{c}}{\delta}, & \text{otherwise.} \end{cases} \tag{A.17}$$

and

$$\partial_{\boldsymbol{c}} H_\delta\left(\sqrt{(\boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c})^2}\right) = \begin{cases} \boldsymbol{\Phi}_1^T \boldsymbol{D}_y^T \boldsymbol{e}_k^T \operatorname{sign}(\boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c}), & \text{if} \quad |\boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c}| \geq \delta, \\ \dfrac{\boldsymbol{\Phi}_1^T \boldsymbol{D}_y^T \boldsymbol{e}_k^T \boldsymbol{e}_k \boldsymbol{D}_y \boldsymbol{\Phi}_1 \boldsymbol{c}}{\delta}, & \text{otherwise.} \end{cases} \tag{A.18}$$

□

A.2 Derivation of Alternating Direction Method of Multipliers

In order to solve problem in (A.1) and consider the complexity, we propose a fast dense disparity reconstruction algorithm by utilizing alternative directional method of multiplier (ADMM). Separating the augmented Lagragian in (3.24) into subproblems, and we can solve each subproblem individually.

A.2.1 $\boldsymbol{x}$-subproblem:

Referring to (3.24), we can solve the x-subproblem by keeping terms with variable $\boldsymbol{x}$, and solve the equation by taking derivative of the subproblem. Therefore, the x-subproblem is,

$$\begin{aligned}\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} & -\boldsymbol{w}^T\left(\boldsymbol{r}-\boldsymbol{x}\right) - \boldsymbol{y}_\ell^T\left(\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T\boldsymbol{x}\right) - \boldsymbol{z}^T\left(\boldsymbol{v}-\boldsymbol{D}\boldsymbol{x}\right)\\ & + \frac{\mu}{2}\|\boldsymbol{r}-\boldsymbol{x}\|^2 + \frac{\rho_\ell}{2}\|\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T\boldsymbol{x}\|^2 + \frac{\gamma}{2}\|\boldsymbol{v}-\boldsymbol{D}\boldsymbol{x}\|^2.\end{aligned}$$

Then, take the derivative of the equation above, and set them to zero. We get,

$$\begin{aligned}\left(\rho_\ell\boldsymbol{\Phi}_\ell\boldsymbol{\Phi}_\ell^T + \mu\boldsymbol{I} + \gamma\boldsymbol{D}^T\boldsymbol{D}\right)\boldsymbol{x}^{(k+1)} &= \boldsymbol{\Phi}_\ell\left(\rho_\ell\boldsymbol{u}_\ell - \boldsymbol{y}_\ell\right) + \left(\mu\boldsymbol{r}-\boldsymbol{w}\right) + \boldsymbol{D}^T\left(\gamma\boldsymbol{v}-\boldsymbol{z}\right),\\ &= \text{r.h.s.}\end{aligned}$$

Since $\boldsymbol{\Phi}_\ell$ is an orthogonal matrix and has a property that $\boldsymbol{\Phi}_\ell^T\boldsymbol{\Phi}_\ell = \boldsymbol{I}$. Also, $\boldsymbol{D}^T\boldsymbol{D}$ is a block circulant matrix, we can use the fourier trick to solve this problem.

$$\boldsymbol{x}^{(k+1)} = \mathcal{F}^{-1}\left[\frac{\mathcal{F}(\text{r.h.s.})}{(\rho_\ell+\mu)\boldsymbol{I} + \gamma\left(|\mathcal{F}(\boldsymbol{D}_x)|^2 + |\mathcal{F}(\boldsymbol{D}_y)|^2\right)}\right] \tag{A.19}$$

Since the difference operator $\mathcal{F}(\boldsymbol{D}_x)$ and $\mathcal{F}(\boldsymbol{D}_y)$ can be precalculated, these precalculated operators can be used for reducing computational complexity.

A.2.2 $\boldsymbol{u}_\ell$-subproblem:

To solve u-subproblems, we also keep terms with variable $\boldsymbol{u}_\ell$ in (3.24). As we consider cases that $\ell = 1, 2$, each $\boldsymbol{u}_\ell$ subproblem can be solved independently.

$$\min_{\boldsymbol{u}_\ell} \;\; \lambda_\ell \|\boldsymbol{W}_\ell \boldsymbol{u}_\ell\|_1 - \boldsymbol{y}_\ell^T \left(\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{x}\right) + \frac{\rho_\ell}{2} \|\boldsymbol{u}_\ell - \boldsymbol{\Phi}_\ell^T \boldsymbol{x}\|^2. \tag{A.20}$$

Based on *shrinkage formula*, the solution is,

$$\boldsymbol{u}_\ell^{(k+1)} = \max\left(\left|\boldsymbol{\alpha}_\ell + \frac{\boldsymbol{y}_\ell}{\rho_\ell}\right| - \frac{\lambda_\ell \tilde{\boldsymbol{w}}_\ell}{\rho_\ell}, 0\right) \cdot \operatorname{sign}\left(\boldsymbol{\alpha}_\ell + \frac{\boldsymbol{y}_\ell}{\rho_\ell}\right), \tag{A.21}$$

where $\tilde{\boldsymbol{w}}_\ell = \operatorname{diag}(\boldsymbol{W}_\ell)$ and $\boldsymbol{\alpha}_\ell = \boldsymbol{\Phi}_\ell^T \boldsymbol{x}$.

Proof. For solving the $\boldsymbol{u}_\ell$-subproblem, taking derivative is the first step. For $\boldsymbol{u}_i^* \neq 0$, the derivative of each element is,

$$\lambda_\ell \tilde{\boldsymbol{w}}_{\ell,i} \operatorname{sign}\left(\boldsymbol{u}_{\ell,i}\right) - \boldsymbol{y}_{\ell,i} + \rho_\ell \left(\boldsymbol{u}_{\ell,i}^* - \boldsymbol{\alpha}_{\ell,i}\right), \tag{A.22}$$

Then, the equation can be written as,

$$\boldsymbol{u}_{\ell,i}^* + \frac{\lambda_\ell \tilde{\boldsymbol{w}}_{\ell,i} \operatorname{sign}(\boldsymbol{u}_{\ell,i}^*)}{\rho_\ell} = \frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}, \tag{A.23}$$

Also, we can get,

$$\left|\boldsymbol{u}_{\ell,i}^*\right| + \frac{\lambda_\ell \tilde{\boldsymbol{w}}_{\ell,i}}{\rho_\ell} = \left|\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}\right|. \tag{A.24}$$

Then,

$$\begin{aligned}
\operatorname{sign}\left(\boldsymbol{u}_{\ell,i}^*\right) &= \frac{\operatorname{sign}(\boldsymbol{u}_{\ell,i}^*)\left|\boldsymbol{u}_{\ell,i}^*\right| + \frac{\lambda_\ell \operatorname{sign}(\boldsymbol{u}_{\ell,i}^*)}{\rho_\ell}}{\left|\boldsymbol{u}_{\ell,i}^*\right| + \frac{\lambda_\ell}{\rho_\ell}} \\
&= \frac{\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}}{\left|\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}\right|} \\
&= \operatorname{sign}\left(\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}\right).
\end{aligned}$$

Finally, we can get the solution of this $\boldsymbol{u}_\ell$-subproblem.

$$\begin{aligned}\boldsymbol{u}^*_{\ell,i} &= |\boldsymbol{u}^*_{\ell,i}|\frac{\boldsymbol{u}^*_{\ell,i}}{|\boldsymbol{u}^*_{\ell,i}|} = |\boldsymbol{u}^*_{\ell,i}| \cdot \text{sign}\left(\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}\right) \\ &= \left(\left|\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}\right| - \frac{\lambda_\ell \tilde{\boldsymbol{w}}_{\ell,i}}{\rho_\ell}\right) \cdot \text{sign}\left(\frac{\boldsymbol{y}_{\ell,i}}{\rho_\ell} + \boldsymbol{\alpha}_{\ell,i}\right).\end{aligned}$$

□

Similarly, we can use *shrinkage formula* to find the closed form solution of $\boldsymbol{v}$-subproblem.

A.3 Parameter Tuning for ADMM

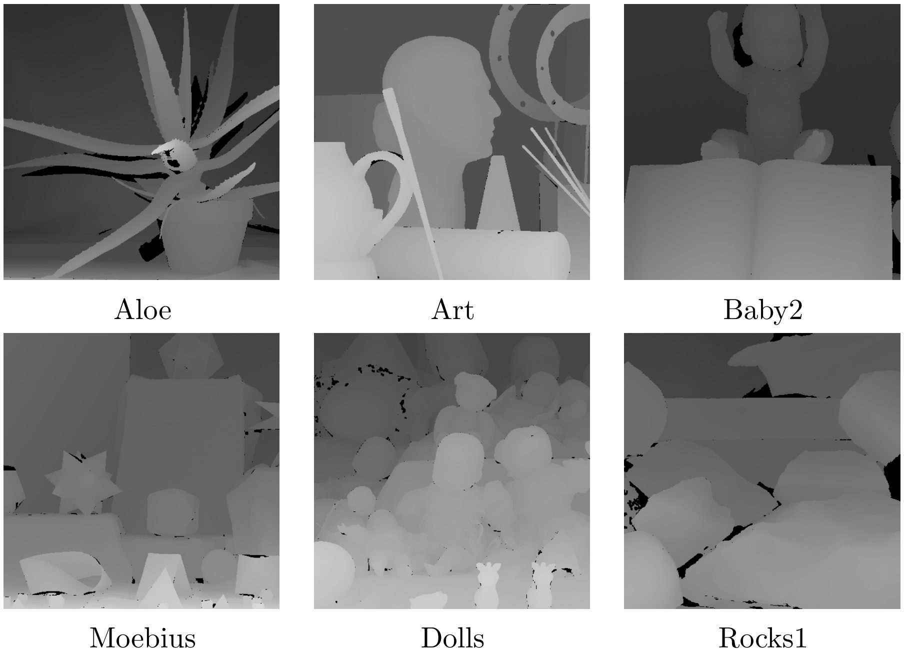
A.3.1 Experimental Configurations

Before presenting results, we first describe our experimental configurations. Testing disparity maps are chosen from Middlebury datasets [46]. All disparity values are normalized to the range [0, 1]. Figure 1.1 shows some examples of disparity maps. For the sampling patterns, we choose the uniformly random samples to minimize any bias towards the sampling. For wavelet dictionary, we use "db2" wavelet function with decomposition level 2, and for contourlet dictionary, we set frequency partition "5, 6". These settings are fixed throughout the experiment.

A.3.2 Regularization Parameters $(\lambda_1, \lambda_2, \beta)$

We empirically evaluate the mean square error (MSE) by sweeping the parameters $(\lambda_1, \lambda_2, \beta)$ from 10^{-6} to 10^0, with a fixed sampling rate of $\xi = 0.2$. The optimal values of the parameters are chosen to minimize the average MSE.

Figure 1.2 shows the MSE curves for various images. For each plot, the MSE is computed by sweeping one parameter while fixing the other parameters. Observing the top row of Figure 1.2, we see that the optimal λ_1 across all images is approximately located in the range of $10^{-6} \leq \lambda_1 \leq 10^{-3}$. Therefore, we select $\lambda_1 = 4 \times 10^{-5}$. Similarly, we can determine $\lambda_2 = 2 \times 10^{-4}$ and $\beta = 2 \times 10^{-3}$. We repeat the above analysis for $\xi = 0.1$. The results are shown in the bottom row of Figure 1.2. The result indicates



Aloe Art Baby2

Moebius Dolls Rocks1

Figure 1.1: Example disparity maps from Middlebury dataset.

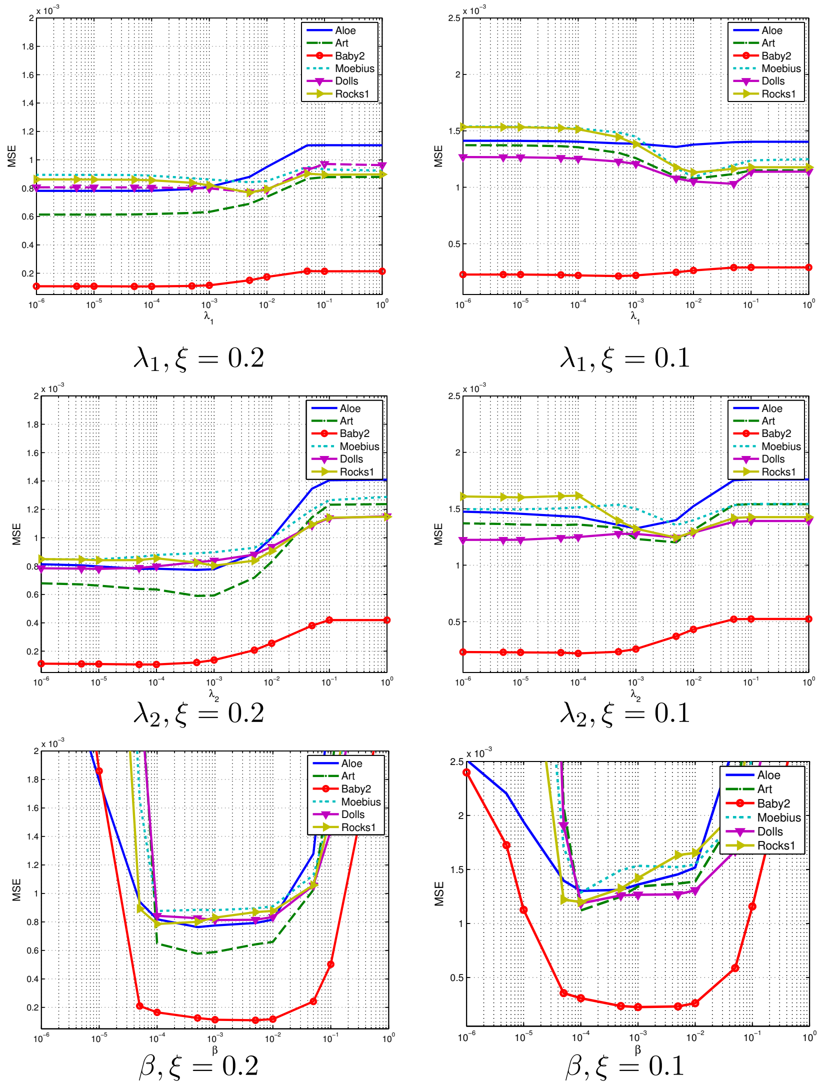
that while there are some difference in the MSE as compared to the top row, the optimal value does not change. Therefore, we keep the parameters using the above settings.

A.3.3 Internal Parameters $(\mu, \rho_1, \rho_2, \gamma)$

For $\mu, \rho_1, \rho_2, \gamma$, we conduct a set of similar experiments as before. The results are shown in Figure 1.3 and Figure 1.4. The criteria to select the parameter is based on the convergence rate. This gives us $\rho_1 = 0.001$, $\rho_2 = 0.001$, $\mu = 0.01$ and $\gamma = 0.1$.

A.3.4 Summary

We summarize our findings in Table A.1. We remark that the values in Table A.1 are “typical” values that correspond to a reasonable MSE on average. Of course, for a specific problem there exists a set of optimal parameters. However, from our experience, this set of parameters seems to be robust over a wide range of problems.



$\lambda_1, \xi = 0.2$

$\lambda_1, \xi = 0.1$

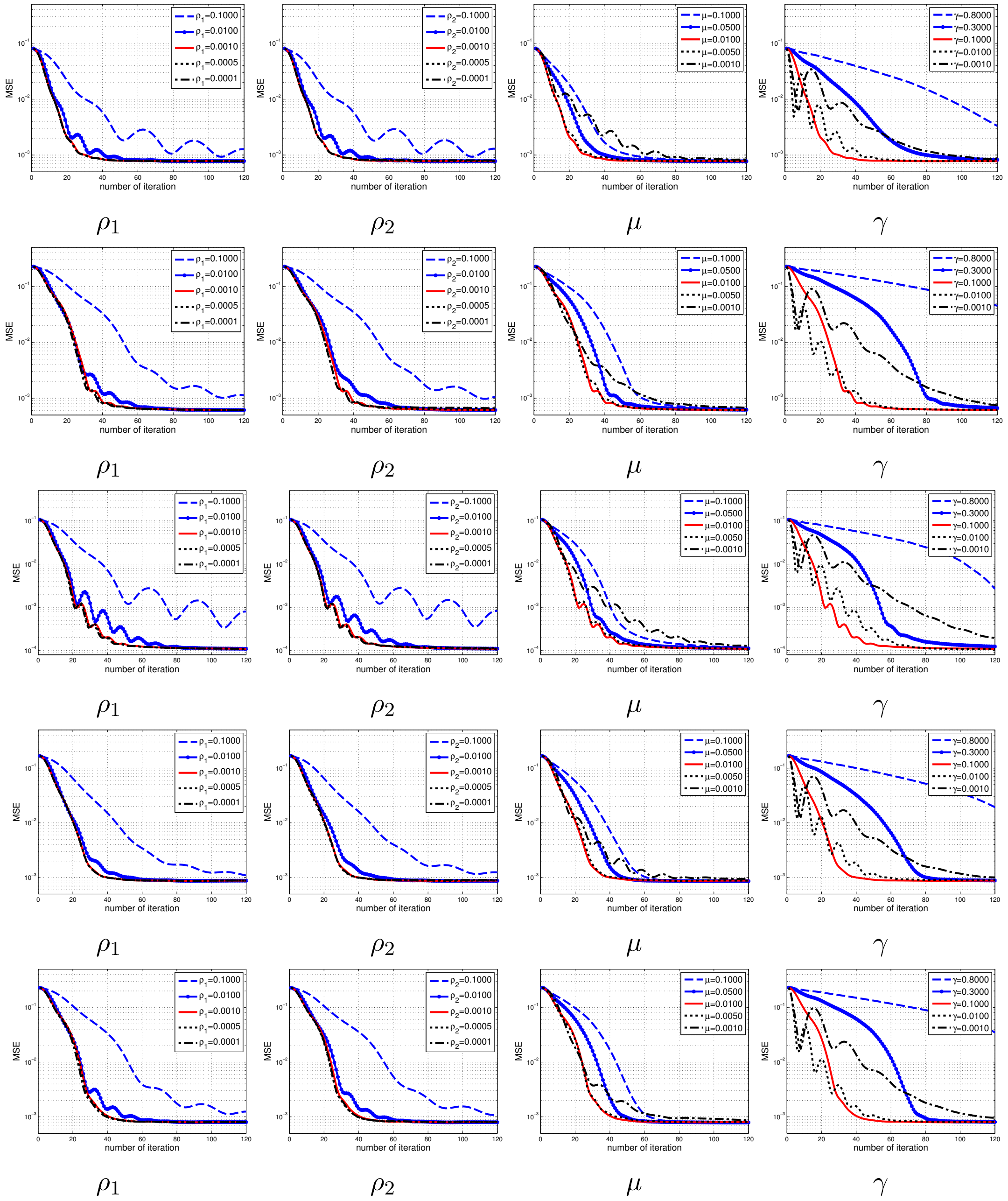
$\lambda_2, \xi = 0.2$

$\lambda_2, \xi = 0.1$

$\beta, \xi = 0.2$

$\beta, \xi = 0.1$

Figure 1.2: Comparison of reconstruction performance with varying regularization parameters and depth images. For each plot, we sweep a parameter from 10^{-6} to 10^{0} while fixing others to be our typical values. We set the sampling rate to be 20% (1st row) and 10% (2nd row). Typical values of regularization parameters are $\lambda_1 = 4 \times 10^{-5}$, $\lambda_2 = 2 \times 10^{-4}$ and $\beta = 2 \times 10^{-3}$.



ρ_1 ρ_2 μ γ

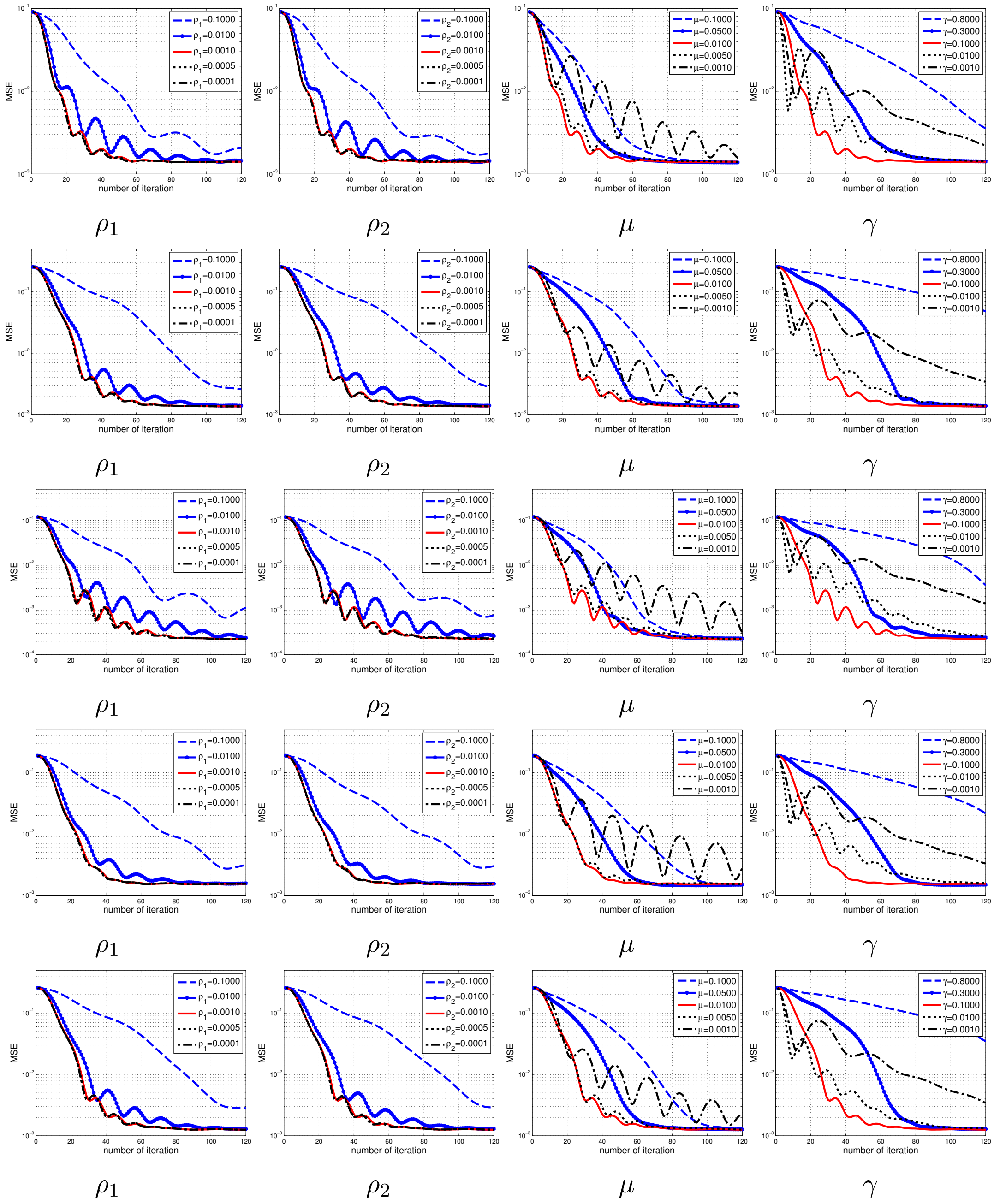
ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

Figure 1.3: MSE for $\xi = 0.2$. Curves in each row are results using Aloe, Art, Baby2, Moebius, Dolls (From Top to Bottom).



ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

ρ_1 ρ_2 μ γ

Figure 1.4: MSE for $\xi = 0.1$. Curves in each row are results using Aloe, Art, Baby2, Moebius, Dolls (From Top to Bottom).

Table A.1: Summary of Parameters and typical values.

Parameter	Functionality	Values
λ_1	Wavelet sparsity	4×10^{-5}
λ_2	Contourlet sparsity	2×10^{-4}
β	Total variation	2×10^{-3}
ρ_1	Half quad. penalty for Wavelet	0.001
ρ_2	Half quad. penalty for Contourlet	0.001
μ	Half quad. penalty for $\boldsymbol{r} = \boldsymbol{x}$	0.01
γ	Half quad. penalty for $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{x}$	0.1

Bibliography

[1] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[2] S. Hawe, M. Kleinsteuber, and K. Diepold, "Dense disparity maps from sparse disparity measurements," in *Proc. IEEE Int. Conf. Computer Vision (ICCV'11)*, Nov. 2011, pp. 2126–2133.

[3] M. A. Lefsky, W. B. Cohen, G. G. Parker, and D. J. Harding, "LIDAR remote sensing of above-ground biomass in three biomes," *Global Ecology and Biogeography*, vol. 11, pp. 393–399, Oct. 2002.

[4] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proc. IEEE Int. Conf. Computer Vision (ICCV'09)*, Sep. 2009, pp. 72–79.

[5] S. Burion, "Human detection for robotic urban search and rescue," M.S. thesis, Carnegie Mellon Univ., 2004, available at http://www.cs.cmu.edu/afs/cs/project/retsina-31/www/Report/Final%20Report.pdf.

[6] R. Khoshabeh, J. Juang, M.A. Talamini, and T.Q. Nguyen, "Multiview glasses-free 3-D laparoscopy," *IEEE Trans. Bio. Eng.*, vol. 59, no. 10, pp. 2859–2865, Oct. 2012.

[7] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[8] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (ToF) cameras: A survey," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, Sep. 2011.

[9] B. Schwarz, "LIDAR: Mapping the world in 3D," *Nature Photonics*, vol. 4, pp. 429–430, Jul. 2010.

[10] C. Niclass, M. Soga, H. Matsubara, S. Kato, and M. Kagami, "A 100-m range 10-frame/s 340 × 96-pixel time-of-flight depth sensor in 0.18- μm cmos," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 2, pp. 559–572, Feb. 2013.

[11] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *Proc. IEEE Int. Conf. Computer Vision (ICCV'11)*, Nov. 2011, pp. 467–474.

[12] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR'06)*, Aug. 2006, vol. 3, pp. 15–18.

[13] Z. Wang and Z. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'08)*, Jun. 2008, pp. 1–8.

[14] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.

[15] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *in Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'97)*, Jun. 1997, pp. 1106–1112.

[16] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Int. Conf. Computer Vision (ICCV'99)*, Sep. 1999, vol. 1, pp. 666–673.

[17] R. S. Feris, J. Gemmell, K. Toyama, and V. Kruger, "Hierarchical wavelet networks for facial feature localization," in *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition (FG'02)*, May 2002, pp. 118–123.

[18] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'04)*, Jun. 2004, vol. 2, pp. 506–513.

[19] L.-K. Liu, S.H. Chan, and T.Q. Nguyen, "Depth reconstruction from sparse samples: Representation, algorithm, and sampling," *IEEE Trans. on Image Process.*, vol. 24, no. 6, pp. 1983–1996, Jun. 2015.

[20] J. Diebel and S. Thrun, "An application of Markov random field to range sensing," in *Advances in Neural Info. Process. System (NIPS'05)*, Dec. 2005, pp. 291–298.

[21] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'07)*, Jun. 2007, pp. 1–8.

[22] J. Li, T. Xue, L. Sun, and J. Liu, "Joint example-based depth map super-resolution," in *IEEE Int. Conf. Multimedia and Expo (ICME'12)*, Jul. 2012, pp. 152–157.

[23] O. M. Aodha, N. D. F. Campbell, A. Nair, and G. J. Brostow, “Patch based synthesis for single depth image super-resolution,” in *Proc. European Conf. Computer Vision (ECCV’12)*, Oct. 2012, pp. 71–84.

[24] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Process. Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[25] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Trans. Info. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.

[26] M. Elad and A. M. Bruckstein, “A generalized uncertainty principle and sparse representation in pairs of bases,” *IEEE Trans. Info. Theory*, vol. 48, no. 9, pp. 2558–2567, Sep. 2002.

[27] S. Schwartz, A. Wong, and D. A. Clausi, “Saliency-guided compressive sensing approach to efficient laser range measurement,” *J. Vis. Commun. Image R.*, vol. 24, no. 2, pp. 160–170, 2013.

[28] S. Schwartz, A. Wong, and D. A. Clausi, “Multi-scale saliency-guided compressive sensing approach to efficient robotic laser range measurement,” in *Proc. IEEE Computer Society Conf. Computer, Robot Vision*, 2012, pp. 1–8.

[29] A. Kirmani, A. Colaco, F.N.C. Wong, and V.K. Goyal, “CODAC: A compressive depth acquisition camera framework,” in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP’12)*, Mar. 2012, pp. 5425–5428.

[30] A. Kirmani, D. Venkatraman, D. Shin, A. Colaco, F.N. C. Wong, J.H. Shapiro, and V.K. Goyal, “First photon imaging,” *Science Magazine*, vol. 343, no. 6166, pp. 58–61, Nov. 2013.

[31] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Mar. 2004.

[32] F. Chung and L. Lu, “Concentration inequalities and martingale inequalities: a survey,” *Internet Mathematics*, vol. 3, no. 1, pp. 79–127, 2006.

[33] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” *Inverse Problems*, vol. 23, pp. 947–968, Apr. 2007.

[34] E. J. Candès and Y. Plan, “A probabilistic and RIPless theory of compressed sensing,” *IEEE Trans. Information Theory*, vol. 57, no. 11, pp. 7235–7254, Nov. 2011.

[35] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Const. Approx.*, vol. 28, no. 3, pp. 253–263, Dec. 2008.

[36] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[37] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[38] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Dec. 2008.

[39] D. D. Y. Po and M. N. Do, "Directional multiscale modeling of images using the contourlet transform," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1610–1620, Jun. 2006.

[40] E. J. Candès and D. L. Donoho, "Recovering edges in ill-posed inverse problems: Optimality of curvelet frames," *Annals of Statistics*, , no. 3, pp. 784–842, Aug. 2002.

[41] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Communications on Pure and Applied Mathematics*, vol. 57, no. 2, pp. 219–266, Feb. 2004.

[42] E. Le Pennec and S. Mallat, "Bandelet image approximation and compression," *Multiscale Model. Simul.*, vol. 4, no. 3, pp. 992–1039, 2005.

[43] M. Vetterli and J. Kovačević, *Wavelets and subband coding*, Prentice Hall, 1995.

[44] E.J. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23(3), pp. 969–985, Nov. 2006.

[45] M.R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, Dec. 1952.

[46] "Middlebury dataset," http://vision.middlebury.edu/stereo/.

[47] J. J. Moreau, "Proximité et dualtité danes un espace hilbertien," *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299, 1965.

[48] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.

[49] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM J. on Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, Jul. 2009.

[50] D. Han and X. Yuan, “A note on the alternating direction method of multipliers,” *J. of Optim. Theory and Applications*, vol. 155, no. 1, pp. 227–238, Oct. 2012.

[51] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[52] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *IEEE Trans. Info. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[53] L. Liu, S. H. Chan, and T. Q. Nguyen, “Depth reconstruction from sparse samples: Representation, algorithm, and sampling (supplementary material),” Available online at http://arxiv.org/abs/1407.3840.

[54] S. H. Chan, T. Zickler, and Y. M. Lu, “Monte Carlo non-local means: Random sampling for large-scale image filtering,” *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3711–3725, Aug. 2014.

[55] F. Li, J. Yu, and J. Chai, “A hybrid camera for motion deblurring and depth map super-resolution,” in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR’08)*, Jun. 2008, pp. 1–8.

[56] J. Park, H. Kim, Y. Tai, M.S. Brown, and I. Kweon, “High quality depth map upsampling for 3D-TOF cameras,” in *Proc. IEEE Int. Conf. Computer Vision (ICCV’11)*, Nov. 2011, pp. 1623–1630.

[57] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. on Computer Vision*, vol. 47, no. 1-3, pp. 7–42, Apr. 2002.

[58] Z. Lee, J. Juang, and T. Q. Nguyen, “Local disparity estimation with three-moded cross census and advanced support weight,” *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1855–1864, Dec. 2013.

[59] E. Kollorz, J. Penne, J. Hornegger, and A. Barke, “Gesture recognition with a Time-of-Flight camera,” *Int. J. of Intell. Systems Tech. and Applications*, vol. 5, no. 3-4, pp. 334–343, Nov. 2008.

[60] M. Van den Bergh and L. Van Gool, “Combining RGB and ToF cameras for real-time 3D hand gesture interaction,” in *IEEE Workshop on Applications of Computer Vision (WACV’11)*, Jan. 2011, pp. 66–72.

[61] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, “3D shape scanning with a time-of-flight camera,” in *IEEE Conf. on Computer Vision and Pattern Recogn. (CVPR’10)*, Jun. 2010, pp. 1173–1180.

[62] S. May, B. Werner, H. Surmann, and K. Pervolz, “3D time-of-flight cameras for mobile robotics,” in *IEEE Int. Conf. on Intel. Robots and Systems*, Oct. 2006, pp. 790–795.

[63] A.K. Jain, L.C. Tran, R. Khoshabeh, and T.Q. Nguyen, “Efficient stereo-to-multiview synthesis,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP'11)*, May 2011, pp. 889–892.

[64] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, “Object stereo -; joint stereo matching and object segmentation,” in *IEEE Conf. on Computer Vision and Pattern Recogn. (CVPR'11)*, Jun. 2011, pp. 3081–3088.

[65] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt, “On-set performance capture of multiple actors with a stereo camera,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 161:1–161:11, Nov. 2013.

[66] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H. Bischof, “Image guided depth upsampling using anisotropic total generalized variation,” in *Proceed. IEEE Int. Conf. on Computer Vision (ICCV'13)*, Dec. 2013, pp. 993–1000.

[67] J. Liu, X. Gong, and J. Liu, “Guided inpainting and filtering for kinect depth maps,” in *21st Int. Conf. on Pattern Recogn. (ICPR'12)*, Nov. 2012, pp. 2055–2058.

[68] Q. Yang, Yang R., J. Davis, and D. Nister, “Spatial-depth super resolution for range images,” in *IEEE Conf. on Computer Vision and Pattern Recogn. (CVPR'07)*, Jun. 2007, pp. 1–8.

[69] J. Lu, D. Min, R.S. Pahwa, and M.N. Do, “A revisit to MRF-based depth map super-resolution and enhancement,” in *IEEE Int. Conf. on Acoust., Speech and Sig. Process. (ICASSP'11)*, May 2011, pp. 985–988.

[70] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[71] L. Chen, H. Lin, and S. Li, “Depth image enhancement for kinect using region growing and bilateral filter,” in *Int. Conf. on Pattern Recogn. (ICPR'12)*, Nov. 2012, pp. 3070–3073.

[72] M. Kiechle, S. Hawe, and M. Kleinsteuber, “A joint intensity and depth co-sparse analysis model for depth map super-resolution,” in *IEEE Int. Conf. on Computer Vision (ICCV'13)*, Dec. 2013, pp. 1545–1552.

[73] O. Mac Aodha, N.F. Campbell, A. Nair, and G.J. Brostow, “Patch based synthesis for single depth image super-resolution,” in *Euro. Conf. on Computer Vision (ECCV'12)*, vol. 7574, pp. 71–84. Springer Berlin Heidelberg, Oct. 2012.

[74] J.M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. on Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.

[75] S.H. Chan, D.T. Vo, and T.Q. Nguyen, "Subpixel motion estimation without interpolation," in *IEEE Int. Conf. on Acoustics Speech and Signal Process. (ICASSP'10),*, Mar. 2010, pp. 722–725.

[76] M.S.C. Almeida and M.A.T. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Trans. on Image Process.*, vol. 22, no. 8, pp. 3074–3086, Aug. 2013.

[77] C.J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, pp. 79–82, Dec. 2005.

[78] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N.A. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *Proceed. of the Euro. Conf. on Comput. Vision (ECCV'10)*, Sep. 2010, vol. 6313, pp. 510–523.

[79] M. Berg, O. Cheong, M. Kreveld, and M. Overmars, "Computational geometry: Algorithms and applications," chapter 9. Springer-Verlag TELOS, 3rd edition, 2008.

[80] L.-K. Liu, Z. Lee, and T. Nguyen, "Sharp disparity reconstruction using sparse disparity measurement and color information," in *IEEE Image Video and Multidimensional. Signal Process. Workshop (IVMSP'13),*, Jun. 2013, pp. 1–4.

[81] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Trans. on Pattern Analysis and Machine Intel.,*, vol. 16, no. 9, pp. 920–932, Sep. 1994.