

A NOVEL COMPRESSION ALGORITHM BASED ON  
SPARSE SAMPLING OF 3-D LASER RANGE SCANS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Oğuzcan Dobrucalı

July 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

---

Prof. Dr. Billur Barshan (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

---

Prof. Dr. Erdal Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

---

Prof. Dr. Gzde Bozdađı Akar

Approved for the Institute of Engineering and Sciences:

---

Prof. Dr. Levent Onural  
Director of Institute of Engineering and Sciences

## ABSTRACT

# A NOVEL COMPRESSION ALGORITHM BASED ON SPARSE SAMPLING OF 3-D LASER RANGE SCANS

Oğuzcan Dobrucalı

M.S. in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Billur Barshan

July 2010

3-D models of environments can be very useful and are commonly employed in areas such as robotics, art and architecture, environmental planning and documentation. A 3-D model is typically comprised of a large number of measurements. When 3-D models of environments need to be transmitted or stored, they should be compressed efficiently to use the capacity of the communication channel or the storage medium effectively. In this thesis, we propose a novel compression technique based on compressive sampling, applied to sparse representations of 3-D laser range measurements. The main issue here is finding highly sparse representations of the range measurements, since they do not have such representations in common domains, such as the frequency domain. To solve this problem, we develop a new algorithm to generate sparse innovations between consecutive range measurements acquired while the sensor moves. We compare the sparsity of our innovations with others generated by estimation and filtering. Furthermore, we compare the compression performance of our lossy compression method with widely used lossless and lossy compression techniques. The proposed method offers small compression ratio and provides a reasonable compromise between reconstruction error and processing time.

*Keywords:* 3-D laser scan, 3-D modeling, 3-D mapping, compressive sensing, compressive sampling, sensor data compression, SICK LMS laser range finder.

## ÖZET

### ÜÇ BOYUTLU LAZER UZAKLIK TARAMALARININ SEYREK ÖRNEKLENMESİNE DAYALI YENİ BİR VERİ SIKIŞTIRMA YÖNTEMİ

Oğuzcan Dobrucalı

Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Prof. Dr. Billur Barshan

Temmuz 2010

Robotbilim, sanat ve mimarlık, çevre planlaması ve dökümantasyonu gibi çeşitli alanlarda ortamların üç boyutlu modellerinden yararlanır. Üç boyutlu modellerin çok fazla sayıda ölçüm içermesinden dolayı, bu modellerin bir yerden bir yere iletilmesi veya bir yerde saklanması gerektiğinde, haberleşme kanalının veya veri depolama alanının kapasitesini verimli kullanmak için verilerin öncelikle etkili bir biçimde sıkıştırılmaları gerekir. Bu tezde, üç boyutlu lazer uzaklık taramalarını sıkıştırmak için, sıkıştırmalı algulamaya dayanan yeni bir yöntem önerilmektedir. Taramaların frekans alanı gibi yaygın kullanılan alanlarda çok seyrek gösterimlerinin olmamasından dolayı, bunların çok seyrek şekilde temsil edilmesi, bu tezde ele alınan ana sorunlardan biridir. Bu sorunun çözümü için taramaları, lazer uzaklık algılayıcısının hareket yönünde peş peşe alınan ölçümler arasında oluşturulan seyrek değişimlerle temsil eden bir yöntem geliştirilmiştir. Oluşturulan değişimlerin seyrekliği, diğer kestirme ve süzgeçleme yöntemleriyle hesaplanan değişimlerle karşılaştırılmıştır. Ayrıca önerilen kayıplı sıkıştırma yönteminin başarımı, yaygın kullanılan kayıpsız ve kayıplı sıkıştırma

yöntemlerinin başarımlarıyla karşılaştırılmıştır. Sonuç olarak, önerilen yöntem kısa sürede ve az kayıpla, büyük oranda sıkıştırma sağlamaktadır.

*Anahtar Kelimeler:* üç boyutlu lazer taraması, üç boyutlu modelleme, üç boyutlu haritalama, sıkıştırılmalı algılama, sıkıştırılmalı örnekleme, sensör verisi sıkıştırma, SICK LMS lazer uzaklık ölçer.

## ACKNOWLEDGMENTS

I would like to thank everyone who contributed to this thesis. First of all, I would like to express my sincere thanks to my thesis supervisor Prof. Dr. Billur Barshan for her supervision, guidance, suggestions, and encouragement throughout the development of this thesis. I am grateful to Prof. Dr. Erdal Arıkan and Prof. Dr. Gözde Bozdağı Akar for showing keen interest to the subject matter and accepting to read and review the thesis. I would also like to thank Prof. Dr. Orhan Arıkan for his inspiration on the subject of this thesis. Finally, I would like to give my special thanks to my officemates for their endless support. I also extend my thanks to TÜBİTAK for funding my graduate studies through an MSc. scholarship.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background on Compressive Sensing</b>	<b>7</b>
2.1	Determining the Sparsifying Basis . . . . .	11
2.2	Determining the Measurement Model . . . . .	12
<b>3</b>	<b>The Proposed Method</b>	<b>15</b>
3.1	The Sparsifying Model . . . . .	21
3.2	The Measurement Model . . . . .	30
3.3	The Reconstruction Model . . . . .	33
<b>4</b>	<b>Comparing Compression Performance of the Proposed Method with Some Well-Known Compression Techniques</b>	<b>36</b>
4.1	Implementation and Comparison with Well-Known Lossless Tech- niques . . . . .	37
4.2	Implementation and Comparison with Well-Known Lossy Tech- niques . . . . .	42



4.3	Implementation and Comparison with the Proposed Method . . .	46
<b>5</b>	<b>Conclusions and Future Work</b>	<b>56</b>
	<b>Appendix</b>	<b>59</b>
<b>A</b>	<b>Well-Known Dictionaries for Forming a Sparsifying Basis</b>	<b>59</b>
<b>B</b>	<b>Methods for Generating Sparse Innovations</b>	<b>62</b>

# List of Figures

1.1	(a) The front view of SICK LMS200, (b) its measurement principle, and (c) its field of view (reprinted from [1]). . . . .	3
2.1	The operation scheme of the single pixel camera (reprinted from [2]).	13
3.1	(a)–(d): Sample data sets collected at the University of Osnabrück AVZ building, and (e)–(h): their reconstructions. . . . .	16
3.2	The percentage of the number of non-zero values to the total number of values in the projections of the 3-D scan illustrated in Figure 3.1(b) onto the bases formed by using (a) Fourier, (b) Gabor, and (c) Haar dictionaries. . . . .	17
3.3	The percentage of the number of non-zero values to the total number of values in the innovations, when the methods referred as (a)–(f) are applied to the 3-D scan illustrated in Figure 3.1(b), respectively. . . . .	19
3.4	The percentage of the number of non-zero values to the total number of values in the sparse representations generated at the sparsifying model for the 3-D scan illustrated in Figure 3.1(b). . . . .	20
3.5	The operation scheme of the proposed method. . . . .	20

3.6	Illustration of (a) the amplitude and phase shifts, and (b) the offset.	21
3.7	The amplitude of $\mathcal{E}^2$ with respect to (a) first- and (b) second-order approximation to $\delta$ for the data set illustrated in Figure 3.1(b).	23
3.8	Illustrations of the difference sequences obtained for the 3-D scans given in Figure 3.1(a)–(d).	23
3.9	The frequencies of appearance of different values in $\mathbf{v}_n$ .	24
3.10	The average of the sample autocorrelation estimate of $\mathbf{v}_n$ with $\pm 2\sigma_{\bar{R}}$ and $\pm 3\sigma_{\bar{R}}$ standard error boundaries.	27
3.11	The flowchart of the sparsifying model algorithm.	28
3.12	The percentage of the number of non-zero values to the total number of values in the sparse representations generated at the sparsifying model for the 3-D scan illustrated in Figure 3.1(b) with white Gaussian noise with zero mean and 0, 1, 2, 3, 4, 5, 10, 20, and 30 cm standard deviation, respectively.	29
3.13	The RMS of the reconstruction error with respect to the number of non-zero values in the sparse data, when the 2-D scans from all 3-D scans in the first data set are sampled using compressive sampling.	31
3.14	The measurement size $M$ in SC and CS with respect to the number of non-zero values of a signal in $\mathfrak{R}^{361}$ .	32
3.15	The flowchart of the measurement model algorithm.	32
3.16	The length of the measurements for the data set illustrated in Figure 3.1(b), when (a) the measurement model and (b) RLE are employed.	33

3.17	The flowchart of the reconstruction model algorithm. . . . .	35
4.1	Two-channel filterbank structure. . . . .	44
4.2	(a) The analysis and (b) the synthesis structures of a 3-level wavelet transform. . . . .	46
4.3	Distortion images for the 3-D scans given in Figure 3.1(a)–(d). . .	49
4.4	The path of the motion and the positions where 3-D scans are acquired at Dagstuhl Castle. . . . .	51
4.5	(a)–(d): Sample data sets collected at Dagstuhl Castle and (e)– (h): their reconstructions. . . . .	52
4.6	The average correlation coefficients between the 2-D scans in the (a) first and (b) second data sets, respectively. . . . .	53

# List of Tables

4.1	Compression ratio (CR), the time required for encoding ( $t_{\text{enc}}$ ) and decoding ( $t_{\text{dec}}$ ) when the raw 3-D scans in the first data set are compressed using Huffman and arithmetic coding. . . . .	39
4.2	Compression ratio (CR), the time required for encoding ( $t_{\text{enc}}$ ) and decoding ( $t_{\text{dec}}$ ) when the raw 3-D scans in the first data set are compressed using ZLIB and GZIP. . . . .	40
4.3	CR, $t_{\text{enc}}$ , and $t_{\text{dec}}$ when the differences between consecutive scans in the first data set are compressed using Huffman and arithmetic coding. . . . .	41
4.4	CR, $\mathcal{D}$ , $t_{\text{enc}}$ , and $t_{\text{dec}}$ when the first data set is compressed using JPEG. . . . .	43
4.5	CR, $\mathcal{D}$ , $t_{\text{enc}}$ , and $t_{\text{dec}}$ when the first data set is compressed using 1-level, 2-level, and 3-level wavelet transforms. . . . .	45
4.6	CR, $\mathcal{D}$ , $t_{\text{enc}}$ , $t_{\text{dec}}$ , number of cases when the signal is encoded with $\{\epsilon, \delta, \Delta, \mathbf{m}\}$ using compressive sampling ( $k_{\text{SHIFT+CS}}$ ), $\{\epsilon, \delta, \Delta, \mathbf{m}\}$ using simple coding ( $k_{\text{SHIFT+SC}}$ ), only $\{\epsilon, \delta, \Delta\}$ ( $k_{\text{SHIFT}}$ ), and the number of cases when the signal is not encoded ( $k_{\text{NOCODING}}$ ), when the first data set is compressed using the proposed method. . . . .	48

4.7	CR, $\mathcal{D}$ , $t_{\text{enc}}$ , and $t_{\text{dec}}$ when the first data set is compressed by different lossless and lossy methods. . . . .	50
4.8	CR, $\mathcal{D}$ , $t_{\text{enc}}$ , and $t_{\text{dec}}$ when the second data set is compressed with different lossless and lossy methods. . . . .	54
4.9	The average percentages of $k_{\text{SHIFT+CS}}$ , $k_{\text{SHIFT+SC}}$ , $k_{\text{SHIFT}}$ , and $k_{\text{NOCODING}}$ , when both data sets are compressed using the proposed method. . . . .	54
4.10	Average signal-to-noise ratio (SNR), CR, $\mathcal{D}$ , $t_{\text{enc}}$ , $t_{\text{dec}}$ , number of cases when the signal is encoded with $\{\epsilon, \delta, \Delta, \mathbf{m}\}$ using compressive sampling ( $k_{\text{SHIFT+CS}}$ ), $\{\epsilon, \delta, \Delta, \mathbf{m}\}$ using simple coding ( $k_{\text{SHIFT+SC}}$ ), only $\{\epsilon, \delta, \Delta\}$ ( $k_{\text{SHIFT}}$ ), and the number of cases when the signal is not encoded ( $k_{\text{NOCODING}}$ ), when the first data set is compressed under the presence of additive white Gaussian noise indicated with its mean and variance. . . . .	55

**Dedicated to my family**

# Chapter 1

## Introduction

Many techniques have been developed to build 3-D models of environments. 3-D modeling techniques allow describing environments including objects with indefinite shapes or patterns, although these techniques can be complex and computationally expensive [3]. The main advantage of using 3-D models of environments is that they are more descriptive and have richer information content than 2-D models in terms of the features extracted from the environments, resulting in less ambiguity in distinguishing features [4]. 3-D models are used in fields varying from robot motion planning and navigation [3, 5, 6, 7], art and architecture [8, 9, 10, 11, 12] to industry/urban planning, water management, and forestry documentation [13, 14, 15]. 3-D models can be obtained using a variety of sensors measuring range or intensity. A commonly used approach in constructing these models is using laser range finders that measure the range between the sensor and the objects along the path of the beam emitted by the sensor. These sensors can supply range measurements within their field of view, as the laser beam is rotated by the sensor. There are several approaches to obtain 3-D models with these sensors: The first is using a conventional 3-D laser scanner. However, since these products are very expensive, this approach is not frequently employed. Another approach is acquiring 3-D range measurements by



translating a 2-D laser range finder that horizontally or vertically scans a field of view of  $180^\circ$ . A third alternative is to acquire the 3-D range information by rotating the 2-D laser range finder around a fixed axis. In the latter two, multiple 2-D laser range finders can be employed where each sensor scans either the horizontal or the vertical axis [6]. The most commonly used laser range finders are the products of SICK AG [16].

Most of the works using 3-D models are in the area of localization and mapping for mobile robots. For instance, Brenneke et al. in [3] proposed a technique for *simultaneous localization and mapping* (SLAM) in outdoor environments. They applied the existing 2-D mapping algorithms to one horizontal layer of a 3-D model. Besides that, maps are obtained and used in 3-D SLAM applications [17, 18, 19]. In order to build either 2-D or 3-D maps from sequentially acquired scans, *iterative closest point algorithm* (ICP) is employed, integrated with odometry measurements [6]. ICP algorithm is also used in the registration of scans of not only planar surfaces, but also curves and non-planar surfaces, as in [20]. Besides ICP, *semantic information* of the range measurements, which is the gradient between the neighbouring measurements, is also used for the same purpose [7]. Apart from deterministic methods for the registration of 3-D objects, parametric methods such as *expectation-maximization* (EM) [4] and *maximum-likelihood* (ML) [21] estimation are employed. Moreover, non-parametric methods, such as the *k-means clustering* algorithm [22], are also used for the same purpose. In addition to modeling with laser scanners only, other devices, such as panoramic cameras, are used integrated with laser scanners [23]. Besides the techniques for modeling indoor environments, especially for robot navigation, terrains are modeled using airborne laser scanners for obtaining terrestrial information as in [24] and [25]. Apart from modeling the environments above the sea level, 3-D models of seafloor are also obtained using autonomous underwater vehicles equipped with a camera, a sonar, and oceanographic sensors [26]. In

summary, many techniques to acquire and process 3-D measurements have been developed, and new techniques are continuously being introduced.

In this thesis, we consider an indoor environment scanned in 3-D with laser range finders. The sensor used in this study is the SICK LMS200 sensor, shown in Figure 1.1(a). This 2-D device measures the range between itself and the objects within its field of view, based on the *time-of-flight* principle. The sweeping laser beam is aligned by the rotating mirror, as illustrated in Figure 1.1(b). The laser has a maximum range of 80 m, field of view of  $180^\circ$  (Figure 1.1(c)), range resolution as low as 1 mm, and a selectable angular resolution of either  $0.25^\circ$ ,  $0.5^\circ$ , or  $1^\circ$ . The measurements have a systematic error of  $\pm 4$  cm, as well as some statistical error that changes with the measurement range, the ambient temperature and illumination, and the reflectivity of the objects in the environment. The sampling frequency of the measurements is 75 Hz [1]. The advantages of using a laser beam is reliable detection of object presence and the independence of the measurements from the amount of ambient light and the colors of the objects. A major disadvantage is that, for proper operation of the sensor, the environment should not contain highly reflective or transparent materials, such as glass. SICK LMS200 is used in various tasks, such as determining volumes and positions of objects, classification of objects, collision prevention for vehicles, and surveillance [1].

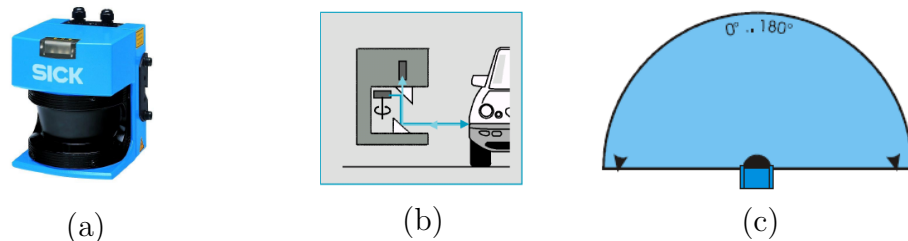


Figure 1.1: (a) The front view of SICK LMS200, (b) its measurement principle, and (c) its field of view (reprinted from [1]).

The scan data collected by any sensor usually needs to be transmitted to a station where the data are processed and analyzed. If there are several such

sensors in the environment, such a system can be categorized as a *wireless sensor network* (WSN), which is a network composed of a number of sensor nodes expanding over a sensor field. The measurements taken at the nodes are transmitted to the sink (i.e., the station) through the WSN. Establishing a WSN over a field to monitor some specific properties is advantageous, since adding nodes to the network, or removing nodes from the network is easy and inexpensive. Despite this advantage, communication between the nodes and the sink in WSNs has limited capabilities in terms of bandwidth, transmission speed, and memory space [27]. Moreover, the scan data of a 3-D model, which is likely to be comprised of hundreds of thousands of range measurements, also needs to be stored in a medium, where the amount of allocated memory is required to be as small as possible. Thus, the data must be written to the medium efficiently in terms of the memory space, as well as the speed in reading and writing operations for the data. This way, fast and accurate autonomous search and scan systems can be developed.

To satisfy all of the requirements mentioned above, the scan data must be compressed before it is transmitted or stored. The amount of data stored in the data storage medium can be increased, and the elapsed time required to transmit the data through the communication channel can be reduced by lowering the size of the data. Although there are many compression techniques developed for different types of data, determining the optimum data compression technique with respect to the following criteria is still an open research field:

An important aspect of data compression is the *compression ratio* (CR), which is the ratio of the size of the compressed output to the size of the original data. The CR is between zero and one (or zero and 100%) for compression operation, and larger than one for expansion operation. The closer the CR is to zero, the greater the amount of compression [28].

Salamon in [28] points out that any data compression method is not perfect; thus compressing any number of bits into one bit, which may be a fictional case, is such a success that even compressing two bits into one bit can be considered as “perfect.” Therefore, a compression method can be considered efficient when the size of the original data is reduced by more than one half. In other words, an efficient compression method, at least halves the storage and communication costs [29].

CRs as low as about 20% are commonly observed, and can be even lower. The CRs for some compression methods used in UNIX operating systems are reported based on several observations on compressing different types of files: 18% for *binary*, 36% for *C source*, 38% for *text*, 42% for *Huffman coding*, 43% for *Pascal source*, and 73% for *arithmetic coding*. Finally, CRs as low as 2% have been reported for specific applications [29].

The amount of *distortion* is the second aspect in data compression [30]. The size of the data is lowered by employing either *lossless* compression techniques in which the whole information in the data is encoded, or *lossy* compression techniques in which the essential part of the information is encoded. Although distortion, which is the difference between the data and its reconstruction using the compressed data, is observed in lossy compression, lossy compression methods are usually preferred, since they result in lower CRs than lossless compression methods. The distortion can be measured in various ways depending on the type of data, and is required to be as low as possible. Let  $\mathbf{x} = \{x_i\}_{i=1}^N$  and  $\hat{\mathbf{x}} = \{\hat{x}_i\}_{i=1}^N$  represent the data sequence and its reconstruction, respectively. Then, some widely used measures of distortion between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  are as follows [30]:

- the *mean squared error* (MSE) between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  is  $\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2$ ,
- the *root mean squared error* (RMSE) between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  is  $\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2}$ ,
- the *signal-to-noise ratio* (SNR) is  $\frac{\sum_{i=1}^N x_i^2}{\sum_{i=1}^N (x_i - \hat{x}_i)^2}$ ,

- the *peak-signal-to-noise ratio* (PSNR) is  $\frac{(\max\{x_i\}_{i=1}^N)^2}{\sum_{i=1}^N (x_i - \hat{x}_i)^2}$ .

*Speed* is another aspect in data compression. It is a measure of how fast the data is compressed (encoding speed) and reconstructed from the compressed data (decoding speed) by using a given compression technique. Speed is inversely proportional to the time required for encoding and decoding the data, and required to be as high as possible.

In this thesis, we propose an effective compression method that can be applied to 3-D laser range measurements as the data is being acquired. To the best of our knowledge, solutions that reduce the cost of transmission and storage of the measurements in 3-D model acquisition do not exist. The main contribution of this thesis is to provide a model to generate sparse representations of laser range measurement sequences. These representations include an incredibly small number of non-zero values compared to the number of measurements in the original sequences. Then, the sparse representations are compressed by applying sparse sampling techniques that have been applied in sampling parametric signals [31], and are based on *compressive sensing*. The proposed method can be considered as a kind of *difference encoding* and is a causal system because it generates sparse representations based on current and previous measurements. Therefore, it can compress even an infinite number of range measurement sequences, in theory.

The rest of this thesis is organized as follows: Compressive sensing is reviewed in Chapter 2. The method is described in detail in Chapter 3 and compared with widely used compression techniques in terms of the CR, distortion, and speed, in Chapter 4. Two sets of experimental data, independently acquired at different institutions, are used for this purpose. Conclusions and directions for future work are provided in the last chapter. A few of the well-known sparsifying dictionaries used in compressive sensing are reviewed in Appendix A. Methods for sparsifying the scan data are described in Appendix B.

## Chapter 2

# Background on Compressive Sensing

Before compressive sensing technique was proposed, classical sampling had been ruled by the Shannon/Nyquist sampling theorem, which requires sampling a signal at a minimum rate of twice its bandwidth, in order not to lose the information content of the signal. Oversampling results in more accurate representation of the signal despite that it is costly. In compressive sensing, the signal is successfully reconstructed with fewer samples than the Shannon/Nyquist sampling theorem requires. Compressive sensing uses a linear sampling model with an optimization procedure for reconstructing the original signal [2].

The signals considered here are range measurement sequences taken within the sensor's field of view, as column vectors in  $\mathfrak{R}^N$ , where  $N$  can be very large. As stated above, compressive sensing focuses on representing the sampled signal with fewer number of measurements, which are actually linear functions of the original signal. To achieve this, compressive sensing relies on *sparsity* and *incoherence* properties. Sparsity property requires the signals to have sparse representations in proper domains. Sparse signals can be represented with a lower sampling

frequency than the Nyquist rate. Furthermore, sparsity enables discrete-time signals to be represented with shorter length than their finite length. In other words, signals can be briefly represented when they are sparsely expressed using a proper basis  $\Psi$ . Incoherence property states that the sparse representation of the signal on basis  $\Psi$  must be extended in the domain in which the signal is sampled [32].

The first step in compressive sensing is to represent the signal using a proper basis onto which the representation is sparse. The basis should contain a set of orthonormal vectors that form a set of waveforms such as the wavelet basis [32]. Let  $\mathbf{x} = [x_1, \dots, x_N]^T$  be the column vector that represents the  $N$  samples of the signal in  $\mathfrak{R}^N$ .  $\Psi = [\Psi_1, \dots, \Psi_N]$  stands for the basis matrix with orthonormal basis vectors  $\{\Psi_i\}_{i=1}^N$ . Here, it is assumed that the basis vectors are column vectors in  $\mathfrak{R}^N$  so that  $\Psi$  is an  $N \times N$  matrix. Thus, we can represent the signal as  $\mathbf{x} = \sum_{i=1}^N s_i \Psi_i = \Psi \mathbf{s}$ , where  $\mathbf{s} = [s_1, \dots, s_N]^T$  in which  $s_i = \langle \mathbf{x}, \Psi_i \rangle$  for  $i = \{1, \dots, N\}$  [2], and  $\langle \cdot, \cdot \rangle$  denotes the *inner product* of two vectors. Note that  $\mathbf{x}$  and  $\mathbf{s}$  are different representations of the same signal in different domains: the time domain and the  $\Psi$  domain, respectively. If the projection of the signal onto the basis  $\Psi$  is sparse, only a small number of coefficients in  $\mathbf{s}$  denoted by  $K$  will have large values, whereas the majority denoted by  $(N - K)$  will be close to zero. When  $K \ll N$ ,  $\mathbf{s}$  is referred as *K-sparse*. The sparsity property defined here is motivated by the assumption that most signals are *compressible* with the choice of a proper basis  $\Psi$ . The approximation of signals with  $K$ -sparse representation is the basis of *transform coding* [2].

At the end of the first step, there are two possible ways to sample the signal: *sample-then-compress framework* and *linear measurement framework*. The first method is an inefficient way since this method requires acquiring all values in the signal, and then determining  $K$  large values among all values. The second method, which is the one used in this thesis, proposes a linear measurement

model without implementing the intermediate step involved in the first method. The linear measurement model computes  $M$  measurements, where  $M \ll N$ . We assume that the measurement model  $\Phi = [\Phi_1^T, \dots, \Phi_M^T]^T$  is an  $M \times N$  matrix, and is composed of basis vectors  $\{\Phi_j\}_{j=1}^M$ , each of which is a column vector in  $\Re^N$ . Let the measurement vector be denoted as  $\mathbf{y} = [y_1, \dots, y_M]^T$  composed of  $\{y_j\}_{j=1}^M$ , where  $y_j = \langle \mathbf{x}, \Phi_j \rangle$ . Thus, the measurement vector can be defined as  $\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \mathbf{s} = \Theta \mathbf{s}$ , which has fewer dimensions than the original signal, referred as the *undersampled* case [32]. Based on the elements of compressive sensing described so far, the objective of compressive sensing can be briefly summarized as determining a measurement model  $\Phi$ , and a sparsifying basis  $\Psi$ , that allow the reconstruction of the signal  $\mathbf{x}$ , which is not damaged despite the dimensionality reduction. More briefly, the objective of compressive sensing is determining  $\Theta$  [2].

The solution to the determination of  $\Theta$  must satisfy two important properties: *Restricted Isometric Property* (RIP) and *incoherence*. RIP requires that  $\zeta$ , a constant between zero and one, should be close to zero by the following statement:

$$(1 - \zeta) \|\mathbf{x}\|_2^2 \leq \|\Theta \mathbf{x}\|_2^2 \leq (1 + \zeta) \|\mathbf{x}\|_2^2 \quad (2.1)$$

where  $\|\cdot\|_2$  is the two-norm of the corresponding vector. The above statement expresses that any vector multiplied by  $\Theta$  cannot be in the null space of  $\Theta$ , so  $\Theta$  must preserve the length of the vectors multiplied by itself. The second requirement for  $\Theta$  is the incoherence property, which indicates uncorrelatedness between the sparsifying basis  $\Psi$  and the measurement model  $\Phi$  [32]. Incoherence states that basis vectors in the measurement model cannot *sparsely* represent the basis vectors in the sparsifying basis [2]. Coherence (i.e., the opposite of incoherence) between  $\Phi$  and  $\Psi$  can be referred as a measurable quantity, computed by

$$\mu(\Phi, \Psi) = \sqrt{N} \max_{1 \leq i \leq N, 1 \leq j \leq M} |\langle \Psi_i, \Phi_j \rangle| \quad (2.2)$$



where  $\mu$  indicates coherence, varying between one and  $\sqrt{N}$  [32]. Low levels of coherence are always preferable for building  $\Theta$ , so that we have maximal incoherence when  $\mu$  is one.

One remaining issue in the design of the compressive sensing structure is determining a lower bound for  $M$ , which is the number of measurements obtained by the measurement model. Since the dimension of the sampled signal  $N$  and the number of non-zero entries in the sparse representation  $K$  are both known, the minimum value of  $M$  can be computed from either

$$M \geq c_1 K \ln \left( \frac{N}{K} \right) \quad [2] \quad (2.3)$$

or

$$M \geq c_2 \mu^2(\Phi, \Psi) K \ln(N) \quad [32] \quad (2.4)$$

where  $c_1$  and  $c_2$  are small positive constants. In Equation (2.3), the minimum number of measurements, which is also the minimum number of basis vectors in the measurement model, is claimed to be proportional to the natural logarithm of the ratio of the size of the sampled signal  $N$  to the sparsity  $K$ . Furthermore, in Equation (2.4), fewer samples in the measurement model are claimed to be sufficient as the coherence decreases. Both Equation (2.3) and (2.4) demonstrate the following facts about compressive sensing [32]:

- No information is lost after sampling as soon as a set of  $M$  samples that satisfy Equation (2.3) or (2.4) is acquired in the measurement model.
- The sampled signal can be recovered without any knowledge of where the zero entries are located in the sparse representation.

After a measurement vector  $\mathbf{y}$ , which has far smaller dimension than the original signal  $\mathbf{x}$ , is obtained, the next step is the reconstruction of the original signal and its sparse representation  $\mathbf{s}$  from the measurement vector. At the end of sampling, we have  $\mathbf{y} = \Theta \mathbf{s}$  where  $\mathbf{s}$  is to be estimated, given  $\mathbf{y}$  and  $\Theta$ . Since

$\Theta$  is an  $M \times N$  matrix with  $M \ll N$ , there are infinitely many  $\tilde{\mathbf{s}}$  that satisfy  $\mathbf{y} = \Theta\tilde{\mathbf{s}}$ . Therefore, optimization techniques are employed to obtain the optimum reconstruction of  $\mathbf{s}$ . The basic idea is to reach the *minimum-norm solution* to  $\mathbf{s}$ , which reduces any part of the null space of  $\Theta$  in the solution that is desired to be as sparse as possible. The optimal solution to  $\mathbf{s}$  is stated as:

$$\hat{\mathbf{s}} = \arg \min \|\tilde{\mathbf{s}}\|_1 \text{ such that } \mathbf{y} = \Theta\tilde{\mathbf{s}} \quad (2.5)$$

where  $\|\cdot\|_1$  is the one-norm of the corresponding vector. Furthermore, if the sparse representation is reconstructed from noisy measurements, the following optimization can be considered:

$$\hat{\mathbf{s}} = \arg \min \|\tilde{\mathbf{s}}\|_1 \text{ such that } \|\mathbf{y} - \Theta\tilde{\mathbf{s}}\|_2 \leq \rho \quad (2.6)$$

where  $\rho$  is the bound on the noise in the measurement vector  $\mathbf{y}$  [33, 34]. Apart from the one-norm solution, a two-norm solution is available in *regularized minimization* for reconstruction [35]. In this case,  $\hat{\mathbf{s}} = \arg \min \|\mathbf{y} - \Theta\tilde{\mathbf{s}}\|_2^2 + c_0\|\tilde{\mathbf{s}}\|_1$ , where  $c_0$  is a small positive constant. One way to solve the given optimization problems is to apply *basis pursuit* algorithms [36]. As soon as the sparse representation of the signal  $\mathbf{s}$  is estimated as  $\hat{\mathbf{s}}$ , the original signal  $\mathbf{x}$  is reconstructed as  $\hat{\mathbf{x}} = \Psi\hat{\mathbf{s}}$  with a small distortion between  $\mathbf{x}$  and  $\hat{\mathbf{x}}$ . In this thesis, we use the solution given by Equation (2.5) because the measurement model  $\Phi$  employed here provides a noiseless measurement vector  $\mathbf{y}$ .

## 2.1 Determining the Sparsifying Basis

As stated above, the first step in compressive sensing is to determine the best sparsifying basis  $\Psi$  for efficient representation of the original signal  $\mathbf{x}$ . Thus, the projection of  $\mathbf{x}$  onto this basis should represent  $\mathbf{x}$  with fewer parameters than  $\mathbf{x}$  has, and allow the reconstruction of  $\mathbf{x}$  with small error. Any sparsifying basis is composed of a set of basis vectors that are actually waveforms. In the

literature, these waveforms are called *atoms*, and the set of atoms that comprise the sparsifying basis is called a *dictionary* [36]. Although there are some readily available dictionaries, such as wavelet packets, cosine packets, Gabor dictionary, Fourier dictionary, chirplets, and warplets, dictionaries can also be designed and tailored according to the signal features. In this thesis, Fourier, Gabor, and Haar dictionaries are tested with the experimental scan data, in order to acquire sufficiently sparse representations. Detailed information on these dictionaries can be found in Appendix A.

## 2.2 Determining the Measurement Model

An interesting application of compressive sensing is the *single pixel camera*, reported in [37]. In the single pixel camera, image data are considered as compressible signals, so that the images are sampled without taking their projections onto a sparsifying basis. In the implementation of sampling, light is projected onto a digital micromirror device (DMD) having an array of  $N$  micromirrors, as shown in Figure 2.1. According to the measurement model used, a random number generator (RNG) selects a set of micromirrors to focus the reflected light onto a photodiode. The measurement model is generated in three different ways: *raster scan*, *basis scan*, and *compressive sampling*. As a result, different combinations of  $M$  pixels out of  $N$  are measured by the photodiode. In the raster scan, the photodiode measures  $N$  pixels one at a time (i.e.,  $M = N$ ), where  $\Phi$  is the  $N \times N$  identity matrix. In the basis scan, the photodiode measures  $M$  pixels, which are determined according to the Walsh basis, one at a time. In this model,  $\Phi$  is the  $N \times N$  Walsh matrix including binary coefficients [38]. In compressive sampling, the photodiode measures  $M$  different linear combinations of  $N$  pixels, using random test functions. It is shown that the smallest distortion on images occurs with the smallest number of measurements, which is achieved when compressive sampling is used as the measurement model.

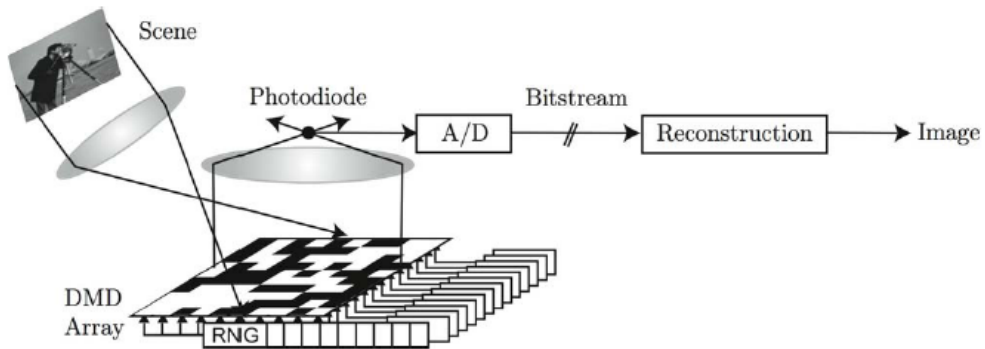


Figure 2.1: The operation scheme of the single pixel camera (reprinted from [2]).

To construct a measurement model using random test functions, as in the compressive sampling model, Candes and Baraniuk propose two alternative methods that are actually somewhat related to each other. Candes suggests in [32] that any random measurement model, which is composed of basis vectors chosen uniformly on the unit sphere, is incoherent with any sparsifying basis with large probability, so that the coherence is expected to be about  $\sqrt{2 \ln N}$ . Supporting this suggestion, Baraniuk suggests in [2] that a measurement model, where all elements in the model matrix are selected independently from a Gaussian distribution with zero mean and  $\frac{1}{N}$  variance, is incoherent with any sparsifying basis with high probability.

Based on the compressive sampling model, we construct the measurement model with the number of basis vectors computed by taking  $c_1 = 1$  in Equation (2.3) such that

$$M = \lceil K \ln \left( \frac{N}{K} \right) \rceil \quad (2.7)$$

where  $\lceil \cdot \rceil$  denotes the *ceiling* function. Following Baraniuk's suggestion in [2], the elements in  $\Phi$  are chosen independently from a Gaussian distribution with zero mean and  $\frac{1}{N}$  variance. Then, the row vectors in  $\Phi$  are orthonormalized by applying the Gram-Schmidt process. Using this measurement model without a sparsifying basis, as in the single pixel camera, where  $\Theta = \Phi$  and  $\Psi$  is  $N \times N$  identity matrix, is advantageous in reconstructing the original signal because

RIP is satisfied, since  $\Phi$  has no null space with high probability where  $\zeta$  is zero in Equation (2.1). Moreover, the incoherence property is satisfied where  $\mu(\Phi, \Psi)$  is found likely to be around  $\sqrt{2 \ln N}$ .

# Chapter 3

## The Proposed Method

We can use compressive sensing to compress any signal using an appropriate sparsifying basis and an incoherent measurement model. This approach is commonly applied in various fields, such as magnetic resonance imaging in medicine [39] and interferometric imaging in astronomy [40]. Although forming the measurement model is a straightforward process, forming the sparsifying basis is a more challenging problem. The main objective in this problem is to find a projection of the signal onto the sparsifying basis, which contains sufficiently sparse critical information to recover the signal with small error [33].

Two different experimental data sets are considered as benchmarks in this thesis. Both of them are comprised of many 3-D scans. Each 3-D scan is acquired by collecting 2-D scans as the sensor is rotated in numerous steps around a horizontal axis above the ground level. Each 2-D scan in the data sets is obtained as the laser beam emitted by the sensor is swept within the sensor's field of view in  $0.5^\circ$  intervals. The first data set contains 29 3-D scans collected at different locations in the University of Osnabrück AVZ building in Osnabrück, Germany [41]. The sensor is rotated in 471 steps to acquire the 2-D scans forming a 3-D scan in this set. The second set is comprised of 82 3-D scans taken at

different locations in the Dagstuhl Castle in Saarland, Germany [42]. Each 3-D scan in this set is acquired by rotating the sensor in 225 steps. As a consequence, every 3-D scan from the first and the second data set constitutes 471 and 225 2-D scans, respectively. The 2-D scans are sequentially acquired as vectors in  $\mathbb{R}^{361}$  (i.e.,  $N = 361$ ). The 3-D scans in the first set are used in this chapter, where different features such as a mannequin, a human being, banisters at the top of the stairs, and chairs are observed, as illustrated in Figure 3.1(a)–(d). In these images, the intensity values are directly proportional to the range measurements such that the white color indicates the maximum range measurement whereas the black color indicates the minimum range measurement.

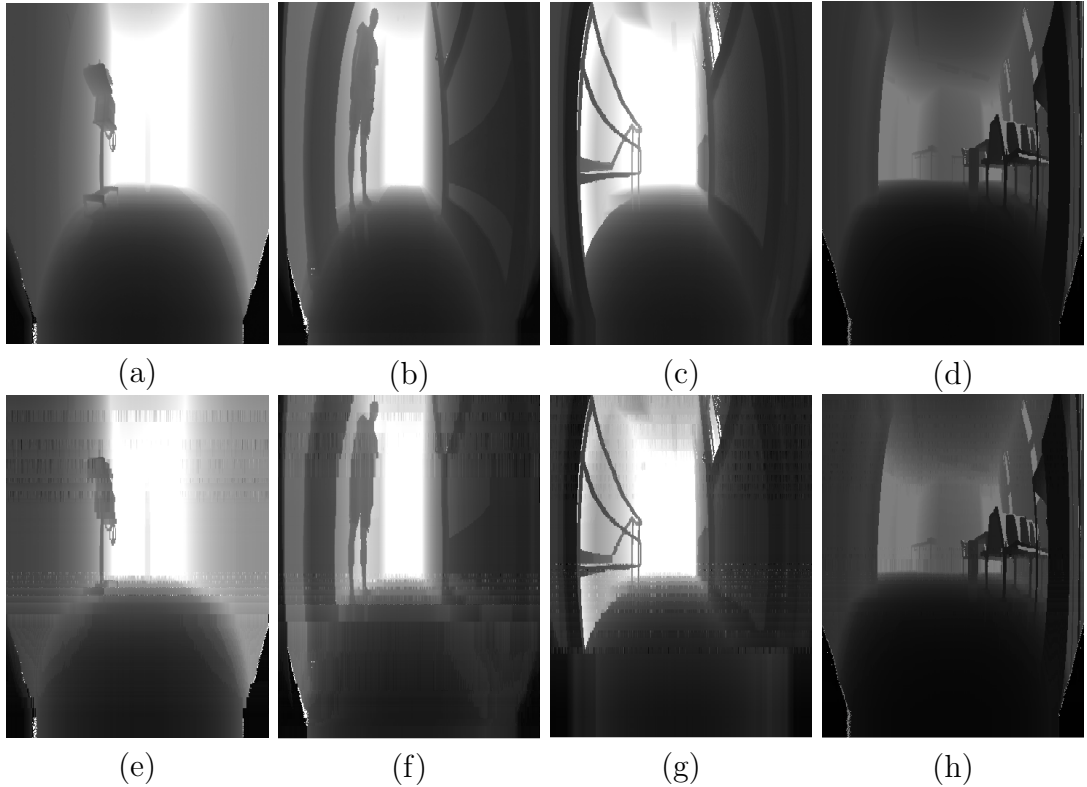


Figure 3.1: (a)–(d): Sample data sets collected at the University of Osnabrück AVZ building, and (e)–(h): their reconstructions.

To apply the sampling model described in Chapter 2, we first consider the projections of a 3-D scan, illustrated in Figure 3.1(b), onto some of the well-known sparsifying bases. The 2-D scans forming the 3-D scan are projected one

at a time onto  $N \times N$  sparsifying bases formed by using the Fourier, Gabor, and Haar dictionaries. According to the parameterization described in Appendix A:

- Fourier dictionary is formed by  $N$  cosine waveforms with frequencies  $\omega = \frac{l\pi}{N}$  where  $l = \{\frac{1}{2}, \frac{3}{2}, \dots, N - \frac{1}{2}\}$ ,
- Gabor dictionary is formed by  $N$  waveforms with no delay, unit standard deviation of the Gaussian envelope of the waveforms, and different frequencies uniformly selected from  $[0, \pi)$ ,
- Haar dictionary is formed by  $N$  wavelets with  $\frac{1}{32}$  dilation and  $\frac{l}{32}$  translation for  $l = \{0, 1, \dots, N - 1\}$ .

The percentages of the number of non-zero values to the total number of values in these projections are plotted in Figure 3.2(a)–(c), respectively. It is observed that the average percentages are around 74.7%, 61.3%, and 88.7%, in the respective parts. It is remarkable that the projections onto the bases described above are not sufficiently sparse, thus both the CR and distortion would be high if compressive sampling were applied to these projections [33].

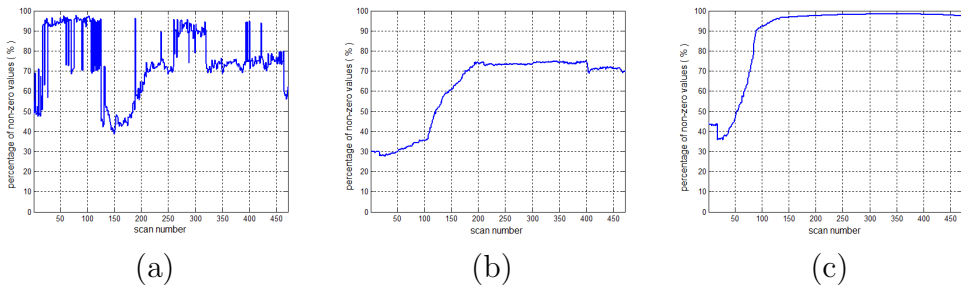


Figure 3.2: The percentage of the number of non-zero values to the total number of values in the projections of the 3-D scan illustrated in Figure 3.1(b) onto the bases formed by using (a) Fourier, (b) Gabor, and (c) Haar dictionaries.

During the process of data acquisition, 2-D scans acquired consecutively have similarities as well as differences. The differences may be caused by changes taking place in a dynamic environment, as well as by the translational or rotational



motion of the sensor because at each step, a different cross-section of the 3-D environment is observed. Since we observe that the raw 2-D scans do not have highly sparse representations in the domains listed above, we attempt to represent them with sparse innovations exploiting the correlation between two consecutively acquired scans, when the sensor is rotated by a small amount before acquiring the next scan. Thus, we define the innovations between:

- (a) two consecutive scans,
- (b) each scan and its estimate using *linear regression* [43] based on the last two scans,
- (c) each scan and its estimate using *second-order polynomial fitting* [43] based on the last three scans,
- (d) each scan and its estimate adding the previous scan to a difference estimate using a *second-order Wiener filter* [43] under the assumption that the differences between consecutive scans form a stationary random sequence,
- (e) each scan and its estimate adding the previous scan to a difference estimate using a 1-D random walk on the previous difference,
- (f) each scan and its estimate using a linear *Kalman filter* with the constant velocity kinematic state model [44], which is also called a polynomial filter, because the mesh points in consecutive scans form piecewise polynomial functions.

The implementation details of the methods (a)–(f) are provided in Appendix B. The percentages of the number of non-zero values to the total number of values in the innovations, when these methods are applied to the 3-D scan illustrated in Figure 3.1(b), are plotted in Figure 3.3. The average percentages are around 43.7%, 92.5%, 99.5%, 71.5%, 27.3%, and 50.9%, respectively. According to these figures, on the average, we obtain the most sparse innovations in (e), but even

this is not found to be sufficient. In this thesis, we propose a method to generate much more sparse innovations with the number of non-zero values being 6.5% of the total number of values on the average, as plotted in Figure 3.4, for the 3-D scan illustrated in Figure 3.1(b).

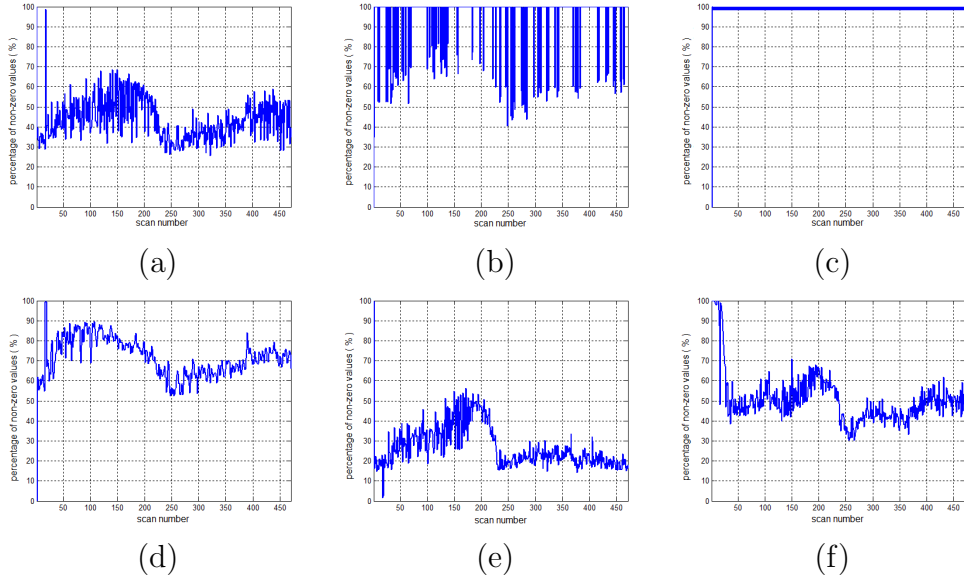


Figure 3.3: The percentage of the number of non-zero values to the total number of values in the innovations, when the methods referred as (a)–(f) are applied to the 3-D scan illustrated in Figure 3.1(b), respectively.

The proposed method is composed of encoder and decoder parts, where the encoder consists of sparsifying, measurement, reconstruction stages, and the decoder involves only the reconstruction stage, as depicted in Figure 3.5. The sparsifying model generates sparse innovations for each scan in the sparsifying stage, and the measurement model samples the innovations with the minimum number of samples in the measurement stage. Finally, the reconstruction model rebuilds each scan from the samples encoded by the measurement model in the reconstruction stage. In the following subsections, these three models are described in more detail.

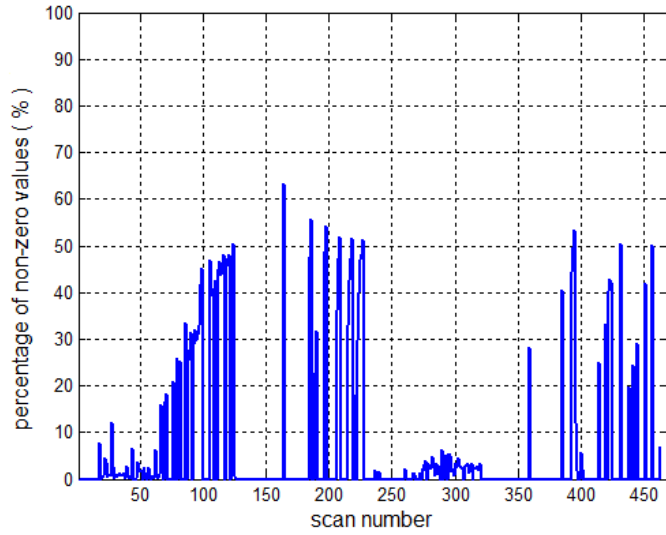


Figure 3.4: The percentage of the number of non-zero values to the total number of values in the sparse representations generated at the sparsifying model for the 3-D scan illustrated in Figure 3.1(b).

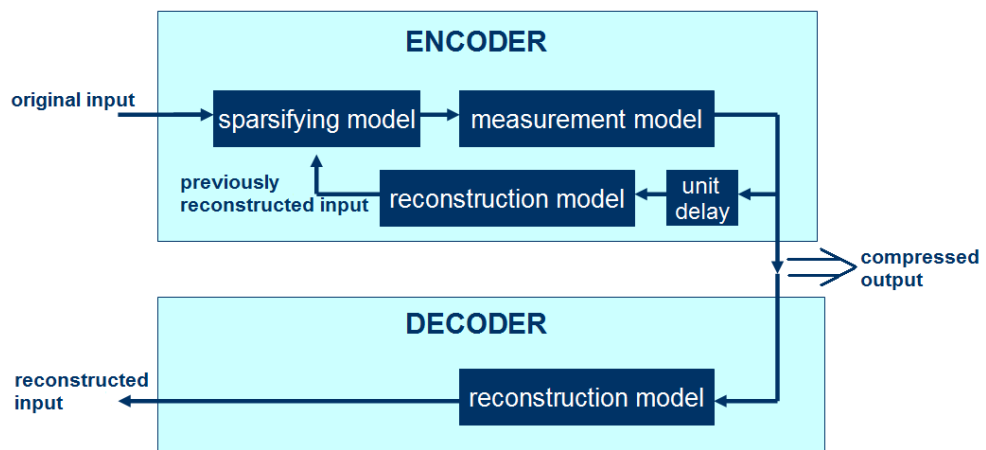


Figure 3.5: The operation scheme of the proposed method.

### 3.1 The Sparsifying Model

In the sparsifying model, we generate the innovations between consecutive scans as follows: Suppose  $\mathbf{r}_n$  is the  $n$ th 2-D scan that is currently acquired, and  $\mathbf{r}_{n-1}$  is the previous one. First,  $\mathbf{r}_{n-1}$  is generated at the encoder by employing the reconstruction procedure in Section 3.3 that the decoder follows, to adapt the sparsifying parameters according to the reconstruction at the decoder. Then,  $\mathbf{r}_{n-1}$  is approximated to  $\mathbf{r}_n$  by shifting  $\mathbf{r}_{n-1}$  along the vertical and horizontal axes by amplitude ( $\epsilon$ ) and phase ( $\delta$ ) shifts, respectively. An example illustrating  $\epsilon$  and  $\delta$  is shown in Figure 3.6(a).

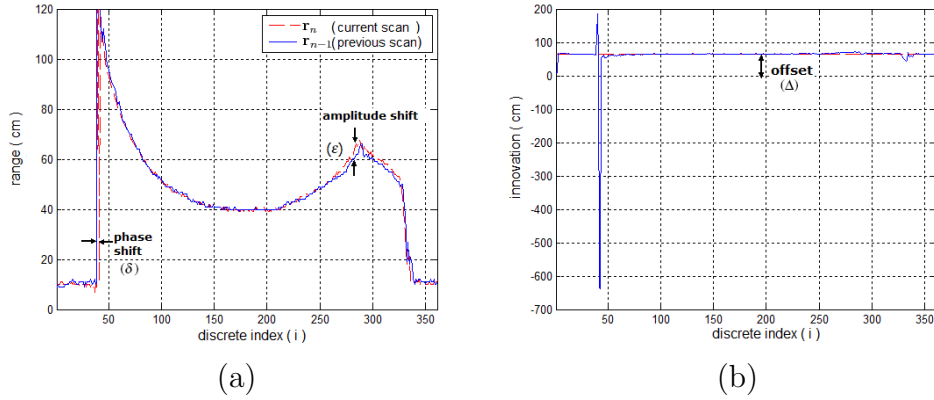


Figure 3.6: Illustration of (a) the amplitude and phase shifts, and (b) the offset.

Assume that the individual range measurements in  $\mathbf{r}_n$  and  $\mathbf{r}_{n-1}$  are denoted by  $r_n[i]$  and  $r_{n-1}[i]$  for  $i = 1, 2, \dots, N$ , respectively. We define an error function  $\mathcal{E}^2 = \sum_{i=1}^N [r_n[i] - (r_{n-1}[i + \delta] + \epsilon)]^2$  and set its partial derivatives with respect to  $\epsilon$  and  $\delta$  to zero to find the optimal values of  $\epsilon$  and  $\delta$ . First, we determine  $\epsilon$  from

$$\frac{\partial \mathcal{E}^2}{\partial \epsilon} = \sum_{i=1}^N [-2r_n[i] + 2(r_{n-1}[i + \delta] + \epsilon)] = 0 \quad (3.1)$$

When we neglect the  $\delta$  term in Equation (3.1), we get a solution for  $\epsilon$ :

$$\epsilon = \frac{1}{N} \sum_{i=1}^N (r_n[i] - r_{n-1}[i]) \quad (3.2)$$

In other words,  $\epsilon$  corresponds to the average amplitude difference between  $\mathbf{r}_n$  and  $\mathbf{r}_{n-1}$ . Then, we determine  $\delta$  from

$$\frac{\partial \mathcal{E}^2}{\partial \delta} = \sum_{i=1}^N \left( -2r_n[i] \frac{\partial r_{n-1}[i + \delta]}{\partial \delta} + 2(r_{n-1}[i + \delta] + \epsilon) \frac{\partial r_{n-1}[i + \delta]}{\partial \delta} \right) = 0 \quad (3.3)$$

The  $r_{n-1}[i + \delta]$  term in Equation (3.3) can be expanded using a *Taylor series* expansion around  $i$ , such that  $r_{n-1}[i + \delta] = r_{n-1}[i] + r'_{n-1}[i]\delta + \frac{1}{2}r''_{n-1}[i]\delta^2 + \dots$ , where  $r'_{n-1}[i]$  and  $r''_{n-1}[i]$  are the first- and second-order differences of the sequence  $\mathbf{r}_{n-1}$  at  $i$ , respectively. Assuming that  $\delta$  is very small compared to  $N$ , we use only the first two terms of the expansion and obtain the following *first-order approximation* to  $\delta$ :

$$\delta = \frac{\sum_{i=1}^N r'_{n-1}[i] (r_n[i] - r_{n-1}[i] - \epsilon)}{\sum_{i=1}^N r'_{n-1}[i]^2} \quad (3.4)$$

If we use the first three terms of the expansion to obtain a more precise expression for  $\delta$ , the *second-order approximation* to  $\delta$  is one of the roots of the following equation, which minimizes  $\mathcal{E}^2$ :

$$\begin{aligned} \frac{\partial \mathcal{E}^2}{\partial \delta} = & \delta^3 \sum_{i=1}^N r''_{n-1}[i] + 3\delta^2 \sum_{i=1}^N r'_{n-1}[i] r''_{n-1}[i] + 2\delta \sum_{i=1}^N (\epsilon r''_{n-1}[i] + r'_{n-1}[i]^2) \\ & + r_{n-1}[i] r''_{n-1}[i] - r_n[i] r''_{n-1}[i] + 2 \sum_{i=1}^N r'_{n-1}[i] (\epsilon + r_{n-1}[i] - r_n[i]) = 0 \end{aligned} \quad (3.5)$$

The value of  $\mathcal{E}^2$  for the 3-D scan illustrated in Figure 3.1(b) is plotted in Figure 3.7 with respect to both approximations to  $\delta$ . We observe that the first- and second-order approximations to  $\delta$  result in nearly the same values of  $\mathcal{E}^2$ . Moreover, computing the first-order approximation to  $\delta$  requires much less time than computing the second-order approximation. Therefore, it seems sufficient to use the first-order approximation to  $\delta$  as given by Equation (3.4).

Shifting  $\mathbf{r}_{n-1}$  along the vertical and the horizontal axes by  $\epsilon$  and  $\delta$ , respectively, we obtain an approximation  $\hat{\mathbf{r}}_n$  to  $\mathbf{r}_n$ . Then, the difference sequence is  $\tilde{\mathbf{v}}_n \triangleq \mathbf{r}_n - \hat{\mathbf{r}}_n$ . Here,  $\tilde{\mathbf{v}}_n$  is a sparse signal representing discontinuities in the

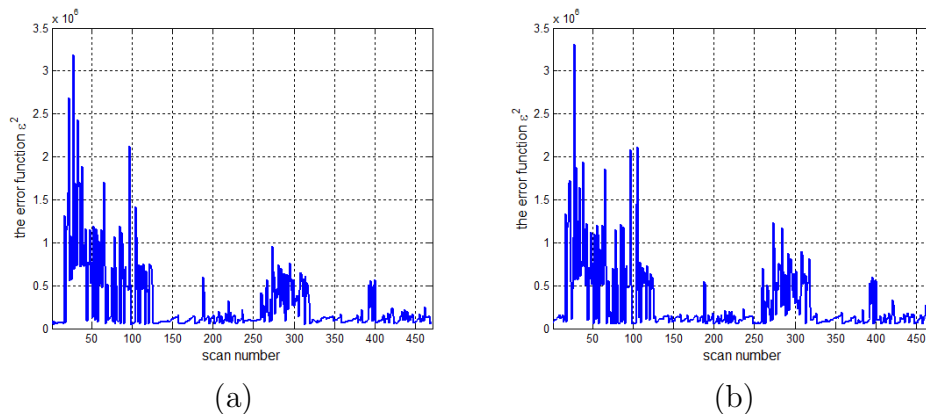


Figure 3.7: The amplitude of  $\mathcal{E}^2$  with respect to (a) first- and (b) second-order approximation to  $\delta$  for the data set illustrated in Figure 3.1(b).

scanned environment. To illustrate this fact, the difference sequences obtained for the 3-D scans given in Figure 3.1(a)–(d) are shown in Figure 3.8(a)–(d), respectively. In these figures, the darker features correspond to larger differences. Note that most of the dark features in these images occur where there is a sudden change in the measured range.

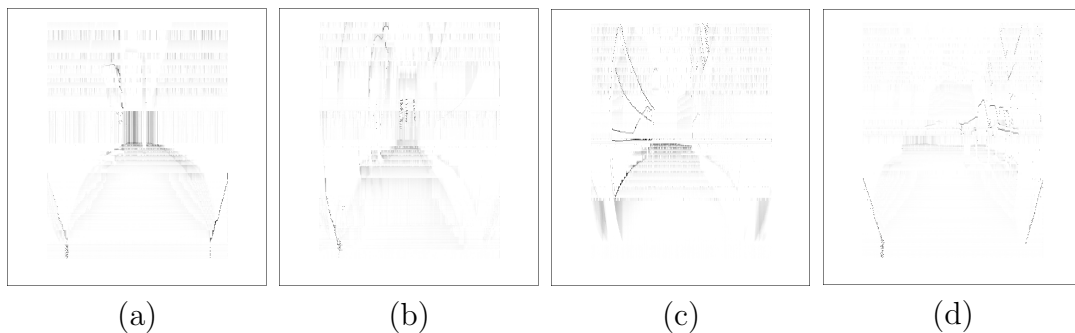


Figure 3.8: Illustrations of the difference sequences obtained for the 3-D scans given in Figure 3.1(a)–(d).

If there is any remaining offset level in  $\tilde{\mathbf{v}}_n$  as in the example given in Figure 3.6(b),  $\tilde{\mathbf{v}}_n$  is further shifted to the zero level either in the positive or the negative vertical direction by the offset value ( $\Delta$ ) indicated in the figure to improve the sparsity. Here,  $\Delta$  is the most frequently appearing value in  $\tilde{\mathbf{v}}_n$ . After shifting the amplitude of  $\tilde{\mathbf{v}}_n$  by  $\Delta$ , we eventually obtain a highly sparse innovation  $\mathbf{v}_n$ . The frequencies of the different values appearing in  $\mathbf{v}_n$ 's generated

during the compression of every 3-D scan in the first data set is given in Figure 3.9. According to the figure, the frequency of zeroes is much higher than the other values, verifying the sparsity of  $\mathbf{v}_n$ .

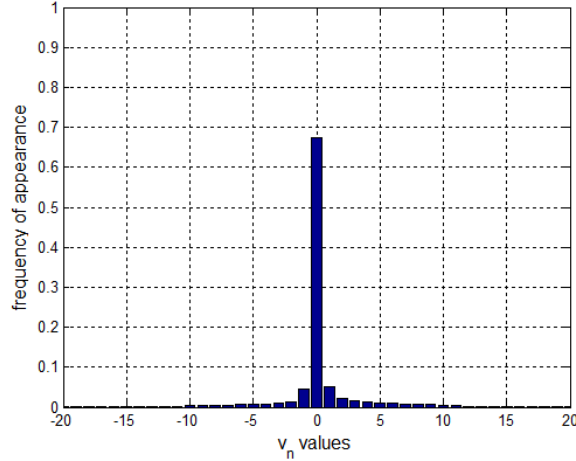


Figure 3.9: The frequencies of appearance of different values in  $\mathbf{v}_n$ .

After we obtain the innovation sequence  $\mathbf{v}_n$ , we need to test whether consecutive innovation sequences, delayed in time by  $\tau$ , are correlated with each other or not. We apply a whiteness test in the autocorrelation domain for this purpose. We assume that the innovation sequence  $\mathbf{v}_n$  of two consecutive 2-D scans consists of  $N$  random variables  $v_n[i]$  where  $i = 1, \dots, N$  and the consecutive random variables *in time* form a white and stationary random sequence. Similarly, an innovation sequence  $\mathbf{v}_{n+\tau}$ , delayed in time by  $\tau$ , is comprised of the elements  $v_{n+\tau}[i]$ . Then, the autocorrelation sequence of the  $i$ th random variable in  $v_n[i]$  is given by  $R_{v[i]}(\tau) = E \{v_n[i]v_{n+\tau}[i]\}$  where  $E\{\cdot\}$  denotes the expectation operator. If the sequence is indeed white, ideally, the autocorrelation sequence  $R_{v[i]}(\tau)$  should be an impulse sequence whose value is  $\sigma_v^2$  for  $\tau = 0$  and zero for  $\tau \neq 0$ .

When only a limited (finite) number of observations of  $v_n[i]$  are available, a biased sample autocorrelation estimate of  $v_n[i]$  can be made as follows:

$$\hat{R}_{v[i]}(\tau) = \frac{1}{N_0} \sum_{n=1}^{N_0-\tau} v_n[i]v_{n+\tau}[i] \quad (3.6)$$

where  $N_0$  is the number of available observations of  $v_n[i]$  in time. (Here,  $N_0$  is same as the number of 2-D scans that constitute a 3-D scan). With finite and fixed number of samples  $N_0$ , the sample autocorrelation estimate will have some fluctuations around the ideal (zero) that need to be tested for statistical significance. If  $N_0$  is sufficiently large ( $N_0 \geq 16$ ), it can be shown that [45] the distribution of the sample autocorrelation estimates around the true value for nonzero  $\tau$  is well approximated by a Gaussian distribution with zero mean and standard error given by

$$\hat{\sigma}_{\hat{R}}(\tau) = \frac{1}{\sqrt{N_0}} \hat{R}_{v[i]}(0) \quad \text{for } \tau \neq 0 \quad (3.7)$$

To smoothen the autocorrelation estimate  $\hat{R}_{v[i]}(\tau)$ , we average  $N$  sample autocorrelation estimates for  $i = 1, \dots, N$  to get:

$$\begin{aligned} \hat{\bar{R}}_v(\tau) &= \frac{1}{N} \sum_{i=1}^N \hat{R}_{v[i]}(\tau) \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{N_0} \sum_{n=1}^{N_0-\tau} v_n[i] v_{n+\tau}[i] \\ &= \frac{1}{N} \frac{1}{N_0} \sum_{n=1}^{N_0-\tau} \sum_{i=1}^N v_n[i] v_{n+\tau}[i] \\ &= \frac{1}{N} \frac{1}{N_0} \sum_{n=1}^{N_0-\tau} \mathbf{v}_n^T \mathbf{v}_{n+\tau} \end{aligned} \quad (3.8)$$

According to the *central limit theorem*,  $\hat{\bar{R}}_v(\tau)$  is also normally distributed with zero mean because it is the average of  $N$  autocorrelation estimates, each of which is Gaussian with finite mean and variance. Therefore, the expected value  $E\{\hat{\bar{R}}_v(\tau)\}$  of  $\hat{\bar{R}}_v(\tau)$  is zero for  $\tau \neq 0$ . Then, the variance of  $\hat{\bar{R}}_v(\tau)$  for  $\tau \neq 0$  is



derived using Equation (3.8) as follows:

$$\begin{aligned}
\sigma_{\bar{R}}^2 &= E \left\{ \hat{R}_v(\tau)^2 \right\} - E^2 \left\{ \hat{R}_v(\tau) \right\} = E \left\{ \hat{R}_v(\tau)^2 \right\} \\
&= E \left\{ \frac{1}{N^2} \frac{1}{N_0^2} \left( \sum_{n=1}^{N_0-\tau} \mathbf{v}_n^T \mathbf{v}_{n+\tau} \right)^2 \right\} \\
&= \frac{1}{N^2} \frac{1}{N_0^2} \sum_{n=1}^{N_0-\tau} \left[ E \left\{ (\mathbf{v}_n^T \mathbf{v}_{n+\tau})^2 \right\} + 2 \sum_{m=1, m \neq n}^{N_0-\tau} E \left\{ \mathbf{v}_n^T \mathbf{v}_{n+\tau} \mathbf{v}_m^T \mathbf{v}_{m+\tau} \right\} \right] \\
&= \frac{1}{N^2} \frac{1}{N_0^2} \sum_{n=1}^{N_0-\tau} \left[ E \left\{ (\mathbf{v}_n^T \mathbf{v}_{n+\tau})^2 \right\} + 2 \sum_{m=1, m \neq n}^{N_0-\tau} E \left\{ \mathbf{v}_n^T \mathbf{v}_{n+\tau} \right\} E \left\{ \mathbf{v}_m^T \mathbf{v}_{m+\tau} \right\} \right]
\end{aligned} \tag{3.9}$$

The last step follows from the assumption of uncorrelatedness of  $\mathbf{v}_n$  in time, and furthermore,  $E \left\{ \mathbf{v}_n^T \mathbf{v}_{n+\tau} \right\} = E \left\{ \mathbf{v}_m^T \mathbf{v}_{m+\tau} \right\} = 0$ . Therefore, the second term in Equation (3.9) does not contribute to  $\sigma_{\bar{R}}^2$  and the equation reduces to:

$$\sigma_{\bar{R}}^2 = \frac{1}{N^2} \frac{1}{N_0^2} \sum_{n=1}^{N_0-\tau} E \left\{ (\mathbf{v}_n^T \mathbf{v}_{n+\tau})^2 \right\} \tag{3.10}$$

According to the  $\sigma_{\bar{R}}^2$  found in Equation (3.10), using the innovation sequences of the first data set, we estimate the standard error  $\sigma_{\bar{R}}$  of the distribution of  $\hat{R}_v(\tau)$  for  $\tau \neq 0$ . If  $\mathbf{v}_n$  is indeed comprised of  $N$  random variables that form a white and stationarity random sequence in time,  $\hat{R}_v(\tau)$  must be zero mean white Gaussian, and 95.4% and 99.6% of  $\hat{R}_v(\tau)$  for  $\tau \neq 0$  must lie within  $\pm 2\sigma_{\bar{R}}$  and  $\pm 3\sigma_{\bar{R}}$ , respectively.  $\hat{R}_v(\tau)$  is estimated using the observations of the innovation sequences acquired during the compression of all 3-D scans, and plotted in Figure 3.10. According to the figure, it is observed that  $\mathbf{v}_n$  is indeed a white sequence in time, since 97.5% and 98.5% of  $R_v(\tau)$  when  $\tau \neq 0$  lie within  $\pm 2\sigma_{\bar{R}}$  and  $\pm 3\sigma_{\bar{R}}$ , respectively.

Under the additional assumption that the elements  $v_n[i], i = 1, \dots, N$  of each innovation sequence  $\mathbf{v}_n$  are uncorrelated with each other as well (i.e., in the direction of the 2-D scan) the standard error in Equation (3.10) becomes:

$$\hat{\sigma}_{\bar{R}}(\tau) = \frac{1}{\sqrt{N}} \frac{1}{\sqrt{N_0}} \hat{R}_{v[i]}(0) \text{ for } \tau \neq 0 \tag{3.11}$$

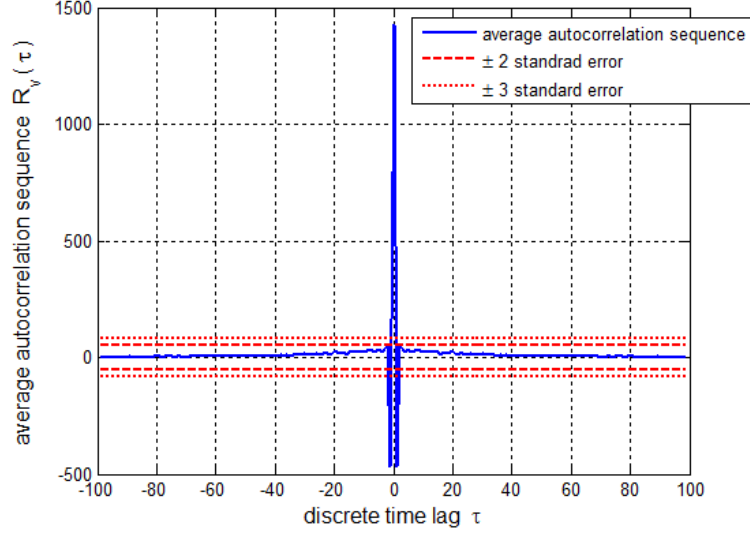


Figure 3.10: The average of the sample autocorrelation estimate of  $\mathbf{v}_n$  with  $\pm 2\sigma_{\bar{R}}$  and  $\pm 3\sigma_{\bar{R}}$  standard error boundaries.

Thus, averaging  $N$  independent autocorrelation estimates reduces the variance by a factor of  $\frac{1}{N}$  and reduces the standard error by a factor of  $\frac{1}{\sqrt{N}}$ .

Consequently,  $\mathbf{r}_n$  is represented with  $\epsilon$ ,  $\delta$ ,  $\Delta$ , and  $\mathbf{v}_n$ . When  $\mathbf{r}_n$  and  $\mathbf{r}_{n-1}$  are highly correlated,  $\mathbf{v}_n$  becomes very small, so  $\mathbf{r}_n$  is represented without  $\mathbf{v}_n$  in that case. On the other hand, when  $\mathbf{r}_n$  and  $\mathbf{r}_{n-1}$  are not sufficiently correlated,  $\mathbf{v}_n$  does not become a sparse sequence. This time,  $\mathbf{r}_n$  is not encoded. The degree of correlation between  $\mathbf{r}_n$  and  $\mathbf{r}_{n-1}$  is measured by comparing the RMSE between  $\mathbf{r}_n$  and  $\hat{\mathbf{r}}_n$  with an experimentally determined threshold that is ten times the maximum allowable distortion that can be tolerated in the reconstruction of  $\mathbf{r}_n$ . The threshold is 200 cm, since 20 cm is determined to be the upper bound on the distortion in the reconstructions. (When the distortion is over 20 cm, it is observed that the objects in the reconstructed 3-D scans are hardly recognized visually.) When  $\mathbf{r}_n$  is encoded, the algorithm followed in the sparsifying model is briefly delineated in the flowchart in Figure 3.11.

Finally, the performance of the sparsifying basis under additive white Gaussian noise is analyzed. The 2-D scans in the 3-D scan illustrated in Figure 3.1(b)

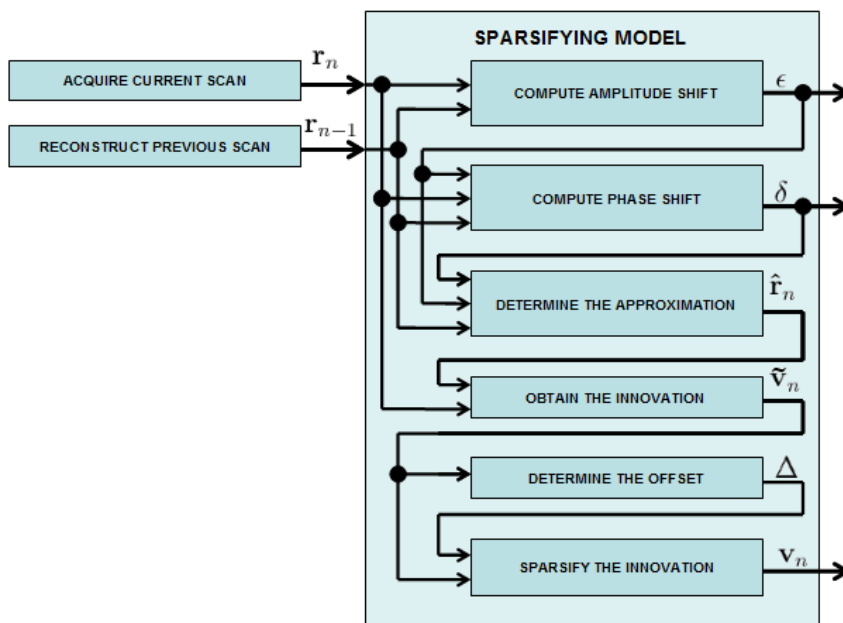


Figure 3.11: The flowchart of the sparsifying model algorithm.

are sparsified after zero mean white Gaussian noise is added to them. The percentage of the non-zero values in the representations, when no noise is added to the 3-D scan, is given in Figure 3.12(a). In this case, 6.5% of the values in the representations are non-zero on the average, the lowest achieved so far. When the standard deviation of the noise is 1, 2, 3, 4, 5, 10, 20, and 30 cm, the percentages of the number of non-zero values to the total number of values in the sparse representations are given in Figure 3.12(b)–(i), respectively. The average percentages of non-zero values in these cases are 5.1%, 4.5%, 7.3%, 11.8%, 13.7%, 20.3%, 73.7%, and 81.9%, respectively. As the sparsity of the representations mentioned above are compared with each other, it is observed that under the presence of noise with standard deviation up to 3 cm, the sparsifying model maintains its performance that is observed in the case without any added noise. Besides that, the model provides representations with acceptable sparsity, under the presence of noise with standard deviation as much as 10 cm. Beyond this level of noise, the representations cannot be considered as sparse.

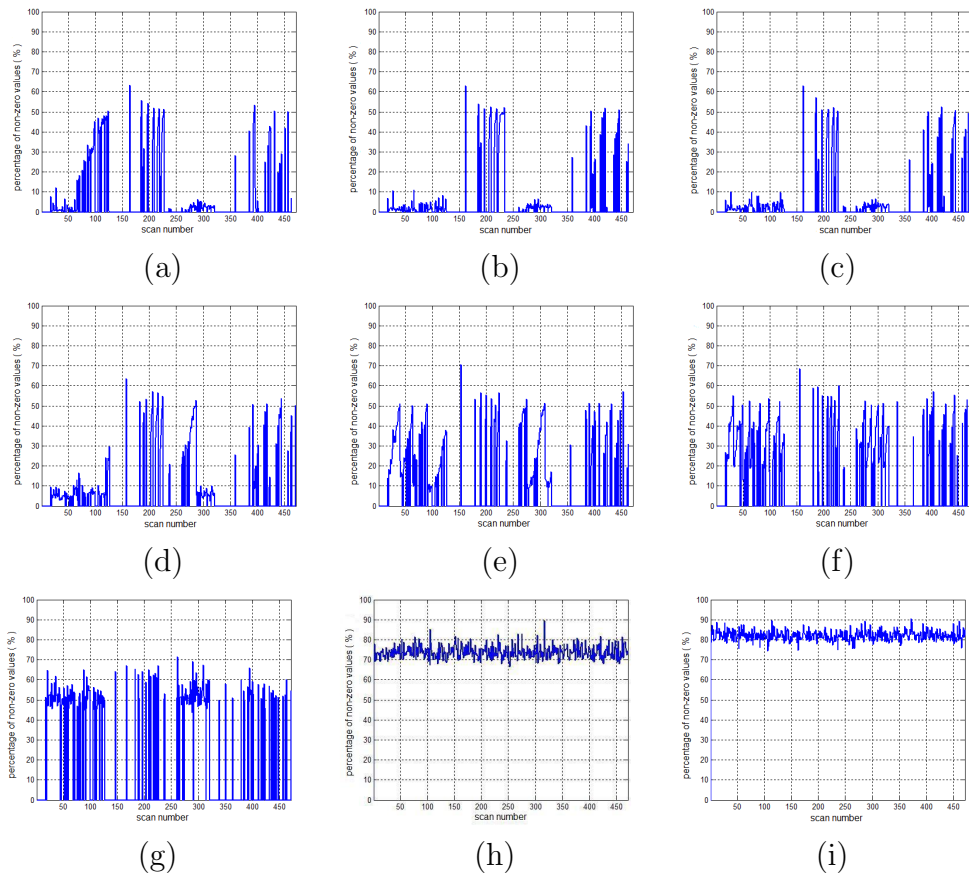


Figure 3.12: The percentage of the number of non-zero values to the total number of values in the sparse representations generated at the sparsifying model for the 3-D scan illustrated in Figure 3.1(b) with white Gaussian noise with zero mean and 0, 1, 2, 3, 4, 5, 10, 20, and 30 cm standard deviation, respectively.

Note that the method proposed here has some similarities with *optical flow* techniques used for motion estimation in image and video processing [46, 47]. In optical flow, spatial and temporal shifts are used to estimate the relative motion between the scene and the camera (the observer). The solution of the following partial differential equation is required:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \quad (3.12)$$

where  $V_x$  and  $V_y$  are the  $x$  and  $y$  components of the velocity of the optical flow of the intensity  $I(x, y, t)$ , and  $\frac{\partial I}{\partial x}$ ,  $\frac{\partial I}{\partial y}$ , and  $\frac{\partial I}{\partial t}$  are the partial derivatives of the image at  $(x, y, t)$  in the corresponding directions. In our method, two spatial shifts  $\delta$  and  $\epsilon$  (and  $\Delta$ ) are involved whose time derivatives correspond to  $V_x$  and  $V_y$  in the optical flow equation, respectively.

## 3.2 The Measurement Model

The measurement model gets the minimum number of samples from  $\mathbf{v}_n$  by using either simple coding (SC) or compressive sampling (CS). Simple coding encodes  $\mathbf{v}_n$  with the pairs of location and amplitude of the non-zero values in  $\mathbf{v}_n$ . The measurement size  $M$  in this case, increases proportionally with the number of non-zero values  $K$ , where  $M = 2K$ . Despite this, the reconstruction error is zero when  $\mathbf{v}_n$  is rebuilt from the measurements taken with SC. Compressive sampling measures arbitrary linear combinations of the values in  $\mathbf{v}_n$ . In this case, the measurement model is determined as described in Section 2.2, and  $M$  is determined using Equation (2.7). Then,  $\mathbf{v}_n$  is encoded with the multiplication  $\Phi\mathbf{v}_n$ . Furthermore, the resulting reconstruction error, which arises when  $\mathbf{v}_n$  is rebuilt from the measurements taken with CS, increases with  $K$ . This fact is illustrated in Figure 3.13 with the graph of the average RMS of the observed reconstruction error with respect to  $K$ , during the compression of all 3-D scans

in the first data set. The measurements obtained from  $\mathbf{v}_n$  using either SC or CS are kept in a column vector  $\mathbf{m}$  in  $\mathfrak{R}^M$ .

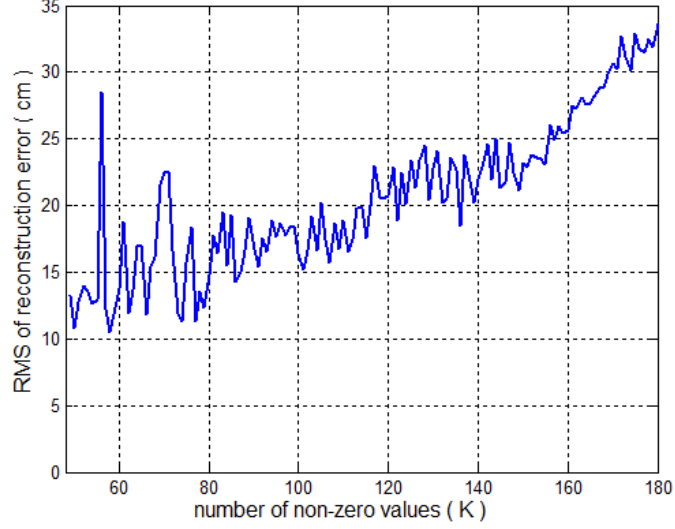


Figure 3.13: The RMS of the reconstruction error with respect to the number of non-zero values in the sparse data, when the 2-D scans from all 3-D scans in the first data set are sampled using compressive sampling.

The measurement size  $M$  for the measurements  $\mathbf{m}$  taken using either SC or CS is illustrated in Figure 3.14. According to the figure, SC seems to be advantageous over CS in terms of  $M$  and the reconstruction error, when  $K$  is below the level indicated by  $K^*$  in the figure. Here,  $K^*$  is the value of  $K$  that makes  $M$  for SC equal to  $M$  for CS. Consequently, we apply SC when  $K \leq K^*$ , and apply CS, otherwise. We include a special character (i.e.,  $\pi$ ) at the beginning of  $\mathbf{m}$  when SC is applied to inform the decoder that we are using SC instead of CS. Besides, when  $K > \frac{N}{2}$ ,  $\mathbf{v}_n$  cannot be considered sparse, since the reconstruction error would be very high if  $\mathbf{v}_n$  were sampled using CS. In that case,  $\mathbf{r}_n$  is not encoded. When  $\mathbf{r}_n$  is encoded, the algorithm followed in the measurement model is given in the flowchart in Figure 3.15.

At the output of the measurement model,  $\mathbf{r}_n$  is represented with  $\{\epsilon, \delta, \Delta, \mathbf{m}\}$  if it is encoded. Otherwise,  $\mathbf{r}_n$  is left as it is, which is indicated by the impulses with amplitude  $N$  in Figure 3.16(a), where the lengths of the representations for

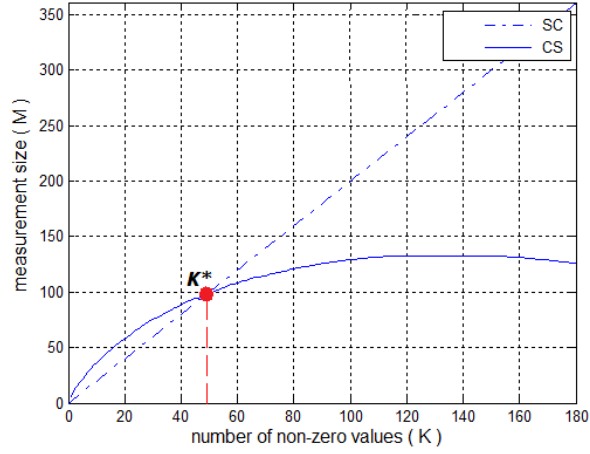


Figure 3.14: The measurement size  $M$  in SC and CS with respect to the number of non-zero values of a signal in  $\mathfrak{R}^{361}$ .

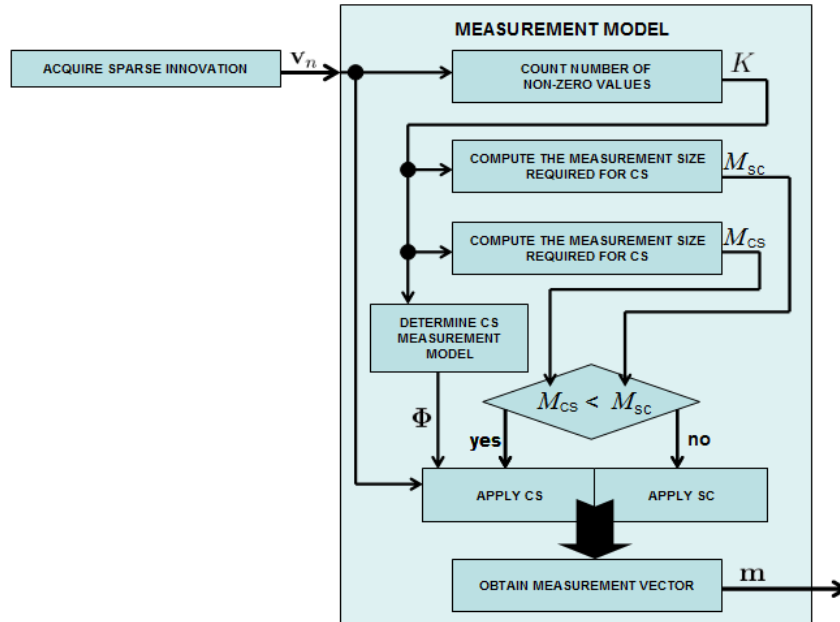


Figure 3.15: The flowchart of the measurement model algorithm.

each 2-D scan in the 3-D scan illustrated in Figure 3.1(b) are shown. Here, one can consider whether it may be possible to achieve measurements with shorter lengths using a simpler method, such as *run length encoding* (RLE) [48]. RLE first determines the sets in the input data, each of which is formed by the repetition of a single character. Then, it encodes each set with its length and the character repeated in this set. This simple method is commonly used in encoding fax images of typical office documents. The lengths of measurements when RLE is employed to encode  $\mathbf{v}_n$  for the same 3-D scan are given in Figure 3.16(b). As the two parts of Figure 3.16 are compared, it is seen that the measurement model of the proposed method provides more efficient measurements than RLE.

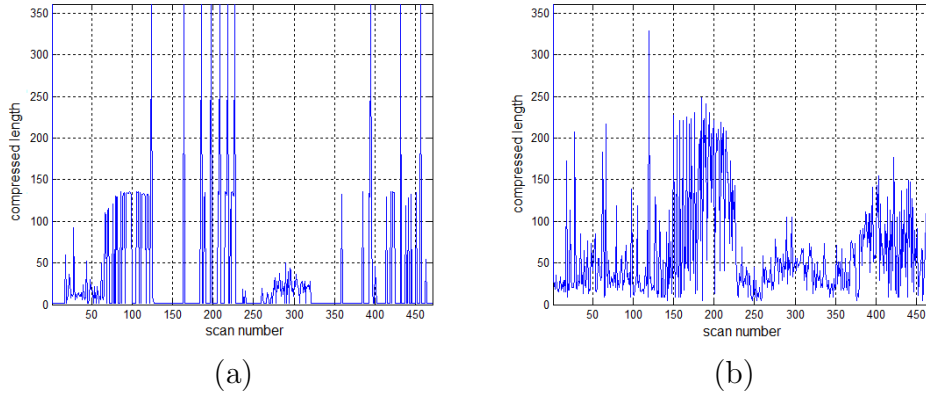


Figure 3.16: The length of the measurements for the data set illustrated in Figure 3.1(b), when (a) the measurement model and (b) RLE are employed.

### 3.3 The Reconstruction Model

The reconstruction model rebuilds  $\mathbf{r}_n$  from the output generated by the encoder. When  $\mathbf{r}_n$  is encoded, the output is composed of  $\{\epsilon, \delta, \Delta, \mathbf{m}\}$ , and its length is  $(M + 3)$ , which is less than  $N$ . If  $\mathbf{r}_n$  is not encoded, the output is  $\mathbf{r}_n$  with length  $N$ . Therefore, the reconstruction procedure starts with determining the length of the encoder output. If the length is  $N$ , the output is stored directly as the reconstruction of  $\mathbf{r}_n$ . Otherwise, the rest of the reconstruction procedure is applied to the output.



In the reconstruction procedure, the output is then decomposed into  $\epsilon$ ,  $\delta$ ,  $\Delta$ , and  $\mathbf{m}$ . After this step,  $\mathbf{r}_{n-1}$ , which is previously reconstructed, is shifted along the vertical and horizontal axes by  $\epsilon$  and  $\delta$ , respectively. The resultant signal  $\hat{\mathbf{r}}_n$  is the approximation to  $\mathbf{r}_n$ . Afterwards,  $\tilde{\mathbf{v}}_n$  is rebuilt from  $\mathbf{m}$  and  $\Delta$ . In this step, if the first value of  $\mathbf{m}$  is  $\pi$ , then  $\mathbf{v}_n$  is rebuilt, decoding the rest of  $\mathbf{m}$  with respect to the SC scheme, which involves filling an empty signal in  $\mathfrak{R}^N$  with the values of location and amplitude pairs given in the measurements. Otherwise,  $\mathbf{v}_n$  is rebuilt, decoding  $\mathbf{m}$  with respect to the CS scheme, which involves solving Equation (2.5), where  $\mathbf{y} = \mathbf{m}$ ,  $\Theta = \Phi$ , and  $\hat{\mathbf{s}} = \mathbf{v}_n$ , following the procedure in [33]. In our implementation,  $\mathbf{v}_n$  is determined using the MATLAB function “*perform\_l1\_recovery*” written by Peyre [49]. Then,  $\tilde{\mathbf{v}}_n$  is obtained by shifting the amplitude of  $\mathbf{v}_n$  by  $-\Delta$ . Eventually,  $\mathbf{r}_n$  is reconstructed by adding  $\tilde{\mathbf{v}}_n$  to  $\hat{\mathbf{r}}_n$ . The algorithm followed in the reconstruction model is summarized in the flowchart in Figure 3.17.

The reconstruction model is used at the decoder, as well as at the encoder to estimate the reconstructions generated by the decoder.

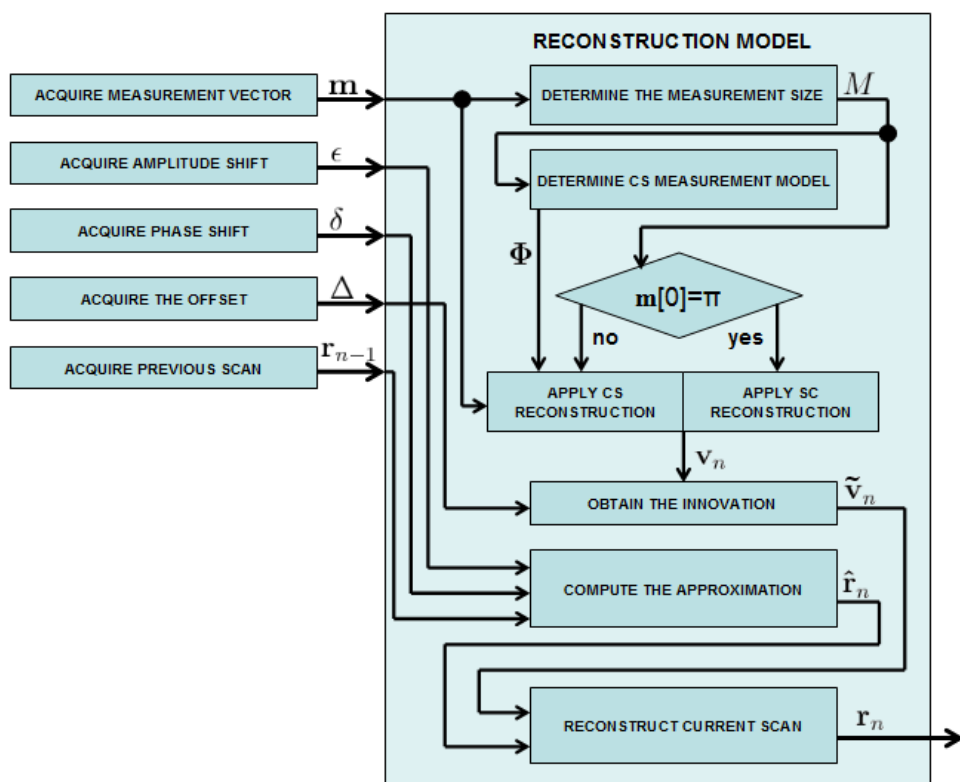


Figure 3.17: The flowchart of the reconstruction model algorithm.

## Chapter 4

# Comparing Compression Performance of the Proposed Method with Some Well-Known Compression Techniques

In this chapter, we compare the compression performance of the proposed method with some well-known and widely used lossless and lossy compression techniques. The 3-D scans referred as scan 01, scan 02, ... etc., are compressed by applying each technique to the 2-D scans forming the 3-D scans individually. For each technique in the comparison, we compare the overall CR, the average distortion ( $\mathcal{D}$ ) that is the average RMSE between the 2-D scans and their reconstructions defined in Chapter 1, and the time required for encoding ( $t_{\text{enc}}$ ) and decoding ( $t_{\text{dec}}$ ) the 3-D scans. These values are found by averaging over the values obtained for all 3-D scans, including 4,930,899 (= 29 3-D scans  $\times$  471 2-D scans  $\times$  361 measurements) range measurements in the first data set, and 6,660,450 (= 82 3-D scans  $\times$  225 2-D scans  $\times$  361 measurements) range measurements in the second data set.

The following implementations are executed on a computer platform with 2 GHz Intel Core2 Duo processor including 2 GB RAM. All executable tasks are run in MATLAB environment installed on Microsoft Windows Vista operating system.

## 4.1 Implementation and Comparison with Well-Known Lossless Techniques

In this section, the 3-D scans in the first data set are compressed using four different lossless techniques, which are *Huffman*, *arithmetic*, *ZLIB*, and *GZIP* coding techniques.

Huffman coding maps every character in the input data to distinct binary patterns based on the frequency of appearance of the characters. It is the optimal lossless coding technique since the characters that appear more frequently are mapped to shorter patterns than the characters that appear less frequently, and the two characters that appear least frequently are mapped to two different patterns having the same length [30]. Arithmetic coding maps *blocks* of characters, instead of single characters, to distinct binary patterns based on how frequently the blocks appear. It is observed that arithmetic coding can sometimes be more efficient than Huffman coding, depending on the nature of the signal to be encoded [30]. The 3-D scans are encoded by Huffman and arithmetic coding using the *huffmanenco* and *arithenco* functions in MATLAB Communications Toolbox, respectively.

ZLIB and GZIP are two popular compression techniques that are variations of *LZ77* [50], which is a widely used compression method that encodes repeated strings in the input data with pairs of *distance* and *length*. Distance is the separation between the beginning of the last location and the previous location of

the repeated string in the data. Length is the size of the corresponding repeated string. Two independent Huffman trees are used in compressing distance and length information, respectively. ZLIB is a general purpose coding library, and can be used in any operating system. ZLIB is reported to provide satisfactory compression on various types of data with optimum use of system resources. ZLIB is also claimed to be able to compress the input data at most by 99.9% in theory [51]. GZIP is a coding technique that is designed to be used instead of *compress*, which is a compression utility used in UNIX operating systems. The files that have been compressed using GZIP carry the suffix “.gz” [52]. The 3-D scans are encoded by ZLIB and GZIP using the functions written by Kleider [53] and Hopkins [54], respectively.

The compression performances of the lossless methods mentioned here are tabulated in Table 4.1 and 4.2. According to the values in the table, arithmetic coding can be said to be efficient in terms of the CR, however it is slow compared to the other techniques except Huffman coding. On the other hand, these techniques compress less than arithmetic coding. Despite the high average CR when Huffman coding is applied to the raw scan data, the CR can be lowered by coding the differences between consecutive 2-D scans, since the range of the distinct characters in the differences is narrower than in the raw scan data. With this approach, both the CR and  $t_{\text{enc}}$  of Huffman coding is reduced to about 12% and 49 seconds, respectively. However, coding the differences instead of the raw scan data may not lower the CR for other compression techniques because their compression performance is not directly related to the range of the distinct characters in the input data, as in Huffman coding. For instance, the average CR for arithmetic coding increases to 16.4%. The compression performances for Huffman coding and arithmetic coding in this case are given in Table 4.3.

3-D scan	Huffman coding			arithmetic coding		
	CR (%)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	CR (%)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
scan 01	41.8	157.4	603.1	4.2	31.6	40.1
scan 02	42.2	182.3	682.8	10.9	36.9	48.0
scan 03	43.0	219.6	798.1	11.3	37.7	49.0
scan 04	40.8	160.5	638.4	11.3	36.5	47.6
scan 05	40.6	156.5	605.1	11.3	35.9	46.7
scan 06	40.2	125.5	510.6	11.3	35.2	45.8
scan 07	40.6	138.0	538.4	11.1	36.7	48.1
scan 08	39.5	123.7	487.0	10.9	38.0	50.1
scan 09	41.2	152.6	600.7	10.9	38.5	50.4
scan 10	41.6	153.5	592.7	11.1	39.7	51.9
scan 11	41.7	186.0	687.7	12.0	38.3	49.4
scan 12	40.5	128.3	488.4	11.2	37.2	48.8
scan 13	41.7	170.2	608.1	11.2	38.4	50.3
scan 14	42.8	196.7	685.8	11.3	38.3	49.6
scan 15	43.3	203.4	709.5	11.4	40.3	52.7
scan 16	42.1	172.6	623.5	11.1	38.0	49.7
scan 17	42.6	187.9	664.9	11.4	38.2	49.7
scan 18	41.7	151.8	568.2	11.4	39.9	52.1
scan 19	40.2	126.6	482.2	11.1	37.4	48.7
scan 20	41.1	135.5	518.1	11.1	38.0	49.2
scan 21	41.3	138.2	525.7	11.2	37.1	48.6
scan 22	40.8	147.4	542.9	11.2	36.9	48.1
scan 23	42.2	191.2	659.7	11.4	37.4	48.8
scan 24	42.1	202.8	691.2	12.0	38.0	48.9
scan 25	43.0	169.8	630.3	11.5	38.0	49.2
scan 26	41.5	166.5	597.2	11.7	37.4	48.6
scan 27	42.7	179.5	645.5	11.9	38.3	49.3
scan 28	42.1	178.2	632.1	11.9	37.9	48.8
scan 29	43.1	200.4	698.0	11.9	39.1	50.5
<b>average:</b>	<b>41.7</b>	<b>165.6</b>	<b>610.6</b>	<b>11.1</b>	<b>37.6</b>	<b>48.9</b>

Table 4.1: Compression ratio (CR), the time required for encoding ( $t_{\text{enc}}$ ) and decoding ( $t_{\text{dec}}$ ) when the raw 3-D scans in the first data set are compressed using Huffman and arithmetic coding.

<b>3-D scan</b>	<b>ZLIB</b>			<b>GZIP</b>		
	CR (%)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	CR (%)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
scan 01	66.3	0.4	0.2	76.7	0.6	0.3
scan 02	67.3	0.4	0.2	76.7	0.5	0.3
scan 03	68.9	0.4	0.2	76.7	0.5	0.3
scan 04	62.5	0.4	0.2	76.7	0.5	0.3
scan 05	63.4	0.4	0.2	76.7	0.5	0.3
scan 06	60.8	0.4	0.2	76.7	0.5	0.3
scan 07	64.6	0.4	0.2	76.7	0.5	0.3
scan 08	60.6	0.4	0.2	76.7	0.5	0.3
scan 09	63.2	0.4	0.2	76.7	0.6	0.4
scan 10	64.2	0.4	0.2	76.7	0.5	0.3
scan 11	70.1	0.4	0.2	76.7	0.5	0.3
scan 12	61.8	0.4	0.2	76.7	0.5	0.3
scan 13	65.2	0.4	0.2	76.7	0.5	0.3
scan 14	65.2	0.4	0.2	76.7	0.5	0.3
scan 15	65.8	0.4	0.2	76.7	0.5	0.3
scan 16	63.3	0.4	0.2	76.7	0.5	0.3
scan 17	65.9	0.4	0.2	76.7	0.5	0.3
scan 18	64.3	0.4	0.2	76.7	0.5	0.3
scan 19	61.7	0.4	0.2	76.7	0.5	0.3
scan 20	64.9	0.4	0.2	76.7	0.5	0.3
scan 21	60.6	0.4	0.2	76.7	0.5	0.3
scan 22	66.8	0.4	0.2	76.7	0.6	0.3
scan 23	64.1	0.4	0.2	76.7	0.5	0.4
scan 24	68.9	0.4	0.2	76.7	0.5	0.3
scan 25	72.8	0.5	0.2	76.7	0.5	0.3
scan 26	64.5	0.4	0.2	76.7	0.5	0.3
scan 27	68.7	0.4	0.2	76.7	0.5	0.3
scan 28	67.6	0.4	0.2	76.7	0.5	0.3
scan 29	70.9	0.4	0.2	76.7	0.5	0.3
<b>average:</b>	<b>65.3</b>	<b>0.4</b>	<b>0.2</b>	<b>76.7</b>	<b>0.5</b>	<b>0.3</b>

Table 4.2: Compression ratio (CR), the time required for encoding ( $t_{\text{enc}}$ ) and decoding ( $t_{\text{dec}}$ ) when the raw 3-D scans in the first data set are compressed using ZLIB and GZIP.

<b>3-D scan</b>	<b>Huffman coding</b>			<b>arithmetic coding</b>		
	CR (%)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	CR (%)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
scan 01	11.5	672.7	12.3	7.8	40.8	50.8
scan 02	12.2	27.8	14.1	12.8	44.2	55.9
scan 03	11.4	24.3	12.0	13.1	44.7	56.7
scan 04	11.4	25.7	12.5	13.2	43.9	55.5
scan 05	11.1	23.4	11.4	13.3	43.9	55.7
scan 06	11.5	24.5	12.1	13.1	43.8	55.6
scan 07	11.7	25.3	12.6	12.8	43.4	55.3
scan 08	11.3	24.1	11.9	12.5	42.5	54.7
scan 09	11.4	25.8	12.7	12.4	43.5	55.8
scan 10	12.1	27.6	13.9	12.8	43.9	56.0
scan 11	13.3	31.5	16.7	14.5	44.8	55.7
scan 12	11.5	25.1	12.5	12.9	43.5	55.3
scan 13	11.9	25.9	13.0	13.0	44.7	56.9
scan 14	11.9	26.5	13.4	13.2	45.0	57.2
scan 15	11.5	25.3	12.6	13.4	45.2	57.2
scan 16	11.7	25.2	12.6	12.9	44.8	56.8
scan 17	11.9	25.9	13.0	13.3	45.1	57.3
scan 18	11.7	25.7	12.9	13.4	45.4	57.4
scan 19	11.4	24.9	12.2	21.0	47.4	61.8
scan 20	11.7	25.4	12.8	21.6	47.7	62.4
scan 21	11.0	23.0	11.2	21.7	47.3	62.1
scan 22	13.7	29.8	16.1	21.7	46.8	61.5
scan 23	12.8	26.5	13.8	22.0	47.8	62.4
scan 24	13.7	29.3	15.8	23.3	48.3	62.0
scan 25	12.0	24.8	12.6	22.2	48.7	63.2
scan 26	13.8	31.0	16.7	22.8	48.1	62.2
scan 27	13.5	33.0	17.6	23.1	49.0	63.0
scan 28	14.1	32.2	17.5	23.1	49.3	63.4
scan 29	14.0	30.8	16.8	23.1	49.2	63.5
<b>average:</b>	<b>12.0</b>	<b>49.0</b>	<b>14.0</b>	<b>16.4</b>	<b>45.6</b>	<b>58.4</b>

Table 4.3: CR,  $t_{\text{enc}}$ , and  $t_{\text{dec}}$  when the differences between consecutive scans in the first data set are compressed using Huffman and arithmetic coding.



## 4.2 Implementation and Comparison with Well-Known Lossy Techniques

Besides the lossless compression techniques, we also apply two lossy compression methods to the 3-D scans in the first data set: *JPEG* and the *wavelet transform*. In this case, we analyze the compression performance also in terms of  $\mathcal{D}$ , which indicates the average distortion on each sample in the reconstructions.

Since the 2-D scans forming a 3-D scan are basically cross-sectional intensity images of the scanned environment where the intensity values represent the depth information, JPEG compression is applied first. JPEG refers to a family of image compression standards including both lossless and lossy techniques. Lossy JPEG techniques are based on the discrete cosine transform (DCT) applied on  $8 \times 8$  blocks of pixels in the image data. The CR for lossy JPEG is claimed to be as low as about 5% in compressing colored images, when the distortion in the reconstructed images is not visually recognizable [55]. In colored images, each pixel is represented with three channels, each of which holds an 8-bit intensity value that corresponds to an unsigned integer between 0 and 255. On the other hand, range measurements in the 2-D scans from the experimental data sets are unsigned integers between 0 and 8000 cm that can be represented with 16-bits. Therefore, before encoding a 2-D scan using JPEG, each range measurement is encoded with three channels such that the most significant eight bits are placed in the first channel, the least significant eight bits are placed in the second channel, and the third channel is left blank. Then, the 2-D scan, which is now a  $1 \times N$  image with three channels, is duplicated eight times to form an  $8 \times N$  image, since JPEG divides the image into  $8 \times 8$  blocks before applying DCT. Eventually, the resultant image data is encoded by lossy JPEG using MATLAB *imwrite* function. The compression performance of JPEG is given in Table 4.4. According to the

table, JPEG can be considered as a fast technique with about 9% CR, but a large distortion of 164.6 cm.

3-D scan	JPEG			
	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
scan 01	8.8	142.2	4.0	7.2
scan 02	9.0	149.8	3.7	6.7
scan 03	8.7	131.0	3.5	6.8
scan 04	8.9	103.0	3.8	6.7
scan 05	8.8	137.5	4.1	7.4
scan 06	8.8	137.3	4.6	7.3
scan 07	8.7	140.1	4.7	7.2
scan 08	8.6	135.9	6.4	8.0
scan 09	8.6	135.1	4.3	7.7
scan 10	9.0	149.0	4.3	7.5
scan 11	9.5	186.6	4.9	7.8
scan 12	8.6	156.4	6.4	9.3
scan 13	8.8	174.7	4.4	8.1
scan 14	8.9	179.0	5.0	8.2
scan 15	9.1	172.8	4.8	7.7
scan 16	8.7	169.2	3.7	7.0
scan 17	8.9	158.9	3.6	6.8
scan 18	9.0	136.6	3.8	6.7
scan 19	8.6	97.9	3.6	6.9
scan 20	8.7	107.5	3.8	6.8
scan 21	8.7	127.9	3.8	6.8
scan 22	9.2	179.0	3.7	6.9
scan 23	8.8	207.3	3.6	7.2
scan 24	9.1	293.6	3.8	7.2
scan 25	8.8	120.6	3.5	6.7
scan 26	9.0	268.7	3.7	6.9
scan 27	9.8	176.6	4.0	7.3
scan 28	9.9	252.7	3.6	7.3
scan 29	9.8	246.6	3.4	7.2
<b>average:</b>	<b>9.0</b>	<b>164.6</b>	<b>4.2</b>	<b>7.3</b>

Table 4.4: CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ , and  $t_{\text{dec}}$  when the first data set is compressed using JPEG.

Besides JPEG, the wavelet transform, which is also widely used in image compression, is applied to the raw scan data. The wavelet transform analyzes the input signal at separate bandwidths by applying the input signal to a specific *filterbank* that is a set of low-pass and high-pass filters connected together in a

network. The basic structure is a two-channel filterbank, which is formed by *analysis* and *synthesis* parts, as depicted in Figure 4.1. There are generally pairs of low-pass ( $H_0$ ) and high-pass ( $H_1$ ) filters in the analysis part, where the input signal is divided into the high frequency ( $x_H$ ) and low frequency ( $x_L$ ) components with half the length of the input signal, after they are downsampled by 2. The signal is then reconstructed from the summation of these components in the synthesis part, where the components are first upsampled by 2, then applied to the synthesis filter pairs ( $F_0$  and  $F_1$ ) [56]. Every 2-D scan in the 3-D scans is compressed using up to 3-level wavelet transform with Haar filterbank, for which the function *dwt* in MATLAB Wavelet Toolbox is employed. The raw scan data is decomposed into a number of frequency components ranging from low frequencies to high frequencies denoted by  $x_L$  and  $x_H$  in 1-level transform;  $x_{LL}, x_{LH}, x_{HL}, x_{HH}$  in 2-level transform;  $x_{LLL}, x_{LLH}, x_{LHL}, x_{LHH}, x_{HLL}, x_{HLH}, x_{HHL}, x_{HHH}$  in 3-level transform as placed in the analysis part given in Figure 4.2(a). For the reconstruction, only the lowest frequency components, which are  $x_L, x_{LL}, x_{LLL}$  in 1-level, 2-level, and 3-level transforms, respectively, are used in the synthesis part given in Figure 4.2(b). Therefore, there appears some distortion on the reconstructions. According to the compression performances given in Table 4.5, the CR decreases exponentially for the wavelet transform as the level increases. On the other hand, the distortion increases simultaneously. Furthermore, the wavelet transform is very fast, such that the time required for encoding or decoding does not exceed one second.

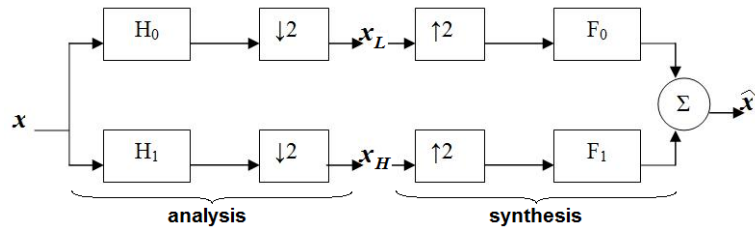


Figure 4.1: Two-channel filterbank structure.

3-D scan	1-level transform				2-level transform				3-level transform			
	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
scan 01	50.2	18.0	0.3	0.3	25.2	25.1	0.6	0.6	12.7	33.5	0.8	1.0
scan 02	50.2	22.0	0.3	0.3	25.2	31.4	0.6	0.6	12.7	38.2	0.8	1.0
scan 03	50.2	24.5	0.3	0.3	25.2	31.2	0.5	0.6	12.7	35.5	0.8	1.0
scan 04	50.2	24.1	0.3	0.3	25.2	34.2	0.5	0.6	12.7	43.8	0.8	0.9
scan 05	50.2	19.0	0.3	0.3	25.2	27.9	0.5	0.6	12.7	36.4	0.8	1.0
scan 06	50.2	18.5	0.3	0.3	25.2	25.1	0.5	0.6	12.7	31.8	0.8	0.9
scan 07	50.2	12.5	0.3	0.3	25.2	17.2	0.6	0.6	12.7	23.0	0.8	1.0
scan 08	50.2	9.2	0.3	0.3	25.2	13.0	0.5	0.6	12.7	17.0	0.8	0.9
scan 09	50.2	9.3	0.3	0.3	25.2	14.7	0.6	0.6	12.7	19.6	0.8	1.0
scan 10	50.2	16.4	0.3	0.3	25.2	25.2	0.5	0.6	12.7	34.7	0.8	1.0
scan 11	50.2	33.3	0.3	0.3	25.2	44.7	0.5	0.6	12.7	54.5	0.8	1.0
scan 12	50.2	15.3	0.3	0.3	25.2	21.3	0.5	0.6	12.7	26.4	0.8	0.9
scan 13	50.2	15.1	0.3	0.3	25.2	20.5	0.5	0.6	12.7	25.9	0.8	0.9
scan 14	50.2	15.7	0.3	0.3	25.2	25.9	0.6	0.6	12.7	31.9	0.8	0.9
scan 15	50.2	22.1	0.3	0.3	25.2	31.6	0.5	0.6	12.7	40.0	0.8	1.0
scan 16	50.2	15.5	0.3	0.3	25.2	22.1	0.5	0.6	12.7	29.0	0.8	0.9
scan 17	50.2	19.9	0.3	0.3	25.2	27.1	0.6	0.7	12.7	31.3	0.9	1.0
scan 18	50.2	22.6	0.3	0.3	25.2	29.9	0.5	0.6	12.7	35.6	0.8	0.9
scan 19	50.2	22.6	0.3	0.3	25.2	28.5	0.5	0.6	12.7	32.3	0.8	0.9
scan 20	50.2	15.2	0.3	0.3	25.2	21.2	0.5	0.6	12.7	27.2	0.8	0.9
scan 21	50.2	20.4	0.3	0.3	25.2	27.7	0.5	0.6	12.7	32.6	0.8	0.9
scan 22	50.2	22.1	0.3	0.3	25.2	31.8	0.5	0.6	12.7	40.4	0.8	0.9
scan 23	50.2	17.5	0.3	0.3	25.2	25.3	0.5	0.6	12.7	32.7	0.8	0.9
scan 24	50.2	26.9	0.3	0.3	25.2	37.2	0.5	0.6	12.7	47.3	0.8	0.9
scan 25	50.2	22.1	0.3	0.3	25.2	28.9	0.5	0.6	12.7	35.3	0.8	0.9
scan 26	50.2	27.4	0.3	0.3	25.2	36.9	0.5	0.6	12.7	45.9	0.8	0.9
scan 27	50.2	43.1	0.3	0.3	25.2	61.1	0.5	0.6	12.7	76.1	0.8	0.9
scan 28	50.2	35.8	0.3	0.3	25.2	51.1	0.5	0.6	12.7	63.1	0.8	0.9
scan 29	50.2	33.3	0.3	0.3	25.2	46.9	0.5	0.6	12.7	60.9	0.8	0.9
<b>average:</b>	<b>50.2</b>	<b>21.3</b>	<b>0.3</b>	<b>0.3</b>	<b>25.2</b>	<b>29.8</b>	<b>0.5</b>	<b>0.6</b>	<b>12.7</b>	<b>37.3</b>	<b>0.8</b>	<b>0.9</b>

Table 4.5: CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ , and  $t_{\text{dec}}$  when the first data set is compressed using 1-level, 2-level, and 3-level wavelet transforms.

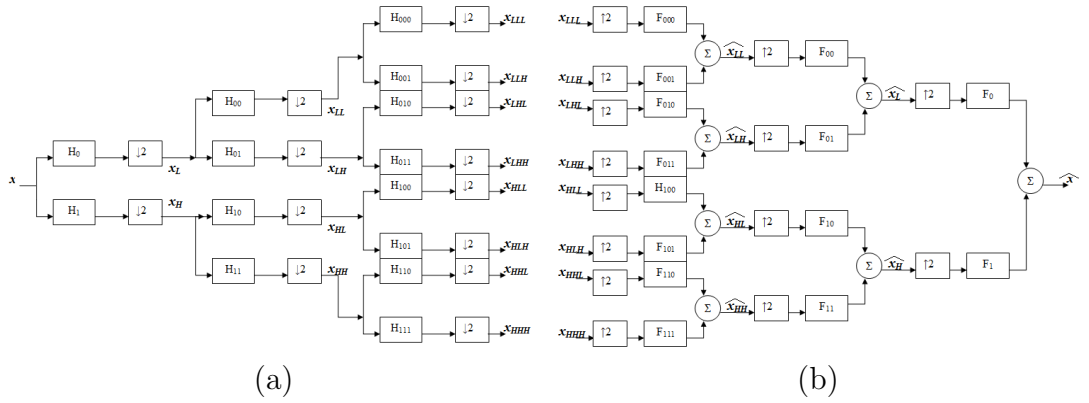


Figure 4.2: (a) The analysis and (b) the synthesis structures of a 3-level wavelet transform.

### 4.3 Implementation and Comparison with the Proposed Method

In this case, the 3-D scans in the first data set are encoded using the proposed method which is a lossy technique. In the implementation, small fluctuations in the compression performance are observed, such that at most  $\pm 2\%$  variations appear in the CR, since the measurement model in CS is determined arbitrarily in each trial. Recall that in the proposed method, each 2-D scan in a 3-D scan is encoded with one of the following:

- $\epsilon$ ,  $\delta$ ,  $\Delta$ , and  $\mathbf{m}$  taken using CS (referred as SHIFT+CS),
- $\epsilon$ ,  $\delta$ ,  $\Delta$ , and  $\mathbf{m}$  taken using SC (referred as SHIFT+SC),
- $\epsilon$ ,  $\delta$ , and  $\Delta$  (referred as SHIFT),
- as itself (referred as NOCODING).

The number of occurrence of each type of code for a given 3-D scan is denoted by  $k_{\text{SHIFT+CS}}$ ,  $k_{\text{SHIFT+SC}}$ ,  $k_{\text{SHIFT}}$ , and  $k_{\text{NOCODING}}$ , respectively. Thus, these numbers change as the CR fluctuates. Moreover, the distortion also changes in this situation. Therefore, every 3-D scan is encoded using the proposed method ten times,

then the average values of the CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ ,  $t_{\text{dec}}$ ,  $k_{\text{SHIFT+CS}}$ ,  $k_{\text{SHIFT+SC}}$ ,  $k_{\text{SHIFT}}$ , and  $k_{\text{NOCODING}}$  are obtained as given in Table 4.6. According to the table, it is observed that the compression performance changes with respect to the complexity of the features in the scanned environment. For instance, the proposed method compresses the 3-D scans illustrated in Figure 3.1(a)–(d) (referred as scan 14, 20, 22, and 28 in the tables, respectively) by about 88%, 90%, 86%, and 85%, respectively. Here, the second scanned environment has less detail than the others, so that 60% of the 2-D scans in this 3-D scan are encoded with only  $\{\epsilon, \delta, \Delta\}$ , and the CR is lower than the others. On the average, all 3-D scans are compressed by 89% with about 13 cm distortion in reconstruction. In this case, 16% of the 2-D scans is encoded with  $\epsilon, \delta, \Delta$ , and  $\mathbf{m}$  obtained using CS; 24% of them is encoded with  $\epsilon, \delta, \Delta$ , and  $\mathbf{m}$  obtained using SC; 57% of the 2-D scans is encoded with  $\epsilon, \delta$ , and  $\Delta$ . Only 3% of them is not encoded. Consequently, it is observed that the CR depends on the correlation between consecutive 2-D scans that is somehow related to the details in the scanned environment. The higher the correlation and the less the detail, the lower the CR.

Moreover, the distortion  $\mathcal{D}$  is dependent on the information provided by the code of  $\mathbf{r}_n$ . At first,  $\mathbf{r}_n$  is defined with the relative vertical and horizontal shifts ( $\epsilon$  and  $\delta$ ) to  $\hat{\mathbf{r}}_n$ . When the RMSE between  $\mathbf{r}_n$  and  $\hat{\mathbf{r}}_n$  is greater than 20 cm,  $\mathbf{r}_n$  is defined with additional information provided by sampling  $\mathbf{v}_n$  through the measurement model. Therefore,  $\mathcal{D}$  is restricted to 20 cm, which is the maximum distortion that can be tolerated in the reconstruction. For instance, when the 3-D scans illustrated in Figure 3.1(a)–(d) are compressed using the proposed method, the resulting average distortions are about 15, 13, 14, and 11 cm, respectively. Their reconstructions are shown in Figure 3.1(e)–(h) for comparison with their originals. Moreover, the differences between these 3-D scans and their reconstructions are illustrated in Figure 4.3(a)–(d) to provide a visual comparison between the distortions obtained for these 3-D scans. According to the figures, the distortion becomes significant in those 2-D scans that are encoded with only  $\{\epsilon, \delta, \Delta\}$ , as indicated by the darker horizontal stripes in the images.

<b>3-D scan</b>	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	$k_{\text{SHIFT+CS}}$	$k_{\text{SHIFT+SC}}$	$k_{\text{SHIFT}}$	$k_{\text{NOCODING}}$
scan 01	8.1	12.1	11.7	10.4	52	95	315	10
scan 02	13.0	13.4	22.7	21.8	110	104	243	14
scan 03	9.7	13.0	17.6	15.9	73	104	283	11
scan 04	11.0	12.2	23.4	21.8	93	144	226	8
scan 05	9.3	13.0	16.4	15.8	71	114	276	11
scan 06	8.8	12.2	12.3	12.2	58	147	257	9
scan 07	11.6	15.2	21.0	19.6	102	46	311	13
scan 08	9.4	14.2	14.2	13.8	77	34	349	12
scan 09	7.5	14.3	10.1	10.1	54	27	379	11
scan 10	11.1	15.1	23.6	23.9	91	56	311	14
scan 11	14.2	11.6	21.2	20.3	107	136	211	17
scan 12	12.3	15.4	16.9	17.0	101	88	267	15
scan 13	10.5	14.0	10.7	10.2	72	94	290	15
scan 14	11.6	15.2	12.3	11.8	84	105	268	15
scan 15	10.3	13.8	10.7	10.2	73	126	260	12
scan 16	11.0	16.6	13.1	12.7	88	47	321	16
scan 17	9.3	12.4	8.8	8.4	57	131	270	12
scan 18	8.3	11.4	7.4	6.9	45	145	270	12
scan 19	7.3	14.0	8.5	8.2	56	47	360	9
scan 20	9.5	12.5	11.1	10.6	73	103	284	11
scan 21	6.5	9.9	5.6	5.1	34	158	274	5
scan 22	14.6	14.4	24.5	22.7	109	83	258	21
scan 23	11.4	14.2	19.7	17.3	77	100	276	19
scan 24	14.0	12.4	19.4	18.4	80	138	226	26
scan 25	8.5	11.4	13.4	12.2	54	143	264	10
scan 26	13.5	10.8	13.3	12.6	61	131	250	30
scan 27	15.9	8.8	21.2	20.5	101	234	118	18
scan 28	15.1	10.5	16.5	15.8	72	227	145	27
scan 29	13.7	10.0	15.1	13.9	67	187	192	26
<b>average:</b>	<b>10.9</b>	<b>12.9</b>	<b>15.3</b>	<b>14.5</b>	<b>76</b>	<b>113</b>	<b>267</b>	<b>15</b>

Table 4.6: CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ ,  $t_{\text{dec}}$ , number of cases when the signal is encoded with  $\{\epsilon, \delta, \Delta, \mathbf{m}\}$  using compressive sampling ( $k_{\text{SHIFT+CS}}$ ),  $\{\epsilon, \delta, \Delta, \mathbf{m}\}$  using simple coding ( $k_{\text{SHIFT+SC}}$ ), only  $\{\epsilon, \delta, \Delta\}$  ( $k_{\text{SHIFT}}$ ), and the number of cases when the signal is not encoded ( $k_{\text{NOCODING}}$ ), when the first data set is compressed using the proposed method.

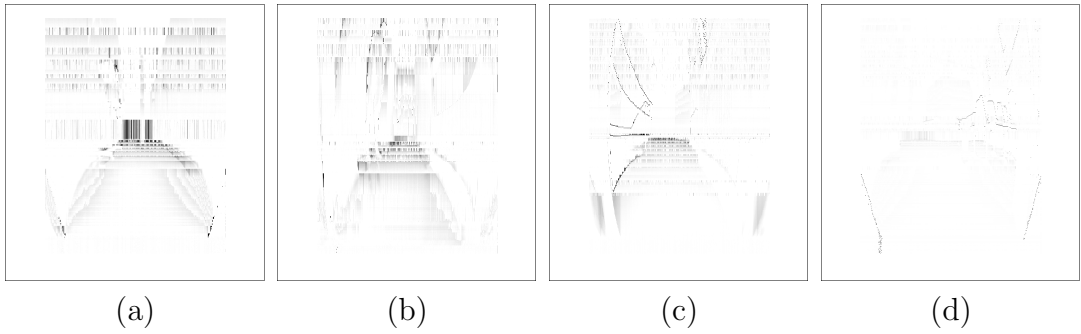


Figure 4.3: Distortion images for the 3-D scans given in Figure 3.1(a)–(d).

The average compression performances of the methods implemented so far are summarized in Table 4.7 for the first data set. When the proposed method is compared with the lossless methods considered here, the proposed method is 2–4 times faster than the variations of Huffman and arithmetic coding, but much slower than ZLIB and GZIP. However, the proposed method compresses the 3-D scans more than all of these methods. When the proposed method is compared with the lossy methods implemented here, the proposed method is much slower than all the methods. However, it results in the least distortion among all the lossy methods. For lossy compression, there always exists a trade-off between reducing the size of the input data, and minimizing the distortion on the reconstructions [30]. Consequently, being a lossy method, the proposed method provides a reasonably good compromise between the CR, accuracy of the reconstructions, and speed when its performance is compared with the performances of the well-known techniques considered in this thesis.

To evaluate the performance of the proposed method further, the second data set is compressed using all of the lossless and lossy methods considered so far. This set is comprised of 82 3-D scans that are taken sequentially at different locations depicted in Figure 4.4, while a robot platform equipped with SICK LMS200 is translated along a hall inside the Dagstuhl Castle in Germany [42]. Besides furniture, features such as buildings outside the hall are observed in this data set, as illustrated in Figure 4.5(a)–(d), because of the windows along the left and the right sides of the hall. Due to the details observed in these scans, it can



	<b>method</b>	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
lossless methods	Huffman coding on $\mathbf{r}_n$	41.7	0	165.6	610.6
	Huffman coding on $(\mathbf{r}_n - \mathbf{r}_{n-1})$	12.0	0	49.0	14.0
	arithmetic coding on $\mathbf{r}_n$	11.1	0	37.6	48.9
	arithmetic coding on $(\mathbf{r}_n - \mathbf{r}_{n-1})$	16.4	0	45.6	58.4
	ZLIB	65.3	0	0.4	0.2
	GZIP	76.7	0	0.5	0.3
lossy methods	JPEG	9.0	164.6	4.2	7.3
	1-level wavelet transform	50.2	21.3	0.3	0.3
	2-level wavelet transform	25.2	29.8	0.5	0.6
	3-level wavelet transform	12.7	37.3	0.8	0.9
	<b>proposed method</b>	<b>10.9</b>	<b>12.9</b>	<b>15.3</b>	<b>14.5</b>

Table 4.7: CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ , and  $t_{\text{dec}}$  when the first data set is compressed by different lossless and lossy methods.

be claimed that the similarity between consecutive 2-D scans in the second data set is less than in the first data set. To prove this, we assume that  $N$  discrete range measurements in a 2-D scan are individual stationary random variables. Let  $r_n[i]$  and  $r_{n+\tau}[i]$  be the  $i$ th range measurements in the 2-D scans  $\mathbf{r}_n$  and  $\mathbf{r}_{n+\tau}$  of a 3-D scan, respectively. The correlation coefficient between  $r_n[i]$  and  $r_{n+\tau}[i]$ , denoted by  $R_{r[i]}(\tau)$ , is a measure of similarity that takes values between  $-1$  and  $1$ .

$$R_{r[i]}(\tau) = \frac{\text{cov}(r_n[i], r_{n+\tau}[i])}{\text{std}(r_n[i]) \text{std}(r_{n+\tau}[i])} = \frac{E \{(r_n[i] - m_i)(r_{n+\tau}[i] - m_i)\}}{\sqrt{E \{(r_n[i] - m_i)^2\} E \{(r_{n+\tau}[i] - m_i)^2\}}} \quad (4.1)$$

where  $m_i$  is the average of  $i$ th measurements in the 2-D scans, and  $\text{cov}(\cdot, \cdot)$ ,  $\text{std}(\cdot)$  denote the covariance and the standard deviation, respectively. Consequently, the similarity between  $\mathbf{r}_n$  and  $\mathbf{r}_{n+\tau}$  is measured from the average of the correlation coefficients between  $r_n[i]$  and  $r_{n+\tau}[i]$  for  $i = 1, \dots, N$  that is denoted by  $\bar{R}_r(\tau)$ .  $\bar{R}_r(\tau)$  is determined from the biased estimates of  $R_{r[i]}(\tau)$  for  $i = 1, \dots, N$  as follows:

$$\bar{R}_r(\tau) = \frac{1}{N} \sum_{i=1}^N \frac{\frac{1}{N_0} \sum_{n=1}^{N_0-\tau} (r_n[i] - m_i)(r_{n+\tau}[i] - m_i)}{\sqrt{\frac{1}{N_0} \sum_{n=1}^{N_0-\tau} (r_n[i] - m_i)^2 \frac{1}{N_0} \sum_{n=1}^{N_0-\tau} (r_{n+\tau}[i] - m_i)^2}} \quad (4.2)$$

where  $N_0$  is the number of 2-D scans comprising a 3-D scan and  $N$  is the number of range measurements in a 2-D scan.  $\bar{R}_r(\tau)$  for the first and second data sets



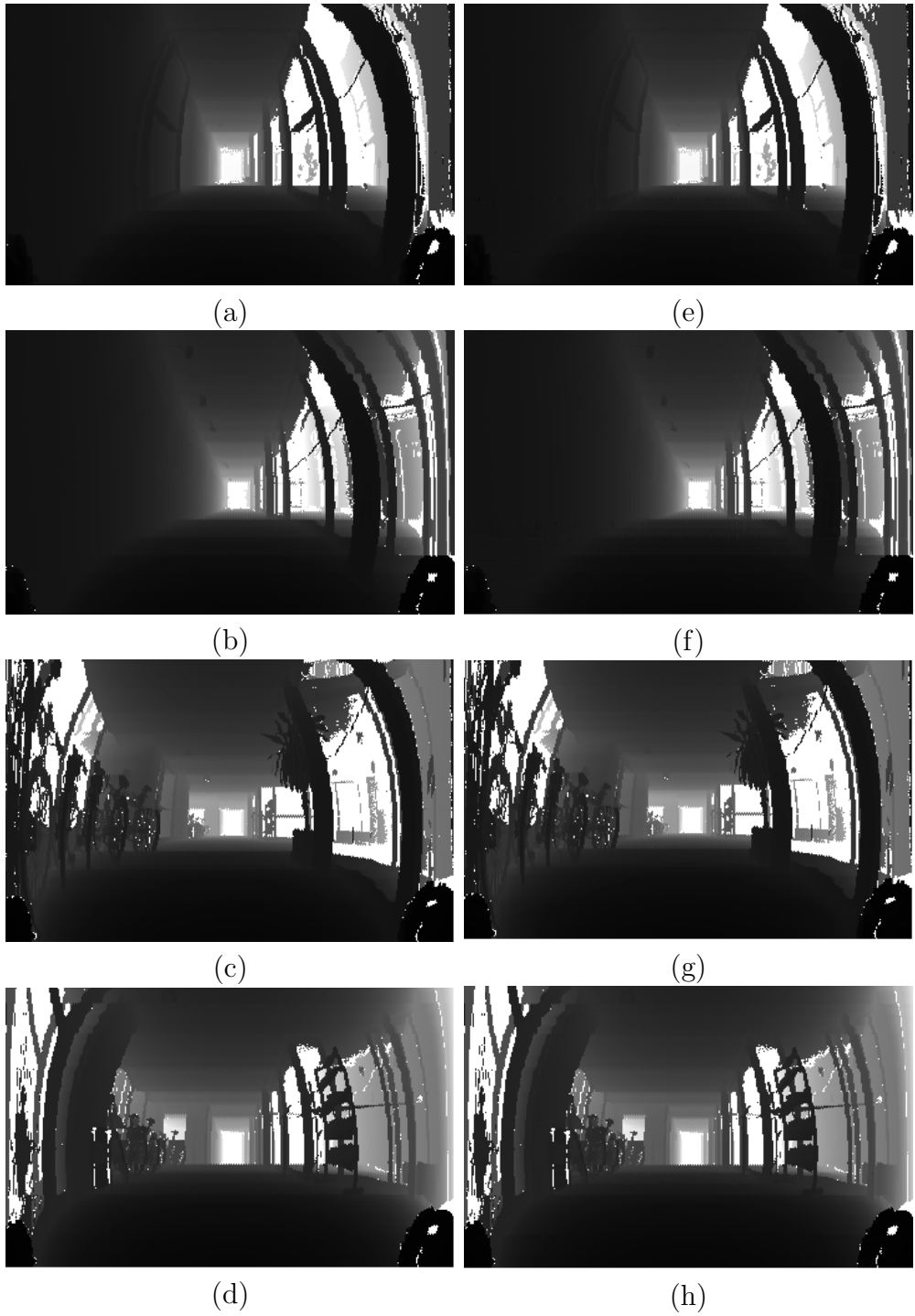


Figure 4.5: (a)–(d): Sample data sets collected at Dagstuhl Castle and (e)–(h): their reconstructions.

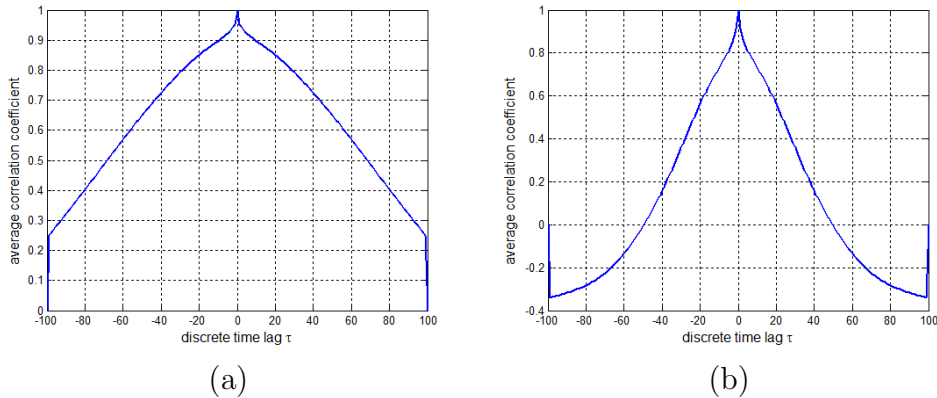


Figure 4.6: The average correlation coefficients between the 2-D scans in the (a) first and (b) second data sets, respectively.

a fast and effective performance with an error of about 5 cm and an encoding time of almost 2 seconds. In conclusion, it is again demonstrated with this experimental data that the proposed method provides satisfactory compression with acceptable distortion and speed. The comparison between the average percentages of  $k_{\text{SHIFT}+\text{CS}}$ ,  $k_{\text{SHIFT}+\text{SC}}$ ,  $k_{\text{SHIFT}}$ , and  $k_{\text{NOCODING}}$ , when both data sets are compressed using the proposed method, is provided in Table 4.9. The table demonstrates how the similarities between the 2-D scans in a 3-D scan affects the compression performance of the proposed method, where the size of the 3-D scans in the first and the second data sets are reduced by 89% and 68%, on the average, respectively.

Finally, the compression performance of the proposed method is observed under the presence of additive white Gaussian noise. In this part, the 3-D scans in the first data set are compressed after zero mean white Gaussian noise is added to them. For different noise levels, each 3-D scan is compressed ten times using the proposed method, then the average SNR, CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ ,  $t_{\text{dec}}$ ,  $k_{\text{SHIFT}+\text{CS}}$ ,  $k_{\text{SHIFT}+\text{SC}}$ ,  $k_{\text{SHIFT}}$ , and  $k_{\text{NOCODING}}$  values are obtained, as given in Table 4.10. Here, SNR is the ratio of the power of the 3-D scan to the power of the noise. According to the table, it is observed that the proposed method maintains its performance with CR around 10% and  $\mathcal{D}$  about 13 cm, under the presence of noise with standard deviation up to 10 cm. Notice that, the sparsifying model

	<b>method</b>	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)
lossless methods	Huffman coding on $\mathbf{r}_n$	683.3	0	101.8	363.1
	Huffman coding on $(\mathbf{r}_n - \mathbf{r}_{n-1})$	253.6	0	12.6	34.2
	arithmetic coding on $\mathbf{r}_n$	27.1	0	21.8	25.7
	arithmetic coding on $(\mathbf{r}_n - \mathbf{r}_{n-1})$	35.8	0	16.4	19.1
	ZLIB	140.8	0	0.3	0.1
	GZIP	143.8	0	0.3	0.2
lossy methods	JPEG	10.0	743.6	1.5	3.6
	1-level wavelet transform	50.2	204.4	0.2	0.2
	2-level wavelet transform	25.2	283.0	0.4	0.3
	3-level wavelet transform	12.7	353.8	0.5	0.5
	<b>proposed method</b>	<b>32.0</b>	<b>4.8</b>	<b>1.9</b>	<b>1.7</b>

Table 4.8: CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ , and  $t_{\text{dec}}$  when the second data set is compressed with different lossless and lossy methods.

	$k_{\text{SHIFT+CS}}$	$k_{\text{SHIFT+SC}}$	$k_{\text{SHIFT}}$	$k_{\text{NOCODING}}$
first data set	16.0%	24.0%	57.0%	3.0%
second data set	4.9%	17.3%	13.8%	64.0%

Table 4.9: The average percentages of  $k_{\text{SHIFT+CS}}$ ,  $k_{\text{SHIFT+SC}}$ ,  $k_{\text{SHIFT}}$ , and  $k_{\text{NOCODING}}$ , when both data sets are compressed using the proposed method.

also performs properly, when the standard deviation of the noise is below the same level. The method provides acceptable compression when the standard deviation remains below 30 cm. Beyond that level, the method cannot perform effective compression. In other words, the method can operate with SNR larger than 23 dB, and works properly with SNR larger than 30 dB.

In this study, it is also investigated whether the compression performance will improve when 2-D scans are encoded with either SHIFT+CS, SHIFT+SC, or NOCODING. In this case, we cannot benefit from encoding a 2-D scan with only three values ( $\epsilon$ ,  $\delta$  and  $\Delta$ ). On the other hand, the distortion will fall, since it becomes significant along the 2-D scans that are encoded with  $\{\epsilon, \delta, \Delta\}$ . For the first data set, the CR increases from about 11% to 16%, and  $\mathcal{D}$  remains almost unchanged on the average. For the second data set, the CR increases from about 32% to 32.6%, and  $\mathcal{D}$  decreases from 4.8 cm to 3.5 cm. As a consequence, the compression performance does not improve significantly.

added noise	SNR (dB)	CR (%)	$\mathcal{D}$ (cm)	$t_{\text{enc}}$ (s)	$t_{\text{dec}}$ (s)	$k_{\text{SHIFT+CS}}$	$k_{\text{SHIFT+SC}}$	$k_{\text{SHIFT}}$	$k_{\text{NOCODING}}$
none	–	10.9	12.9	15.3	14.5	76	113	267	15
$\mathcal{N}(0, 1)$	52.7	10.8	12.9	12.1	10.9	74	115	268	15
$\mathcal{N}(0, 4)$	46.6	10.8	12.9	14.6	13.5	73	115	268	14
$\mathcal{N}(0, 9)$	43.1	10.5	12.8	11.3	10.2	70	117	270	14
$\mathcal{N}(0, 10)$	42.7	10.6	12.7	11.3	10.2	70	118	269	14
$\mathcal{N}(0, 16)$	40.6	10.3	12.6	11.1	9.9	66	120	271	14
$\mathcal{N}(0, 25)$	38.7	10.2	12.6	10.5	9.4	64	121	272	14
$\mathcal{N}(0, 100)$	32.7	11.2	12.5	10.2	9.1	62	117	276	16
$\mathcal{N}(0, 400)$	26.6	20.0	13.4	23.4	22.3	147	0	286	38
$\mathcal{N}(0, 900)$	23.1	27.5	13.5	14.2	13.3	88	0	287	96
$\mathcal{N}(0, 2500)$	18.7	88.1	6.7	2.5	0.4	0	0	56	415
$\mathcal{N}(0, 10000)$	12.7	100.0	0.0	2.9	0.4	0	0	0	471

Table 4.10: Average signal-to-noise ratio (SNR), CR,  $\mathcal{D}$ ,  $t_{\text{enc}}$ ,  $t_{\text{dec}}$ , number of cases when the signal is encoded with  $\{\epsilon, \delta, \Delta, \mathbf{m}\}$  using compressive sampling ( $k_{\text{SHIFT+CS}}$ ),  $\{\epsilon, \delta, \Delta, \mathbf{m}\}$  using simple coding ( $k_{\text{SHIFT+SC}}$ ), only  $\{\epsilon, \delta, \Delta\}$  ( $k_{\text{SHIFT}}$ ), and the number of cases when the signal is not encoded ( $k_{\text{NOCODING}}$ ), when the first data set is compressed under the presence of additive white Gaussian noise indicated with its mean and variance.

Another modification we have considered in the algorithm is combining  $\epsilon$  and  $\Delta$  into one vertical shift value in the encoder. Thus,  $\mathbf{r}_n$  can now be represented with only two parameters that are the vertical and the horizontal shift values. In this case, the CR is reduced by 0.3% and 0.1% on the average for the first and the second data sets, respectively.

Furthermore, we also analyze whether  $\epsilon$  and  $\delta$  can be encoded using *difference encoding* to reduce the CR even further. Therefore, the correlations between consecutive values of  $\epsilon$  and  $\delta$ , generated while encoding a 3-D scan, are observed. Although consecutive values of  $\delta$  are found to be somewhat correlated with each other, consecutive values of  $\epsilon$  are not. Consequently, it will not much reduce the CR, when  $\epsilon$  and  $\delta$  are further encoded using difference encoding.

## Chapter 5

# Conclusions and Future Work

In this thesis, we consider a task that can be vital for 3-D representation of indoor environments, using the SICK LMS200 laser range finder. The task involves compressing the 2-D range scans forming the 3-D model simultaneously with the acquisition of the scans, to be able to use the capacity of the communication channel or the data storage medium effectively during transmission or storage of the scan data. From this perspective, we propose a compression technique based on compressive sensing for correlated 2-D scans acquired sequentially.

In general, the proposed technique involves sampling the sparse signal efficiently. Each sparse signal is obtained from the difference between the current scan and its estimate, which is generated by shifting the previous scan along the horizontal and vertical axes by certain amounts. The amount of displacements along these axes are formulized with respect to the current and previous scans. Then, the amplitude of the difference signal is offset to improve the sparsity. Compression is achieved by sampling the sparse signal using either simple coding or compressive sampling. Therefore, the proposed method is a kind of difference encoding technique.

The compression performance of the proposed method relies on the similarity between consecutive 2-D scans in the input data. The higher the correlation

between consecutive 2-D scans and the less the detail in the scanned environment, the lower the CR. For instance, the proposed method compresses a 3-D scan in the first data set which includes approximately 170 thousand range measurements (about 1.36 MB data<sup>1</sup>), within 15 seconds and by about 89% on the average, with the average distortion per measurement being about 13 cm. Moreover, the proposed method maintains this performance under the presence of zero mean white Gaussian noise added to the scan data when the SNR is larger than 30 dB. However, for the second data set where the similarity is somewhat less than in the first one, the proposed method compresses a 3-D scan from this set which includes approximately 81 thousand range measurements (about 650 KB data), within two seconds and by about 68% on the average, with the average distortion per measurement being about 5 cm. The amount of distortion seems to be reasonable with the provided compression ratio, since the measurements acquired from the laser range finder have already  $\pm 4$  cm systematic error, besides the statistical error. Therefore, the proposed method is fast and efficient according to the criteria described in Chapter 1. The proposed method is recommended for applications where both the CR and speed are crucial. However, a lossless compression technique, such as arithmetic coding, can be used in applications where the accuracy of the range measurements is more important.

In summary, the proposed method provides acceptable CR compared with the alternative compression techniques that we have considered, and as it provides a reasonably good compromise between reconstruction accuracy and speed, it can be properly used for 3-D representation of environments [57]. The proposed method can be improved by coding the encoder output further via lossless coding techniques such as arithmetic coding or LZ77 [50], to reduce the CR even more. Moreover, our future work involves hardware implementation of the proposed method on FPGAs so that the method can be used in real-time applications.

---

<sup>1</sup>Each range measurement is stored within 8 byte variables in MATLAB environment.



We will also extend its application to other types of data sets comprised of measurement sequences, such as video data.

# Appendix A

## Well-Known Dictionaries for Forming a Sparsifying Basis

This section includes brief descriptions of some of the well-known and widely used dictionaries employed in forming the sparsifying basis. The dictionaries reviewed in this section are the Dirac, Fourier, Gabor, and Haar dictionaries.

A fundamental dictionary in the time domain is the *Dirac* dictionary, which is parameterized by  $\gamma \in \{0, 1, \dots, L-1\}$ , where  $L$  is the number of atoms. Then, the atoms in the Dirac dictionary are defined as  $\psi_\gamma(t) = \mathbf{1}(t - \gamma)$ , where  $t$  is the continuous or discrete-time variable and  $\mathbf{1}(t - \gamma)$  is the unit impulse sequence in discrete time or the Dirac delta function in continuous time. This function is zero except when  $t = \gamma$  [36]. A linear combination of atoms may also be referred to as atoms for a modified dictionary. It is remarkable that apart from all atoms in the Dirac dictionary being orthogonal to each other, all linear combinations of these atoms are also orthogonal. Therefore, it is trivial to obtain an orthonormal basis from these linear combinations.

A fundamental dictionary in the frequency domain is the *Fourier* dictionary, which is parameterized by  $\gamma = \{\omega, v\}$ , where  $\omega$  indicates the frequency within

$[0, 2\pi)$ , and  $v$  indicates the type of the waveform ( $v = 0$  indicates a cosine, whereas  $v = 1$  indicates a sine waveform). The atoms in the Fourier dictionary are expressed with either  $\psi_{(\omega,0)} = \cos(\omega t)$  or  $\psi_{(\omega,1)} = \sin(\omega t)$  for a specific value of  $v$ . For  $L$  waveforms in the dictionary, the frequency sequence parameterizing discrete waveforms is defined as  $\omega = \frac{l\pi}{L}$ , where  $l = \{\frac{1}{2}, \frac{3}{2}, \dots, L - \frac{1}{2}\}$ . The atoms are orthogonal to each other. The Fourier dictionary is efficient and effective in representing *smooth* signals, but its efficiency fails in representing signals with *singularities* [36].

*Gabor* dictionaries can represent the signal in both the time and the frequency domains. A Gabor dictionary is parameterized by  $\gamma = \{\omega, t_0, \theta, \sigma_E\}$ , where  $\omega$  indicates the frequency within  $[0, \pi)$ ,  $t_0$  indicates the location of the center of the waveform on the time axis,  $\theta$  indicates the phase, and  $\sigma_E$  indicates the standard deviation of the Gaussian envelope of the waveform. Atoms in Gabor dictionaries are expressed as  $\psi_\gamma(t) = e^{\frac{-(t-t_0)^2}{\sigma_E^2}} \cos[\omega(t - t_0) + \theta]$ , where  $t$  is again the continuous or the discrete-time variable.

A popular dictionary that scales the signal in the time domain is the *Haar* dictionary. This dictionary is also known as the Haar filter bank that is commonly employed with the wavelet transform, since it is easy and practical to implement. Haar dictionary is composed of waveforms, represented by  $\psi(\frac{t-b}{a})$  and referred as wavelets. In the definition of wavelets, the dilation parameter  $a$  specifies the frequency of the wavelet, where large values indicate low frequencies, and small values indicate high frequencies. Furthermore, the translation parameter  $b$  specifies the location of the wavelet on the time axis. Therefore, these parameters characterize time-frequency localization of wavelets. A basic Haar wavelet with  $a = 1$  and  $b = 0$  is the waveform such that  $\psi_{\{1,0\}}(t) = \{1 \text{ for } t \in [0, \frac{1}{2}); -1 \text{ for } t \in [\frac{1}{2}, 1); 0 \text{ elsewhere}\}$ . In discrete time,  $a$  and  $b$  take discrete values, such that  $a = a_0$  for fixed  $a_0 \geq 1$ , besides  $b = lb_0a_0$  for fixed  $b_0 > 0$ , where  $l = \{0, 1, \dots, L - 1\}$  for  $L$  atoms in the dictionary. The

possible values of  $b$  depend on the values used in  $a$ , so that narrow wavelets, with high frequencies must be translated in small steps on the time axis to cover the whole time domain. In contrast, wider wavelets, with low frequencies are translated in large steps for the same reason. It is remarkable that translated wavelets with constant rate are orthogonal to each other. To obtain orthonormal wavelets, a wavelet function can be defined with a scalar factor such as  $|a|^{-\frac{1}{2}}\psi(\frac{t-b}{a})$  [58].

In any wavelet dictionary, like the Haar dictionary, wavelets are formed as linear combinations of translated mother wavelets at fixed dilation. Like wavelets, mother wavelets are characterized by dilation and translation parameters. In the Haar dictionary, the basic mother wavelet with one dilation, and zero translation is expressed as  $\psi(t)_{\{1,0\}} = \{1 \text{ for } t \in [0, 1); 0 \text{ elsewhere}\}$  [58]. Therefore, one can propose that signals can be represented by using mother wavelets instead of wavelets; indeed it is the representation of the signal at a certain resolution. Let  $W_a$  and  $V_a$  denote two spaces spanned by wavelets and mother wavelets, respectively, at fixed dilation,  $a$ . These spaces intersect only at the origin, and they form the upper space  $V_{a+1}$ . Based on this information, it can be stated that a signal can be represented on a basis composed of either wavelets or mother wavelets. These representations are the projections of the signal onto spaces  $W_a$  and  $V_a$ , respectively. Thus, the projection of the signal cannot include all of the information on the signal; part of the information can be projected onto  $V_a$ , and the remaining part onto  $W_a$ . As a consequence, the reconstruction of the signal from projections will be close to, but not exactly equal to the signal. In the upper space,  $V_{a+1}$ , the signal is represented with higher resolution, which results in better reconstruction than the one from the lower space. Although representing the signal with higher resolution is better than in lower resolution, it should be noted that the number of translated wavelets increases exponentially, which adds computational cost and increases the dimension of the representation.

# Appendix B

## Methods for Generating Sparse Innovations

This section includes the implementation details of the methods referred as **(a)**–**(f)** in Chapter 3. In the following, the models and parameters used in the methods are described.

For the following methods, let  $\{\mathbf{r}_n, \mathbf{r}_{n-1}, \mathbf{r}_{n-2}, \mathbf{r}_{n-3}, \dots\}$  be the sequentially acquired 2-D scans in a data set. Each scan is considered to be a sequence of range measurements denoted by  $\mathbf{r}_n = \{r_n[1], r_n[2], \dots, r_n[N]\}$ ,  $\mathbf{r}_{n-1} = \{r_{n-1}[1], r_{n-1}[2], \dots, r_{n-1}[N]\}$ ,  $\mathbf{r}_{n-2} = \{r_{n-2}[1], r_{n-2}[2], \dots, r_{n-2}[N]\}$ ,  $\mathbf{r}_{n-3} = \{r_{n-3}[1], r_{n-3}[2], \dots, r_{n-3}[N]\}$ ,  $\dots$ , etc. Furthermore, let  $\mathbf{S}$  be the matrix where the 2-D scans acquired until  $\mathbf{r}_n$  are accumulated as row vectors, such that:

$$\mathbf{S} = \begin{bmatrix} r_{n-1}[1] & r_{n-1}[2] & \dots & r_{n-1}[N] \\ r_{n-2}[1] & r_{n-2}[2] & \dots & r_{n-2}[N] \\ r_{n-3}[1] & r_{n-3}[2] & \dots & r_{n-3}[N] \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (\text{B.1})$$

### (a) Differences between Consecutive Scans

When  $n > 1$ , the differences between two consecutive scans are defined as  $\mathbf{d}_n = \mathbf{r}_n - \mathbf{r}_{n-1}$ . When  $n = 1$ ,  $\mathbf{d}_1 = \mathbf{r}_1$ . Thus,  $\mathbf{r}_n$  is represented by  $\mathbf{d}_n$ .

### (b) Linear Regression Using the Last Two Scans

In this method,  $r_n[i]$ , where  $i = 1, 2, \dots, N$ , is predicted by the following procedure: At first, the  $i$ th column of  $\mathbf{S}$  is differentiated to determine the critical points in the corresponding slice of the scan. The critical points appear to be spikes in the differential signal. The spikes indicate the separations between the linear surfaces on the corresponding slice of the scan. If  $r_{n-1}[i]$  and  $r_{n-2}[i]$  are not separated by a critical point,  $r_n[i]$  is assumed to be on the linear surface formed by  $r_{n-1}[i]$  and  $r_{n-2}[i]$ . Therefore, it is estimated to be the point on the line that satisfies  $r_{n-1}[i] = p_1(n-1) + p_0$  and  $r_{n-2}[i] = p_1(n-2) + p_0$ , where  $p_0$  and  $p_1$  are line parameters. These equations can be expressed in compact form, such that:

$$\begin{bmatrix} r_{n-1}[i] \\ r_{n-2}[i] \end{bmatrix} = \begin{bmatrix} 1 & n-1 \\ 1 & n-2 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} \quad (\text{B.2})$$

which is denoted by  $\mathbf{u} = \mathbf{A} \mathbf{p}$ . The parameters placed in  $\mathbf{p}$  are computed from  $\mathbf{p} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{u}$ . Eventually,  $r_n[i]$  is estimated to be  $p_1 n + p_0$ . If  $r_{n-1}[i]$  and  $r_{n-2}[i]$  are separated by a critical point,  $r_n[i]$  is estimated to be  $r_{n-1}[i]$ . Consequently,  $\mathbf{r}_n$  is represented by the difference between itself and its estimate.

### (c) Second-order Polynomial Fitting Using the Last Three Scans

In this method,  $r_n[i]$ , where  $i = 1, 2, \dots, N$ , is predicted by the following procedure: As in (b), the  $i$ th column of  $\mathbf{S}$  is differentiated to determine the critical

points in the corresponding slice of the scan. The critical points appear to be spikes in the differential signal. The spikes indicate the separations between the continuous surfaces on the corresponding slice of the scan. If  $r_{n-1}[i]$ ,  $r_{n-2}[i]$ , and  $r_{n-3}[i]$  are not separated by a critical point,  $r_n[i]$  is assumed to be on the polynomial interpolated by  $r_{n-1}[i]$ ,  $r_{n-2}[i]$ , and  $r_{n-3}[i]$ . Therefore, it is estimated to be the point on the polynomial that satisfies  $r_{n-1}[i] = q_0 + q_1(n-1) + q_2(n-1)^2$ ,  $r_{n-2}[i] = q_0 + q_1(n-2) + q_2(n-2)^2$ , and  $r_{n-3}[i] = q_0 + q_1(n-3) + q_2(n-3)^2$ , where  $q_0$ ,  $q_1$ , and  $q_2$  are polynomial coefficients. These equations can be expressed in compact form as:

$$\begin{bmatrix} r_{n-1}[i] \\ r_{n-2}[i] \\ r_{n-3}[i] \end{bmatrix} = \begin{bmatrix} 1 & n-1 & (n-1)^2 \\ 1 & n-2 & (n-2)^2 \\ 1 & n-3 & (n-3)^2 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix} \quad (\text{B.3})$$

which is denoted by  $\mathbf{u} = \mathbf{A} \mathbf{q}$ . The coefficients placed in  $\mathbf{q}$  are computed from  $\mathbf{q} = (\mathbf{A}^T \mathbf{A} + \mathbf{I}_3)^{-1} \mathbf{A}^T \mathbf{u}$ , where  $\mathbf{I}_3$  is a  $3 \times 3$  identity matrix. A small term is added to the diagonal coefficients of  $\mathbf{A}^T \mathbf{A}$  because  $\mathbf{A}$  is obtained to be close to singular matrix.<sup>1</sup> Eventually,  $r_n[i]$  is estimated to be  $q_0 + q_1 n + q_2 n^2$ . If  $r_{n-1}[i]$ ,  $r_{n-2}[i]$ , and  $r_{n-3}[i]$  are separated by a critical point,  $r_n[i]$  is estimated to be  $r_{n-1}[i]$ . Consequently,  $\mathbf{r}_n$  is represented by the difference between itself and its estimate.

#### (d) Second-order Wiener Filtering on Differential Scans

In this method, let  $\{\mathbf{d}_1, \dots, \mathbf{d}_{n-1}\}$  be available for the scan  $\mathbf{r}_n$ . We assume that every difference sequence is stationary and is comprised of  $N$  zero mean random variables, such that  $\mathbf{d}_{n_1} = \{d_{n_1}[1], \dots, d_{n_1}[N]\}$  and  $\mathbf{d}_{n_2} = \{d_{n_2}[1], \dots, d_{n_2}[N]\}$ , where  $1 \leq \{n_1, n_2\} \leq n-1$ . Then, the biased autocorrelation between  $d_{n_1}[i]$  and

---

<sup>1</sup>See A. Tikhonov, *On the stability of inverse problems* (1943) for a detailed explanation.

$d_{n_2}[i]$  for  $i = 1, \dots, N$  is estimated as follows:

$$R_{d[i]}(n_1 - n_2) = R_{d[i]}(\tau) = \frac{1}{N_0} \sum_{m=1}^{N_0-\tau} d_m[i] d_{m+\tau}[i] \quad (\text{B.4})$$

where  $N_0$  is the number of observations of the differences, and  $\tau \triangleq n_1 - n_2$ . Consequently, the estimate of  $\mathbf{d}_n$ , denoted by  $\hat{\mathbf{d}}_n$ , is obtained by predicting  $d_n[i]$  for  $i = 1, \dots, N$  over  $d_{n-1}[i]$  and  $d_{n-2}[i]$  using a second-order Wiener filter. Each pair of filter coefficients  $\{f_1^i, f_2^i\}$  used to estimate  $d_n[i]$  for  $i = 1, \dots, N$  is derived from the following equation:

$$\begin{bmatrix} R_{d[i]}[1] \\ R_{d[i]}[2] \end{bmatrix} = \begin{bmatrix} R_{d[i]}[0] & R_{d[i]}[1] \\ R_{d[i]}[1] & R_{d[i]}[0] \end{bmatrix} \begin{bmatrix} f_1^i \\ f_2^i \end{bmatrix} \quad (\text{B.5})$$

which is denoted by  $\mathbf{u} = \mathbf{A} \mathbf{f}^i$ . Since  $\mathbf{A}$  is a positive-definite matrix,  $\mathbf{f}^i = \mathbf{A}^{-1} \mathbf{u}$ . Eventually,  $\mathbf{r}_n$  is predicted to be  $\mathbf{r}_{n-1} + \hat{\mathbf{d}}_n$ . Consequently,  $\mathbf{r}_n$  is represented by the difference between itself and its prediction.

### (e) 1-D Random Walk on the Previous Difference

In this method,  $\mathbf{d}_n$  is estimated as

$$\hat{\mathbf{d}}_n = \alpha \mathbf{d}_{n-1} + \mathbf{w}_n \quad (\text{B.6})$$

where  $\alpha$  is the correlation coefficient between  $\mathbf{d}_n$  and  $\mathbf{d}_{n-1}$ , and  $\mathbf{w}_n$  is white Gaussian noise vector with zero mean and an identity covariance matrix. If we multiply both sides of Equation (B.6) by  $\mathbf{d}_{n-1}^T$ , and take the expected value,  $\alpha$  is determined to be  $\frac{E\{\mathbf{d}_{n-1}^T \mathbf{d}_n\}}{E\{\mathbf{d}_{n-1}^T \mathbf{d}_{n-1}\}}$ , where  $E\{\cdot\}$  is the expectation operator. Using all the differential signals determined from all of the 3-D scans in the first data set,  $\alpha$  is estimated to be  $-0.4$ . After  $\mathbf{d}_n$  is estimated,  $\mathbf{r}_n$  is predicted to be  $\mathbf{r}_{n-1} + \hat{\mathbf{d}}_n$ . Consequently,  $\mathbf{r}_n$  is represented by the difference between itself and its prediction.



## (f) Linear Kalman Filtering with the Constant Velocity Kinematic State Model

In this method, we estimate  $\mathbf{r}_n$  denoted by  $\hat{\mathbf{r}}_n$ , using a linear Kalman filter with the constant velocity kinematic state model. In this model, the 2-D scans are considered to be the position state in  $N$ -dimensional space at discrete instants, and the position is assumed to change in time without acceleration.  $[\mathbf{r}_m, \mathbf{d}_m]^T$  is the state vector for  $m = \{n, n-1, n-2, \dots\}$ , where  $\mathbf{d}_m$  indicates the velocity. The slight changes in velocity are modeled as state estimate error  $e_n$  that is white Gaussian noise with zero mean and unit variance. Consequently,  $\mathbf{r}_n$  is estimated as  $\hat{\mathbf{r}}_n$  in the state estimate model that is expressed as

$$\begin{bmatrix} \hat{\mathbf{r}}_n \\ \mathbf{d}_n \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{0}_N & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{r}_{n-1} \\ \mathbf{d}_{n-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\mathbf{I}_N \\ \mathbf{I}_N \end{bmatrix} e_n \quad (\text{B.7})$$

where  $\mathbf{I}_N$  and  $\mathbf{0}_N$  are  $N \times N$  identity matrix and zero matrix, respectively.

The measurement vector at instant  $n-1$  is  $\mathbf{r}_{n-1}$  having measurement noise  $w_{n-1}$  that is white Gaussian noise with zero mean and standard deviation 10 times that of the state estimate error. The measurement model can be expressed as:

$$\begin{bmatrix} \mathbf{r}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N \end{bmatrix} \begin{bmatrix} \mathbf{r}_{n-1} \\ \mathbf{d}_{n-1} \end{bmatrix} + \begin{bmatrix} \mathbf{I}_N \end{bmatrix} w_{n-1} \quad (\text{B.8})$$

The ratio of the standard deviation of the measurement error to the standard deviation of the state estimate error is determined such that the error in predicting  $\mathbf{r}_n$  is minimized. Eventually,  $\mathbf{r}_n$  is represented by the innovation that is the difference between itself and its estimate,  $\mathbf{r}_n - \hat{\mathbf{r}}_n$ .

In the Kalman filter structure, both the state estimate and measurement errors are supposed to be zero mean white Gaussian noises that are uncorrelated with each other. In this case, the Kalman filter becomes the optimal *minimum mean square error state estimator*, otherwise it is the best *linear state estimator* [59]. However, the noise in 3-D laser range measurements are declared to

be correlated at adjacent scans, and not quite Gaussian, since the noise in the range measurements includes the noises generated by the optical components of the sensor, for instance the lens, the mirror driver, and the laser, besides the electronic noise [60]. Despite this fact, we model the noise as zero mean white Gaussian to satisfy the consistency of the state estimation [44], which requires zero mean and white innovation, and zero mean state estimate error.

# Bibliography

- [1] SICK AG, “Technical description. LMS200/211/221/291 laser measurement systems,” 2006. <https://mysick.com/saqqara/pdf.aspx?id=im0012759>.
- [2] R. G. Baraniuk, “Compressive sensing,” *IEEE Signal Processing Magazine*, vol. 24, p. 118, July 2007.
- [3] C. Brenneke, O. Wulf, B. Wagner, “Using 3-D laser range data for SLAM in outdoor environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1–4, (Las Vegas, Nevada, U.S.A.), pp. 188–193, IEEE Robotics and Automation Society, October 2003.
- [4] Y. Liu, R. Emery, D. Chakbarti, W. Burgard, S. Thrun, “Using EM to learn 3-D models of indoor environments with mobile robots,” in *Proceedings of International Conference on Machine Learning*, (Williamstown, Massachusetts, U.S.A.), The International Machine Learning Society, June-July 2001.
- [5] H. Surmann, A. Nüchter, J. Hertzberg, “An autonomous mobile robot with a 3-D laser range finder for 3-D exploration and digitalization of indoor environments,” *Robotics and Autonomous Systems*, vol. 45, pp. 181–198, December 2003.

- [6] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, J. Hertzberg, “Globally consistent 3-D mapping with scan matching,” *Robotics and Autonomous Systems*, vol. 56, pp. 130–142, July 2007.
- [7] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, H. Surmann, “3-D mapping with semantic knowledge,” *Lecture Notes in Artificial Intelligence*, vol. 4020, pp. 335–346, June 2006.
- [8] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, “The digital Michelangelo project: 3-D scanning of large statues,” in *Computer Graphics Annual Conference*, (New Orleans, Louisiana, U.S.A.), pp. 131–144, ACM SIGGRAPH, July 2000.
- [9] F. Bernardini, H. Rushmeier, I. M. Martin, J. Mittleman, G. Taubin, “Building a digital model of Michelangelo’s Florentine Pieta,” *IEEE Computer Graphics and Applications*, vol. 22, pp. 59–67, January-February 2002.
- [10] C. Chevrier, J. P. Perrin, “Interactive 3-D reconstruction for urban areas—An image based tool,” in *Proceedings of Computer Aided Architectural Design Futures*, (Eindhoven, Netherlands), pp. 753–765, CAAD Futures, July 2001.
- [11] G. Guidi, M. Pieraccini, S. Ciofi, V. Damato, J. A. Beraldin, C. Atzeni, C. Atz, “Tridimensional digitizing of Donatello’s Maddalena,” in *Proceedings of IEEE International Conference on Image Processing*, vol. 1, (Thessaloniki, Greece), pp. 578–581, IEEE Signal Processing Society, October 2001.
- [12] G. S. Spagnolo, R. Majo, M. Carli, A. Neri, “3-D scanner and virtual gallery of small cultural heritage objects,” in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 5302, (San Jose, California,

- U.S.A.), pp. 148–155, Society of Photo-Optical Instrumentation Engineers, January 2004.
- [13] B. Ergun, “Creating a 3-D model by terrestrial laser scanners and photogrammetry techniques: A case study on the historical peninsula of Istanbul,” in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 6618, (Munich, Germany), p. 6180, Society of Photo-Optical Instrumentation Engineers, June 2007.
- [14] Y. Arayici, “An approach for real world data modelling with the 3-D terrestrial laser scanner for built environment,” *Automation in Construction*, vol. 16, pp. 816–829, September 2007.
- [15] P. J. Watt, D. N. M. Donoghue, “Measuring forest structure with terrestrial laser scanning,” *International Journal of Remote Sensing*, vol. 26, pp. 1437–1446, April 2005.
- [16] SICK AG, “SICK Partner Portal,” March 2010. <http://www.mysick.com/>.
- [17] L. P. Ellekilde, S. Huang, J. V. Miro, G. Dissanayake, “Dense 3-D map construction for indoor search and rescue,” *Journal of Field Robotics*, vol. 24, pp. 71–89, February 2007.
- [18] W. Zhou, J. V. Miro, G. Dissanayake, “Information-efficient 3-D visual SLAM for unstructured domains,” *IEEE Transactions on Robotics*, vol. 24, pp. 1078–1087, October 2008.
- [19] K. Pervözl, A. Nüchter, H. Surmann, J. Hertzberg, “Automatic reconstruction of colored 3-D models,” in *Proceedings of Robotik 2004*, pp. 215–222, 2004.
- [20] P. J. Besl, N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, February 1992.

- [21] D. Hähnel, W. Burgard, “Probabilistic matching for 3-D scan registration,” in *Proceedings of the VDI-Conference Robotik*, vol. 1679, (Ludwigsburg, Germany), pp. 137–143, VDI Gesell, June 2002.
- [22] H. Zhou, Y. Liu, “3-D modeling from multi-view registered range images using K-means clustering,” in *IEEE International Conference on Industrial Technology*, vol. 1–6, (Bombay, India), pp. 977–982, IEEE, December 2006.
- [23] P. Biber, S. Fleck, F. Busch, M. Wand, T. Duckett, W. Strasser, “3-D modeling of indoor environments by a mobile platform with a laser scanner and panoramic camera,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Sendai, Japan), IEEE Robotics and Automation Society, September 2004.
- [24] G. Vosselman, S. Dijkman, “3-D building model reconstruction from point clouds and ground plans,” *International Archives of Photogrammetry and Remote Sensing*, vol. 34, pp. 37–43, October 2001.
- [25] S. Filin, “Surface clustering from airborne laser scanning data,” *Computers and Geosciences*, vol. 30, pp. 1033–1041, November-December 2004.
- [26] M. Johnson-Roberson, O. Pizarro, S. B. Williams, I. J. Mahon, “Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys,” *Journal of Field Robotics*, vol. 27, pp. 21–51, January 2010.
- [27] N. Kimura, S. Latifi, “A survey on data compression in wireless sensor networks,” in *Proceedings of IEEE International Conference on Information Technology: Coding and Computing*, vol. 2, (Las Vegas, Nevada, U.S.A.), pp. 8–13, IEEE Computer Society, April 2005.
- [28] D. Salamon, *A Guide to Data Compression Methods*. New York, U.S.A.: Springer, 2002.

- [29] D. A. Lelewer, D. S. Hirschberg, “Data compression,” *Computing Surveys*, vol. 19, pp. 261–296, September 1987.
- [30] K. Sayood, *Introduction to Data Compression*. San Diego, U.S.A.: Academic Press, 2000.
- [31] T. Blu, P. L. Dragotti, M. Vetterli, P. Marziliano, L. Coulot, “Sparse sampling of signal innovations,” *IEEE Signal Processing Magazine*, vol. 25, pp. 31–40, March 2008.
- [32] E. J. Candes, M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, March 2008.
- [33] E. Candes, J. Romberg, “Signal recovery from random projections,” in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol. 5674, (San Jose, California, U.S.A.), pp. 76–86, Society of Photo-Optical Instrumentation Engineers, January 2005.
- [34] C. Candes, J. Romberg, T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure Applied Mathematics*, vol. 59, pp. 1207–1223, August 2006.
- [35] C. Zhu, “Stable recovery of sparse signals via regularized minimization,” *IEEE Transactions on Information Theory*, vol. 54, pp. 3364–3367, July 2008.
- [36] S. S. Chen, *Basis Pursuit*. PhD thesis, Stanford University, Department of Statistics, California, U.S.A., November 1995.
- [37] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, R. G. Baraniuk, “Single-pixel imaging via compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, pp. 83–91, March 2008.
- [38] K. G. Beauchamp, *Walsh Functions and Their Applications*. New York, U.S.A.: Academic Press, Inc., 1975.

- [39] M. Lustig, D. L. Donoho, J. M. Santos, J. M. Pauly, “Compressed sensing MRI,” *IEEE Signal Processing Magazine*, vol. 25, pp. 72–82, March 2008.
- [40] Y. Wiaux, L. Jacques, G. Puy, A. M. M. Scaife, P. Vandergheynst, “Compressed sensing imaging techniques for radio interferometry,” *Monthly Notices of the Royal Astronomical Society*, vol. 395, pp. 1733–1742, April 2009.
- [41] A. Nüchter, “Osnabrück University and Jacobs University knowledge-based systems research group repository,” March 2010. <http://kos.informatik.uni-osnabrueck.de/3Dscans/> (8<sup>th</sup> data set as of July 2010).
- [42] A. Nüchter, K. Lingemann, “Osnabrück University and Jacobs University knowledge-based systems research group repository,” March 2010. <http://kos.informatik.uni-osnabrueck.de/3Dscans/> (13<sup>th</sup> data set as of July 2010).
- [43] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. U.S.A.: John Wiley & Sons, 1996.
- [44] Y. Bar-Shalom, T. E. Fortmann, *Tracking and Data Association*. San Diego, U.S.A.: Academic Press, 1988.
- [45] R. L. Anderson, “Distribution of the serial correlation coefficient,” *The Annals of Mathematical Statistics*, vol. 13, pp. 1–13, March 1942.
- [46] S. S. Beauchemin, J. L. Barron, “The computation of optical flow,” *ACM Computing Surveys*, vol. 27, pp. 433–467, September 1995.
- [47] D. J. Fleet, Y. Weiss, *Optical flow estimation*, ch. 15, pp. 239–258. Handbook of Mathematical Models in Computer Vision, Berlin, Germany: Springer, 2006.
- [48] R. Base, *Information Theory, Coding and Cryptography*. New Delhi, India: Tata McGraw-Hill, 2008.



- [49] G. Peyre, “Toolbox sparsity,” July 2009. <http://www.mathworks.com/matlabcentral/fileexchange/16204>.
- [50] J. Ziv, A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, May 1977.
- [51] G. Roelofs, M. Adler, “The ZLIB Homepage,” August 2009. <http://www.zlib.net/>.
- [52] J. Gailly, M. Adler, “The GZIP Homepage,” July 2003. <http://www.gzip.org/>.
- [53] M. Kleder, “Rapid lossless data compression of numerical or string variables,” November 2005. <http://www.mathworks.com/matlabcentral/fileexchange/8899>.
- [54] J. Hopkins, “Compression routines,” October 2009. <http://www.mathworks.com/matlabcentral/fileexchange/25656-compression-routines>.
- [55] W. B. Pennebaker, J. L. Mitchell, *JPEG Still Image Data Compression Standard*. The Netherlands: Kluwer Academic Publishers, 2004.
- [56] G. Strang, T. Nguyen, *Wavelets and Filterbanks*. Wellesley MA, U.S.A.: Wellesley-Cambridge Press, 1997.
- [57] O. Dobrucali, B. Barshan, *A compression method based on compressive sampling for 3-D laser range scans of indoor environments*, vol. 62 of *Lecture Notes in Electrical Engineering*, ch. 7. Berlin, Germany: Springer, October 2010. in Proceedings of the 25th International Symposium on Computer and Information Sciences (ISCIS 2010), E. Gelenbe, R. Lent, G. Sakellari, A. Sacan, H. Toroslu, A. Yazici (editors).
- [58] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics, 1992.

- [59] Y. Bar-Shalom, X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Boston, U.S.A.: Artech House, 1993.
- [60] X. F. Sun, P. L. Rosin, R. R. Martin, F. C. Langbein, “Noise in 3-D laser range scanner data,” in *Proceedings of IEEE International Conference on Shape Modeling and Applications*, (Stony Brook, New York, U.S.A.), pp. 37–45, IEEE Computer Society, June 2008.