Ph.D. Dissertation of Engineering

A Timing-Aware Sampling Algorithm for Dual-Mirror LiDAR Imaging

듀얼미러 라이다 이미징을 위한 타이밍이 고려된 샘플링 알고리즘

February 2019

Graduate School of Engineering
Seoul National University
Electrical and Computer Engineering Major

NGUYEN XUAN TRUONG

A Timing-Aware Sampling Algorithm for Dual-Mirror LiDAR Imaging

Supervisor: Professor Lee Hyuk-Jae

Submitting a Ph.D. Dissertation of Engineering

February 2019

Graduate School of Engineering
Seoul National University
Electrical and Computer Engineering Major

Nguyen Xuan Truong

Confirming the Ph.D. Dissertation written by
Nguyen Xuan Truong
February 2019

Chair 최기영 (Seal)
Vice Chair 이혁재 (Seal)
Examiner 조남익 (Seal)
Examiner 채수익 (Seal)
Examiner 김현 (Seal)

Abstract

In recent years, active sensor technologies such as light detection and ranging (LIDAR) have been intensively studied in theory and widely adopted in many applications, i.e., self-driving cars, robotics and sensing. Generally, the spatial resolution of a depth-acquisition device, such as a LiDAR sensor, is limited because of a slow acquisition speed. To accurately reconstruct a depth image from a limited spatial resolution, a two-stage sampling process has been widely used. However, two-stage sampling uses an irregular sampling pattern for the sampling operation, which requires a large amount of computation for reconstruction. A mathematical formulation of a LiDAR system demonstrates that the existing two-stage sampling does not satisfy its timing constraint for practical use. Therefore, designing a LiDAR system with an efficient sampling algorithm is a significant technological challenge.

Firstly, this thesis addresses the problem of adopting the state-of-art laser marking system of a dual-mirror deflection scanner when creating a high-definition LIDAR system. Galvanometer scanners are modeled and parameterized based on concepts of their controllers and the well-known raster scanning method. The scanning strategy is then modeled and analyzed considering the physical scanning movement and the minimum spanning tree. From this analysis, the link between the quality of the captured image of a field of view (FOV) and the scanning speed is revealed. Furthermore, sufficient conditions are derived to indicate that the acquired image fully covers the FOV and that the captured objects are well aligned under a specific frame rate. Finally, a sample LIDAR system is developed to illustrate the proposed concepts.

Secondly, to overcome the drawbacks of two-stage sampling, we propose a new sampling method that reduces the computational complexity and memory requirements by generating the optimal representatives of a sampling pattern in down-sample data. A sampling pattern is derived from a k-NN expanding operation from the downsampled representatives. The proposed algorithm is designed to preserve the object boundary by restricting the expansion-operation only to the object boundary or complex texture. In addition, the proposed algorithm runs in linear-time complexity and reduces the memory requirements using a down-sampling ratio. Experimental results with Middlebury datasets and Brown laser-range datasets are presented.

Thirdly, state-of-the-art adaptive methods such as two-step sampling are highly effective while addressing indoor, less complex scenes at a moderately low sampling rate. However, their performance is relatively low in complex on-road environments, particularly when the sampling rate of the measuring equipment is low. To address this problem, this thesis proposes a region-of-interest-(ROI)-based sampling algorithm in on-road environments for autonomous driving. With the aid of fast and accurate road and object detection algorithms, particularly those based on convolutional neural networks (CNNs), the proposed sampling algorithm utilizes the semantic information and effectively distributes samples in road, object, and background areas. Experimental results with KITTI datasets are presented.

Keywords : LiDAR, Scanning sampling patterns, Sampling, Reconstruction.
Student Number : 2015-30752

Table of Contents

List of Figures

Fig. 3.4 Block diagram of the proposed sample LIDAR system. LD- Laser diode, PD- Photodiode, TIA- Transimpedance Amplifier, PGA – Programmable operational amplifier, TDC- Time-to-digital Converter... 31

List of Tables

Chapter 1: Introduction

1.1. Overview

In recent years, autonomous driving or self-driving cars is considered the next logical step of the car industry. Not only granting more personal freedom, allow people to free their hands and mind while driving, self-driving cars are also expected to vastly reduce road accidents, congestion, pollution; and eliminate the huge cost of owning personal vehicles when integrating with blooming share-driving services [1]. There are five self-driving levels, starting with level 0 of no automation to level 5 of complete automation where drivers simply just enter the destination and let the systems do the rest of work. While the ultimate goal is a safe and fully autonomous driving system, current commercial systems now can constantly sense the surrounding environments and immediately alert the drivers of danger, traffic situations, and road conditions. Being considered a key to the future, autonomous driving and advanced driver-assistance systems, or ADAS, have been actively studied and developed over two decades [2]-[6].

From the sensing aspect, 3-D cameras such as RGB-D cameras and LiDAR (Light Detection and Ranging) sensors are becoming more affordable, enabling both academic studies and industrial applications, such as self-driving cars employing video analytics on LiDAR captured data for path planning as well as obstacle detection [7], [8]. To mimic the complicated natural sensing system of humans, a vehicle is installed with different types of sensors such as grayscale/color cameras, inertial and GPS navigation sensors, radio detection and ranging (RADAR), and light detection and ranging (LiDAR) systems [9], [10].

One of the most critical and challenging tasks in autonomous driving is the generation of a local map of objects (i.e., road, vehicles, and pedestrians) surrounding a car. This task directly relies on the depth sensing technologies. In recent years, a widely used solution for estimating a depth map is to use an active sensor (i.e., LiDAR), which generally scans objects in a field of view (FOV) and measures their distances. Whereas classical stereo vision techniques are only capable of estimating distances of close objects [11]-[13], a LiDAR sensor can produce rich information of a wide and broad FOV with a range up to 120 m [14]-[18]. However, there are two challenging problems in utilizing LiDAR sensors. Firstly, compared to camera sensors, it has a limited resolution owing to gradual data acquisition; hence, it is challenging to represent object-level information rather than pixel-level masks in an FOV. Secondly, although a LiDAR is capable of constructing a high-definition map of objects, this requires hardware resources to process and store a large-scale point cloud. In addition, it is noteworthy that a high-resolution LiDAR is highly expensive. To solve these two problems, efficient and accurate sampling is required to reduce the spatial resolution of a LiDAR, particularly for autonomous driving in on-road environments, where the spatial and temporal resolution of a LiDAR sensor is significantly sparser than that of an image sensor.

1.2. Scope and contributions

In this thesis, we are interested in three fundamental questions.

1- *How to design a LiDAR that can scan different sampling patterns*? This thesis addresses the problem of adopting the state-of-art laser marking system of dual-mirror scanners when creating a high-definition LIDAR system. This dual-mirror LiDAR's advantage is to scan a field of view with a designated scanning pattern, for example, when increasing a vertical resolution or track an object area.

2- *What are limitations of existing sampling algorithms on LiDAR*? This thesis demonstrates that the conventional sampling problem in theory oversimplifies the timing behaviors in a LiDAR. To address this problem, timing behaviors in a typical LiDAR are intensively characterized and a new sampling problem with timing constraints are proposed.

3- *How to solve the new sampling problem for LiDAR in different cases*? This thesis demonstrates that state-of-art sampling algorithms violate the timing constraints in the new sampling problem. In this thesis, three fast and accurate sampling algorithms are proposed to address the LiDAR scanning problem under different scenarios: 1) Only depth information is used; and 2) RGB images from a CMOS camera and regions of interest are provided.

1.3. Thesis Outlines

This thesis is organized as follows. Chapter 2 briefly reviews some related work. A dual-mirror LiDAR system and its characteristics are presented in Chapter 3. In Chapter 4, a new sampling model for LiDAR is presented and a timing-aware sampling algorithm is proposed. Chapter 5 presents an ROI-based LiDAR sampling algorithm in on-road environment for autonomous driving. Chapter 6 presents some implementation issues to design a LiDAR. Finally, Chapter 7 concludes this thesis.

Chapter 2: Related work

This thesis are related three areas of LiDAR: circuits and devices, sampling and reconstruction as in Fig. 2.1.

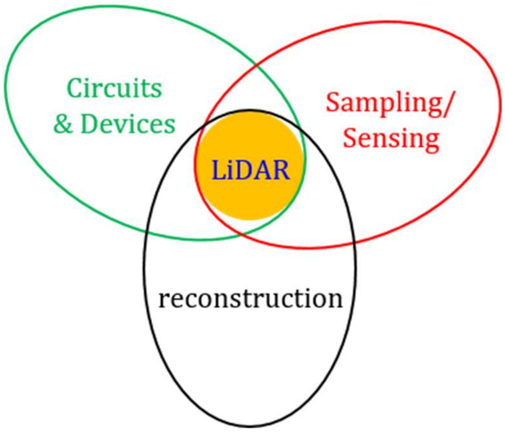


Fig. 2.1. LiDAR and three fundamental areas: circuits and devices, sampling/ sensing, and reconstruction.

2.1. LiDAR sensors

To measure distances, a LIDAR system estimates the time interval between the emission of the light photons from the LIDAR and the arrival of the light reflected from a distant object. According to the number of emitter and detector pairs, LIDAR systems are categorized into two types: those which use one pair and those which use multiple pairs as shown in Fig. 2.2. Both types have their own advantages and disadvantages.

The well-known commercial product Velodyne LIDAR [19]-[21] was motivated by a simultaneous use of multiple emitter/detector channels in the vertical direction. As those channels are rotated by a motor, the system can capture an image with multiple lines of data. For example, 16-, 32- or 64-channel versions

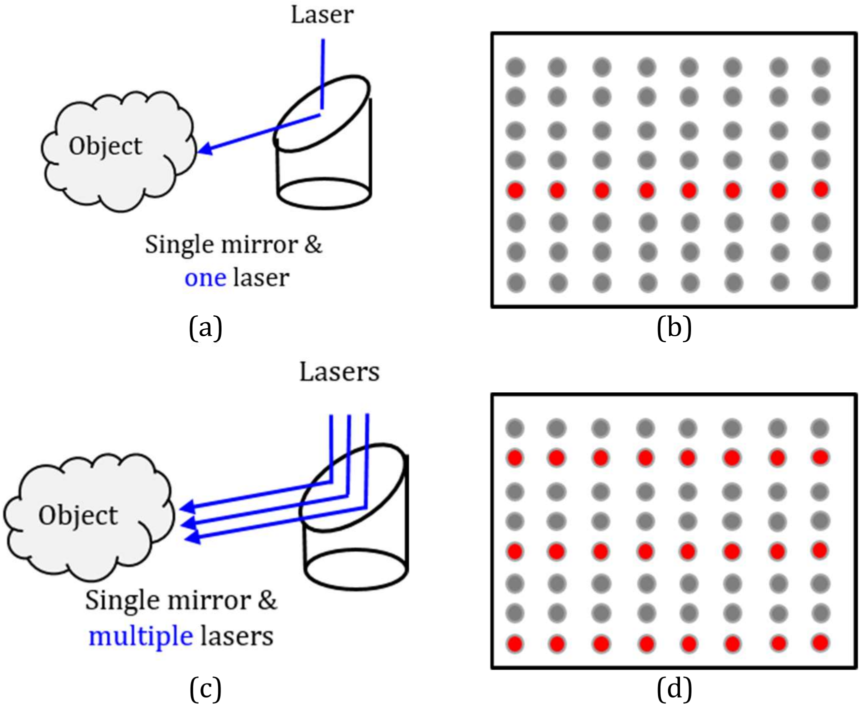


Fig. 2.2. LiDAR Scanning mechanism. (a)-(b) 2D LiDAR with single rotating mirror and one lasers and its scanning pattern; and (c)-(d) 3D LiDAR with single mirror and multiple lasers and its scanning pattern.

can produce 16, 32 or 64 lines of point cloud data, respectively, corresponding to 300,000, 700,000 or 1,200,000 measurements per second. Moreover, they achieve a relatively high frame rate of about 10 frames per second (fps). Despite these advantages, their prices are relatively high due to the hardware cost of multiple channels. On the other hand, a single-channel LIDAR is widely used in laser marking systems [22]-[25]. Because there is only a single emitter and detector pair, the price of this system is relatively low. Moreover, the system is easy to control and flexible to scan an arbitrarily point in a field of view (FOV). As a result, it can be used to scan various patterns and to increase the vertical resolution. However, the scanning speed and frame rate are not very high, and consequently, it takes a few seconds to obtain patterns in a laser marker.

2.2. Sampling

In general, an ultimate goal of a sampling problem is to select a sampling matrix to minimize the reconstruction error in an entire scene or a specific region-of-interest (ROI). This problem is directly related to the compressive sensing theory, which has intensively studied in many decades. Several approaches to find a sampling matrix have been presented in [26]-[31]. Generally, for a given sampling budget, a sampling algorithm should pick samples along gradient (as in Fig. 2.3(a)) than a regular grid sampling (as in Fig. 2.3(b)).

2.2.1. Sampling problem definition

Let $x \in R^N$ be a $N \times 1$ vector representing the depth map of an entire scene in a field of view (FOV) of a capturing device such as LiDAR. For straightforwardness, x is normalized such that $0 \leq x_i \leq 1$ for $i = 1, \dots, N$. In general, a sensor device cannot acquire data for all the locations in the FOV such that the depth map of the entire FOV is reconstructed from the sampled data. Let M denote the number of samples that a sensor device can acquire. The *sampling problem* is an optimization problem of selecting the samples in the FOV to minimize the reconstruction error with the constraint that the number of the samples satisfies the target budget M. For mathematical formulation, let {1,…,N} denotes the set of indexes that correspond to the locations of the entire FOV, while {i_1,…, i_M} represents the set of the indexes that correspond to the sample locations among {1,…,N}.

Problem P1 (*Sampling problem*)*:* The sampling problem is to derive {i_1,…, i_M} to

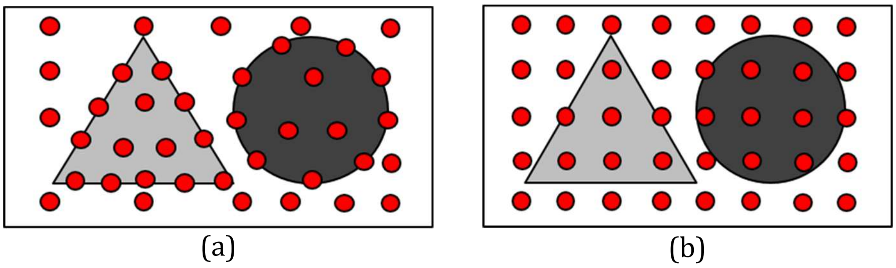


Fig.2.3. Two sampling patterns on a scene: (a) Edge-aware sampling; and (b) Regular grid sampling.

minimize the following objective function

$$\min_{i_1,\ldots,i_M} \frac{1}{N}\sum_{j=1}^{N} \left\|x_j - \widetilde{x_j}\right\|_L \tag{2.1}$$

where $x_1, \ldots, x_N$ are real values and $\widetilde{x_1}, \ldots, \widetilde{x_N}$ are the values that are estimated from M measurements $x_{i_1}, \ldots, x_{i_M}$.

Because it is not feasible to obtain a solution in a brute-force search manner, a heuristic method is most likely used. The next subsection presents a heuristic algorithm called Oracle random sampling or gradient-based sampling, which is derived in [28].

2.2.2. Sampling model

In general, a probabilistic model is used to represent the sampling problem. For N locations in an FOV, a diagonal matrix $S \in R^{N\times N}$ is used to represent the sampling operation with the $(i,i)^{\text{th}}$ entry of S being

$$S_{ii} = \begin{cases} 1, & \text{with probability} \quad p_i, \\ 0, & \text{with probability} \quad 1 - p_i, \end{cases} \tag{2.2}$$

where $0 \leq p_i \leq 1$ for $i = 1, \ldots, N$ is a sequence of predefined probabilities.

Given S, the sampled data $b \in R^{N\,\times\,1}$is defined by:

$$b = Sx. \tag{2.3}$$

where the i^{th} entry b_i is zero if $S_{ii} = 0$.

The target budget is defined by the target sampling ratio ξ with $0 < \xi < 1$, which represents the average sampling frequency. The following constraint is then obtained as follows:

$$\frac{1}{N}\sum_{i=1}^{N} p_i = \xi. \tag{2.4}$$

For a large N, the standard concentration inequality guarantees that the average number of ones in S is approximately ξN (i.e., $\xi N = M$) [28].

2.2.3. Oracle Random sampling (Gradient-based sampling)

Similar to [28], this study uses the gradient-based sampling method (called Oracle random sampling in [28]) to identify the edges or the highly textured areas of a given depth image. Let $a = [a_1, \dots, a_N]^T$be a vector that represents the magnitude of the gradient of the depth map:

$$a = \nabla x = \sqrt{(D_x x)^2 + (D_y x)^2}. \tag{2.5}$$

The intuition of the gradient-based sampling method is that the average gradient computed by all N samples is similar to the average gradient computed from a subset of ξN samples [13]. Let $\{p_j\}_{j=1}^{N}$ be the optimal sampling probability for defining the sampling map S. For a specified sampling ratio ξ and a gradient map, the derivation of the optimal sampling probability $\{p_j\}_{j=1}^{M}$is formulated as the following optimization problem:

$$\min_{p_1,\ldots,p_N} \frac{1}{N}\sum_{j=1}^{N}\frac{a_j^2}{p_j} \ . \tag{2.6}$$

subject to $\frac{1}{N}\sum_j^N p_j = \xi$ and $0 \ \leq \ p_j \leq \ 1$. In [32], the solution is formulated as follows:

$$p_j = \min(\tau a_j, 1). \tag{2.7}$$

where τ is the solution of $g(\tau) = 0$ and $g(\tau)$ can be calculated as follows:

$$g(\tau) = \sum_j^N \min(\tau a_j, 1) - \xi N. \tag{2.8}$$

Note that $g(\tau)$ is a piecewise linear and monotonically increasing function, with $g(+\infty) = N(1-\xi)$ and $g(0) \leq 0$ [32]. Therefore, τ can be uniquely determined as the root of $g(\tau)$. Moreover, an efficient solution for the derivation of τ is available.

2.3. Reconstruction

To evaluate a sampling pattern, a reconstruction or depth completion method is used. The problem of reconstructing a dense image from a sparse number of measurement is the compressive sensing (CS) which has been intensively studied for two decades [33]-[37]. While Shannon's theorem states that to reconstruct a signal (e.g., a depth profile) a sampling rate (e.g., the spatial resolution of our sensor) is required to be at least twice the maximum frequency of the signal, CS revolutionized signal processing by showing that a signal can be reconstructed from a much smaller set of samples if it is sparse in some domain. For depth estimation from sparse measurement, S. Hawe *et al.* [26] exploit the sparsity of the disparity maps in the Wavelet domain. The dense reconstruction problem is then posed as an optimization problem that simultaneously seeks a sparse coefficient vector in the Wavelet domain while preserving image smoothness. They also introduce a conjugate subgradient method for the resulting large-scale optimization problem. Liu *et al.* [28] empirically show that a combined dictionary of wavelets and contourlets produces a better sparse representation of disparity maps, leading to more accurate reconstruction. Ma et al. [38], [39] consider the case in which a robot has to navigate in an unknown environment but does not have enough on-board power or payload to carry a traditional depth sensor (e.g., a 3D LiDAR) and thus can only acquire a few (point-wise) depth measurements. Hence, they solve the construction problem with an assumption that the second-order derivative of depth map is sparse, leading to faster and more accurate reconstruction, especially when the number of sampling points are small. In recent years, inspired by convolutional neural network (CNN), the hand-craft dictionaries have been replaced by deep CNN in the depth completion task [40]-[43]. J. Uhrig et al. [40] introduces a sparsity-invariant CNN to reconstruct a depth from LiDAR measurements with a sparse convolution operator. N. Chodosh et al. [41] presents an CNN-based compressive sensing method for LiDAR depth completion using 1,800 parameters and an iterative manner. Ma et al. [43] introduces a LiDAR depth completion method based on a deeper network (Resnet-18) and achieve a superior reconstruction performance.

Chapter 3: Dual-Mirror LiDAR

3.1. Introduction

This study addresses the problem of adopting a state-of-the-art laser marking system [22]-[25], a dual-mirror deflection scanner in this case, to create a high-definition LIDAR system. The goal of this study is to model galvanometer scanners; and to analyze the performance of a LIDAR system based on this type of scanners.

Three major contributions are described below.

1- *Modeling*: Section 3.2 presents the controller modeling of a galvanometer scanner in a LIDAR system by deriving the speed constraints affecting the scanner controller. Both timing constraints on the physical speed of the scanner and the latency of the communication interface are addressed. Furthermore, notations and common terminologies are introduced for the rest of the study.

2- *Scanning problem*: Section 3.3 defines the laser scanning problem, shows an intuitive solution, and proves its optimality. Hence, the relationship between a field of view (FOV) of a captured image and the frame rate is derived. Moreover, sufficient conditions are derived to check whether the obtained image fully covers the FOV and includes well-aligned objects for given frame rates. Experimental results show that the scanners can achieve frame rates of 17.6, 9.0 and 4.6 fps for image sizes of 240×16, 240×32 and 240×64, respectively.

3- *LiDAR system*: Section 3.4 presents a sample LIDAR system which was developed based on the proposed solutions. The proposed system, able to provide images at various resolutions depending on target frame rates, is evaluated in terms of its speed and the resulting visual quality. The system achieves nearly 97,000 measurements per second while only using a single emitter/detector channel. In addition, given an FOV, scalable frame rates (i.e., 600×600, 300×300, or 150×150) can be achieved by the proposed system.

3.1.1. Related work

The demand for LIDAR sensors has been increasing due to the growing number of autonomous vehicles [2]-[5]. These sensors play a critical role in self-driving cars by Google or Tesla. This subsection briefly reviews existing LIDAR sensors which are available as commercial products. To this end, exiting LIDAR products are compared with important performance parameters such as the number of channels, the scan frequency, and the horizontal/vertical angles. Table 3.1 summarizes the product specifications as given by the relevant companies. LEDDAR 16M [17] uses sixteen segments simultaneously to measure the distances of objects at sixteen angles. As its rate is 50Hz, the number of measurements per second is 800. Riegl-VUX-1UAV Lidar [18] uses a single pair of an emitter and a detector which can be rotated in a 330 ° FOV. This 2D LIDAR is able to scan 550,000 measurements per second, and the motor speed can be configured at different speeds between 10Hz and 200Hz. IBEO ALASCA [16] also adopts a similar rotating method, but it uses four channels. Therefore, it can scan a vertical angle of 3.2°. Velodyne LIDAR sensors [19]-[21]

TABLE 3.1
LIST OF SOME LIDAR PRODUCTS ON THE MARKET

Model	Type	# of channels	Speed (Hz)	Horizontal Angle	Vertical Angle	# of Meas.
IBEO ALASCA [16]	3D	4	8-40	160°	3.2°	N/A
LEDDAR 16M [17]	2D	16	50	9°-95°	×	800
PUCK™ VLP-16 [19]	3D	16	10	360^{0}	+15° to -15°	300,000
HDL-32E [20]	3D	32	10	360^{0}	+10° to -30°	700,000
HDL-64E [21]	3D	64	10	360^{0}	26.9°	2,200,000
Riegl-VUX-1UAV [18]	2D	1	10-200	330°	×	550,000

also use a rotation module to extend the vertical scanning angle. They increase the number of measurements proportionally by increasing the number of emitter/detector channels.

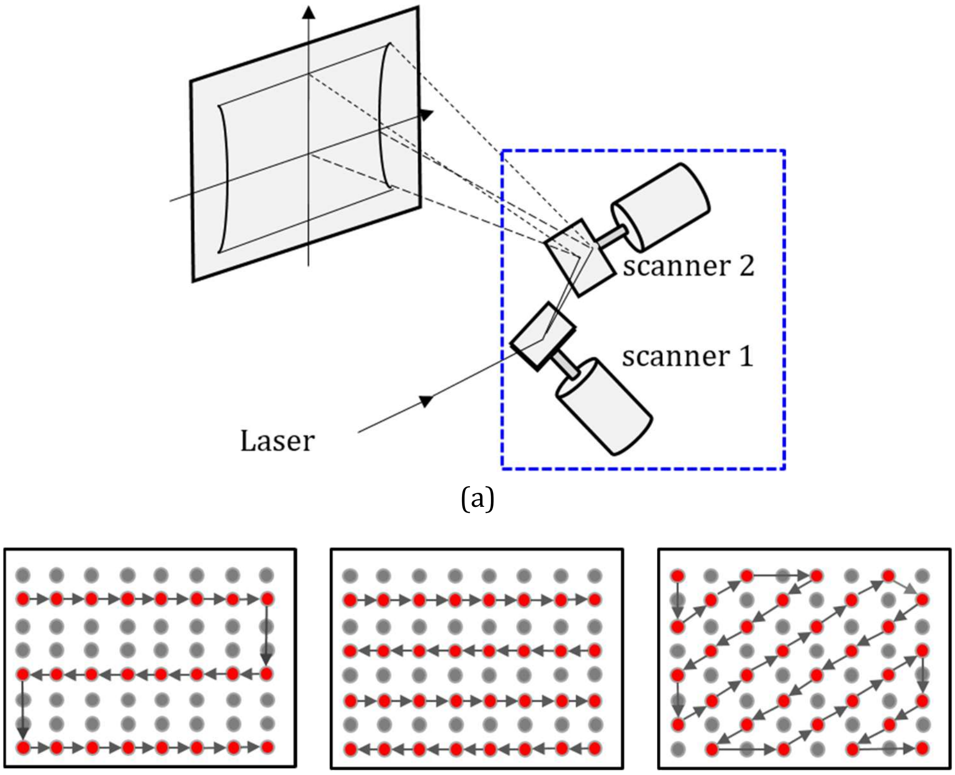
These LIDAR products share a similar scanning structure in which only one motor is used. Thus, any increase in the vertical resolution must depend on the number of optical channels. This study suggests a solution which adopts the scanners used in laser marking systems. This LIDAR system only requires a single emitter/detector pair while offering various resolutions. To show the effectiveness of this LIDAR system, this chapter aims to investigate its performance in terms of the frame rate, resolution and FOV and also creates a sample LIDAR system based on the proposed concept.

3.2. Modelling a controller of dual-mirror scanners

The common structure of a single-channel LIDAR system is a combination of 1) a scanning module and 2) a single-point measurement module. This study focuses on the scanning module, which is directly related to the FOV and the frame rate of the system. Therefore, optical parts and depth measurements are only briefly covered in Section 3.4.

3.2.1. Dual-mirror scanners

The single-channel LIDAR system in this study adopts the state-of-the-art laser



(a)

(b) (c) (d)

Fig. 3.1. Dual-mirror laser scanners. (a) System; and three scanning patterns: (b) Vertical raster scanning pattern (sampling 3 out of 8 lines); (c) Vertical raster scanning pattern (sampling 4 out of 8 lines); and (d) Diagonal raster scanning pattern.

marking system, i.e., the dual-mirror deflection system of the type widely used in industrial applications [22], [44]. As shown in Fig. 3.1, the scanners control two motors to move their corresponding mirrors to a point at a specific position in the FOV. Each mirror is responsible for scanning the direction in the FOV. In Fig. 3. 1, scanning angles which represents the angle differences with respect to the origins are defined by ϕ_1 and ϕ_2 for the x and y axes, respectively. In practice, it is known that dual-mirror deflection systems are very accurate. For example, the scanners adopted in this study have an angle step of 12 μrad, with each capable of scanning a maximum of 60,000 steps. Therefore, the system can cover a FOV of 41.2^o×41.2^o and represent a scene with very high-resolution images of 60,000×60,000.

3.2.2. Controller Model

3.2.2.1. FOV representation

Initially, let x_{min}, y_{min}, x_{max}, and y_{max} denote the minimum and maximum ranges in the vertical and horizontal directions of the interest FOV, respectively. In this case, these variables are expressed as follows:

$$0 \leq \ x_{min,}\ x_{max,} y_{min}, y_{max} \leq \ 60{,}000 \tag{3.1}$$

In practice, the entire FOV is usually represented by a set of regular grid points. In other words, any two consecutive points are aligned by a fixed step. Let $xstep, ystep$ denote regular steps on the x-axis and y-axis of the FOV, respectively. Hence, the width and height of the obtained image can be derived as follows:

$$width = \frac{x_{max} - x_{min}}{xstep} \tag{3.2}$$

$$height = \frac{y_{max} - y_{min}}{ystep} \tag{3.3}$$

Example 1: It is assumed that the LIDAR system fully covers the FOV (*i.e.,* x_{min} = y_{min} = 0 and x_{max} = y_{max} = 60,000), which is represented by a frame 480×240 in size (*i.e.,* width = 480 and height = 240). The regular scanning steps are then as follows:

$$xstep = \frac{60{,}000}{480} = 125 \tag{3.4}$$

$$height = \frac{y_{max} - y_{min}}{ystep} \tag{3.5}$$

3.2.2.2. Timing constraints

To control the scanner, it is necessary to send a destination coordinate (x_D, y_D) so that it can move to there from its current position (x_C, y_C). If the LIDAR system measures the distance at each position, the number of measurements is equal to the number of positions. Obviously, these numbers depend on how many new positions are sent to the scanner and its maximum scan speed.

First, dual-mirror deflections scanners are interfaced by a communication interface (i.e., XY2-100 industrial protocol [45]). An illustration of the protocol is shown in the upper part of Fig. 3. 2. To control two motors, a driver must send 16-bit coordinates via the x and y channels (XCHN, YCHN) for an update. Moreover, the protocol includes four additional control bits to form a 20-bit packet to handle the new position. Hence, it takes 20 cycles to send the destination to the scanners. The specification shows that the clock frequency (CLK) is limited by 2MHz, resulting in a minimal period of 500ns. Eventually, an update requires $10\mu s$ (*i.e.,* $20 \times 500ns$). This implies that the maximum number of updates is limited by 100,000 positions.

Second, the maximum speed of a galvanometer scanner is bounded by the maximum frequency f_{max} at which it is able to finish the travel of x_{min} → x_{max} → x_{min} in one second. It should be noted that the maximum frequency parameter is usually given in the product specifications. However, the scanning angle is not clearly stated in these specifications. Both $f_{max} = 150$ and $f_{max} = 1000$ are correct if their corresponding scanning angles are not provided. Intuitively, setting a narrower scanning angle results in a faster scan speed. This intuition offers a solution to increase the frame rate by narrowing

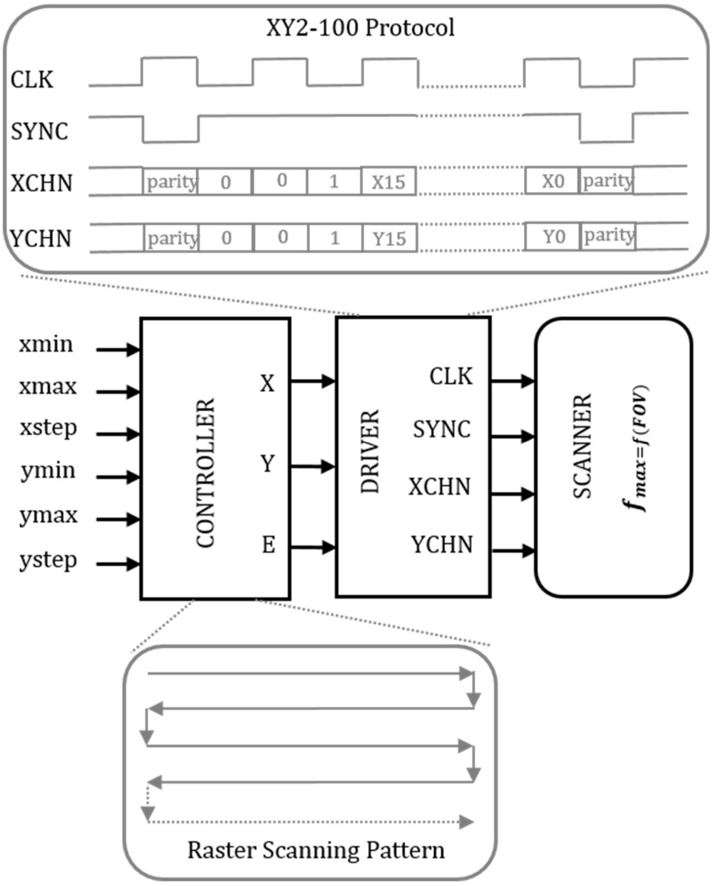


Fig. 3.2. Controller design for galvanometer scanners using the XY2-100 protocol and a raster scanning pattern.

the scanning FOV.

3.2.2.3. Maximum Speed of LiDAR scanners

In this subsection, the maximum speed of the galvanometer scanner is formally derived. Due to inertia, updating of the position starts by slowly increasing its speed, reaching to a stable speed, and finally slowly decreases to zero when moving to the destination. This study assumes that the speed is constant during the movement if it is still under the maximum speed, as this approach does not sacrifice generality.

$$v \leq v_{max} = \frac{2f_{max}(x_{max} - x_{min})}{1s} \tag{3.6}$$

Here, v_{max} is the maximum average speed of a scanner, and the numerator, $2f_{max}(x_{max} - x_{min})$, is the path length that the scanner traverses in one second.

Example 2: Assume that the galvanometer scanner is able to scan at a maximum frequency $f_{max} = 150$ at an angle of 45^{o} with x_{max} = 60,000 and x_{min} = 0. Then, its speed is bounded as follows:

$$v \leq v_{max} = \frac{2 \times 150 \times 60{,}000}{1s} = \frac{18}{\mu s} \tag{3.7}$$

This indicates that the scanner is not able to move along a path longer than 18 within one microsecond. For example, if each step unit is 12 μrad, the scanner cannot move through an angle longer than 18 step units during that time.

3.3. LiDAR scanning optimization problem

Existed double galvanometers are often used in laser marking systems [22]-[25] which are often required to handle an arbitrary pattern. The arbitrary pattern is derived by a complicated algorithm such as a genetic-based approach [24]. This complex scanning pattern is possible because a marking system does not require a fast scanning speed as the marking repetition rate is only about 1kHz to 5kHz [22]. On the other hand, the scanning speed is very important for a LIDAR system so that this study attempts to find the scanning pattern that increases the scanning speed. To this end, this study proposes to use a special scanning mode which is a simple yet fast scanning mode. More importantly, our approach gives the optimal scanning pattern for a LIDAR scanning system.

3.3.1. Scanning Problem

This subsection addresses the scanning problem of finding the optimal path along which scanners run at the maximum speed. The formal definition of the problem is as follows:

Definition 1 **(Scanning Problem):** Given a set of N positions in the FOV, the scanning problem is to find the Eulerian trail which visits every position exactly once.

Let $\{p_i\}_{i=1}^{N}$ denote the positions in the FOV. The scanning problem is to find a trail, $q_1 \to q_2 \to \cdots \to q_N$, where the set $\{q_j\}_{j=1}^{N}$ is equal to $\{p_i\}_{i=1}^{N}$. Although two sets may have identical elements, the orders of their elements can differ. This condition does not indicate that the coordinates of q_i must be identical to those of p_i for any $i = 1, \ldots, N$. In other words, the problem to solve is to find the scanning order in which all positions are visited exactly once. It should be noted that $\{q_j\}_{j=1}^{N}$ is a permutation of $\{p_i\}_{j=1}^{N}$ and that any such permutation results in a valid order for scanners. Additionally, the condition which holds that every position should be visited exactly once reflects the actual case, in which it is desired to have only one distance measurement at a single position for an image.

When N positions are given, there are $N!$ possibilities to obtain a valid trail for scanners, as $N!$ permutations of $\{p_i\}_{j=1}^{N}$ exist. The natural demand is to find the trail along which the scanners can travel in the shortest time or along the shortest path. Given the assumption that the speed of the scanner is constant during the movement, the traversing time is proportional to the path length. Therefore, the optimal scanning optimization is defined as follows:

***Problem 1* (Optimal Scanning Problem):** Given a set of N positions $\{p_i\}_{i=1}^{N}$ in the FOV, find the trail $q_1 \to q_2 \to \cdots \to q_N$ that minimizes the total length of the traversing path,

$$L_q = \sum_{\mathrm{i=1}}^{\mathrm{N-1}} \left|\left|q_i - q_{i+1}\right|\right|_2, \tag{3.8}$$

where the set $\{q_j\}_{j=1}^{N}$ is equal to $\{p_i\}_{i=1}^{N}$ and where $||.||_2$ denotes a Euclidian distance (i.e., L2-norm), which is formulated as follows:

$$\left|\left|q_i - q_{i+1}\right|\right|_2 = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{3.9}$$

3.3.2. Optimal scanning pattern

3.3.2.1. Grid-graph representation of Field of View

An FOV is usually represented by a grid graph in which any two consecutive points are aligned at fixed steps. Similar to Section 3.2.1, let $xstep, ystep$ denote regular steps on the x-axis and y-axis of the FOV, respectively. Without a loss of generality, it is assumed here that $xstep$ is equal to or smaller than y$step$. The optimal scanning problem is then to find an optimal trail on the grid graph. First, a feasible solution is considered, i.e., the raster scanning pattern, as shown in the lower part of Fig. 3.2. Starting from point (x_{min}, y_{min}), scanners visit the next point (x_{min} + *xstep*, y_{min}). They then iteratively visit others along the

Algorithm 3.1: Raster Scanning Algorithm with Laser Scanners

```
Inputs: $x_{min}, x_{max}, y_{min}, y_{max}, xstep, ystep$
Outputs: $x, y$

$x=x_{min}, y=y_{min}, dir = 0$
While $y < y_{max}$

    /* Even line */
    If $x==0$
        While $x < x_{max} - xstep$
            Measure an object's distance at the scanning angle
            $x := x + xstep$
        End while
        $dir = 1$
    Else

        /* Odd Line*/
        While $x > x_{min}$
            Measure an object's distance at the scanning angle
            $x := x - xstep$
        End while
        $dir = 1$
    End if
    $y := y + ystep$
End while
```

same line until they reach the right border (x_{max} – $xstep$, y_{min}). After going over all points on the first line, the scanners scan the next line in a similar manner. This procedure is repeated until all points on the grid graph are visited. The raster scanning pattern is shown in Alg. 3.1 and explained in detail in the next subsection.

3.3.2.2. Optimal scanning pattern

This subsection proves the optimality of the raster scanning pattern by the proposed theorem, as follows:

***Theorem 1* (Optimality):** If N positions forming the grid graph in the FOV are given and $xstep$ is equal to or smaller than $ystep$, the scanning pattern in Alg. 1 provides an optimal solution to *Problem 1*.

Proof: The scanning pattern derived by Alg. 3.1 forms a tree because it does not include any cycle. In addition, the tree consists of all vertices in the graph. Therefore, the proof for *Theorem 1* is equivalent to the problem of showing that the tree which is the derived raster scanning pattern is the minimum spanning tree (MST) [46] of the graph. In this study, the procedure of the well-known Prim algorithm [47] for the MST problem is adopted to address this problem. Let q_1 = (x_{min}, y_{min}) be the first node in the MST. In the Prim algorithm, the next node is the candidate of q_2 which is closest to q_1. A candidate solution is the point (x_{min} +x_{step}, y_{min}) due to $xstep$ being equal to or smaller than $ystep$. The procedure of choosing nodes for the MST on the first line is repeated until the rightmost node (x_{max}, y_{min}) is reached. Because there is no remaining node in this current line, the next node must be on the second line. According to the Prim algorithm, the MST can be expanded to any node on the second line. However, in order to guarantee the conditions of the MST, the rightmost one (x_{max}-x_{step}, y_{min} + x_{step}) is selected when its current position is (x_{max} - x_{step}, y_{min}). This procedure will be iteratively applied until all nodes in the grid graph are added to the tree to continuously form the MST.

Example 3: Consider an example with six positions, as shown in Fig. 3.3. Every two nodes are connected by an edge, which is the Euclidean distance between them. The problem is to find the MST over the graph. Here, we push node “1”

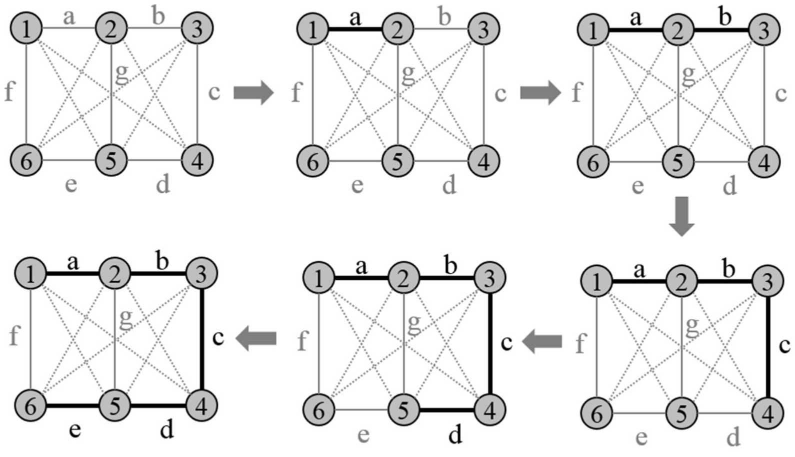


Fig. 3.3. An example of the Prim algorithm to find a raster scanning pattern on a grid graph.

into the set. Clearly, the minimum edge is "a," which should be added to the MST, after which node "2" is inserted into the set. The next edge will be the minimum weight edge connected to the set {"1", "2"}. Therefore, edge "b" is inserted into the MST and node "3" is added to the set. At this point, a minimum edge connected to the set {"1", "2", "3"} is required. There are three possible options: "c", "g" and "f". Mathematically, it is possible to select any of them, as the graph considered is an undirected graph. However, in reality, it is clear that the current position of the galvanometer is now at node "3." Eventually, edge "c" is selected and the node "4" is added to the set. Next, edge "d" will be added to the MST similarly and node "5" is inserted into the set. Finally, edge "e" is added to the MST, and the MST is completely constructed.

Although the raster scanning is a series of heuristic steps, it is the optimal solution in this specific case. It should be noted that the assumption that $xstep$ is equal to or smaller than $ystep$ is only used for expressing the proof clearly without a loss of generality. In case $xstep$ is larger than $ystep$, the requirements of Theorem 1 can still be met by slightly modifying Alg. 3.1 by exchanging the x-axis and the y-axis.

3.3.2.3. Combining an optimal sampling pattern with timing constraints

Here, we reconsider the controller in the center of Fig. 3.2. It requires six parameters: x_{min}, x_{max}, y_{min}, y_{max}, $xstep$ and $ystep$. x_{min}, x_{max}, y_{min} and y_{max} define the FOV of interest, whereas $xstep$ and $ystep$ determine the resolution of the captured image. The goal of this subsection is to investigate the link between the frame rate and the resolution factors.

3.3.2.3.1. Constraining in moving steps

Let t_x and t_y denote the time for scanners to travel the distances of $xstep$ and $ystep$, respectively. The speed of Galvanometer scanners in each small step should satisfy the condition in (3.6), which states that the maximum speed constraint and thus the speeds of the scanners, v_x and $v_{y,}$, must satisfy the following conditions:

$$v_x = \frac{xstep}{t_x} \leq v_{max} = \frac{2f_{max}(x_{max} - x_{min})}{1s} \tag{3.10}$$

$$v_y = \frac{ystep}{t_y} \leq v_{max} = \frac{2f_{max}(y_{max} - y_{min})}{1s} \tag{3.11}$$

These conditions are defined based on the speed constraints; thus, they can be directly rewritten under the timing constraints, which are derived as follows:

$$\frac{1s}{2f_{max}} \leq \frac{x_{max} - x_{min}}{xstep} \times t_x \tag{3.12}$$

$$\frac{1s}{2f_{max}} \leq \frac{y_{max} - y_{min}}{ystep} \times t_y \tag{3.13}$$

Intuitively, the right sides of (3.12) and (3.13) indicate the times required for

Galvanometer scanners to complete the scan of the horizontal and vertical lines, respectively. Therefore, each frame requires the following time,

$$\frac{1}{FPS} = \frac{y_{max}-y_{min}}{ystep} \times \frac{x_{max}-x_{min}}{xstep} \times t_x + \frac{y_{max}-y_{min}}{ystep} \times t_y, \tag{3.14}$$

where *FPS* denotes the frame rate. The first term on the right side of (3.14) represents the x-axis scanning time, which is determined by multiplying the number of lines by the horizontal-line scanning time, whereas the second term on the right side of (3.14) is the vertical-line scanning time. It should be noted that the raster scanning pattern in Alg. 3.1 is used for this derivation.

3.3.2.3.2. Two constrains on frame rate

Based on (3.12), (3.13), and (3.14), the first constraint on the frame rate is derived as follows:

$$\frac{1}{FPS} \geq \left(\frac{y_{max} - y_{min}}{ystep} + 1\right) \times \frac{1s}{2f_{max}} \tag{3.15}$$

This implies that the frame rate depends on the number of lines in the acquired image. If the scanners scan more lines, $\left(\frac{y_{max}-y_{min}}{ystep}\right)$, *FPS* becomes smaller. On the other hand, if they scan fewer lines, *FPS* becomes larger.

Next, the second constraint is derived to compensate for (3.15), as (3.15) does not include x-axis parameters which represent the number of pixels on a line (*width*). It should be noted that t_x and t_y are the timing intervals to update the positions on the horizontal and vertical directions, respectively. Therefore, they must be equal to or greater than the timing unit τ by the communication interface (*i.e.*, $t_x \geq \tau$ and $t_y \geq \tau$). Thus, the second condition is directly derived from (3.14), as follows:

$$\frac{1}{FPS} \geq \left(\frac{y_{max} - y_{min}}{ystep} \times \frac{x_{max} - x_{min}}{xstep} + \frac{y_{max} - y_{min}}{ystep}\right) \times \tau \qquad (3.16)$$

The term on the right of (3.16) is obtained by multiplying the number of pixels in the image by the time unit for updating the position in the FOV. The inequalities in (3.15) and (3.16) provide conditions which link the frame rate, the solution, and the FOV. The following subsection investigates these conditions with numerical examples.

3.3.2.3.3. Numerical examples

Example 4: For the example of the frame rate, the settings in Example 1 in which the galvanometer scanners scan at the maximum frequency ($f_{max} = 150$) and the frame image has the size of 480×240 are reused. Then, (3.15) can be changed as follows:

$$\frac{1}{FPS} \geq (height + 1) \times \frac{1s}{2f_{max}} \Rightarrow FPS \leq 1.25 \qquad (3.17)$$

Consider the case in which scanners use the XY2-100 industrial protocol [12] for the communication interface. In this case, each update requires $10\mu s$ (*i.e.,* $\tau = 10\mu s$), and (3.16) is changed as follows:

$$\frac{1}{FPS} \geq (height \times width + height) \times \tau \quad \Rightarrow FPS \leq \ 0.867 \qquad (3.18)$$

From (3.17) and (3.18), the maximal frame rate is determined to be 0.867 because (3.18) has a smaller inequality criterion. However, (3.17) only depends on the "*height*" parameter. If the frame width is decreased to 240, (3.16) is changed as follows:

$$\frac{1}{FPS} \geq (height \times width + height) \times \tau \quad \Rightarrow FPS \leq \ 1.73 \qquad (3.19)$$

From (3.17) and (3.19), the maximal frame rate is determined to be 1.25 because (3.17) has a smaller inequality criterion. This example clearly shows

TABLE 3.2
FRAME RATES OVER DIFFERENT RESOLUTIONS OF AN ACQUIRED IMAGE

height	*width*	*FPS* by (3.15) (fps)	*FPS* by (3.16) (fps)	***Final FPS (fps)***
16	240	17.647	25.934	**17.647**
16	480	17.647	12.994	**12.994**
16	640	17.647	9.750	**9.750**
16	1280	17.647	4.879	**4.879**
32	240	9.091	12.967	**9.091**
32	480	9.091	6.497	**6.497**
32	640	9.091	4.875	**4.875**
32	1280	9.091	2.440	**2.440**
64	240	4.615	6.483	**4.615**
64	480	4.615	3.248	**3.248**
64	640	4.615	2.438	**2.438**
64	1280	4.615	1.220	**1.220**

that either (3.15) or (3.16) can be dominant according to the setting of the resolution.

Recall that existing LIDAR systems usually adopt multiple channels of emitters and detectors. Therefore, multiple lines can be scanned simultaneously. The following part investigates the performance of the proposed controller with the same line settings used in the other cases. Consider the Velodyne systems VLP-16 [19], HDL-32E [20], and HDL-64E [21], which simultaneously scan 16, 32 and 64 data lines, respectively. Thus, *height* is set to 16, 32, and 64 for each corresponding system. In Table 3.2, the corresponding frame rates according to different resolutions are presented. It should be noted that the FPS by (3.15) in the third column is only proportional to *height*; whereas FPS by (3.16) in the fourth column is proportional to both *width* and *height*. The lower values for these two columns are selected for the final FPS in the fifth column. These results show that the scanners are able to achieve an fps of approximately 17.647 when acquiring an image resolution of

240×16, whereas acquiring an image of 1280×16 leads to a frame rate of 4.879 fps. This outcome indicates that a higher resolution results in a lower frame rate. In other words, the frame rate decreases linearly as the number of scanning lines increases.

Example 5: For the example of the FOV, the FOV factor which is not investigated in Example 3 is considered here. In particular, the frame rate and resolution do not indicate the coverage area in the FOV of interest. One assumption in Example 3 is expressed as follows:

$$t_x = t_y = \tau = 10\mu s \tag{3.20}$$

This indicates that the update time for the position is 10 μs. Hence, the maximum moving step is 180 due to the constraints on the speed of the scanners. This correlation can be expressed as follows:

$$\max\left(\frac{xstep}{t_x}, \frac{ystep}{t_y}\right) \leq v_{max} = 180 \tag{3.21}$$

$$\Rightarrow xstep \leq 180 \text{ and } ystep \leq 180 \tag{3.22}$$

Table 3.3 shows the FOVs corresponding to the different frame sizes and frame rates. The first and second columns represent the frame *height* and *width*, respectively. From (3.22), the maximum angles can be derived by multiplying the *width* and *height* with the moving steps. Columns from 3 to 6 show the angles of the FOV. The vertical angle gradually decreases due to the maximum speeds of Galvanometer scanners. In practice, the FOV of interest can be extended by increasing t_y in (3.20). Eventually, y_{step} is increased and the vertical angle is extended accordingly. Note that t_y represents the time unit for updating a scanning line. In general, it only needs *height*-1 times to switch from the current line to the next one. Thus, the FPS only increases slightly.

TABLE 3.3

FOVs over Different Resolutions of an Acquired Image

$height$	$width$	FOV_x		FOV_y	
		x_{max}	$angleX_{max}$	y_{max}	$angleY_{max}$
16	240	43,200	29.7°	2,880	**1.98°**
	480	60,000	41.2°		
	640	60,000	41.2°		
	1280	60,000	41.2°		
32	240	43,200	29.7°	5,760	**3.95°**
	480	60,000	41.2°		
	640	60,000	41.2°		
	1280	60,000	41.2°		
64	240	43,200	29.7°	11,520	**7.91°**
	480	60,000	41.2°		
	640	60,000	41.2°		
	1280	60,000	41.2°		

3.4. LiDAR system Prototype

This section presents the sample LIDAR system developed under the proposed concepts.

3.4.1. System overview

In addition to idealistic concepts, it is important to consider an actual verification framework, as it is directly linked to practical applications. Based on this objective, a sample LIDAR system is proposed. Its system block diagram is shown in Fig. 3.4. The central component of the proposed system is a controller which is built with a combination of a Raspberry PI III board [48] and a LOGI-PI board [49]. The Raspberry PI serves as a high-level communication interface bridge to send the depth measurements from the LOGI-PI board to the display PC via the TCP-IP Ethernet protocol [50]. All low-level interfaces are handled on the FPGA board, LOGI-PI, which uses a low-cost device, a Spartan 6-XC6SLX9-2TQG144C from Xilinx. The controller is connected to the Galvanometer scanners (motor and motor drivers) [44], a laser source (i.e., a laser diode, LD), a programmable gain amplifier circuit (PGA), and a time-to-digital converter (TDC) [51]. Signals reflected from objects are captured by a photodetector (PD) [52]. The distance measurement methods [53]-[54] are likely to be integrated in our framework. However, it should be noted that this study concentrates on the interface between the controller and the Galvanometer scanners. The selection of elements of other optical elements and the timing measurement circuits may be different, which is out of the scope of this study.

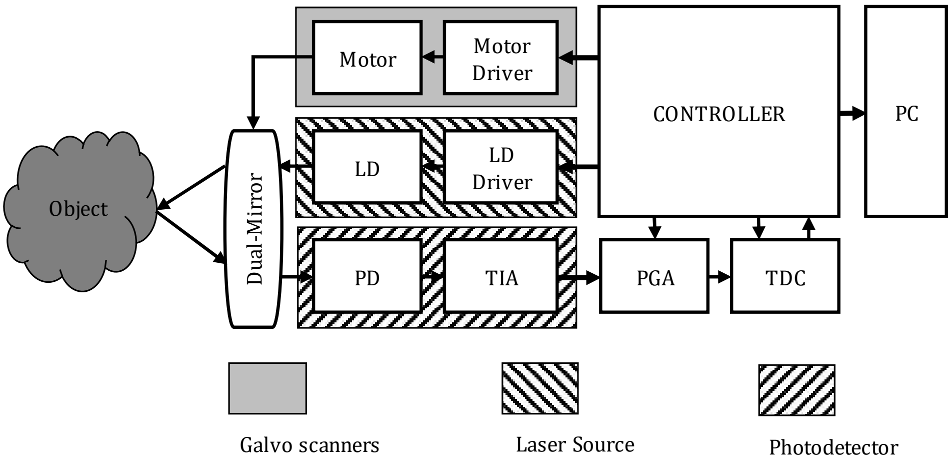
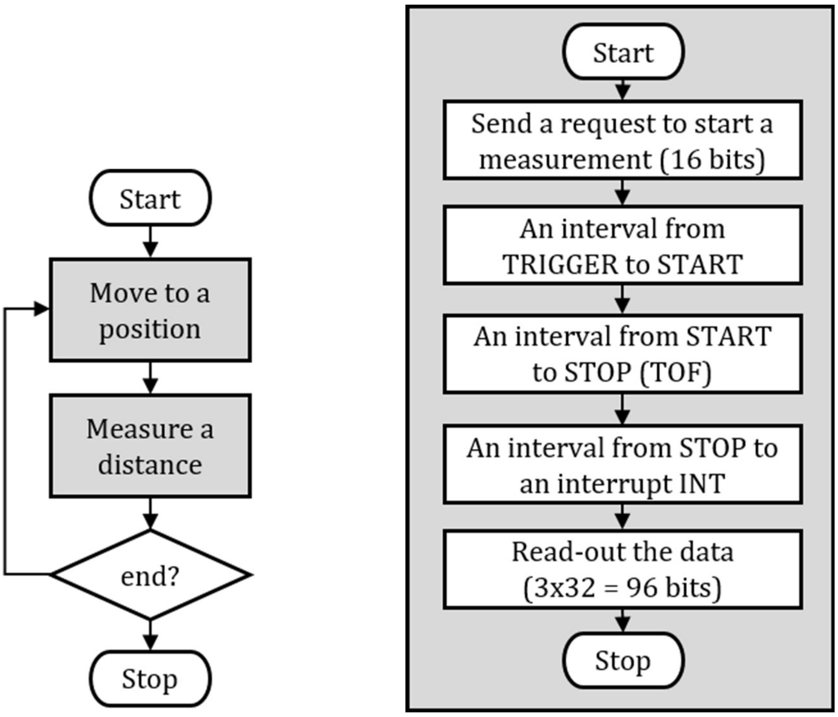


Fig. 3.4. Block diagram of the proposed sample LIDAR system. LD- Laser diode, PD- Photodiode, TIA- Transimpedance Amplifier, PGA – Programmable operational amplifier, TDC- Time-to-digital Converter

Fig. 3.5 illustrates the main routine of measuring the distances. The raster scanning algorithm in Alg. 1 is simplified in Fig. 3.5(a), in which the measurement of a single point is conducted by iteratively mapping the position to the measured distance. Each measurement starts by triggering an LD to emit light at the position and also enabling a TDC by the START signal. It should be noted that the emitted light comes to the object and that a reflected signal returns to the system and reaches the detector PD (referred to as a STOP signal). The TDC then measures the time interval between two signals, which is read out to the controller and translated into the distance. This procedure is summarized in Fig. 3.5(b).


Start
Move to a position
Measure a distance
end?
Stop
Start
Send a request to start a measurement (16 bits)
An interval from TRIGGER to START
An interval from START to STOP (TOF)
An interval from STOP to an interrupt INT
Read-out the data (3x32 = 96 bits)
Stop

(a) (b)

Fig. 3.5. Routine of measuring distances in a LiDAR system. (a) Simplified measurement flow in a LiDAR; and (b) Flowchart to measure a distance of a single point.

In this system, considering the interface between the controller and the LiDAR scanners, two subjects, the efficiency of various resolutions and the frame rate adjustment, should be addressed. These are explained in the following subsections.

3.4.2. Speed evaluation

This subsection discusses experiments with the proposed LIDAR system at different frame rates. The results of the experiments are presented in Table 3.4. The proposed LIDAR system achieves various frame rates by setting the corresponding frame size. For example, from the second row to the sixth row, the *width* in the second column is fixed at 360, but the *height* in the third column

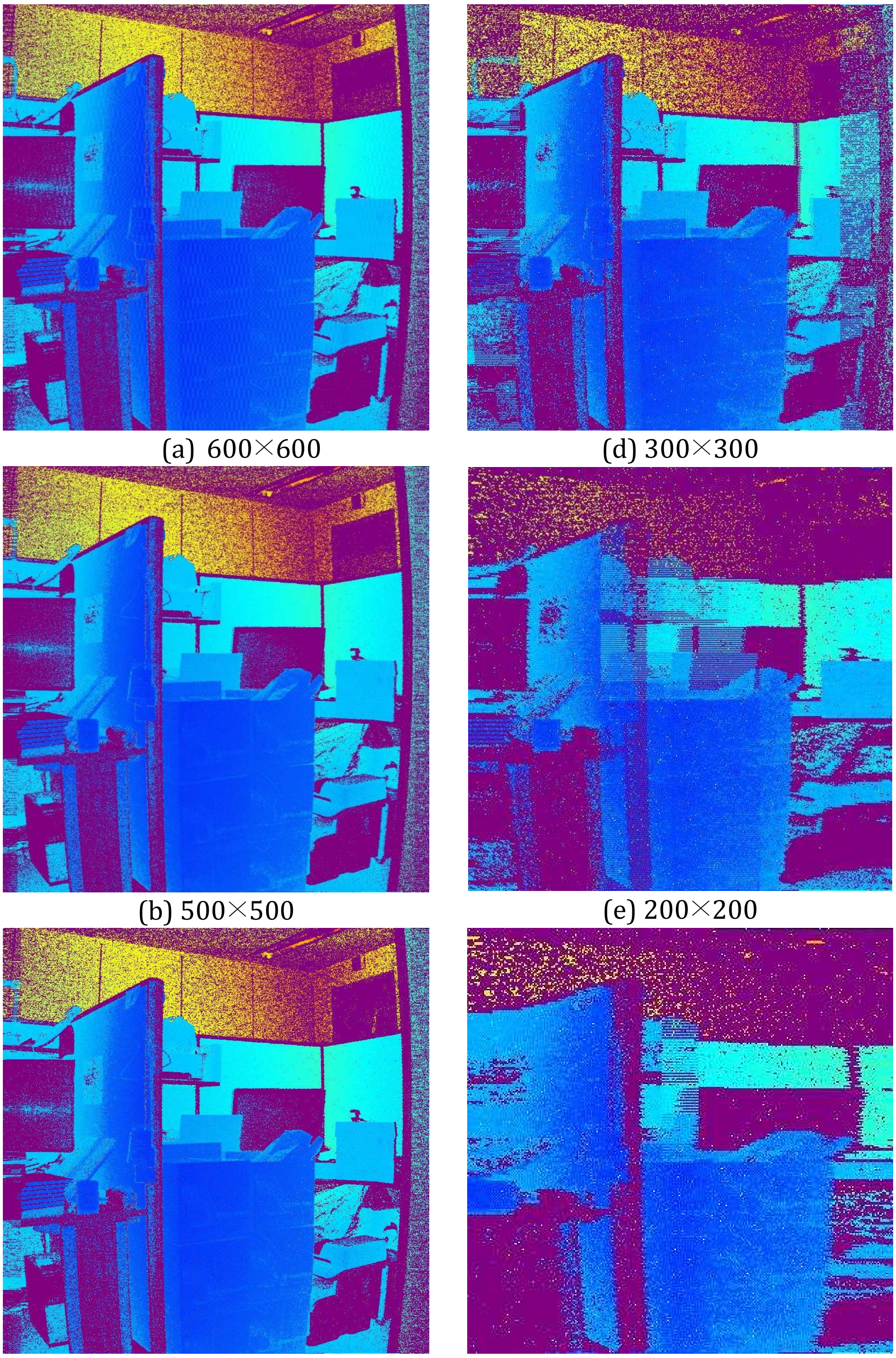
TABLE 3-IV
PARAMETER SETTINGS WITH THE PROPOSED SAMPLE LIDAR SYSTEM FOR A GIVEN FRAME RATE

Frame rate	Width	Height	Number of pixels per frame	Number of measurements
20	360	12	4,320	86,400
10	360	27	9,720	97,200
5	360	54	19,440	97,200
2	360	135	48,600	97,200
1	360	270	97,200	97,200
0.5	450	440	198,000	99,000

is adjusted to vary the frame rates in the first column to 20, 10, 5, 2, and 1. The fourth and last columns present the number of pixels per frame and the number of measurements, respectively. It should be noted that the number of measurements must be equal to or less than 100,000, which is the maximum number of updates. The results show that the proposed system can achieve nearly 100,000 measurements per second despite the fact that it only uses one emitter and detector pair. On the other hand, the Velodyne LIDAR sensors VLP-16, HDL-32E, and HDL-64E, which use 16, 32, and 64 channels, respectively, correspondingly achieve 300,000, 700,000, and 2,200,000 measurements per second. These results indicate that each channel for these sensors only provides approximately 20,000 measurements per second. Therefore, the proposed system shows much better performance and is also able to adjust the parameters to fit the desired frame rate.

3.4.3. Subjective Evaluation

Fig. 3.6 shows the experimental images provided by the proposed sample LIDAR system. Nine images were taken at various resolutions of 600×600, 500×500, 400×400, 300×300, 200×200, and 150×150. The results show that the images 600×600, 500×500, and 400×400 in size nicely capture the same FOV and clearly identify the objects at different distances. Meanwhile, as the resolutions decrease, the FOVs are gradually narrowed in the captured images of 300×300, 200×200 and 150×150. These images also contain boundary artifacts. These phenomena can be explained as follows. Consider the images of 400×400 and 300×300. If the FOV is 60,000×60,000, the moving steps will be 150 and 200, respectively. Recall the speed constraint in (3.22), which is derived from (3.6) and (3.7). Therefore, the moving step when obtaining a 400×400 image encounters (3.2), whereas the step when obtaining a 300×300 image does not. In practice, the scanners cannot exceed the maximum speed. Eventually, the FOV decreases. Furthermore, boundary artifacts arise because the scanners change the scanning directions in those areas and then require more time for scanning. In summary, the images show the visual impact of the derivations in this study. The message is that violating the conditions leads to a narrowing of the FOV and to boundary artifacts. In other words, derivations in (3.22) provide a sufficient condition to achieve a “good” image.



(a) 600×600 (d) 300×300

(b) 500×500 (e) 200×200

(c) 400×400 (f) 150×150

Fig. 3.6. Images according to various resolutions by the proposed sample LIDAR system.

3.4.4. Accuracy Evaluation

This subsection briefly discusses the accuracy of our LIDAR system based on two criteria: the vertical resolution and ranging accuracy. The vertical resolution is an important factor to define an object for general object detection applications. Table 3.5 reports the vertical resolutions of various LIDAR systems including our LIDAR prototype. The results show that our system outperforms the existing systems in terms of vertical resolution. By utilizing the high resolution of galvanometers, our LIDAR system can achieve 12μrad (=0.0007^{o}) vertical resolution. Moreover, the vertical resolution is likely to be adjusted for specific applications as shown in Section III-C. In particular, the resolutions are configured to achieve the desired frame rate and FOV.

The evaluation for the ranging accuracy of our LiDAR system is conducted on the obtained images in Fig. 3.6 of the original manuscript as follows. First, a box at center or close points is cropped. Second, six positions in the small box are selected and sixes 8×8 corresponding blocks are taken as shown in Fig. 3.7. Finally, for each small block, the mean and deviation are calculated. This simple experiment makes an assumption that distances in a small block are similar. Therefore, each position is marked with a mean and deviation which are considered to the measured distance and error, respectively. The detailed results are reported in Table 3.6. For each position, its mean and deviation of distances are reported. For example, the position 1 in the image 600×600 is at 3.407m with the error is 0.037m (or 3.7cm). On average, the error is about 0.055m or 5.5cm. To this end, the experimental results show that the average error is about 5.5cm, which is suitable for many practical applications.

TABLE 3.5
VERTICAL RESOLUTION COMPARISON OF VARIOUS LIDAR SYSTEMS

Model	Numbers of channels	Vertical Angle	Vertical Resolution
IBEO ALASCA [16]	4	3.2°	0.8°
LEDDAR 16M [17]	16	×	×
PUCK™ VLP-16 [19]	16	+15° to -15°	2°
HDL-32E [20]	32	+10° to -30°	1.33°
HDL-64E [21]	64	26.9°	0.4°
Riegl-VUX-1UAV [18]	1	×	×
Our LIDAR system	**1**	**41.2°**	**up to 0.0007°**

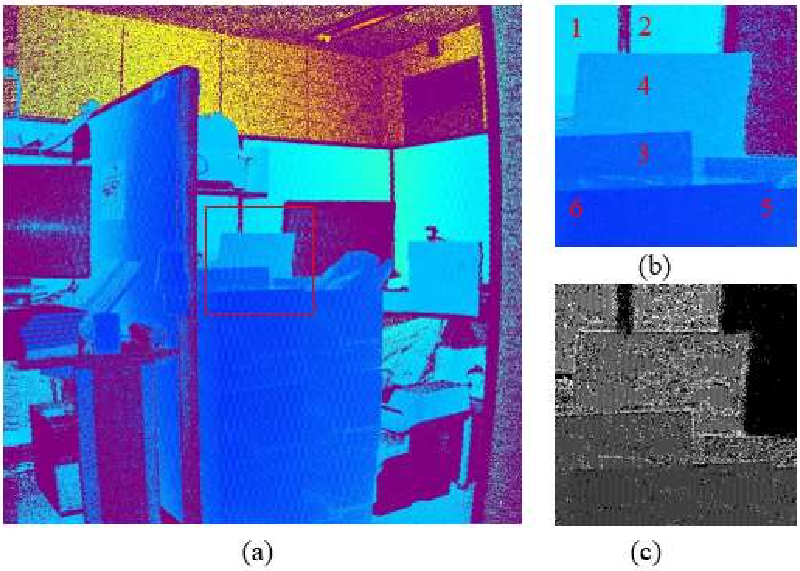


Fig. 3.7. An experiment setup for the ranging accuracy estimation. (a) The center-cropped position. (b) The cropped region of interest and positions. (c) The corresponding map.

TABLE 3.6
MEAN/DEVIATION OF DISTANCE MEASUREMENTS AT DIFFERENT POSITIONS AND RESOLUTIONS.

Image	Pos. 1 (m)	Pos. 2 (m)	Pos. 3 (m)	Pos. 4 (m)	Pos. 5 (m)	Pos. 6 (m)
600×600	3.409/ 0.037	3.481/ 0.048	2.934/ 0.046	2.982/ 0.052	2.246/ 0.072	2.088/ 0.043
500×500	3.432/ 0.035	3.493/ 0.048	2.927/ 0.056	3.011/ 0.060	2.226/ 0.065	2.078/ 0.062
400×400	3.436/ 0.041	3.469/ 0.068	2.959/ 0.057	3.019/ 0.049	2.266/ 0.068	2.101/ 0.089

Chapter 4: Sampling for Dual-Mirror LiDAR: Sampling Model and Algorithm

4.1. Introduction

Efficient and accurate sampling is eventually required to reduce the spatial resolution of such scanning systems. For a broad use in depth-data acquisition systems, a sampling method should have the following properties:

1. Perceptually capture groups or regions that generally reflect the global aspects of a depth image. Given a sample budget, a sampling method should be capable of capturing details in the object boundary while omitting details in the smooth areas. The definition of sampling is formulated to represent these properties for an enhanced understanding of the method and to facilitate the comparison of different techniques.
2. Being computationally efficient implies having a computational complexity of $O(n)$, where n is the number of image pixels. For practical use, sampling methods must run at a speed that is similar to that of gradient computation or other low-level visual-processing methods, implying approximately linear time with low constant factors.

Uniform random or grid sampling is the most straightforward approach that is highly efficient and satisfies the second property. However, this method is generally inefficient in capturing perceptually critical non-local properties of an image such as the object boundary. Therefore, reconstruction quality is relatively poor and does not satisfy the quality requirements in several applications. On the other hand, a non-uniform sampling strategy can significantly enhance the signal to noise ratio (SNR) [26]-[30]. An effective sampling approach is proposed in [26] under the assumption that the global properties of a depth image (e.g. its gradients) are available. This method efficiently samples the depth image along edges. Unfortunately, the gradient is not available prior to sampling, which renders the assumption unrealistic in practical uses. Nonetheless, the sampling method provides strong evidence of the feasibility of a more effective sampling method to capture the global aspects of an image; thereby enhancing the SNR or reconstruction quality. To obtain the global properties, several sampling schemes are proposed in [28]-[30] according to which the sampling operations are performed in two stages. In the first stage, the scheme adopts a uniform strategy using only a part of the sample budget. The sampled data are used to reconstruct an image and then extract global information such as gradient map [28] and object saliency map [29], [30]. Non-uniform sampling approaches are used in the refinement stage, and then the sampling result is merged into those of the first one.

Although previous two-step sampling methods significantly enhance the SNR, they exhibit two drawbacks. First, they invoke an intermediate reconstruction that is complicated, which makes it challenging to reconstruct an image in real time even though numerous approaches for efficient reconstruction such as convex optimization and greedy methods [26], [28], [33]-[39] have been extensively studied. Second, an irregular sampling pattern usually requires additional storage space or transmission bandwidth, which must be included in the budget of samples and therefore reduces the number of feasible samples. These two challenging issues limit the use of a two-step approach to practical applications such as data acquisition or laser measurement systems, which strictly require an efficient sampling method.

To address the above two drawbacks, this study proposes a new mathematical formulation of the constraints for a practical sampling method in a LiDAR system. Based on the proposed mathematical formulation, it is shown that existing two-stage sampling approaches are not suitable for a practical LiDAR system. Therefore, this study presents a novel sampling method to efficiently perform non-uniform random sampling. The proposed algorithm extends the two-step method in the previous designs [28]-[30] to reduce the computational complexity and the requirements of additional storage or bandwidth while still achieving high SNR quality. The proposed method performs uniform sampling at the pilot stage and non-uniform sampling at the refinement stage. However, unlike in the previous methods, the proposed technique efficiently derives non-uniform sampling based on the gradient of the downsampled image. Consequently, the proposed method follows implicit global properties notwithstanding decision-making using a greedy approach. More critically, the proposed method for computing the gradient and refinement-sampling map is substantially faster than other methods because it does not require intermediate reconstruction. Consequently, it is computationally efficient with $\mathrm{O}(n)$ complexity for n image pixels. In addition, the proposed method reduces the requirement of additional memory (or bandwidth) to store (or transmit) the sampling pattern. To this end, the proposed method outperforms grid sampling by at most 5.92 dB. As a result, the proposed sampling achieves a reconstructed quality that is similar to the optimal sampling in the previous design, while substantially reducing the computation time and memory requirements.

4.2. Sampling Model for Dual-Mirror LiDAR

The conventional sampling model in Chapter II is intuitive. However, it over-simplifies a practical LiDAR system because a timing constraint is not fully considered. The reason is that the derivation of an optimal sampling pattern is time-consuming, which consequently increases the overall data acquisition time even though the number of sampling points is reduced. Furthermore, a practical LiDAR system demands the minimal use of hardware resources such as memory footprint. To address these issues in the design of a practical LiDAR system, this section discusses the constraint required by a LiDAR system and presents a modified formulation of the sampling problem discussed in the previous section.

4.2.1. Timing constraints

A LiDAR system usually operates by performing multiple point-wise measurements in a FOV. A block diagram of a LiDAR is illustrated in Fig. 4.1. A typical measuring procedure of the LiDAR system is described as follows. A controller in the LiDAR system starts by computing a target location in the FOV, which requires a computation time t^{pos}. In the next step, the target position is transmitted to a mechanical scanner that controls motors and mirrors to direct the emitted light. This step requires the communication and motor control time, which is denoted by t^{scan}. After the mirror is aimed at the target, the laser diode in the LiDAR system emits a laser beam in time t^{emit}. Next, the LiDAR waits until the laser reaches an object and its reflected signal arrives at a photodetector.

The time interval between the emitted and detected signals is generally referred to as time of flight (TOF) and is denoted as t^{TOF}. Finally, the measurement of t^{TOF} is converted to an electric signal and transmitted to the optical device controller that calculates the TOF from the signal. This time is denoted by t^{calc}. In the last step, the result is transmitted to the main controller that reads the signal in time t^{read}. For a given position, q_k, in an FOV, let t_k denote the time required to measure its distance. Therefore, t_k is a function $f(.)$ of the variables t^{pos}, t^{scan}, t^{emit}, t^{TOF}, t^{calc}, and t^{read}. Obviously, the upper bound of t_k is the summation of all variables when all steps are operated in a sequential manner. Meanwhile, the lower bound of t_k is the maximum among all variables assuming that all steps are operated in a pipelined manner. To this end, t_k must satisfy both upper and lower bounds.

$$t_k = f\left(t_k^{pos}, t_k^{scan}, t_k^{emit}, t_k^{TOF}, t_k^{calc}, t_k^{read}\right) \tag{4.1-a}$$

$$t_k \geq \max\left\{t_k^{pos}, t_k^{scan}, t_k^{emit}, t_k^{TOF}, t_k^{calc}, t_k^{read}\right\} \tag{4.1-b}$$

$$t_k \leq t_k^{pos} + t_k^{scan} + t_k^{emit} + t_k^{TOF} + t_k^{calc} + t_k^{read} \tag{4.1-a}$$

Three variables t_k^{emit}, t_k^{calc} and t_k^{read} are likely to be fixed as their operations are the same for all measurements. Therefore, t_k usually depends on three remaining variables, t_k^{pos}, t_k^{scan}, and t_k^{TOF}.

The derivation of a sampling pattern affects t_k^{pos}, which indicates the time to determine the sampling point. If a sampling pattern is predetermined, it does not require the time to compute a target location (or $t_k^{pos} = 0$ for all k). On the other hand, the derivation of a complex sampling pattern might require a considerable amount of time so that t_k^{pos} becomes very large.

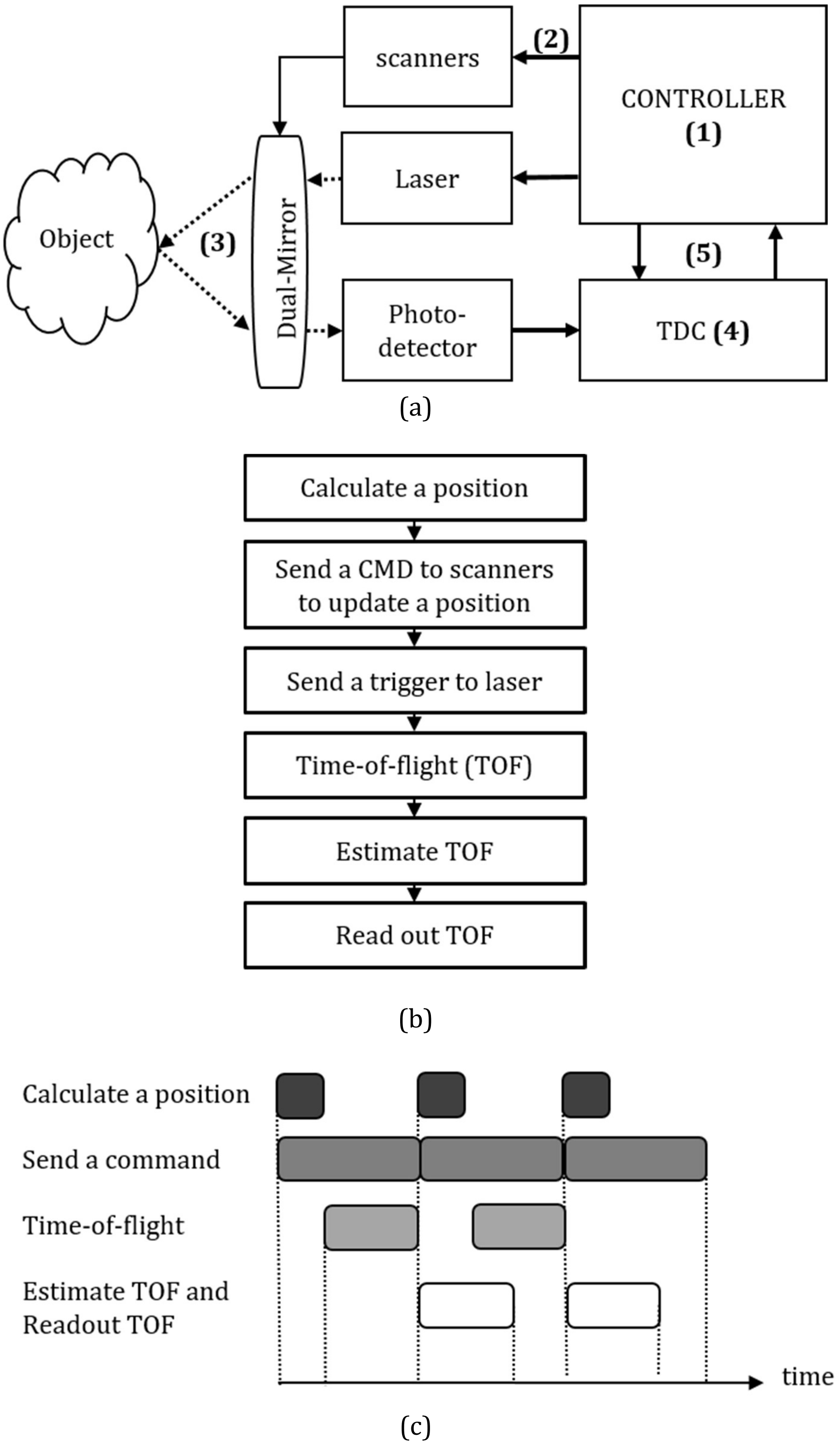


Fig. 4.1. A typical LiDAR system, (a) Block diagram; (b) Dataflow to measure a distance; and (c) Pipelined timing schedule to measure a distance.

When the LiDAR system measures at M locations corresponding to indexes $i_1, \ldots, i_{M,}$ the total time is expressed as follows:

$$t_{\Sigma}^{M} = \sum_{k=1}^{M} f\left(t_{i_k}^{pos}, t_{i_k}^{scan}, t_{i_k}^{emit}, t_{i_k}^{TOF}, t_{i_k}^{calc}, t_{i_k}^{read}\right). \tag{4.2}$$

Given a time budget T for scanning M locations, the following constraint must be satisfied:

$$t_{\Sigma}^{M} = \sum_{k=1}^{M} f\left(t_{i_k}^{pos}, t_{i_k}^{scan}, t_{i_k}^{emit}, t_{i_k}^{TOF}, t_{i_k}^{calc}, t_{i_k}^{read}\right) \leq T. \tag{4.3}$$

In practice, a LiDAR captures an image frame by frame, and therefore, the time budget T is usually set for a single frame. For example, T is 33 milliseconds if 30 frames are captured for every second. For the two-stage sampling in [13], the $M/2$ samples in the first stage are predefined so that they do not require time to calculate their locations. Meanwhile, the $M/2$ remaining samples require complex computation to derive their patterns, which results in longer computation times to generate the pattern (i.e., 20 seconds as reported in [28]). Therefore, this sampling does not satisfy the timing constraint:

$$\begin{aligned} t_{\Sigma}^{M} &\geq \sum_{k=1}^{M} max\left(t_{i_k}^{pos}, t_{i_k}^{scan}, t_{i_k}^{emit}, t_{i_k}^{TOF}, t_{i_k}^{calc}, t_{i_k}^{read}\right) \geq \sum_{k=1}^{M} t_{i_k}^{pos} \\ &\geq 20s. \end{aligned} \tag{4.4}$$

Obviously, t_{Σ}^{M} becomes much larger than the time budget T that is, in general, a fraction of second in practice. Consequently, the two-stage sampling in [28] cannot be used for practical LiDAR sampling with the timing constraint considered.

4.2.2. Memory-space constraint

This subsection presents an analysis of memory and bandwidth in a LiDAR system. Consider the practical case in which LiDAR is integrated in a system and a sampling method must satisfy the memory or bandwidth constraint of the system. In particular, let C denote the available memory capacity (or an available transmission bandwidth) for storing (or transmitting) LiDAR data. Meanwhile, let $G(.)$ be a function that represents the amount of stored/transferred data. Therefore, $G(.)$ depends on the sampling budget ξ, depth resolution n, and a sampling pattern S.

$$G(\xi, n, S) = G(M, n, i_1, \ldots, i_M\}. \tag{4.5}$$

In addition, the amount of data $G(\xi, n, S)$ must satisfy the following inequality:

$$G(\xi, n, S) \leq C. \tag{4.6}$$

The straightforward derivation of $G(\xi, n, S)$ for a given non-uniformly random sampling (i.e., two-stage sampling in [28]) is described as follows. Because each pixel in b consists of n bits, the amount of data becomes $n \times \xi \times N$ bits. In addition, the sampling pattern S is also stored and/or transmitted because it is non-uniform and irregular. Because one bit is necessary for each pixel in the input image of size N, the amount of data for S is N. Therefore, the total amount of data $G(\xi, n, S)$ is derived as follows:

$$G(\xi, n, S) = n \times \xi \times N + N. \tag{4.7}$$

Generally, for a practical capturing device, the memory space inside the device and/or the transmission bandwidth to the external system is limited. Therefore, it is necessary to select the sampling ratio to satisfy the memory and/or bandwidth requirement. By combining (4.6) and (4.7), with a given memory and/or bandwidth capacity C, the amount of data $G(\xi, n, S)$ must satisfy the following inequality:

$$n \times \xi \times N + N \leq C. \tag{4.8}$$

From (3.8), the target sampling ratio ξ is limited by the available memory space C, resolution n, and image size N:

$$\xi \leq \frac{C - N}{n \times N}. \tag{4.9}$$

A new terminology target compression ratio, χ is defined to represent the ratio of the size of the available memory space C to the size of the input image $(n \times N)$:

$$\chi = \frac{C}{n \times N}. \tag{4.10}$$

The compression ratio represents the extent to which the original data should be compressed to satisfy the available memory capacity. The relationship between the sampling ratio ξ and compression ratio χ can be obtained from (4.9) and (4.10) and can be expressed as follows:

$$\xi \leq \chi - \mathrm{N}/(n \times N). \tag{4.11}$$

Uniform grid sampling does not require the storage of the sampling map S because the pattern is fixed. Therefore, the required memory space $G(\xi, n, S)$ is modified as follows:

$$G(\xi, n, S) = n \times \xi \times N \leq C. \tag{4.12}$$

where the second term in (4.8) is removed. In this case, the sampling ratio becomes

$$\xi \leq \frac{C}{n \times N} = \chi. \tag{4.13}$$

Example 1: A depth image of size 512×512 with 8-bit resolution (data size = 256KB) is sampled and stored in memory of size 64 KB (i.e., N = 32 KB, n = 8, and C = 64 KB). This implies that 32 KB (= 64 KB - 32 KB) is used to store the sampled depth map b. If b is obtained from the non-uniform sampling, the sampling ratio ξ is limited to the following value:

$$\xi = \frac{C - N}{n \times N} = \frac{64 - 32}{8 \times 32} = 12.5\%. \tag{4.14}$$

On the other hand, the uniform grid uses 64 KB to store b, thereby resulting in the following value of ξ

$$\xi = \frac{C}{n \times N} = \frac{64}{8 \times 32} = 25\% . \tag{4.15}$$

This result demonstrates that the uniform grid stores twice as many samples as the non-uniform grid does. This implies that the uniform grid is likely to achieve higher image quality than that of the non-uniform grid when memory space is limited, which is true in numerous real-world applications.

4.2.3. New sampling problem with constraints

Based on those two constraints discussed in the previous two subsections, Conventional sampling problem is modified as follows:

Problem 2 *(New sampling problem for LiDAR)*: The sampling problem is to derive $\{i_1, \ldots, i_M\}$ to minimize the following objective function:

$$\min_{i_1,\ldots,i_M} \frac{1}{N}\sum_{j=1}^{N}\left(x_j - \tilde{x}_j\right)^2. \tag{4.16}$$

subject to the following two constraints:

a. (timing constraint)

$$t_{\Sigma}^{M} = \sum_{k=1}^{M} f\left(t_{i_k}^{pos}, t_{i_k}^{scan}, t_{i_k}^{emit}, t_{i_k}^{TOF}, t_{i_k}^{calc}, t_{i_k}^{read}\right) < T$$

b. (memory-space constraint)

$$G(M, n, i_1, \dots, i_M) \ \leq \ C$$

where $x_1, \dots, x_N$ are the real values and $\widetilde{x_1}, \dots, \widetilde{x_N}$ are values that are estimated from M measurements $x_{i_1}, \dots, x_{i_M}$. Problem 2 is modified into two variations by ignoring either the timing or memory constraint. Problem 2a is the same problem as Problem 2 with the removal of the memory-space constraint and Problem 2b is derived from Problem 2 by removing the timing constraint.

While the use of non-uniform sampling enhances the image quality of the reconstructed image, it involves considerable computational complexity and additional memory space. This illustrates the trade-off between higher image quality and faster execution time/larger memory requirement. This study proposes a novel algorithm that improves image quality while reducing computational complexity and memory requirements.

4.3. Proposed sampling Algorithm and Its Properties

4.3.1. Downsampling and *k*-NN expanding operator

To reduce the required memory space, the proposed algorithm attempts to reduce the second term on the right side in (4.7). The concept is explained using an example illustrated in Fig. 4.2. Fig. 4.2(a) illustrates an "Aloe" image in the Middlebury benchmarks [11], [12]. Down-sampling this image by 3:1 yields the image illustrated in Fig. 4.2(c), which is one ninth of the original image.

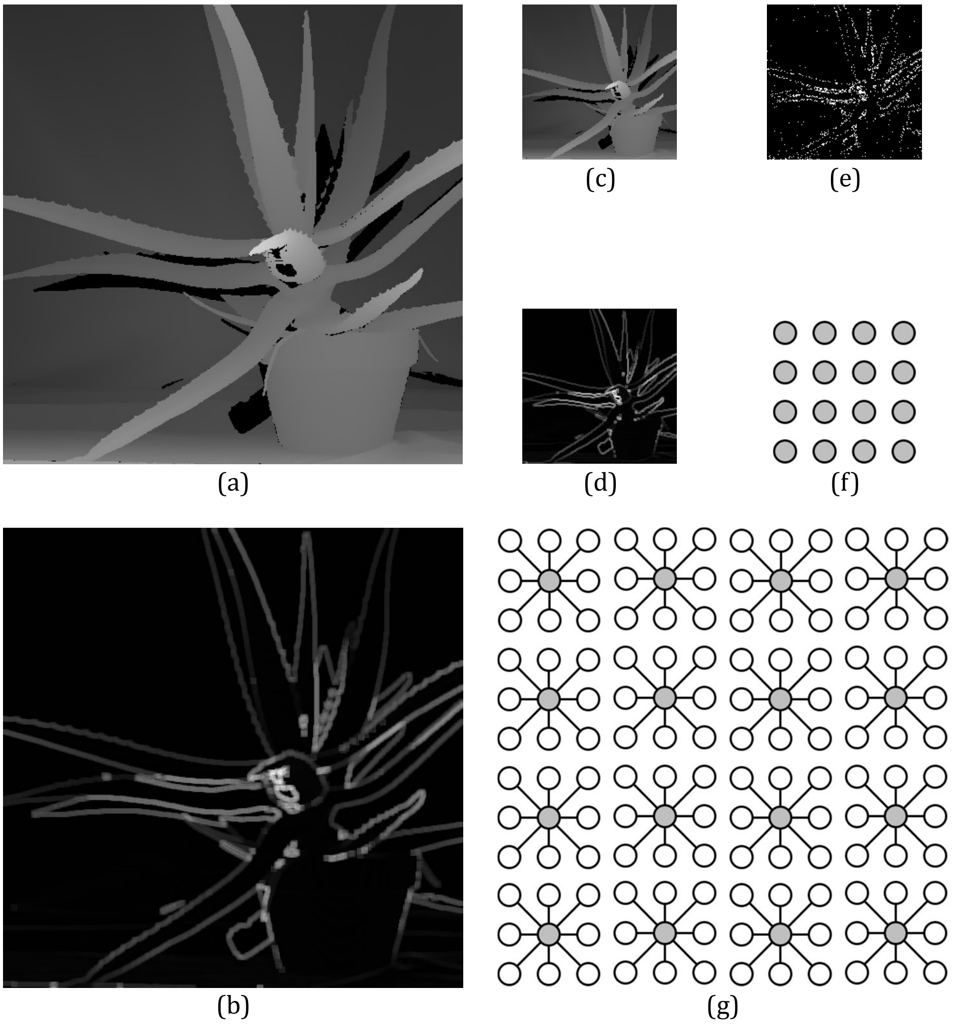


Fig. 4.2. Example of a down-sampling operator and its graph representation. (a) –(b) original "Aloe" image and its gradient; (c)-(d) a down-sampled image (1:3), its gradients and its gradient-based sampling map. (e)-(f) an example of a 4 × 4 down-sampled image and its corresponding 12 × 12 image, which can be considered as sixteen disjointed sets represented by 16 representatives.

Notwithstanding the down-sampling, the smaller image perceptually captures groups or regions. These image characteristics are illustrated in Figs. 4.2(b) and (d), which present the gradient images of Figs. 4.2(a) and (c), respectively. It should be also noted that in the downsampled image, object boundaries and textured patterns are perceptually captured in most regions. Fig. 4.2(e) demonstrates a sampling map generated from the "downsampled" gradient image in Fig. 4.2(d) in which sampled points are densely located in objects boundary. It suggests that a “downsampled” sampling map in Fig. 4.2(e) can be considerred as an indicator to detect texture areas in a scene. Fig. 4.2(f)-(g) show an example of a graph respresentation to visualize this indicator described above. Fig. 4.2(f) shows a set of representatives, playing a role of indicator in Fig. 4.2(e). Each point in Fig. 4.2(g) is mapped to a resprensentative point in Fig. 4.2(f) marked with gray. This mapping allows to copy a characteristics from Fig. 4.2(f) to Fig. 4.2(g). For example, if a point in Fig. 4.2(f) is a texture area, its mapping point and neighboring points in Fig. 4.2(g) are likely to be in a texture area. This is named as a k-NN expanding operator explained in Fig. 4.3.

The proposed algorithm captures the gradient information ∇x from the downsampled image and then derives sampled depth map b from the downsampled gradient information. If the image is downsampled by 3:1 in both the horizontal and vertical directions, the size of the sampled depth map is reduced to 1/9 of the original image. The second term of the right side in (4.7) is also decreased to $N/9$. On the other hand, the down-sampling results in aliasing artifact in the highly textured or boundary regions. This implies that image quality will likely degrade because of the loss of information by down-sampling. Increasing the number of samples facilitates the capture of more information (e.g., details of the boundary object or highly textured areas). To reduce this artifact, the proposed algorithm uses additional samples in the textured region. In case that a pixel is in a highly textured region, its neighboring pixels are also likely to be in the highly textured region because of the non-local

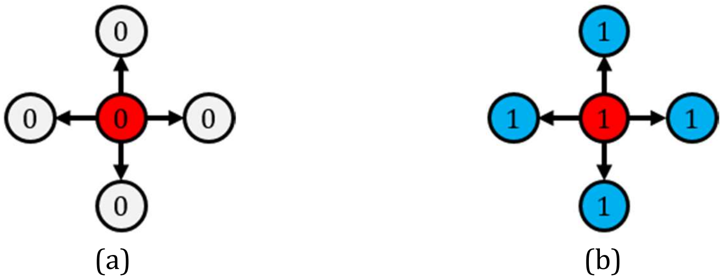


Fig. 4.3. An example of the proposed expanding operator.

image characteristic. Utilizing this characteristic, the proposed algorithm selects samples in the neighbors of the sampled data observed in a highly-texture region. To achieve this, the algorithm uses the *k*-NN expanding operation, which is explained in Fig. 4.3.

Example 2: Fig. 4.3 illustrates an example of the proposed expanding operation. Each central node is representative of its respective set, which includes the four of its neighbors. In Fig. 4.3(a), the central node is marked with label 0, indicating that it is in a smooth area. The operator predicts that the four neighbors of this node are also in the smooth area by assigning zero to their labels. On the other hand, the central node in Fig. 4.3(b) is marked as one, indicating that it is in a highly textured area or on the boundary of an object. The proposed algorithm sets one to its neighbors and predicts them to be in a highly textured area.

The sampling map is constructed as follows: The sampling map consists of all the representatives and their neighbors marked by non-zero labels. In Fig. 4.3(a), the four neighbors have not been included in the sampling map as they are marked zero. Meanwhile, the four neighbors in Fig. 4.3(b) are added to the sampling set. An intuitive approach is to undertake dense sampling in highly textured areas and sparse sampling in smooth areas.

4.3.2. Proposed Sampling Algorithm with *k*-NN Expanding

The proposed sampling procedure consists of two stages: a pilot stage to obtain a coarse regular sampling map and a refinement stage to enhance the sampling map. The pilot stage selects the partial ratio α $(0 < \alpha < 1)$ from the budget. A uniform grid sampling is used in this stage, resulting in sampling map $S^{(1)}$ with $\alpha \times \xi \times N$ non-zero elements. The sampling period is identical in both the horizontal and vertical directions such that it is straightforwardly defined by step as follows:

$$step = \sqrt{\frac{1}{\alpha \times \xi}} \ . \tag{4.17}$$

For a specified image, let W and H denote its width and height, respectively. Then, N=W×H can be obtained. In uniform sampling with the step expressed in (4.17), S_{ii}=1 if and only if the index i satisfies the following condition:

$$i = \lfloor i_H \times step \rfloor \times W + \lfloor i_W \times step \rfloor. \tag{4.18}$$

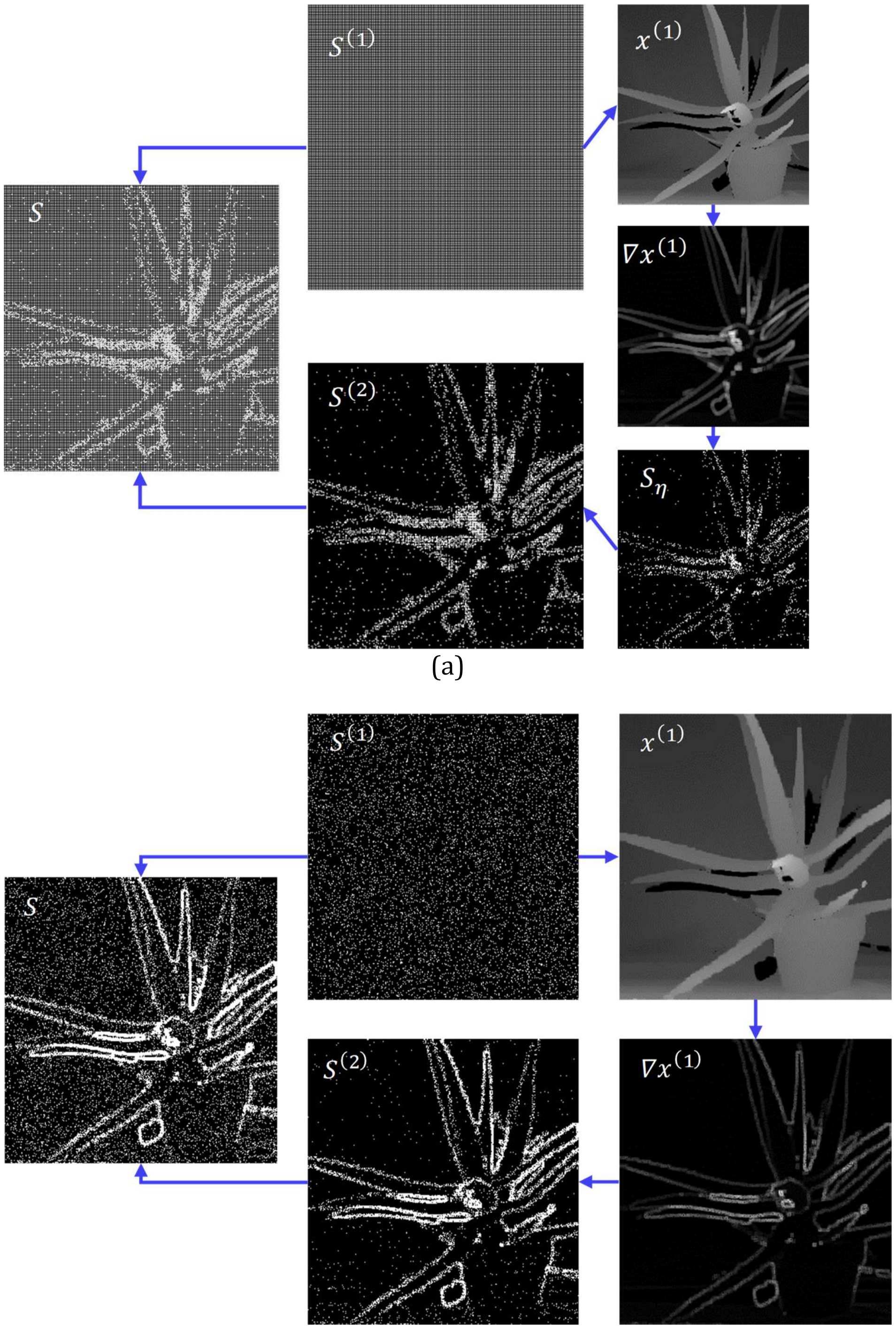
where $\lfloor . \rfloor$ represents the floor operation, and i_H and i_W are integer numbers such that the corresponding $\lfloor i_H \times step \rfloor$ and $\lfloor i_W \times step \rfloor$ are the coordinates of the pixel in the 2D image. Apparently, i_H and i_W satisfy the following conditions, $i_H \in \left\{1,2,\dots,\left\lfloor \frac{H}{step} \right\rfloor\right\}$ and $i_W \in \left\{1,2,\dots,\left\lfloor \frac{W}{step} \right\rfloor\right\}$.

Given $S^{(1)}$, the sampled depth map $b^{(1)}$is derived as follows:

$$b^{(1)} = S^{(1)}x. \tag{4.19}$$

and its corresponding downsampled map $x^{(1)}$ is downsampled by the step in (4.17). The size of $x^{(1)}$ corresponds to $M_\alpha = \alpha \times \xi \times N$, where M_α represents the size of $x^{(1)}$ hereafter in this study. It must be noted that this is the main variation between the proposed study and that in [13]. The map $x^{(1)}$

in the proposed study is straightforwardly a downsampled image of a substantially smaller size than that of the original image. Fig. 4.4 illustrates a visual comparison between the proposed sampling scheme and that of [28]. Fig. 4.4(a) illustrates the proposed sampling scheme where $x^{(1)}$ is directly derived by down-sampling operation. Meanwhile, the $x^{(1)}$ in [28] is the reconstructed image derived from $b^{(1)}$ as illustrated in Fig. 4.4(b).



(a)

(b)

Fig. 4.4. Comparison of two sampling algorithms: (a) the proposed sampling algorithm and (b) two-stage algorithm in [28].

In the second stage, the downsampled image $x^{(1)}$ is used as a guide to compute the gradient $\nabla x^{(1)}$. The gradient map $\nabla x^{(1)} = \left[a_1, \dots, a_{M_\alpha}\right]^T$ is derived as explained in Chapter 2. By formulating the optimization problem of (2.6), the optimal probability and sampling map S_η are obtained. By applying the k-NN expanding operator to S_η, where $x^{(1)}$ is used as the set of representatives for the original image, $S^{(2)}$ can be obtained. The sampling ratio η is separate from ξ, and it is derived as follows.

Recall that the downsampled image $x^{(1)}$ is the set of representatives for the original image. Based on the k-NN expanding operator in Section IV-B, for each representative, k neighboring points are added into the final sampling map. Note that the number of remaining samples is $(1 - \alpha) \times \xi \times N$ because $M_\alpha = \alpha \times \xi \times N$ samples are used in the pilot stage. Using the k-NN expanding operator, each of the representatives are extended to their k neighboring pixels. Consequently, the number of representatives added in this refinement stage can be derived by dividing the remaining samples by k and can be expressed as follows:

$$\frac{(1 - \alpha) \times \xi \times N}{k}. \tag{4.20}$$

This results in the sampling ratio η for the downsampled image $x^{(1)}$, which can be expressed as follows:

$$\eta = \frac{\frac{(1 - \alpha) \times \xi \times N}{k}}{M_\alpha} = \frac{(1 - \alpha) \times \xi \times N}{k \times \alpha \times \xi \times N} = \frac{1 - \alpha}{k \times \alpha}. \tag{4.21}$$

Algorithm 4.1: Proposed Sampling Algorithm

```
Input: N, $\xi, b, k,\ \alpha$
Output: $S$

Pilot Stage
Obtain $S^{(1)}$ by a uniform grid sampling with a budget ratio $\alpha \times \xi$.
Compute $b^{(1)}$ from $S^{(1)}$.
Compute $x^{(1)}$ from $b^{(1)}$.

Refinement Stage
Compute $\nabla x^{(1)}$.
Conduct the optimal sampling on $\nabla x^{(1)}$ to obtain the set $S_\eta$ with the sampling ratio $\eta = \frac{1 - \alpha}{k \times \alpha}$.
Compute $S^{(2)}$by k-expanding of $S_\eta$.
Compute $S = S^{(1)} + S^{(2)}$.
```

Fig. 4.5. Proposed sampling algorithm with k-NN expanding operator.

Given S_η, the proposed k-NN expanding operator is applied for the derivation of the refined sampling map $S^{(2)}$. As each representative expands only to its neighbors, $S^{(1)}$ and $S^{(2)}$ are exclusive. Therefore, the final sampling map S, is obtained by S=$S^{(1)}$+$S^{(2)}$. The algorithm is summarized in Fig. 4.5.

Example 3: Assume that the target sampling ratio is 20% ($\xi = 0.2$) and half of the budget is used at the first stage ($\alpha = 0.5$). Then, the size of the downsampled image in the first stage includes 10% of the original image's size. The refinement stage applies the straightforward four-node pattern (k=4*).* To obtain the remaining 10% of the samples, 2.5% of the representatives must be selected from the downsampled image. From (4.20), the sampling ratio η in the downsampled image is selected as 25%.

$$\eta = \frac{1 - 0.5}{4 \times 0.5} = 0.25. \tag{4.22}$$

4.3.3. Example with Synthetic Data

An example of the implementation of the proposed sampling algorithm is illustrated in Fig. 4.6. The synthetic image has 25 × 25 pixels, and the sampling ratio is 20% (ξ=0.2). Fig. 4.6(a) illustrates the sampling map $S^{(1)}$ when 10% is used by a uniform grid sampling (α=0.5). From (4.17), the sampling period is derived as $step$=$\sqrt{10}$. Without loss of generality, the starting index is selected as 2 (=1+$step$/2) such that a pattern is derived (see Fig. 4.6(a)). Therefore, the 8 × 8 downsampled image $x^{(1)}$ is obtained in Fig. 4.6(b). For convenience, $x^{(1)}$ exhibits a straightforward shape having a smooth 4 × 4 square in its center. The gradient $\nabla x^{(1)}$ is illustrated in Fig. 4.6(c). The refinement stage applies the straightforward 4-NN pattern (k=4) in Section IV. 16 representatives are then selected from the available 8 × 8 gradient image. The feasible solution of (2.6) is displayed in set S_η in Fig. 4.6(d), wherein the black pixels represent highly-textured areas, and the white pixels indicate smooth ones. It is random in a general case. However, a feasible solution is selected as an example because the randomness is not likely to be true if the number of samples is substantially small. The use of a random solution will likely result in bias such that the selected samples are not located on the object boundary. Finally, an expanding operator is applied to S_η and then the derived sampling $S^{(2)}$ is merged to $S^{(1)}$ to form the final sample S, which is illustrated in Fig. 4.6(e). In S, the white squares are omitted while the remaining ones are sampled. Meanwhile, the uniform grid sampling pattern is illustrated in Fig. 4.6(f) as the entire sampling budget is used to derive the pattern. The sampling pattern in Fig. 4.6(e) preserves more points in the boundary area than that in Fig. 4.6(f). Note that the difference

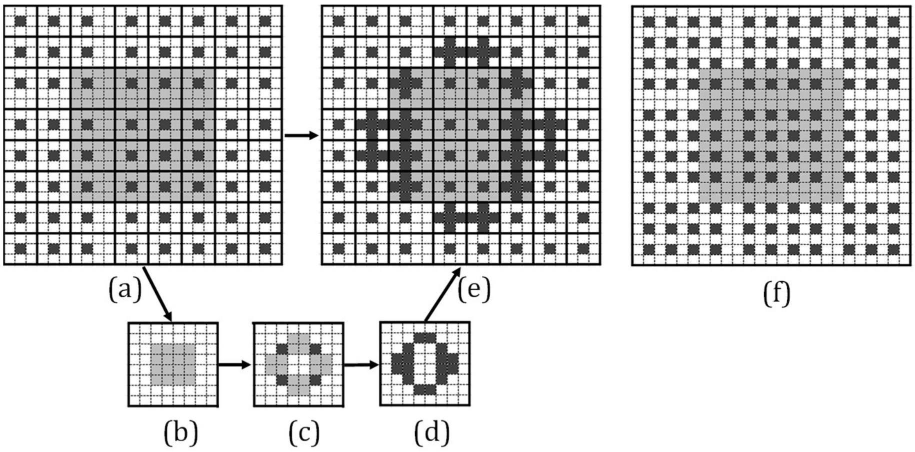
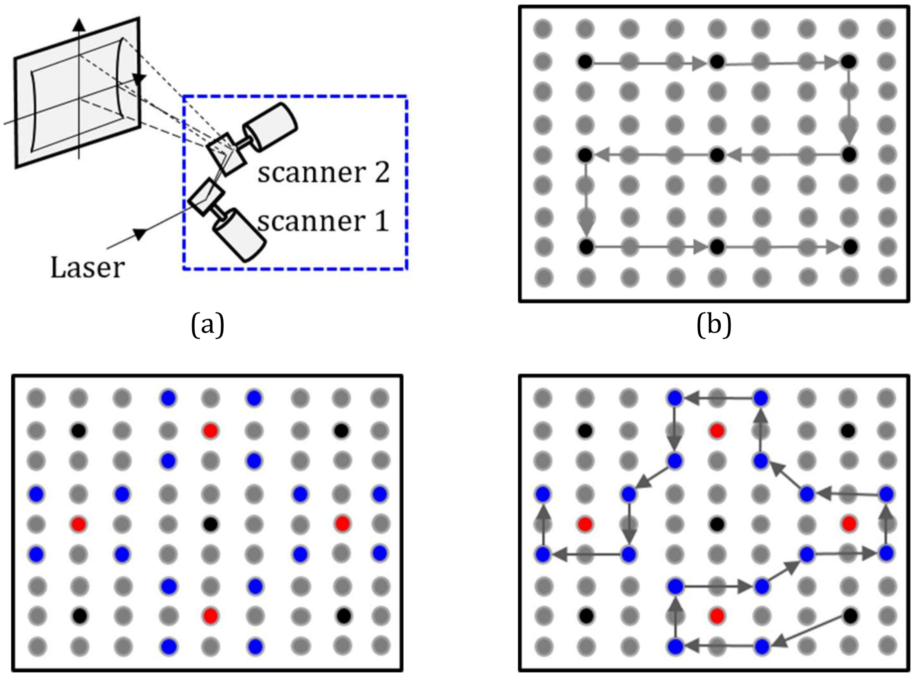


Fig. 4.6. Example of the operation results by the proposed algorithm: (a) $S^{(1)}$ (b) $x^{(1)}$ (c) $\nabla x^{(1)}$ (d) S_η (e) $S = S^{(1)} + S^{(2)}$ and (f) the grid sampling.

between $S^{(1)}$ and $S^{(2)}$ is that $S^{(1)}$ is the representative obtained from $S\eta$ and $S^{(2)}$ represents the pixels expanded from $S^{(1)}$, which does not include $S^{(1)}$. For example in the figure below, $S^{(1)}$ represents the shaded pixels in Fig. 4.6(a) and $S^{(2)}$ represents the neighboring pixels around the shaded pixels in Fig. 4.6(e). Because $S^{(2)}$ does not include the shaded pixels, $S^{(2)}$ is exclusive to $S^{(1)}$.

Fig. 4.7 demonstrates a synthetic example in which our two-stage sampling method is used in a LiDAR. At the first stage, LiDAR scanners in Fig. 4.7(a) scans a grid sampling pattern as shown Fig. 4.7(b) in which sampled points are marked with black (i.e. nine black points in this case). Fig. 4.7(c) shows the way to generate a sampling pattern for the second stage. At first, a gradient map is computed, and four points with big gradients are selected out of nine points and marked with red. For each red point, its four neighbors are scanned in the second stage and marked with blue. Finally, the LiDAR scanners scans all blue points. All colored points in Fig. 4.7(d) forms measured map.



(a) (b) (c) (d)

Fig. 4.7. An example of the operation results by the proposed algorithm: (a) $S^{(1)}$ (b) $S = S^{(1)}+ S^{(2)}$ and (d) Sampling pattern.

4.3.4. Proposed sampling algorithm with interpolation

In Algorithm 4.1, the gradients are computed on a very small down-sampled image $x^{(1)}$ to find the optimal sampling map $S\eta$, which significantly reduces the time complexity compared to [28]. However, this may lead to finding an incorrect $S\eta$ as a large information of the image might be lost in down-sampling. To address this problem, an interpolation-based two-stage algorithm is proposed by modifying two steps in lines 9 and 10 of the Algorithm 4.1 to derive the sampling $S^{(2)}$. First, instead of obtaining a set $S\eta$ in line 10 of Algorithm 4.1, a probability p is obtained from $p^{(1)}$. First, a "downsampled" probability map $p^{(1)}$ is computed from $\nabla x^{(1)}$. Next, a probability map p is derived from $p^{(1)}$ by interpolation. Finally, $S^{(2)}$ is derived from p. Algorithm 2 is summarized in Fig. 4.8.

Fig. 4.9 demonstrates a visual explanation of interpolation-based sampling algorithm in LiDAR. At the stage, LiDAR naturally does a grid sampling to obtain a "downsampled" image $x^{(1)}$ of an FOV in step (1). Downsampled gradient $\nabla x^{(1)}$ and its corresponding probability $p^{(1)}$ are derived in steps (2) and (3). Because a large information of the image might be lost in down-sampling, the proposed algorithm computes a probability map p from $p^{(1)}$ by interpolation in step (4). It should be noted that p captures texture regions of an image, results in an find sampling map $S^{(2)}$ having many sampling points in object boundaries.

Algorithm 4.2: Proposed Sampling Algorithm (Interpolation)

```
Input: $N, \xi, b, k, \alpha$
Output: $S$

Pilot Stage
Obtain $S^{(1)}$ by a uniform grid sampling with a budget ratio $\alpha \times \xi$.

Compute $b^{(1)}$ from $S^{(1)}$.
Compute $x^{(1)}$ from $b^{(1)}$.

Refinement Stage
Compute $\nabla x^{(1)}$.
Compute down-sampled probability $p^{(1)}$ from $\nabla x^{(1)}$.
Compute probability $p$ from $p^{(1)}$ by a linear interpolation.
Obtain $S^{(2)}$ from $p$.
Compute $S = S^{(1)} + S^{(2)}$.
```

Fig. 4.8. Proposed sampling algorithm with interpolation.

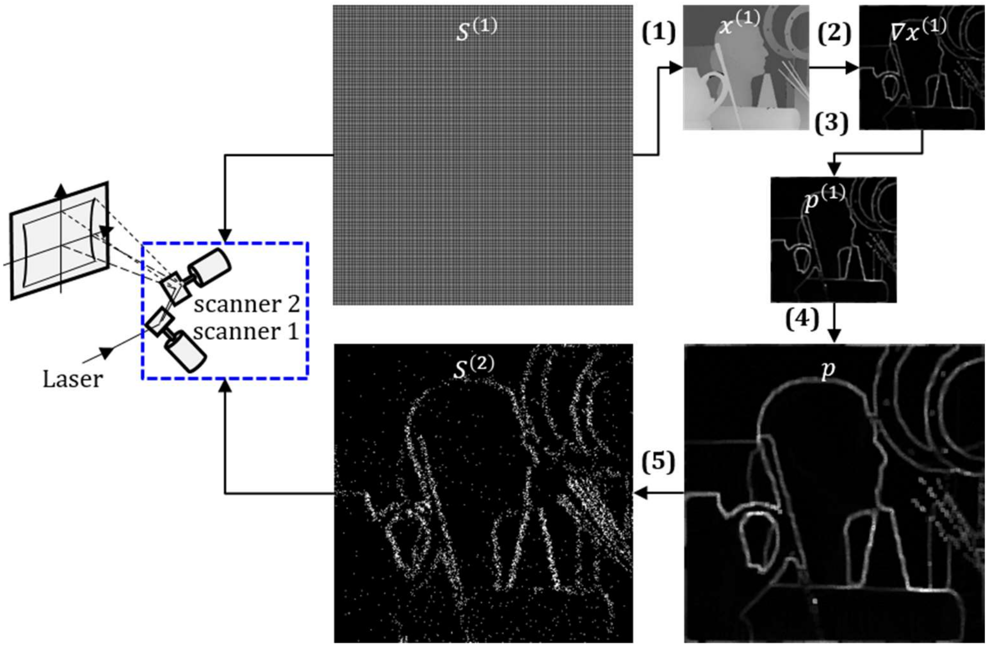


Fig. 4.9. Proposed sampling algorithm.

4.3.5. Timing and memory constraints

This subsection discusses the memory requirements and the time complexity of the proposed algorithm to test whether the proposed algorithm satisfies the two constraints in Problem 2.

4.3.5.1. Timing constraint

Among the timing parameters discussed in Section 4.2, t^{pos} indicates the time for generating a refinement sampling pattern $S^{(2)}$. The derivation of $x^{(1)}$ in the pilot stage (lines from 3 to 6 in Fig. 4.5) is not time-consuming because it is a result of uniform grid sampling. Next, the operation in line 8 in Fig. 4.5 is a simple derivation of a gradient and takes less than 10 cycles in most hardware circuits. In addition, this computation can be executed on-the-fly if the input is received in the raster-scan order. The computation of the line 9 in Fig. 4.5 aims to obtain a root of $g(\tau)$ that is a piecewise linear and monotonically increasing function, with $g(+\infty) = N(1 - \xi)$ and $(0) \leq 0$. Therefore, the root is unique and its derivation is performed in an iterative way. This step, in practice, is likely to obtain a root within a small number of iterations (i.e., 64 iterations) and therefore, it can be processed within 1,000 cycles in most hardware. For more information, a flowchart of the iterative algorithm is given in Appendix B. Note that experimental results also demonstrate that the number of iterations is less than 32 for all locations to be derived. The last step in Line 10 is simply an expanding operator. Given a sampling pattern in line 9 in Fig. 4.5, it is straightforward that this step can executed within a single hardware cycle. In summary, the proposed algorithm can be implemented in hardware within 1,000 cycles, taking less than 10 microseconds for an operating clock frequency of 100MHz.

In general, the time complexity of the proposed algorithm can be analyzed as follows. A sequential visit of sampling points with the steps defined above permits the simultaneous derivation of $S^{(1)}$, $b^{(1)}$, and $x^{(1)}$. This stage is executed in linear time, $O(M)$, with respect to the number of pixels. In the refinement stage, the gradient computation and the gradient-based optimization are executed in linear time, $O(M)$. These steps are performed using the downsampled image, which further reduces the complexity. The expanding step is also fast and executes in linear time, $O(M)$. In summary, the proposed sampling algorithm satisfies the timing constraints of the LiDAR system.

4.3.5.2. Memory constraint

The memory space (or communication bandwidth) required in the proposed algorithm can be analyzed in a manner similar to that in Section III-B. The sampling map, $S^{(1)}$ in the pilot stage is uniform grid such that it does not require a memory space for storage. Meanwhile, the sampling map, S_η, in the refinement stage must be stored because it is a non-uniform pattern. The size of S_η is equal to that of the downsampled image, which requires $\alpha \times \xi \times N$ bits. In addition, the sampled depth map requires memory space of size, $n \times \xi \times N$, where n indicates the resolution of the depth map. Therefore, the total memory space can be expressed as follows:

$$G(\xi, n, S) = n \times \xi \times N + \alpha \times \xi \times N. \tag{4.23}$$

Given the available memory space C, as the sampling ratio is limited, it can be expressed by the following inequality:

$$\xi \le \frac{C}{(n + \alpha) \times N}. \tag{4.24}$$

When compared with the sampling ratio ξ in (4.9), the value in (4.24) is significantly bigger because α is substantially smaller than n, and N is substantially larger than n and α. This analytical comparison demonstrates that the proposed algorithm has a sampling ratio ξ, which is substantially larger than that of the previous algorithm in [28]. This illustrates that the proposed algorithm has a larger number of pixels in the sampled depth map, thereby resulting in higher reconstruction quality.

Example 1 (continued): For an equivalent memory space and the depth image specified in Example 1, the proposed algorithm is used to derive the sampled depth map. If half of the samples is selected in the first stage (i.e., $\alpha = 0.5$), the sampling ratio is derived from (4.24):

$$\xi \le \frac{64}{(8 + 0.5) \times 32} = 23.5\%. \tag{4.25}$$

This value is much larger than that for [28] in Example 1 ($\xi = 12.5\%$) and is close to the value for the uniform grid sampling ($\xi = 25\%$).

For various cases, see Table 4-III for details.

4.4. Experimental results

This section presents an evaluation of the proposed sampling method in comparison to three reference algorithms, uniform grid sampling, gradient-based optimal sampling in [26], and two-stage sampling in [28]. From Sections 4.4.1 to 4.4.4, four sampling methods are compared for Problem 2b, which is derived from Problem 2 by maintaining the memory space constraint but ignoring the timing constraint. On the other hand, Section 4.4.5 compares the proposed and uniform grid sampling approaches that satisfy the timing constraint. Experiments are conducted using the six testbeds in the Middlebury datasets[①]: Aloe, Art, Baby2, Moebius, Dolls, and Rocks [11], [12].

4.4.1. Comparison on the conventional sampling problem

4.4.1.1. Subjective comparison

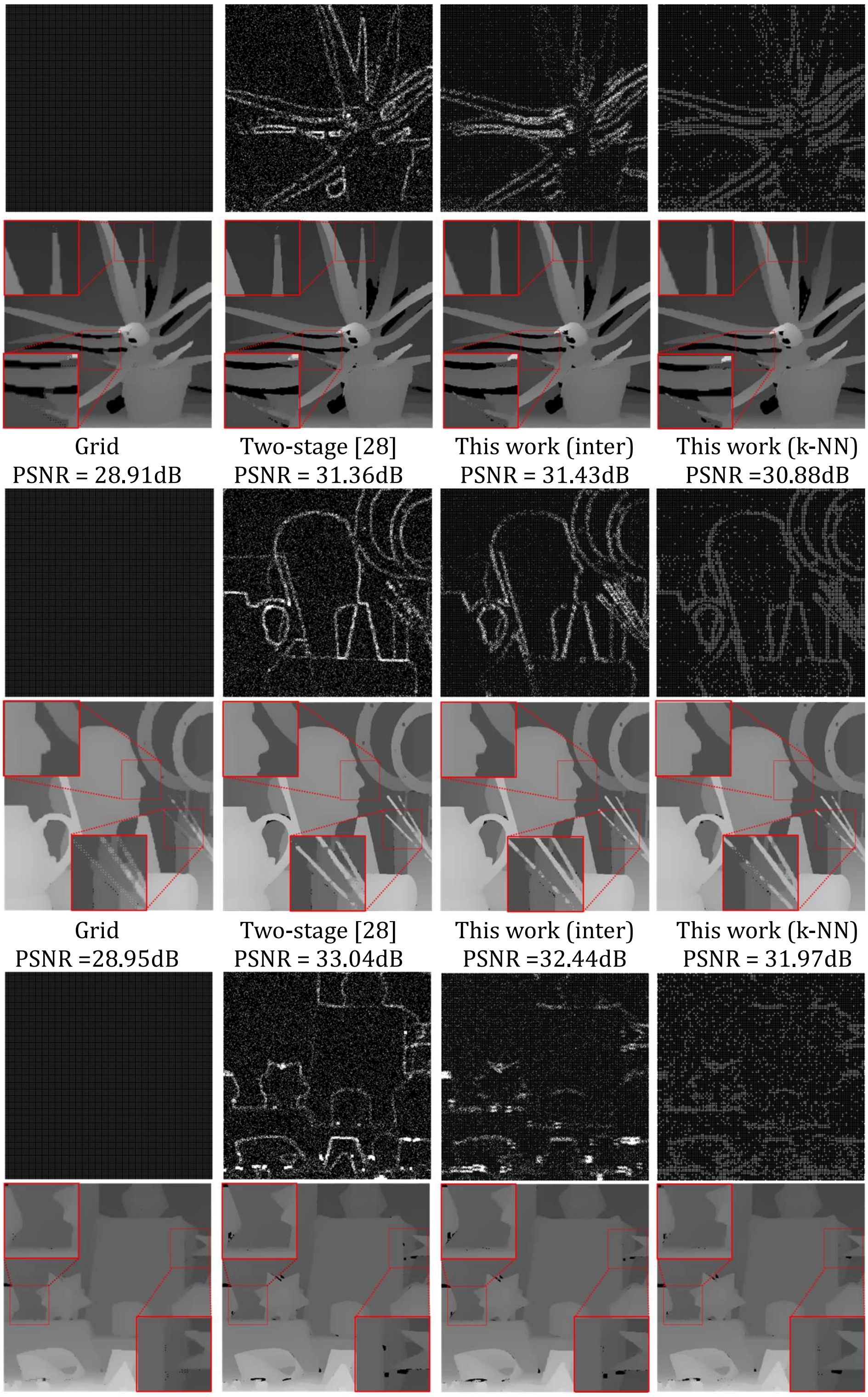
This subsection compares grid sampling, [28], and two proposed sampling methods on "Aloe", "Art" and "Moebius" at sampling rate 10%. To evaluate the reconstruction quality, both timing and memory-space constraints are ignored in this case. On each reconstructed image, two small regions are cropped and zoomed in for comparison. Experimental results are demonstrated in Fig. 4.10. For each test image, grid sampling gives the lowest reconstruction quality. Meanwhile, both of two proposed sampling method only slightly degrades a reconstruction performance when being compared with [28]. Fig. 4.10 shows an example to compare four sampling methods at sampling ratio $\xi = 0.1$. Fig. 4.10(a)-(d) shows the sampling maps of grid, [28], our Alg. 1 with k-NN and our Alg. 2 with interpolation. Fig. 4.10(e)-(h) shows the reconstructed images of four sampling maps with zoom-in cropped areas. The results demonstrate that our proposed methods significantly improve reconstruction quality of a grid method; achieving PSNRs of 31.97 dB and 32.44 dB compared to 28.95 dB. Meanwhile, they only slightly degrade reconstruction quality of the state-of-art sampling method in [28].

4.4.1.2. Quantitative comparison

PSNR results of four sampling algorithms are presented in Table 4-I. Table 4-I.A

[①] http://vision.middlebury.edu/stereo/data/

shows the PSNR results of all six testbeds, while Table 4.1.B shows the average PSNR differences of our approaches over grid sampling and that in [28]. Note that for fair comparison, PSNR results of the two-stage sampling method in [28] are reported in two cases: 1) use a gradient of an RGB image to sample the first half as reported in [28] (marked with "RGB"); and 2) perform uniformly random sampling in the first stage by running the code by the authors. Experimental results demonstrate that both of two proposed methods outperforms the grid sampling. Particularly, as shown in Table 4.1.A, compared to the grid, our proposed method with interpolation improves PSNR by 2.23, 3.77, 5.21, 5.88 and 6.89 (dB) corresponding to percentage of samples of 5%, 10%, 15%, 20% and 25%. Our proposed sampling method with an *k*-NN expanding operator also achieves 0.93, 2.24, 3.58, 3.76, and 3.87 dB improvements when being compared with the grid. Meanwhile, our proposed method only slightly degrades the state-of-art two-stage ones in [28] as it only degrades PNSRs by 0.15, 0.30, 0.77, 1.23 and 2.11 dB for sampling ratios of 5%, 10%, 15%, 20% and 25%, respectively.



Grid
PSNR = 28.91dB

Two-stage [28]
PSNR = 31.36dB

This work (inter)
PSNR = 31.43dB

This work (k-NN)
PSNR =30.88dB

Grid
PSNR =28.95dB

Two-stage [28]
PSNR = 33.04dB

This work (inter)
PSNR =32.44dB

This work (k-NN)
PSNR = 31.97dB

Grid
PSNR = 28.72dB

Two-stage [28]
PSNR = 32.86dB

This work (inter)
PSNR = 32.93dB

This work (k-NN)
PSNR = 30.63dB

Fig. 4.10. Subjective comparisons on “Aloe”, “Art” and “Moebius” images.

TABLE 4.1.A

COMPARISON OF PSNRS OF THE RECONSTRUCTED IMAGE

Test Image	Sampling Methods	Percentage of samples/ Reconstruction quality (PSNR)				
		5%	10%	15%	20%	25%
Aloe	Grid	25.32	28.91	30.09	31.30	32.35
	[28] (RGB)	27.60	31.39	33.37	36.41	38.63
	[28] (Depth)	27.87	31.36	34.12	36.31	39.33
	This work (inter)	27.95	31.43	33.49	35.85	37.99
	This work k-NN	27.13	30.88	33.41	35.24	36.32
Art	Grid	27.52	28.95	30.84	32.52	33.71
	[28] (RGB)	30.87	34.15	37.28	42.97	48.00
	[28] (Depth)	29.55	33.04	35.77	37.54	39.60
	This work (inter)	29.07	32.44	35.30	38.50	40.65
	This work k-NN	28.34	31.97	35.10	36.21	37.83
Baby	Grid	34.44	36.80	37.67	39.05	40.07
	[28] (RGB)	39.70	44.90	48.66	52.50	52.00
	[28] (Depth)	38.55	43.05	46.51	49.17	54.59
	This work (inter)	38.50	42.54	46.38	47.89	49.08
	This work k-NN	35.81	39.13	42.17	43.25	44.24
Dolls	Grid	28.49	29.05	30.09	30.81	31.67
	[28] (RGB)	29.51	32.53	34.00	36.27	37.65
	[28] (Depth)	29.32	31.52	32.92	36.88	38.20
	This work (inter)	29.22	31.48	33.48	34.30	37.26
	This work k-NN	28.80	30.59	32.03	33.13	34.85
Moebius	Grid	27.69	28.72	29.85	31.17	32.24
	[28] (RGB)	31.07	35.11	37.76	39.92	41.89
	[28] (Depth)	29.95	32.86	36.11	38.44	40.79
	This work (inter)	30.13	32.93	34.56	36.44	38.20
	This work k-NN	28.64	30.63	33.21	34.72	35.53
Rock	Grid	27.69	28.72	29.85	31.17	32.24
	[28] (RGB)	30.77	35.40	37.51	40.45	42.51
	[28] (Depth)	30.17	33.75	38.83	40.35	43.75
	This work (inter)	29.66	32.97	36.46	38.32	40.43
	This work k-NN	28.01	31.36	33.95	36.03	36.72

TABLE 4.1.B

THE PSNR COMPARISON OF THE RECONSTRUCTED IMAGE

Sampling Methods	Percentage of samples/ PSNR				
	5%	10%	15%	20%	25%
Grid (baseline)	28.52	30.19	31.40	32.67	33.71
This work (inter)	2.23	3.77	5.21	5.88	6.89
This work k-NN	0.93	2.24	3.58	3.76	3.87
Two-stage [28] (baseline)	30.90	34.26	37.38	39.78	42.71
This work (inter)	-0.15	-0.30	-0.77	-1.23	-2.11
This work k-NN	-1.45	-1.84	-2.40	-3.35	-5.13

4.4.2. Comparison on the new sampling problem for LiDAR

4.4.2.1. Compression ratios

In these images, the depth is represented in 8-bit resolution (n = 8). The target compression ratios, χ, in (4.10) are selected as 5%, 10%, 15%, 20%, and 25% of the size of the original image. The sampling ratio, ξ, is derived from (4.9), (4.13), and (4.24), and the results are presented in Table 4.2. The first column presents the sampling methods. From the second to the sixth columns, the sampling ratios are reported for various target compression ratios (χ). For the uniform grid sampling, ξ and χ are equivalent because no additional data is necessary to store the sampling pattern. On the other hand, the previous sampling methods in [26] and [28] require memory space for their sampling patterns. As discussed in Chapter 2.3, these methods require 12.5% of the space required by the original image when the depth uses 8-bit resolution. Therefore, it is not feasible to use those methods if the target compression ratio is either 5% or 10%. When the target compression ratios are 15%, 20%, or 25%, their sampling ratios are 2.50%, 7.5%, and 12.5%, respectively. The sampling ratios of the proposed sampling method are presented in the last row of Table 4.2, which evidently illustrates that the sampling ratio approaches the available memory space because the amount of data needed to store the sampling pattern is significantly smaller than that for the previous sampling methods.

Because of the randomness in the selection of data samples in (2.1) and (2.2), the amount of sampled data may not be equal to the target sampling ratio ξ. Therefore, the experiment is conducted to demonstrate the extent to which the sampling operation satisfies the target sampling ratio ξ. In this regard, Table

TABLE 4.2
THE SAMPLING RATIO (ξ) MATHEMATICALLY DERIVED FROM (4.9), (4.13), AND (4.24) FOR A GIVEN TARGET COMPRESSION RATIO (χ)

Sampling methods	Target Compression Ratio (χ) 5%	10%	15%	20%	25%
Uniform grid	5%	10%	15%	20%	25%
[26]	N/A	N/A	2.5%	7.5%	12.5%
[28]	N/A	N/A	2.5%	7.5%	12.5%
Proposed	4.71%	9.41%	14.12%	18.82%	23.53%

TABLE 4.3
THE MEASURED SAMPLING RATIO (ξ) AVERAGED OVER SIX TEST IMAGES FOR A GIVEN TARGET COMPRESSION RATIO (χ)

Sampling methods	Target Compression Ratio (χ) 5%	10%	15%	20%	25%
Uniform grid	5.0376	10.0319	14.9563	20.0663	25.0566
[26]	N/A	N/A	2.5069	7.5041	12.5043
[28]	N/A	N/A	2.4965	7.4968	12.5102
Proposed	4.7321	9.5067	14.198	18.8839	23.0648

4.3 presents the sampling ratio ξ measured by experiments with Middlebury six test images for a specified target compression ratio. The measurement results are averaged over six images. The numbers in Table 4.3 are comparable to those in Table 4.2. This demonstrates that the sampled data ratios are approximately equal to the target sampling ratios.

4.4.2.2. Quantitative evaluation with Peak-signal-to-noise-ratio

The conventional approach to evaluating a sampling pattern is to reconstruct the estimation of the original image from its samples and compare the reconstructed image with the original image. For this evaluation, the alternating direction method of multipliers with the wavelet dictionary is used as the reconstruction algorithm. The details of this metric are available in [28]. The MATLAB toolbox[2] of [28] is publicly provided by the authors. For evaluating

[2] http://videoprocessing.ucsd.edu/~leekang/projects.html

the quality of the reconstructed image, the PSNR (Peak-signal-to-noise ratio) of the reconstructed image with the six testbeds in the Middlebury datasets is reported in Table 4.4. As reported in Tables 4.2 and 4.3, the target compression ratios are selected as 5%, 10%, 15%, 20%, and 25%. The experimental results demonstrate that the proposed sampling is superior to grid sampling in terms of the PSNRs at most target compression ratios. Furthermore, at the compression ratios of 10% and 15%, the proposed sampling achieves the best reconstruction quality among all the methods. In the case of the target compression ratios of 20% or 25%, the optimal sampling in [26] achieves the best quality. However, it should be noted that the optimal sampling approach is proposed in [26] under the assumption that the global properties of a depth image (e.g. its gradients) are available, but in practice, the gradient is not available prior to data sampling, which renders the assumption unrealistic in practical applications. The proposed sampling achieves quality that is similar to the method in [28], while the proposed sampling is substantially faster and requires much less memory than the sampling method in [28]. It must be noted that the experimental results with the method in [28] are adjusted to solve Problem 2b. Particularly, the results are obtained with the target-sampling ratio ξ instead of target compression ratios (χ) such that they are different from those reported in [28]. For example, with χ =15% the sampling ratio ξ is only 2.5% as shown in Table 4.2 such that the corresponding PNSR results in Table 4.3 become much smaller than those in [28].

There are some cases in which the PSNRs are degraded notwithstanding the increase in the sampling ratio (χ). For example, the PSNR with “Baby” in the grid

TABLE 4.4

THE PSNRS OF THE RECONSTRUCTED IMAGE FOR A SPECIFIED MEMORY SPACE

Test Images	Sampling Methods	Target Compression Ratio (χ)				
		5%	10%	15%	20%	25%
Aloe	Grid	**24.02**	27.15	28.52	30.53	31.41
	Proposed	23.65	**28.13**	**30.63**	32.08	32.83
	[28]	NA	NA	23.25	28.41	31.48
	[26]	NA	NA	23.66	**32.82**	**37.03**
Art	Grid	**24.83**	26.51	27.08	30.05	30.15
	Proposed	23.59	**27.60**	**29.20**	30.65	31.16
	[28]	NA	NA	23.50	28.18	30.74
	[26]	NA	NA	21.01	**30.74**	**34.60**
Baby	Grid	29.18	31.66	32.21	36.71	35.26
	Proposed	**29.64**	**36.08**	**38.13**	39.64	37.57
	[28]	NA	NA	30.09	37.43	41.84
	[26]	NA	NA	29.99	**41.04**	**49.17**
Dolls	Grid	**27.51**	27.89	28.22	31.59	30.32
	Proposed	26.51	**30.52**	**31.96**	33.07	32.21
	[28]	NA	NA	28.55	31.09	32.46
	[26]	NA	NA	28.62	**36.31**	**41.36**
Moebius	Grid	**26.65**	27.24	27.62	30.76	30.59
	Proposed	25.03	**30.55**	**31.49**	34.00	32.89
	[28]	NA	NA	27.51	31.28	33.81
	[26]	NA	NA	26.72	**38.00**	43.34
Rocks	Grid	24.54	26.81	27.64	29.85	30.44
	Proposed	**24.94**	**28.77**	**31.03**	33.08	32.54
	[28]	NA	NA	25.10	29.52	32.60
	[26]	NA	NA	25.96	**35.96**	**41.09**
Average	Grid	**26.12**	27.88	28.55	31.58	31.36
	Proposed	25.56	**30.28**	**32.07**	33.75	33.20
	[28]	NA	NA	26.33	30.99	33.82
	[26]	NA	NA	25.99	**35.81**	**41.10**

sampling and the proposed sampling is degraded with χ varying from 20% to 25%. This occurs when the sampling grid matches the object boundary at χ = 20% but not at χ = 25%.

4.4.2.3. Quantitative evaluation with Percentages of bad pixels

This subsection evaluates the reconstruction quality with another metric, the

Percentage of Bad Pixels (*PBP*). A pixel is labeled as a “bad pixel” if the variation between its true and estimated values is larger than a certain threshold th. Then, PBP is defined as follows [1]:

$$PBP = \frac{1}{N} \sum_{i\,=\,1}^{N} (|xout_i - x_i| \;\; > \;\; th). \tag{4.26}$$

where $xout_i$ is the reconstructed image and x_i is the original image. In general, the *PBP* is used to measure the quality of the reconstructed image for representing the absolute difference metric apart from the PSNR that compares the mean squared metric. It should be noted that lower PBP values denote better quality. Each test is performed 10 times, and the reported results are the average values because the random sampling generates a different result for each trial. Three values of the threshold th are used for this experiment, varying among 1, 2, and 3.

The experimental results with the PBP metric are shown in Table 4.4, which demonstrates that the proposed sampling achieves optimal performance in a majority of the cases. Non-uniform sampling yields lower *PBP* than uniform sampling and the proposed sampling because the non-uniform sampling effectively addresses blurring artifacts on the object boundary and consequently enhances the PSNRs. However, it may result in a loss of detail in smooth areas. As the smooth areas are generally larger than the boundary areas, non-uniform sampling is adversely affected by a large PBP.

TABLE 4.5

THE PBPs OF THE RECONSTRUCTED IMAGE FOR A SPECIFIED MEMORY SPACE

Test Images	Sampling Methods	% of Bad Pixels [$th = 1$] Target Compression Ratio (χ)				
		5%	10%	15%	20%	25%
Aloe	Grid	**30.02**	18.53	12.80	9.43	6.89
	Proposed	33.53	**17.53**	**10.25**	**7.27**	**5.41**
	[28]	NA	NA	50.68	19.39	10.06
	[26]	NA	NA	83.45	39.63	23.57
Art	Grid	**32.01**	**16.81**	11.78	8.87	6.68
	Proposed	37.93	18.03	**11.53**	**8.28**	**6.14**
	[28]	NA	NA	55.24	18.92	9.76
	[26]	NA	NA	82.33	37.69	21.91
Baby	Grid	**15.72**	**6.72**	4.96	3.15	2.32
	Proposed	20.71	7.86	**4.02**	**2.54**	**1.93**
	[28]	NA	NA	28.56	8.76	4.10
	[26]	NA	NA	59.81	21.60	6.89
Dolls	Grid	**31.53**	**17.68**	**13.05**	**9.10**	**6.52**
	Proposed	40.27	21.15	13.74	9.79	7.75
	[28]	NA	NA	55.81	23.31	13.69
	[26]	NA	NA	70.90	45.35	24.38
Moebius	Grid	23.63	**12.54**	9.45	6.93	4.94
	Proposed	**25.48**	12.80	**8.37**	**5.68**	**4.15**
	[28]	NA	NA	41.09	13.60	7.12
	[26]	NA	NA	67.84	26.08	11.71
Rocks	Grid	**18.47**	**8.83**	**5.96**	4.26	3.17
	Proposed	24.40	11.04	6.28	**3.71**	**2.89**
	[28]	NA	NA	38.57	12.09	5.83
	[26]	NA	NA	64.27	38.00	12.79
Average	Grid	**30.02**	18.53	12.80	9.43	6.89
	Proposed	33.53	**17.53**	**10.25**	**7.27**	**5.41**
	[28]	NA	NA	50.68	19.39	10.06
	[26]	NA	NA	83.45	39.63	23.57

TABLE 4.5 (CONT')

THE PBPS OF THE RECONSTRUCTED IMAGE FOR A SPECIFIED MEMORY SPACE

Test Images	Sampling Methods	% of Bad Pixels [$th = 2$] Target Compression Ratio (χ)				
		5%	10%	15%	20%	25%
Aloe	Grid	**21.22**	11.58	8.23	5.94	4.31
	Proposed	23.49	**11.36**	**6.65**	**5.03**	**3.85**
	[28]	NA	NA	35.45	11.17	5.65
	[26]	NA	NA	74.47	25.77	12.44
Art	Grid	**19.25**	**10.68**	8.34	**6.00**	**4.29**
	Proposed	24.61	12.13	**8.18**	6.09	4.75
	[28]	NA	NA	39.29	10.99	6.27
	[26]	NA	NA	69.69	23.90	12.12
Baby	Grid	**8.24**	4.04	3.16	2.18	1.71
	Proposed	11.52	**3.88**	**2.16**	**1.32**	**1.17**
	[28]	NA	NA	16.46	3.89	1.61
	[26]	NA	NA	38.70	11.06	2.16
Dolls	Grid	**17.18**	**8.71**	**6.48**	**4.44**	**3.25**
	Proposed	23.29	10.98	6.60	4.72	3.89
	[28]	NA	NA	35.82	10.67	5.88
	[26]	NA	NA	55.22	24.43	8.57
Moebius	Grid	**13.68**	7.77	6.09	4.51	3.05
	Proposed	15.72	**7.58**	**5.14**	**3.56**	**2.74**
	[28]	NA	NA	24.61	6.74	3.66
	[26]	NA	NA	49.27	11.27	3.41
Rocks	Grid	**9.10**	**4.86**	3.91	2.74	2.18
	Proposed	13.37	5.30	**2.97**	**1.96**	**1.77**
	[28]	NA	NA	22.49	4.96	2.22
	[26]	NA	NA	55.47	23.33	4.32
Average	Grid	**21.22**	11.58	8.23	5.94	4.31
	Proposed	23.49	**11.36**	**6.65**	**5.03**	**3.85**
	[28]	NA	NA	35.45	11.17	5.65
	[26]	NA	NA	74.47	25.77	12.44

TABLE 4.5 (cont')

The PBPs of the Reconstructed Image for a Specified Memory Space

Test Images	Sampling Methods	% of Bad Pixels [$th = 3$] Target Compression Ratio (χ)				
		5%	10%	15%	20%	25%
Aloe	Grid	**18.56**	9.45	7.07	5.04	3.76
	Proposed	19.82	**9.39**	**5.70**	**4.42**	**3.47**
	[28]	NA	NA	27.26	8.88	4.65
	[26]	NA	NA	68.72	17.24	7.96
Art	Grid	**16.09**	**9.47**	7.58	**5.15**	**3.88**
	Proposed	19.97	10.72	**7.29**	5.47	4.32
	[28]	NA	NA	31.55	9.35	5.55
	[26]	NA	NA	62.58	18.62	7.98
Baby	Grid	**6.09**	3.46	2.87	1.99	1.58
	Proposed	8.43	**2.80**	**1.69**	**1.06**	**1.01**
	[28]	NA	NA	11.66	2.49	0.97
	[26]	NA	NA	31.20	6.88	0.95
Dolls	Grid	**11.81**	**6.03**	4.75	3.36	**2.57**
	Proposed	16.80	7.41	**4.49**	**3.32**	2.85
	[28]	NA	NA	25.55	6.84	3.70
	[26]	NA	NA	44.50	14.04	3.95
Moebius	Grid	**10.91**	6.34	4.97	3.71	2.53
	Proposed	12.84	**5.93**	**4.06**	**2.83**	2.31
	[28]	NA	NA	18.41	4.99	2.83
	[26]	NA	NA	39.91	5.31	**2.06**
Rocks	Grid	**6.63**	4.06	3.42	2.41	1.94
	Proposed	9.06	**3.84**	**2.28**	**1.61**	**1.52**
	[28]	NA	NA	13.84	3.17	1.56
	[26]	NA	NA	49.35	15.10	1.59
Average	Grid	**18.56**	9.45	7.07	5.04	3.76
	Proposed	19.82	**9.39**	**5.70**	**4.42**	**3.47**
	[28]	NA	NA	27.26	8.88	4.65
	[26]	NA	NA	68.72	17.24	7.96

4.4.3. Subjective evaluation

Fig. 4.11 illustrates the subjective quality of the reconstructed image produced by the proposed algorithm in experimenting with the “Aloe” image. The target compression ratios, χ, are selected as 5% and 25%, resulting in the two sampling patterns illustrated in Figs. 4.11(a) and (d), respectively. Evidently, the sampling map at a ratio of 25% appears denser than that at 5%. The reconstructed images are evidently different. The reconstructed image with 5% in Fig. 4.11(b) suffers from blurring artifacts at the object boundaries, resulting in a significant variation from the image in Fig. 4.11(c). On the other hand, the reconstructed image with 25% appears similar to that in Fig. 4.11(e), and the variation in Fig. 4.11(f) is mostly black. The variations are intuitively understandable considering that the large compression ratio is a trade-off with the degradation in the reconstructed image quality.

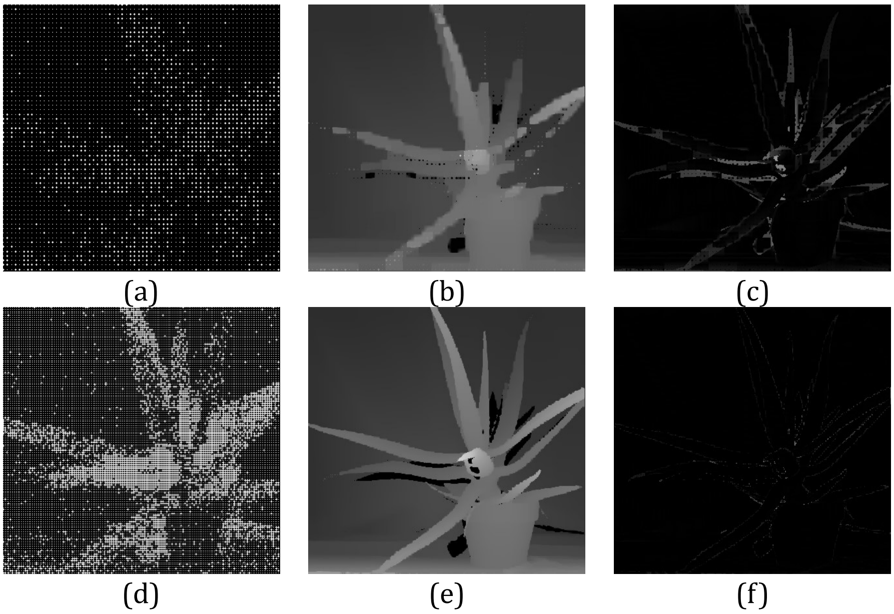


Fig. 4.11. Subjective quality of the image reconstructed by the proposed sampling algorithm with two target compression ratios. (a), (b), (c) Sampling pattern with 5% compression ratio, its reconstructed image and Variation from the original image with 5% compression ratio. (d), (e), (f) Sampling pattern with 25% compression ratio, its reconstructed image and variation from the original image.

4.4.4. Proposed grid sampling and grid sampling method

Recall that the two-stage sampling algorithm in [28] is too time-consuming to satisfy the timing constraint of Problem 2. Therefore, the sampling method in [28] is not compared in this subsection, and only the grid sampling and the proposed two-stage sampling are compared. The experiments are conducted with 24 disparity images from the Middlebury datasets [11], [12]. The reconstruction methods[③] in [39]-[39] are selected for the sake of running time and reconstruction quality.

4.4.4.1. Middlebury datasets

Fig. 4.12 demonstrates a comparison of reconstructed images that are obtained from the uniform grid and the proposed sampling schemes with Aloe, and Art from the Middlebury sets at a sampling rate of 20%. For each set, the first, second, and third rows present the ground truth images and the reconstructed images obtained from the uniform grid and the proposed samplings, respectively. In the "Art" image, especially in the regions of face and sticks, the proposed sampling pattern produces much better reconstruction quality than in the case of uniform grid sampling. In particular, the region surrounding the face in the reconstructed image of the uniform grid sampling suffers from large artifacts. On the contrary, the proposed sampling efficiently includes more samples in the same area so that its reconstructed image looks better.

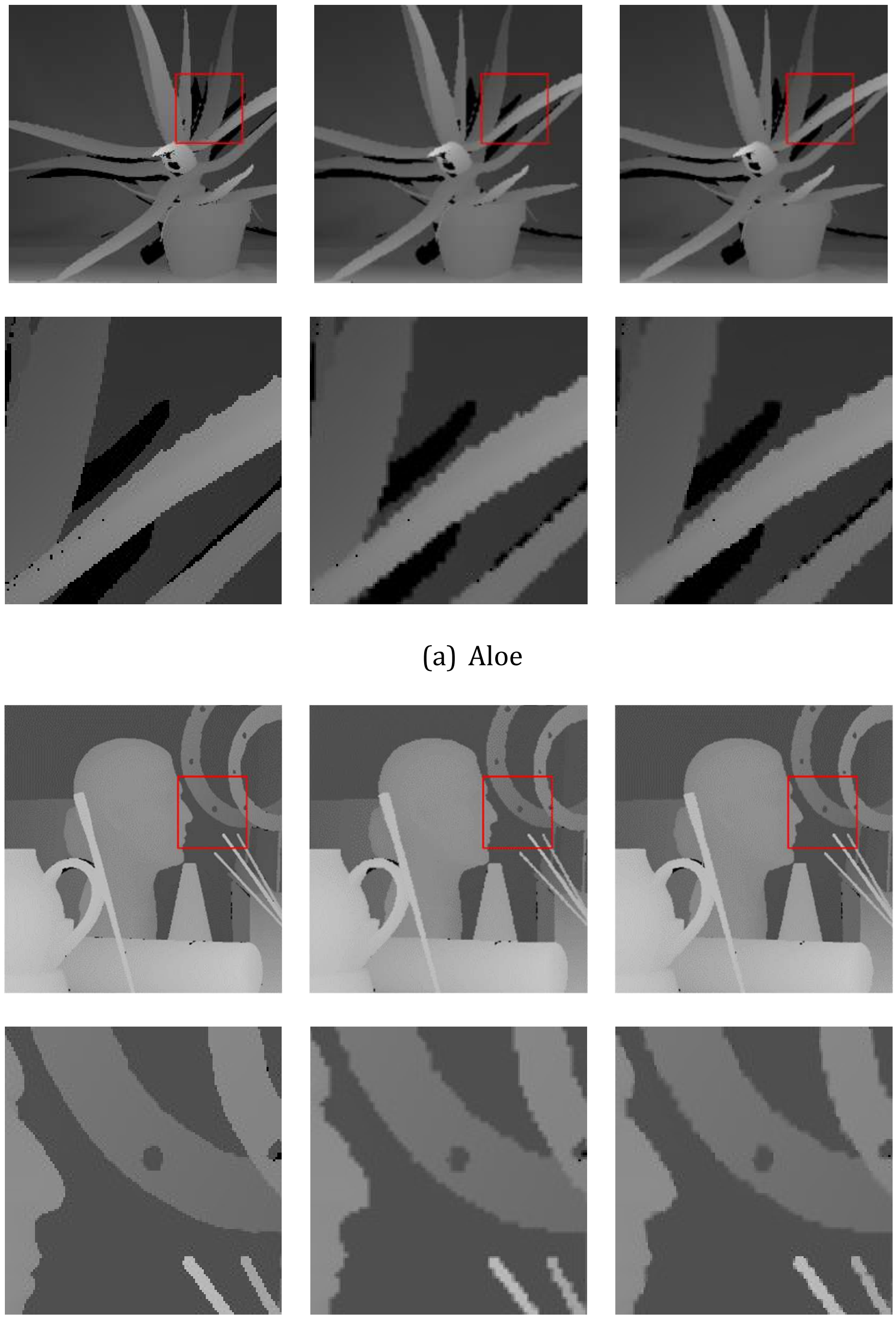
Fig. 4.13 presents the PSNR comparison between the proposed and the uniform grid samplings with the Middlebury datasets. Fig. 4.13(a) and Fig.4.

[③] https://github.com/sparse-depth-sensing/sparse-depth-sensing

13(b) demonstrate the PSNR results of the uniform grid and the proposed samplings, respectively, while Fig. 4.13(c) presents the PSNR enhancement by the proposed sampling. Experimental results demonstrate that the proposed sampling consistently outperforms that of the uniform grid for all test images and at all five sampling rates. The best PSNR improvement of about 16dB is achieved for “Wood2” image at a given sampling rate of 25%. At the sampling rates of 5%, 10%, 15%, 20%, and 25%, the proposed sampling achieves the averaged PSNR improvements of 1.16, 1.50, 2.54, 3.79, and 6.27dB, respectively.

4.4.4.2. Brown Laser range datasets

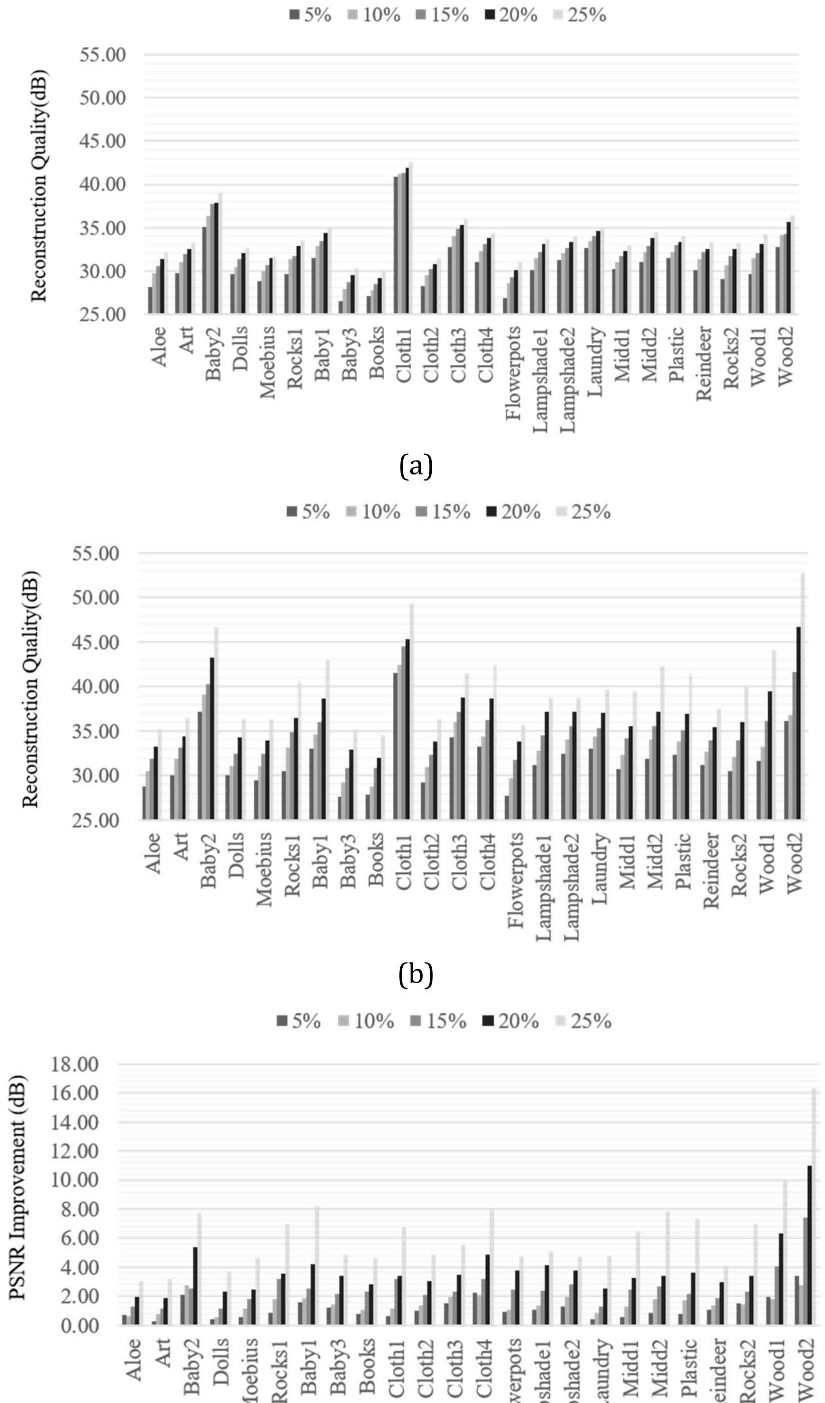
Fig. 4.14 presents the PSNR comparison between the proposed and the uniform grid samplings with the Brown laser range datasets [55]. Fig. 4.14(a) and Fig. 3.14(b) demonstrate the PSNR results of the uniform grid and the proposed samplings, respectively, while Fig. 4.14(c) presents the PSNR enhancement by the proposed sampling. Experimental results demonstrate that the proposed sampling consistently outperforms that of the uniform grid for all test images and at all five sampling rates. The best PSNR improvement of about 16dB is achieved for “Wood2” image at a given sampling rate of 25%. At the sampling rates of 5%, 10%, 15%, 20%, and 25%, the proposed sampling achieves the averaged PSNR improvements of 1.26, 1.48, 2.04, 2.93, and 6.97dB, respectively.



(a) Aloe

(b) Art

Fig. 4.12. Subjective comparison of the ground truth (first row), the grid sampling (second row), and the proposed method (third row) on (a) "Aloe", (b) "Art".

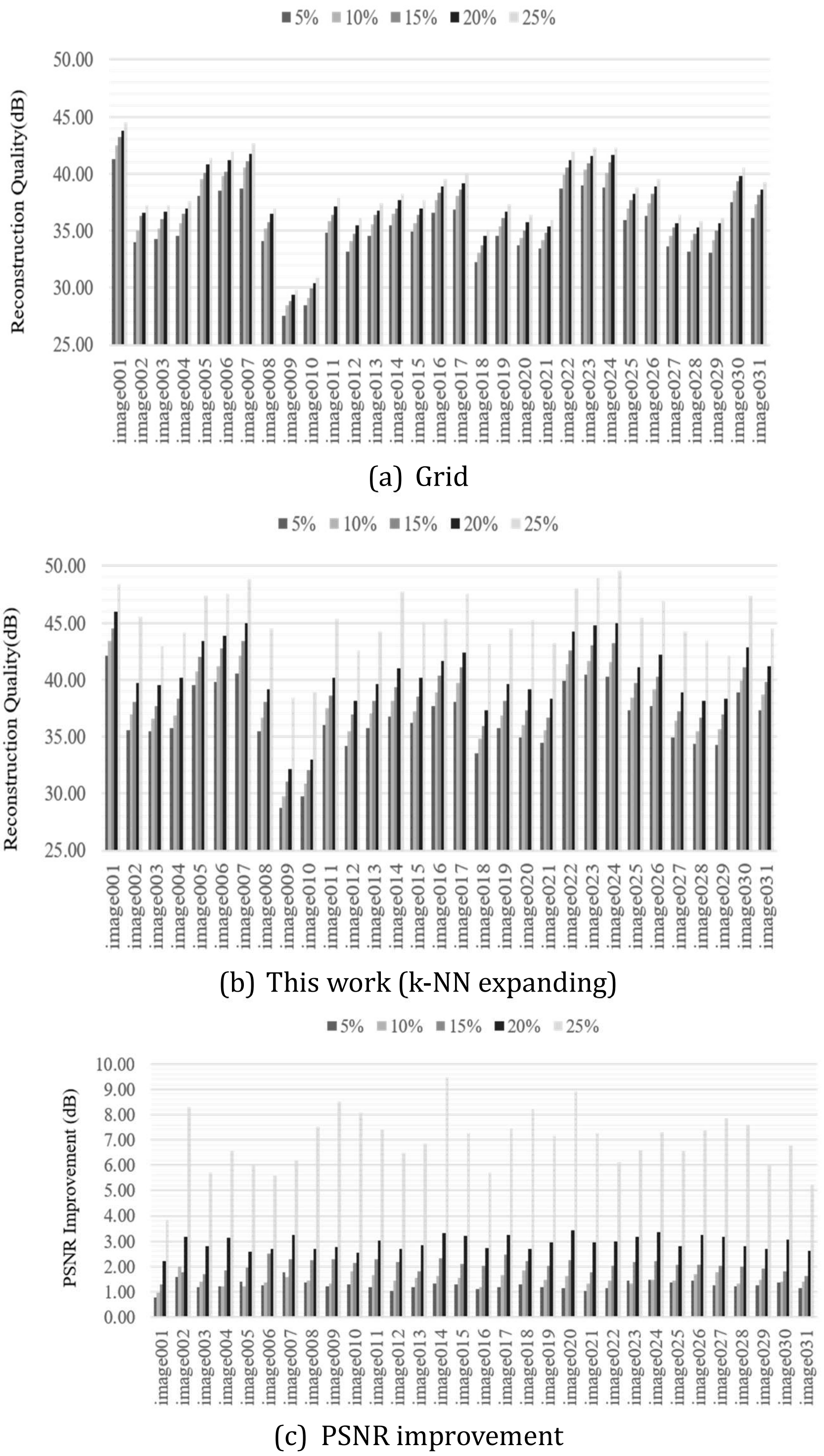


(a)

(b)

(c)

Fig. 4.13. PSNR improvements of the proposed method over the uniform grid sampling on Middlebury datasets. (a) Grid sampling; (b) Proposed sampling; and (c) Comparison.



(a) Grid

(b) This work (k-NN expanding)

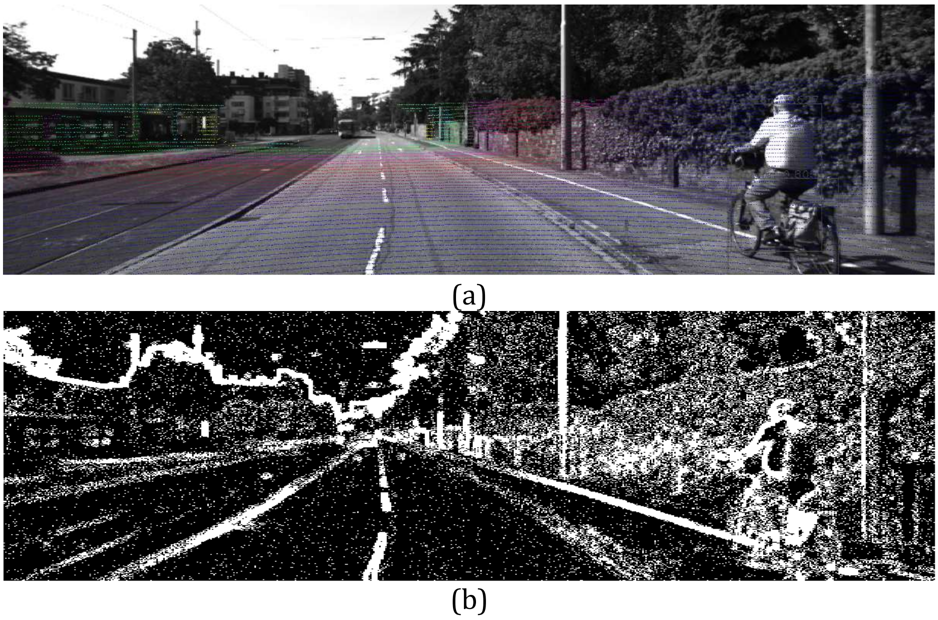
(c) PSNR improvement

Fig. 4.14. PSNR improvements of the proposed method over the uniform grid sampling on Brown laser range datasets. (a) Grid sampling; (b) Proposed sampling; and (c) Comparison.

Chapter 5: ROI-based LiDAR Sampling in On-Road Environment for Autonomous Driving

5.1. Introduction

An ultimate goal of a sampling problem is to select a sampling matrix to minimize the reconstruction error in an entire scene or a specific region-of-interest (ROI). This problem is directly related to the compressive sensing theory, which has intensively studied in many decades. Several approaches to find a sampling matrix have been presented in [26]-[31]. Motivated by the property of the wavelet transform that the relevant coefficients coincide with discontinuities, S. Hawe et al. [26] recommended that a data acquisition system should pick samples at the discontinuities or along gradients. However, this approach is not practical for two reasons. First, the gradient of the disparity map is not available prior to sampling. Therefore, all the gradient information needs to be inferred from the color image. Second, the gradient of a color image could be significantly different from that of the disparity map. Thus, inferring the disparity gradient from the color image is challenging. L. K. Liu et al. [27] recommended using outlier elimination prior to edge disparity estimation. Meanwhile, S. Schwartz et al. [29], [30] proposed a saliency-guided sampling approach to perform sampling in a two-stage manner. First, approximately 10% of the samples are sampled randomly, and an approximate depth map is derived from those sampled data. Subsequently, object information or saliency is extracted from the estimated depth to select better locations with the remaining sample budget. Following this approach, L. K. Liu et al. proposed a similar two-step sampling in [28]. Particularly, at the pilot stage, half of the sample budget is sampled randomly or along the gradients of a color image. In the second stage, called the refinement stage, sampled points are used to estimate a round disparity map and then to compute locations for the remaining sample budgets. However, these approaches [28]-[30] involve time-consuming rough disparity estimation. In [31], L. K. Liu proposed a sampling framework for acquiring a depth video. Considering the merit of spatial information, this method estimates the motions between two color frames and uses them to compute a depth map from a previous estimated depth frame. A rough estimated map is used as a guide for a gradient-based sampling; so that the sampling procedure is completed in one stage rather than two. However, this method still involve time-consuming disparity estimation. In addition, the method is verified only with relatively simple synthetic datasets for which motion estimation becomes highly accurate, rendering accurate-sampling relatively convenient. However, in out-door environments; it is challenging to estimate motion accurately, owing to illumination, noise, and other camera factors. Furthermore, the sampling schemes in [26]-[31] are inappropriate for autonomous driving in on-road environments for the following two reasons. Firstly, as shown in Fig. 5.1, a scene in an outdoor environment generally consists of a complex background; this caused previous methods to allocate excessively high number of samples into non-interested areas (i.e., trees in Fig. 5.1). Secondly, it is challenging to obtain a reliable gradient image of a scene in an outdoor scenario because generally, its RGB image is complicated and its raw depth image too sparse to estimate a reliable gradient map (i.e., 1–2% sparse compared to an RGB image).



(a)

(b)

Fig. 5.1. Example of a sampling mask in on-road scenario. a) RGB image and sampling points. b) Sampling mask for a given sampling rate of 25%.

Inspired by the dual-mirror LiDAR in [56], Nguyen et al. in [57] proposed a sampling method to distribute samples in road and object regions based on the ratio between their areas. Given a fixed number of samples in road and ROI areas, the method efficiently distributes samples from road to object regions and thus significantly enhances the reconstruction quality of objects. However, in a road environment, it is challenging to fix the number of samples in road and object regions.

As a single type of sensor is likely to be inadequate for mimicking the sensing system of humans during driving, the main objective of this study is to present a framework exploiting the semantical information from camera sensors to enhance the data acquisition of a LiDAR. The proposed framework addresses a general LiDAR sampling case, where the budget of an entire frame is fixed. In addition, the proposed sampling method extends the state-of-the-art Oracle random sampling method in [28]; hence, it is highly efficient and applicable for various scenarios, i.e., on-road environments. To this end, the contributions of this work are as follows:

1. A systematical framework of depth acquisition in an on-road environment is presented. Unlike previous approach, our proposed detects the objects in a road and segments a scene into background, ROI, and object areas. From the segmentation result, the approach distributes samples across the segmented areas.
2. An ROI-based sampling problem is proposed to optimize a depth sensing system in a LiDAR for an on-road environment. The optimization problem has an optical solution, which effectively addresses the two problems of the LiDAR discussed above. Experimental results demonstrate that the proposed approach significantly reduces the mean-absolute-error (MAE) in the object area by at most 52.8%. Moreover, it achieves robust reconstruction quality at a very low sampling rate of 1%. In addition, the proposed sampling is remarkably fast (i.e., within a few milliseconds), rendering it applicable to a real-time acquisition system.

5.2. Proposed ROI-based sampling algorithm

This section presents the main concept and mathematical derivation of the proposed sampling.

5.2.1. Motivating example

Before addressing the sampling problem, particularly for on-road scenarios, it is necessary to determine the type of sampling pattern that should be used in

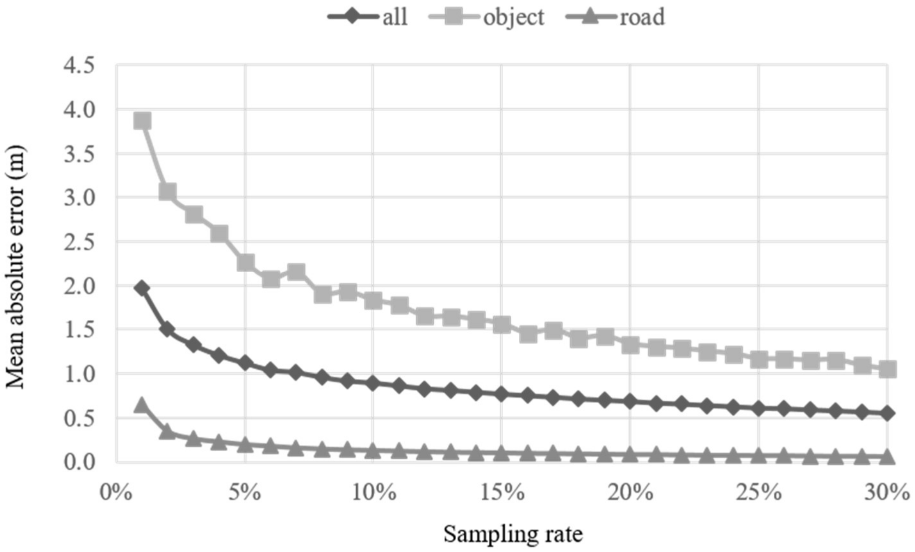
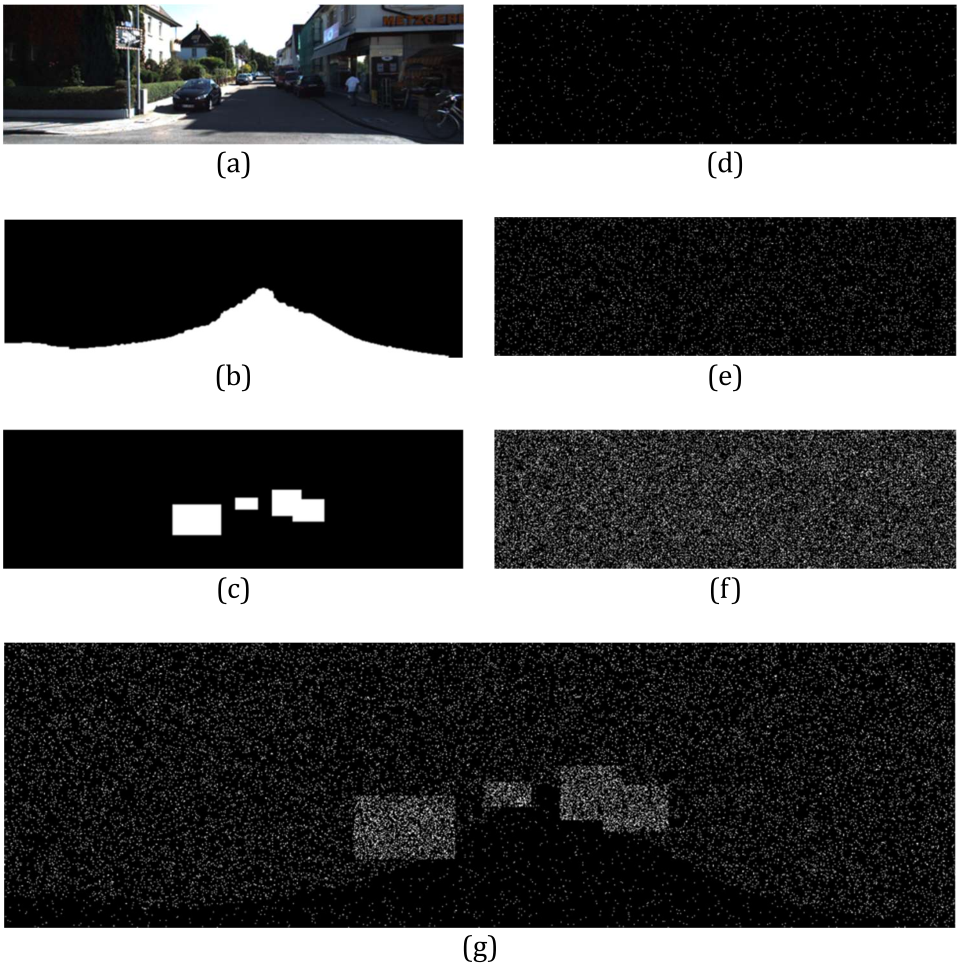


Fig. 5.2. Profiling of mean absolute error (m) on object, road and overall areas over sampling rates.

this case. This subsection demonstrates the characteristics of road, object, and background regions and the utilization of these characteristics to generate a sampling pattern. Fig. 5.2 shows an average MAE of test images for specified sampling rates ranging from 1% to 30%. Errors are measured in three regions: an entire scene, on-road, and object areas. Fig. 5.2 illustrates that three error lines gradually decrease when the sampling rate increases. In particular, the overall error of the entire scene decreases from 1.97 m to 0.55 m when the sampling rate increases from 1% to 30%. In addition, the number of errors on the road areas is approximately two times that on the entire scene. For example, at the sampling rate of 1%, the MAE error on the object area is approximately 3.88 m, which is approximately two times of that on the entire area (1.97 m). Meanwhile, the number of MAE errors on the road is relatively small. When the sampling rate increases from 1% to 30%, the error decreases from 0.65 m to

0.06 m. The profiling results in Fig. 5.2 indicate a strong message: For a given sampling budget, 1) it is likely to give more samples on an object area as this area plays a critical role for self-driving tasks, for example, obstacle detection and path planning; 2) a sampling budget on a road area is likely to be decreased without a significant degradation on MAE error.

Fig. 5.3 demonstrates a sampling pattern constructed under the above characteristics. First, thanks to various fast and accurate road and object detection algorithms [58]-[64], a single scene is assumed to be segmented into road, object, and background regions as shown in Fig. 5.3(a), (b), and (c), respectively. Meanwhile, three random patterns at sampling rates of 1%, 5%, and 20% are given in Fig. 5.3(d), (e), and (f), respectively. The synthetic sampling pattern is derived in the bottom image as follows. The sampling locations on the road region are obtained by applying locations on the road mask in Fig. 5.3(b) and those in the random sampling pattern in Fig. 3(d). Next, the locations on the object region are derived from the pattern at the rate of 20% (Fig. 5.3(f)) with the object mask in Fig. 5.3(c). Finally, the remaining locations in a background region are selected from the pattern at a sampling rate of 5% (i.e., Fig. 5.3(e)). The final sampling map is shown in Fig. 5.3(g); it has a sampling rate of 5.1%, which is close to that in Fig. 5.3(e). However, the MAE error on a road area is approximately 1.39 m, which is significantly smaller than that of the random error (i.e., 2.49 m). For this particular case, the overall MAE is approximately 0.82 m, which is smaller than that of the uniformly random sampling (0.89 m).



(a) (d)

(b) (e)

(c) (f)

(g)

Fig. 5.3. Motivational example for ROI-based sampling. (a) RGB image; (b) Road mask; (c) Object mask; (d), (e), (f) Random sampling masks at sampling rates of 1%, 5%, and 20%, respectively; and (g) Expected sampling map.

5.2.2. ROI-based Sampling Problem

To address the sampling problem in an on-road environment, a scene is assumed to be segmented into three regions: roads, objects, and background ones. Aided by convolutional neural networks (CNNs), various road and object detection algorithms have been rapidly developed in recent years; consequently, their accuracy and speed have improved significantly [58]-[64]. Numerous road detection approaches submitted to the KITTI road detection benchmark have precisions of over 96%, whereas the runtime is approximately 0.06 s [61]. Considering the effectiveness of available state-of-the-art road and object detection algorithms, it is reasonable to assume that both the road and object masks yielded by an RGB image of a scene are specified prior to the sampling operation in a LiDAR. Fig. 5.4 illustrates a system configuration for LiDAR sampling in an on-road environment. The three input images for a sampling module are an RGB image, an object mask, and a road mask. The sampling procedure operates as follows: First, an RGB image is captured, requiring approximately 16.7 ms for a high-speed 60-fps camera sensor. Next, both the object and road masks are derived from the specified RGB image, requiring less than 20 ms because the object and road detections can be performed at the frame rate of 50 fps [58, 61]. Finally, a sampling pattern is derived in 1 or 2 ms to guide the data acquisition in a LiDAR.

Given this configuration, the sampling problem in a LiDAR is derived as follows. Let S_B, S_R, and S_O denote the index sets of the points in background,

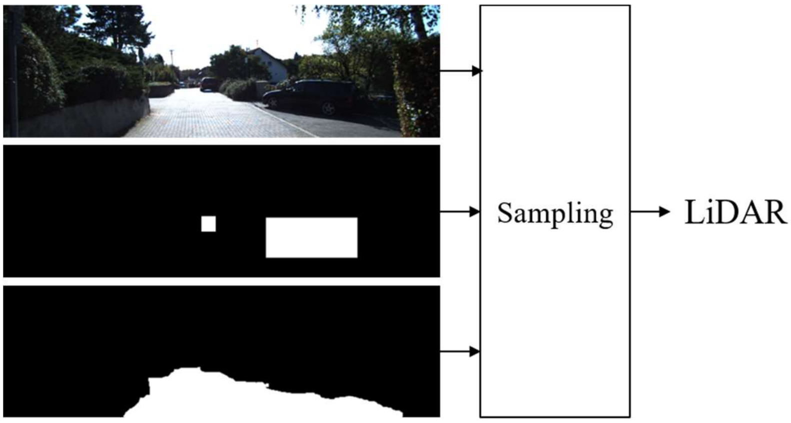


Fig. 5.4. System configuration for LiDAR sampling in an on-road environment. It is assumed that object and road masks are specified prior to sampling. The output sampling is used by a LiDAR sensor.

road, and object areas, respectively. Thus, the union of three sets is equal to $\{1, \dots, N\}$, and an intersection of any two sets is an empty set.

$$S_B \cup S_R \cup R_O = \{1, \dots, N\}.. \tag{5.1-a}$$

$$S_B \cap S_R = S_B \cap S_O = S_O \cap S_R = \emptyset. \tag{5.1-b}$$

The points in the background, road, and object areas have different characteristics so that they are likely to be sampled with different priorities. Let λ_B, λ_R, and λ_O be weighted-parameters representing sampling priorities for the background, road and object areas, respectively. Therefore, the sampling problem in Chapter 2 is modified as follows:

Problem P2 (*Sampling problem for a LiDAR*)*:* Given λ_B, λ_R, and λ_O, the sampling problem is to derive $\{i_1, \dots, i_M\}$ that minimizes the following objective function:

$$\min_{i_1,\ldots,i_M} \frac{1}{N}\left(\lambda_B \sum_{i\in S_B} \left|\left|x_i - \widetilde{x_\iota}\right|\right| + \lambda_R \sum_{j\in S_R} \left|\left|x_j - \widetilde{x_\jmath}\right|\right| + \lambda_O \sum_{k\in S_O} \left|\left|x_k - \widetilde{x_k}\right|\right|\right) \tag{5.2}$$

where $x_1, \ldots, x_N$ are real values and $\widetilde{x_1}, \ldots, \widetilde{x_N}$ are the values estimated from the M measurements $x_{i_1}, \ldots, x_{i_M}$.

Compared to Problem P1 in Chapter 2, Problem P2 has additional weight parameters λ_B, λ_R, and λ_O. Apparently, when λ_B, λ_R, and λ_O are equal to one, the objective function in (5.2) becomes that in (2.1). That is, the background, road, and object areas have identical sampling priority. As mentioned above, it is not feasible to determine a solution in a brute-force search manner; therefore, a heuristic method is necessary. Similar to the derivation in Chapter 2, Problem 2 is transformed into a gradient-based sampling in the following section.

5.2.3. Proposed ROI-based sampling algorithm

For consistency, let $a = [a_1, \ldots, a_N]^T$ be a vector indicating the prior information of the depth map. It should be noted that the vector a may be defined as the gradient of the depth map as shown in (2.5) in Chapter 2. However, it represents any prior information. Using the probabilistic model in [28], the gradient-based sampling is modified as follows. For a specified sampling ratio ξ, the prior map, and hyper-parameters λ_B, λ_R, and λ_O, the derivation of the optimal sampling probability $\{p_j\}_{j=1}^{M}$ is formulated as the following optimization problem:

$$\min_{p_1,\ldots,p_N} \frac{1}{N}\left(\lambda_B \sum_{i\in S_B} \frac{a_i^2}{p_i} + \lambda_R \sum_{j\in S_R} \frac{a_j^2}{p_j} + \lambda_O \sum_{k\in S_O} \frac{a_k^2}{p_k}\right) \tag{5.3}$$

subject to $\frac{1}{N}\sum_j^N p_j = \xi$ and $0 \leq p_j \leq 1$; here, S_B, S_R, and S_O are defined as (5.1-a) and (5.1-b).

Compared to (2.6), the optimization in (5.3) has additional parameters, λ_B, λ_R, and λ_O, which represent the sampling priority in the background, road, and object areas, respectively. When λ_B, λ_R, and λ_O are equal to one, the optimization problem in (5.3) becomes that in (2.6). Therefore, the proposed optimization can be considered as a generalized variation of the conventional problem.

5.2.4. Practical considerations

As described in Section I, a gradient generally becomes excessively noisy in an on-road environment so that it is challenging to generate a "desired" sampling pattern from the problem in (2.6). Therefore, this study considers an extreme and widely used case where the gradient information is unspecified or is under the following condition:

$$a_1 = a_2 = \cdots = a_N = 1 \tag{5.4}$$

Recall that a road area is generally flat so that the MAE does not change significantly as the sampling rate changes. Therefore, the parameter λ_R for a road area is likely to be smaller than the one for a background area. Considering this, λ_B and λ_R must satisfy the following equation:

$$\lambda_R = \alpha \times \lambda_B \tag{5.5}$$

where α is a constant and $\alpha \leq 1$. The parameter α indicates that the sampling rate in a road area is likely to be α times that in a background area.

Meanwhile, points in an object area are more likely to be sampled rather than those in a road area. Therefore, λ_O and λ_R must satisfy the following equation:

$$\lambda_O = \beta \times \lambda_R \tag{5.6}$$

where β is a constant bigger or equal to one, such that the sampling rate in a road area is likely to be β times that in a background area.

Without loss of generality, the optimization problem in (5.3) is modified as follows. For a given sampling ratio ξ, the prior map, and parameters α and β, the derivation of the optimal sampling probability $\{p_j\}_{j=1}^{M}$ is formulated as the following optimization problem:

$$\min_{p_1,\ldots,p_N} \frac{1}{N}\left(\sum_{i\in S_B} \frac{a_i^2}{p_i} + \alpha \sum_{j\in S_R} \frac{a_j^2}{p_j} + \beta \sum_{k\in S_O} \frac{a_k^2}{p_k}\right) \tag{5.7}$$

subject to $\frac{1}{N}\sum_j^N p_j = \xi$ and $0 \leq p_j \leq 1$; here, S_B, S_R, and S_O are defined as (5.1-a) and (5.1-b).

5.2.5. Distortion optimization problem

Obviously, given weight parameters α and β, solving (5.8) is similar to solving (2.6) with an exaction solution described in (2.7) and (2.8). This subsection presents an optimization problem to derive these parameters.

Problem P3 (α, β – distortion problem): Parameters α, β are derived by solving the following optimization problem:

$$\min_{\alpha,\beta} MAE_{obj} + \lambda MAE_{all} \tag{5.8}$$

where MAE_{obj} and MAE_{all} are reconstruction errors on road an overall regions, respectively; and λ is a weight factor.

Problem P3 clearly demonstrates that for α, β selection, it is necessary to consider both a quality enhancement on an ROI and a degradation on the other. However, similar to the sampling problem, it is a chicken-and-edge problem, which is not feasible to determine a solution in a brute-force search manner. In practice, a numerical solution is obtained as shown in the following subsection.

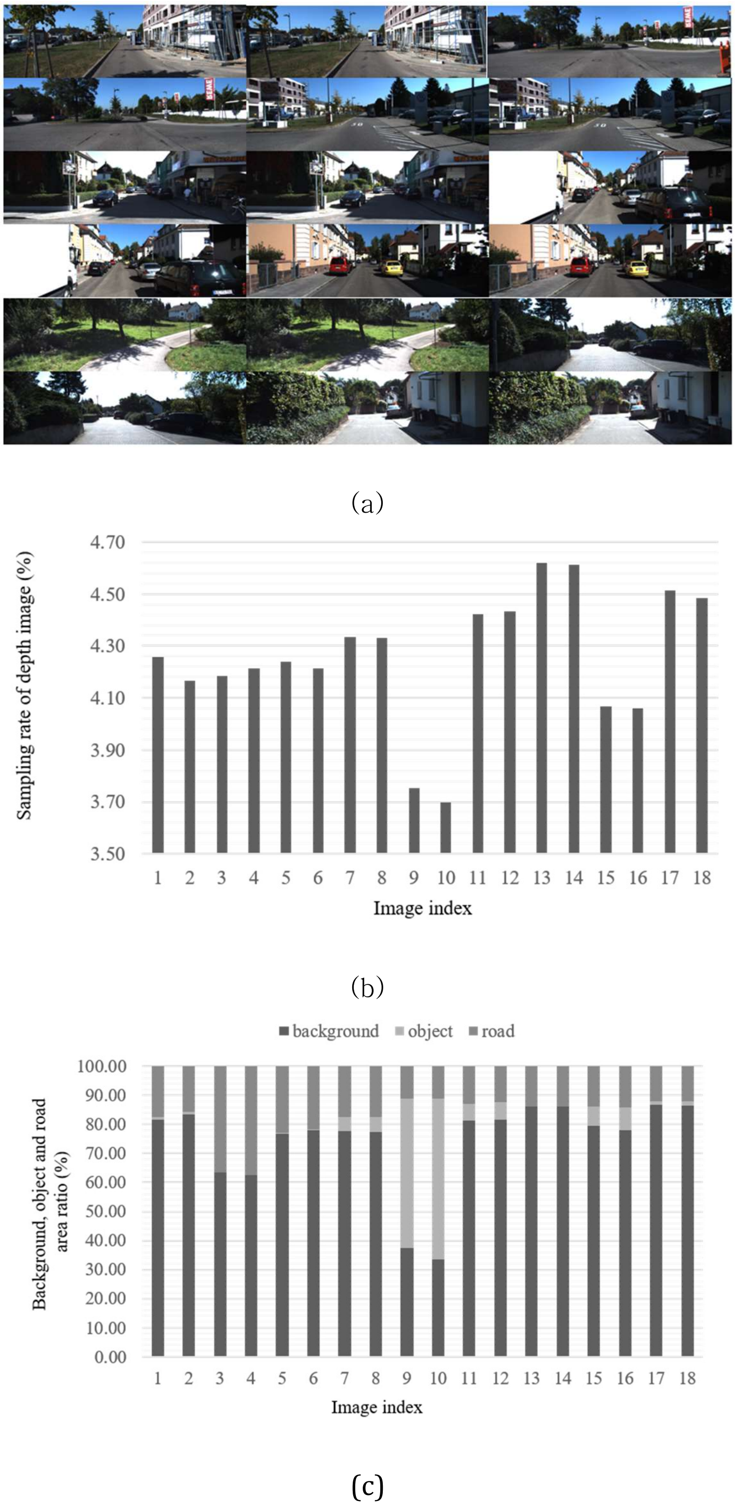
5.3. Experimental results

This section describes the experimental environments and then demonstrates the results of the proposed design. Firstly, Sections 5.3.1 and 5.3.2 describe the datasets and experimental results of the proposed ROI-based sampling method with the parameters α and β. Subsequently, a detailed comparison with the previous results is presented. Sampling rate ξ is set to either 5%, 10%, 15% or 20%, and 18 datasets in the KITTI dataset are used in the experiments.

5.3.1. Datasets

A set of 18 test images is acquired from the well-known KITTI dataset [9], [10] to cover several typical on-road scenarios. The set profiles are included in Fig. 5.5. Firstly, Fig. 5.5(a) shows the captured color images of these datasets; these images display various typical on-road scenarios such as a clear scene without any object and a scene with close and distant vehicles. Fig. 5.5(b) shows the sampling rates of the groundtruth depth images. As shown in [9]-[10], a depth image is obtained by projecting a point cloud data provided by a LiDAR sensor into a color image domain. However, the number of measurements of the LiDAR (i.e., a 64-channel Velodyne LiDAR) is significantly sparser than that of an RGB image. For these datasets, the sampling rate is approximately 4.26% on an average and ranges from 3.70% to 4.62%, as shown in Fig. 5.5(b). It is noteworthy that the sampling method in [15] is based on a wavelet, and the contour-based reconstruction method performs ineffectively when the sample budget is small (i.e., 1% or 2%) [18], [19]. That is, it is challenging to obtain a reliable gradient image of a scene in an outdoor scenario because generally, its RGB image is complicated and its raw depth image excessively sparse to estimate a reliable gradient map (i.e., 1%~2% sparse compared to an RGB image).

In addition, Fig. 5.5(c) demonstrates the profiling area ratios of the background, road, and object regions in the datasets. In particular, the ratio of the object areas is approximately 8.24% on average and ranges from 0% to 55.7%. Meanwhile, the ratio of the road areas is approximately 17.24% on average and ranges from 10.94% to 37.5%. The remaining background occupies an average area ratio of 74.33% and ranges from 33.35% to 86.81%. Two inferences can be noted from this numerical analysis. Firstly, it is too tight to fix a budget sample for the road and object areas as in [56]. Secondly, distributing a specified budget based on the computing areas of the road and object regions as in [56] cannot address various practical cases.


(a)

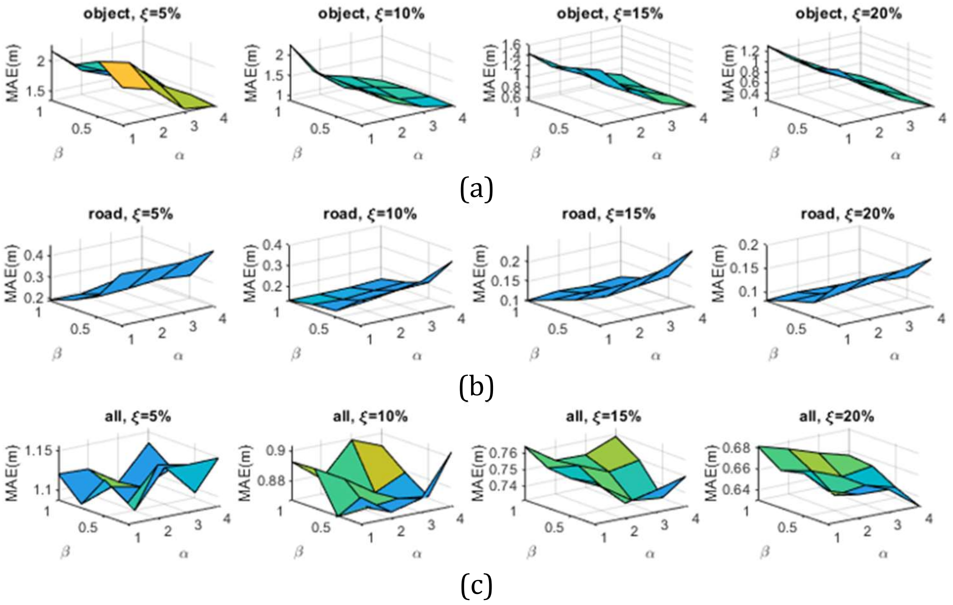
(b)

(c)

Fig. 5.5. Test images from KITTI datasets [1, 2]. (a) Color images; (b) Sampling rates of depth images; and (c) Background, object, and road area ratio.

5.3.2. Evaluation with different parameters

This subsection presents the experimental results with various values of the parameters α and β. As the sampling rate is set to 5%, 10%, 15%, or 20%, the minimum value of α is set as 0.25 to maintain a considerable sampling rate in a road area (approximately 1.25%, 2.5%, 3.75%, or 5%, respectively). Meanwhile, the maximum value of β is set as four as the maximum sampling rate in the object regions is approximately 80% (= 20% × 4). In particular, α is set to 0.25, 0.5, 0.75, or one, whereas β is set to one, two, three, or four. Fig. 5.6(a), Fig. 5.6(b), and Fig. 5.6(c) show the average MAEs of the object, road, and entire areas, respectively. The first, second, and third rows display the MAEs of the object, road, and overall areas, respectively. In each row, the first, second, third, and fourth columns display the MAEs at the sampling rates of 5%, 10%, 15%, and 20%, respectively. It should be noted that the MAEs on the object areas significantly decrease when the value of the parameter β increases. For example, considering the baseline sampling $\alpha = \beta = 1$ and the sampling $\alpha = 0.25$, $\beta = 4$. Note that with $\alpha = \beta = 1$, the sampling method becomes a uniform random sampling. The MAEs on the object area with this setting are shown in Fig. 5.6(a); here, the MAEs decrease by 0.83 m, 0.97 m, 0.87 m, and 0.48 m at the sample rates of 5%, 10%, 15%, and 20%, respectively. Meanwhile, the overall MAEs shown in Fig. 5.6(c) do not change significantly when the MAE differences between the various parameters shown in the third row of Fig. 6 are likely less than 0.1 m in all the cases. In addition, the average MAEs on the road area shown in Fig. 5.6(b) increase to an average of approximately 0.18 m and ranges from 0.1 m to 0.26 m. To reflect the importance of the object regions,



(a)

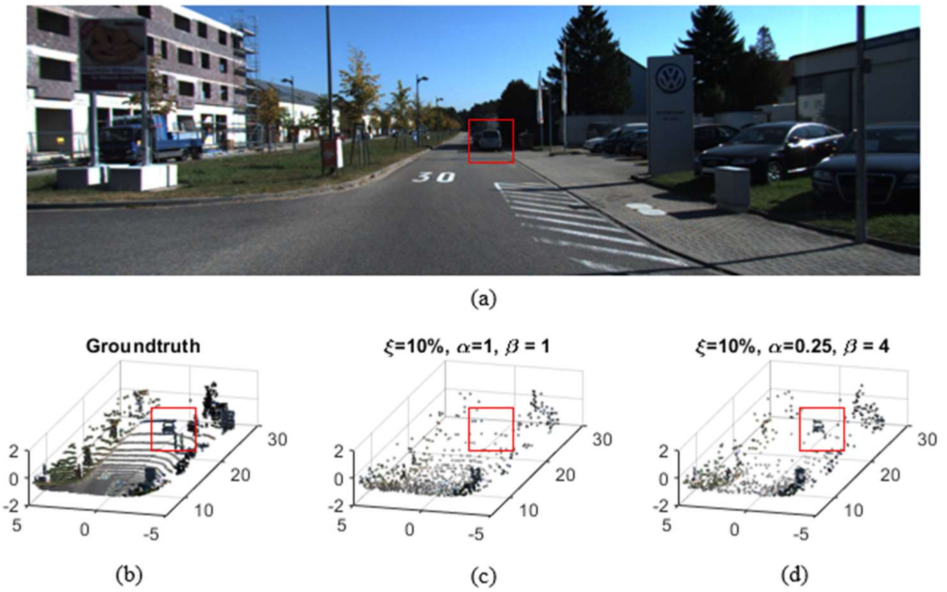
(b)

(c)

Fig. 5.6. MAEs at the sampling rate of 5%, 10%, 15%, and 20% on different regions: (a) Object; (b) Road; and (c) Overall.

$\beta = 4$ is used for comparison. Meanwhile, the lower and upper values of α are experimentally selected as 0.25 and one, respectively. These experimental results demonstrate that the proposed sampling method with $\alpha = 0.25$, $\beta = 4$ significantly improves the reconstruction quality on the object area. Similarly, the proposed sampling with $\alpha = 1$, $\beta =$ 4 also achieves a large improvement on the road area compared to the baseline. In this case, the average MAE on the road areas decrease to an average of approximately 0.84 m and ranges from 0.67 m to 0.98 m for various sample rates. It should be noted that 0.84 m is highly critical when compared to the sizes of the cars on the road.

Fig. 5.7 demonstrates the subjective comparison between the proposed sampling with $\alpha = 0.25$, $\beta = 4$ and the uniformly random sampling at the sampling rate of 10%. Fig. 5.7(a) shows the color view of a scene, because a car is considered as an object of interest. The 3D point cloud view of a scene is



(a)

(b) (c) (d)

Fig. 5.7. Subjective comparison between the proposed sampling ($\boldsymbol{\alpha} = \mathbf{0.25}$, $\boldsymbol{\beta} = \mathbf{4}$) with the baseline ($\boldsymbol{\alpha} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{1}$) or uniformly random sampling. (a) RGB image; (b) Ground truth point cloud; (c) Samples obtained by random sampling at the sampling rate 10%; and (d) Samples obtained by the proposed sampling.

displayed in Fig. 5.7(b). Meanwhile, the sampling points of the random sampling and the proposed method are shown in Fig. 5.7(c) and (d), respectively. It is apparent that the number of samples in Fig. 5.7(c) and (d) are substantially less than that in Fig. 5.7(b); this results in a substantial reduction in memory storage and computational power in an actual case. However, visually, the samples in Fig. 5.7(d) appear significantly better than those in Fig. 5.7(b) because the samples surrounding the car are effectively captured in this sampling. It should be noted that only very few points of the car are captured in Fig. 5.7(c); this is likely to cause ambiguity in obstacle avoidance or path planning in practical autonomous driving. This subjective comparison clearly demonstrates the advantage of the proposed ROI-based sampling over the uniformly random sampling.

5.3.3. Object and quantitative comparisons

This subsection compares the proposed sampling method with three previous sampling approaches: random sampling, color-image-guided sampling [26], and two-stage sampling [28]. To reflect the importance of the object regions, $\beta = 4$ is used for comparison. The lower and upper values of α are selected as 0.25 and one, respectively. In particular, for the proposed method, two settings are $\alpha = 1$, $\beta = 4$ and $\alpha = 0.25$, $\beta = 4$. It should be noted that the sampling methods in [26], [28] are used with synthetic disparity datasets [11], [12] rather than on-road KITTI datasets [9], [10]. Therefore, for a fair comparison, they are modified to be used for KITTI datasets. In particular, the color-image-guided sampling [26] uses a half of the sample budget used for random sampling. For the remaining budget, it computes the gradient of a gray image and computes the remaining locations based on the gradient-based sampling in Chapter 2. Moreover, the two-stage sampling [28] is modified by using the interpolation method in Matlab for estimating a depth map because the reconstruction quality of the method in [28] is excessively low when the sparsity of the depth map is approximately 1–4%.

The comparison of the MAEs on the object, road, and overall areas are reported in Tables 5-I, II, and III, respectively. On each table, the first, second, and third rows display the experimental results with uniformly random sampling, color-image-guided sampling [26], and two-stage sampling [28], respectively; the fourth and fifth rows display the results with two variations in the proposed methods. In each row, the second, third, fourth, and fifth columns report the results with the sampling rates of 5%, 10%, 15%, and 20%, respectively. Table 5.1 shows that the proposed sample achieves the highest performance among all the methods on the object area, which is critical in autonomous driving. Compared to [26], the variation in the proposed method in the fifth row reduces the MAE by 0.852 m on average, ranging from 0.79 m to 0.955 m. A reconstruction error or an MAE is commonly used to verify the effectiveness of a sampling pattern. Objects refer to cars, trucks, or pedestrians, whose size is relatively small. Thus, a reduction in the MAE by approximately 1 m is significant.

This study also presents a visual example in Fig. 5.7. Apparently, the absence of measurements in the object areas in Fig. 7(c) causes significant degradation of an MAE. When compared to the two-stage sampling [28], the variation of the proposed method reduces the MAE by 0.382 m on an average (ranging from 0.258 m to 0.631 m). That is, the proposed method achieves 35.75% reduction on an average (ranging from 26.84% to 52.8%) in the MAE on the object area when compared to [28]. Tables 5.2 and 5.3 present the MAEs on the road and overall areas of less importance. The results indicate that the two-stage sampling [28] achieves the highest performance with these criteria. However, compared to [28], the proposed method decreases the MAE on the road and entire areas by at most 0.265 m and 0.194 m, respectively. It should be noted that the MAE degradation mainly occurs on the background region; it consists of trees or buildings, which are less important for obstacle detection and localization. Hence, the proposed sampling method provides an effective trade-off between errors on the object and those on the remaining areas.

TABLE 5.1

COMPARISONS OF MAEs (M) AMONG VARIOUS SAMPLING ALGORITHMS ON **OBJECT** AREA AT SAMPLING RATES OF 5%, 10%, 15%, AND 20%

	5%	10%	15%	20%
Random	2.176	1.871	1.433	1.315
Color image-guided [26]	2.306	1.702	1.350	1.136
Two-stage [28]	1.981	1.230	0.819	0.581
This work $\boldsymbol{\alpha = 1},\ \boldsymbol{\beta = 4}$	1.506	0.892	0.667	0.365
This work $\boldsymbol{\alpha = 0.25},\ \boldsymbol{\beta = 4}$	**1.350**	**0.900**	**0.561**	**0.274**

TABLE 5.2

COMPARISONS OF MAEs (M) AMONG VARIOUS SAMPLING ALGORITHMS ON **ROAD** AREA

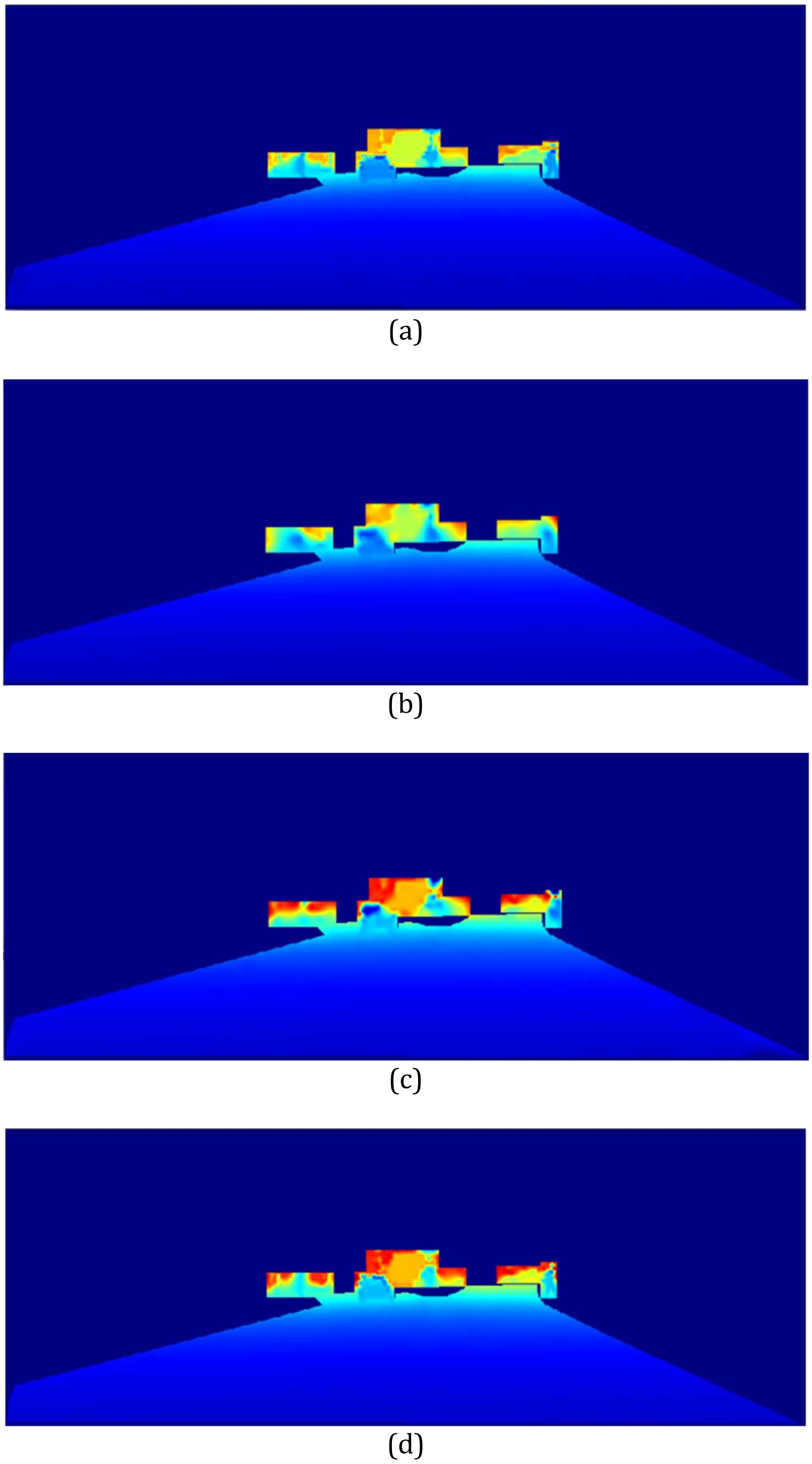
	5%	10%	15%	20%
Random	0.193	0.135	0.100	0.083
Color image-guided [26]	0.211	0.137	0.103	0.084
Two-stage [28]	**0.189**	**0.116**	**0.086**	**0.067**
This work $\boldsymbol{\alpha = 1},\ \boldsymbol{\beta = 4}$	0.211	0.144	0.110	0.083
This work $\boldsymbol{\alpha = 0.25},\ \boldsymbol{\beta = 4}$	0.454	0.350	0.243	0.183

TABLE 5.3

COMPARISONS OF MAEs (M) AMONG VARIOUS SAMPLING ALGORITHMS ON **OVERALL** AREA

	5%	10%	15%	20%
Random	1.121	0.893	0.764	0.682
Color image-guided [26]	1.123	0.877	0.735	0.627
Two-stage [28]	**1.041**	**0.748**	**0.573**	**0.437**
This work $\boldsymbol{\alpha = 1},\ \boldsymbol{\beta = 4}$	1.137	0.891	0.759	0.655
This work $\boldsymbol{\alpha = 0.25},\ \boldsymbol{\beta = 4}$	1.149	0.903	0.750	0.631

The final reconstructed images are shown in Fig. 5.8. Fig. 5.8(a) shows the reconstruction result from a raw depth map. Note that a raw depth is also sparse. As demonstrated in Fig. 5.8(b), (c), and (d), the road areas are effectively reconstructed for all the three methods, i.e., random sampling, modified two-stage sampling [28], and this work, respectively. In addition, the proposed algorithm generates the closest output compared to the upper-bound performance, whereas the random sampling yields the lowest performance. Fig. 5.8(e), Fig. 5.8(f), Fig. 5.8(g), and Fig. 5.8(h) show the reconstruction results and zoomed-out object regions from a raw depth image, random sampling, modified two-stage sampling [28], and this work, respectively. Evidently, the proposed sampling method yields the best performance on an object area.



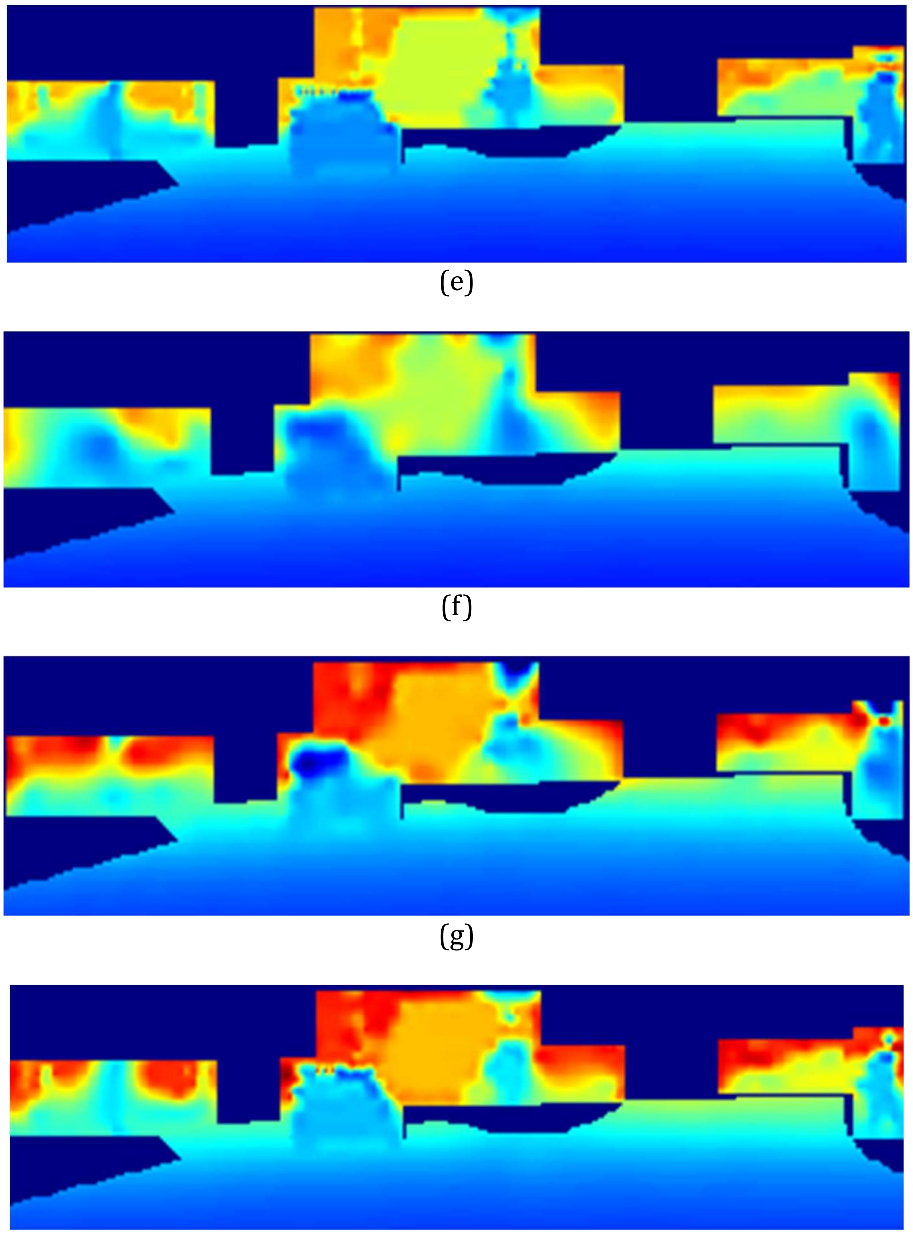
(a)

(b)

(c)

(d)

Fig. 5.8. Example of reconstruction results at road and object areas by various sampling methods. (a) raw depth; (b) random sampling, (c) two-step sampling [28]; (d) this work.



(e)

(f)

(g)

(h)

Fig. 5.8 (cont'). Example of reconstruction results at road and object areas by various sampling methods. (a) raw depth; (b) random sampling, (c) two-step sampling [28]; (d) this work and (e), (f), (g), and (h) the zoom-out results of the object areas from of (a), (b), (c), and (d), respectively.

Chapter 6: Implementation Issues

6.1. Implementation of gradient-based sampling

This subsection presents an efficient method for solving the gradient-based sampling. For a specified sampling ratio ξ and a gradient map, the derivation of the optimal sampling probability $\{p_j\}_{j=1}^{M}$ is formulated as the following optimization problem:

$$\min_{p_1,\ldots,p_N} \frac{1}{N}\sum_{j=1}^{N}\frac{a_j^2}{p_j} \ . \qquad (6.1)$$

subject to $\frac{1}{N}\sum_j^N p_j = \xi$ and $0 \ \leq \ p_j \leq \ 1$. In [32], the solution is formulated as follows:

$$p_j = \min\left(\tau a_j, 1\right). \qquad (6.2)$$

where τ is the solution of $g(\tau) = 0$ and $g(\tau)$ can be calculated as follows:

$$g(\tau) = \sum_{j}^{N} \min(\tau a_j, 1) - \xi N. \qquad (6.3)$$

Fig. 6.1 presents the Matlab code for determining a sampling pattern for a specified gradient map. A detailed flowchart is illustrated in Fig. 6.2. In general, the flow chart presents an iterative binary search. The number of pixels N, sampling rate ξ, and number of iterations are the inputs. To compute the output P, two variables A and B are first initialized as a lower bound and an upper bound. In each iteration, a variable C is computed by averaging A and B; P is computed from C using Eq. (6.2). Note that this flowchart does not include multiplication, so that it can be efficiently implemented in hardware at a high clock speed (i.e., 100 MHz) Fig. 6.3 and Fig. 6.4.

Matlab Code

```
% objfunc
function out = objfun(x,b,k)
out = sum(min(1,b*x))-k;
end

% Gradient sampling
function Sout = gradient_sampling(grad, xi)
        [H, W]    = size(grad);
        N                  = W*H;
        myfun     =
@(v)objfun(v,grad(:),round(N*xi));
        tau                = fzero(myfun,[0,1e16]);
        p                  = max(1e-16, min(1,grad
        Sout          = (rand(H,W)<=p);
end
```

Fig. 6.1. Matlab code to determine a sampling pattern for a specified gradient map.

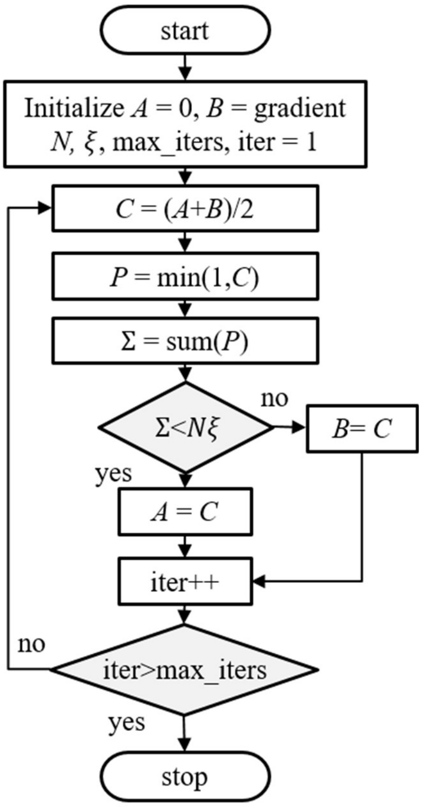


Fig. 6.2. Flowchart of efficient solution for gradient-based sampling.

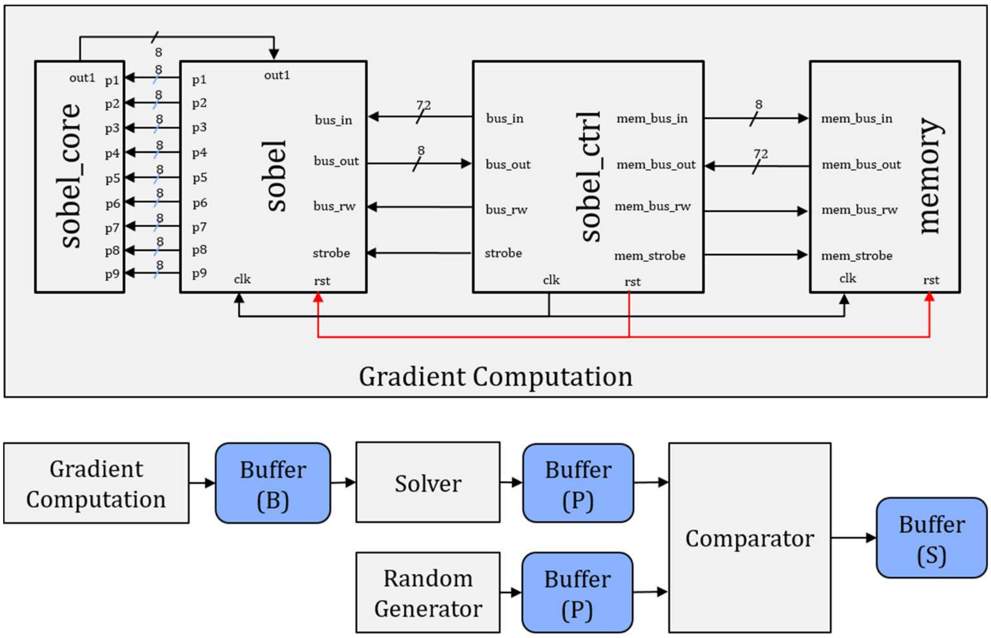
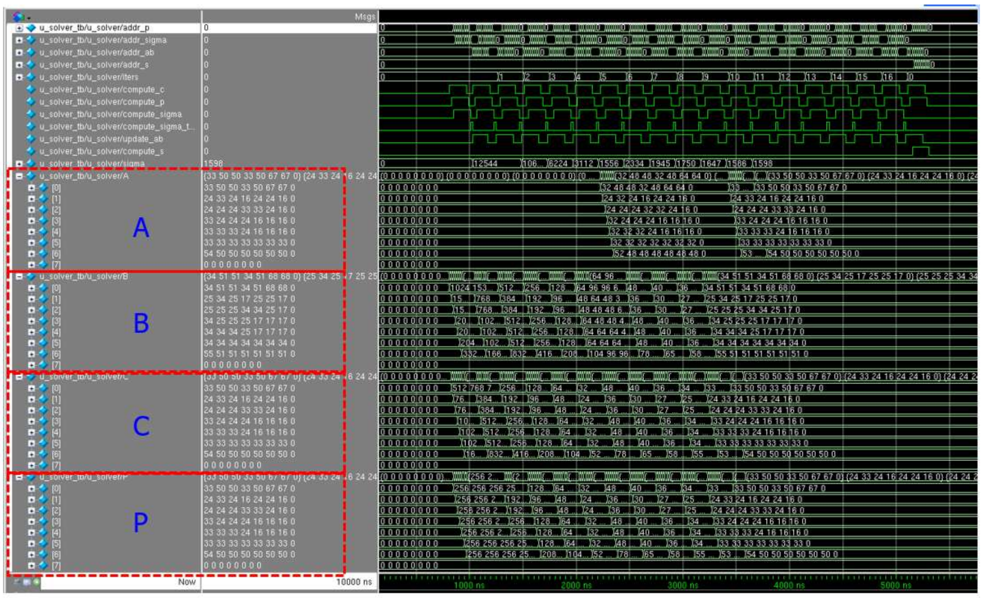


Fig. 6.3. Block diagram to solve the gradient-based sampling problem.



```
u_solver_tb/u_solver/addr_p                0
u_solver_tb/u_solver/addr_sigma            0
u_solver_tb/u_solver/addr_ab               0
u_solver_tb/u_solver/addr_s                0
u_solver_tb/u_solver/iters                 0
u_solver_tb/u_solver/compute_c             0
u_solver_tb/u_solver/compute_p             0
u_solver_tb/u_solver/compute_sigma         0
u_solver_tb/u_solver/compute_sigma_t...    0
u_solver_tb/u_solver/update_ab             0
u_solver_tb/u_solver/compute_s             0
u_solver_tb/u_solver/sigma                 1598
u_solver_tb/u_solver/A                     {33 50 50 33 50 67 67 0} {24 33 24 16 24 24
  [0]                                      33 50 50 33 50 67 67 0
  [1]                                      24 33 24 16 24 24 16 0
  [2]                                      24 24 24 33 33 24 16 0
  [3]                                      33 24 24 24 16 16 16 0
  [4]                                      33 33 33 24 16 16 16 0
  [5]                                      33 33 33 33 33 33 33 0
  [6]                                      54 50 50 50 50 50 50 0
  [7]                                      0 0 0 0 0 0 0 0
u_solver_tb/u_solver/B                     {34 51 51 34 51 68 68 0} {25 34 25 17 25 25
  [0]                                      34 51 51 34 51 68 68 0
  [1]                                      25 34 25 17 25 25 17 0
  [2]                                      25 25 25 34 34 25 17 0
  [3]                                      34 25 25 25 17 17 17 0
  [4]                                      34 34 34 25 17 17 17 0
  [5]                                      34 34 34 34 34 34 34 0
  [6]                                      55 51 51 51 51 51 51 0
  [7]                                      0 0 0 0 0 0 0 0
u_solver_tb/u_solver/C                     {33 50 50 33 50 67 67 0} {24 33 24 16 24 24
  [0]                                      33 50 50 33 50 67 67 0
  [1]                                      24 33 24 16 24 24 16 0
  [2]                                      24 24 24 33 33 24 16 0
  [3]                                      33 24 24 24 16 16 16 0
  [4]                                      33 33 33 24 16 16 16 0
  [5]                                      33 33 33 33 33 33 33 0
  [6]                                      54 50 50 50 50 50 50 0
  [7]                                      0 0 0 0 0 0 0 0
u_solver_tb/u_solver/P                     {33 50 50 33 50 67 67 0} {24 33 24 16 24 24
  [0]                                      33 50 50 33 50 67 67 0
  [1]                                      24 33 24 16 24 24 16 0
  [2]                                      24 24 24 33 33 24 16 0
  [3]                                      33 24 24 24 16 16 16 0
  [4]                                      33 33 33 24 16 16 16 0
  [5]                                      33 33 33 33 33 33 33 0
  [6]                                      54 50 50 50 50 50 50 0
  [7]                                      0 0 0 0 0 0 0 0
Now                                        10000 ns
```

Fig. 6.4. Captured waveform by the solver module.

6.2. System overview

Our LiDAR system prototype are presented in Fig. 6.5. The system consists of following main components: laser source, galvanometer, optical lenses with band-pass filter, detector circuit, LiDAR controller module with FPGA and a PC. The simplified flow from a laser source to a detector is illustrated in Fig. 6.6. Captured images at resolutions of 80×80, 160×160 and 320×320 are presented in Fig. 6.7.

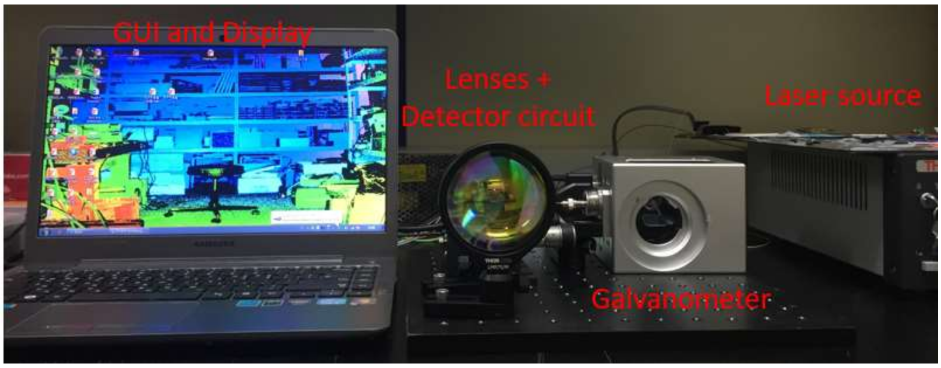


Fig. 6.5. LiDAR system prototype.

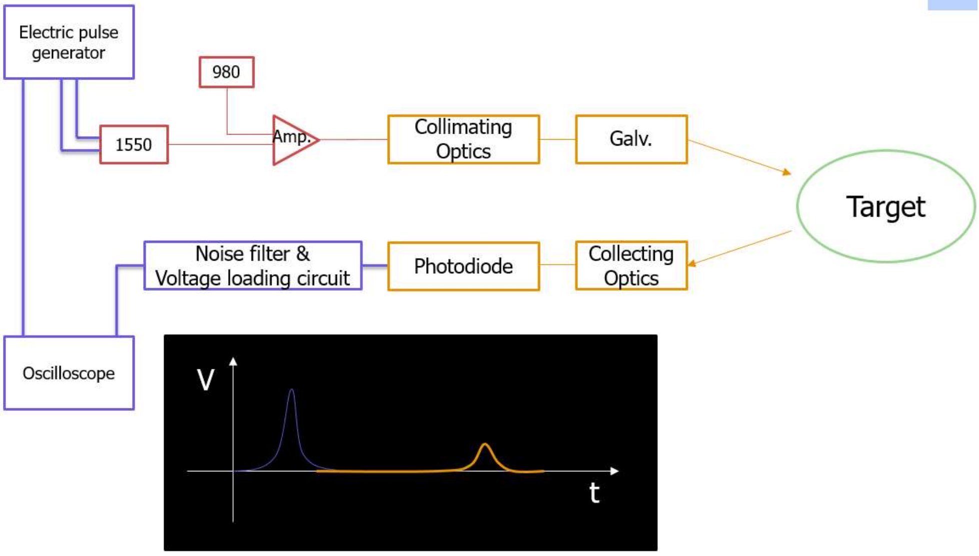
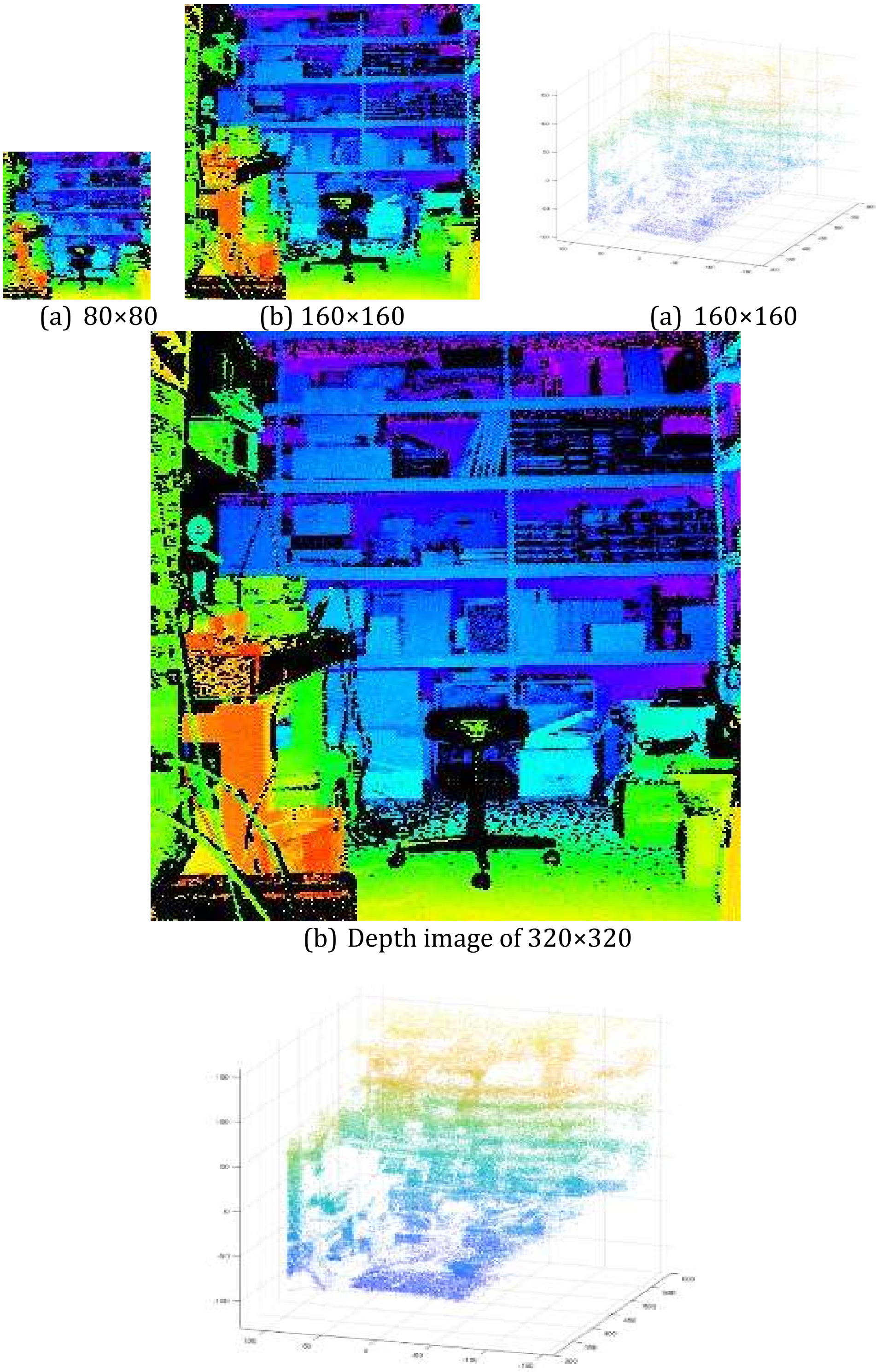


Fig. 6.6. Simplified flow for laser and detection modules.



(a) 80×80

(b) 160×160

(a) 160×160

(b) Depth image of 320×320

(c) Pointcloud of 320×320

Fig. 6.7. Captured images at resolutions of 80×80, 160×160 and 320×320.

Chapter 7: Conclusion

This thesis investigates dual-mirror LiDAR imaging under three aspects: System, Sampling model, and Sampling algorithm.

Firstly, by replacing the conventional scanning scanners by dual-mirror ones, we demonstrates that a LiDAR can obtain different scanning patterns for a given timing budget. When sampling points form a grid graph, a sampling problem is defined, and its optimal solution is obtained by exploiting a minimum spanning tree concept. Hence, the relationship between an FOV of a captured image and the frame rate is derived. Sufficient conditions are derived to check whether the obtained image fully covers the FOV and includes well-aligned objects for given frame rates. Experimental results show that the scanners can achieve frame rates of 17.6, 9.0 and 4.6 fps for image sizes of 240×16, 240×32 and 240×64, respectively.

Secondly, this study proposes a new mathematical formulation of the constraints for a practical sampling method in a LiDAR system. Based on the proposed mathematical formulation, it is shown that existing two-stage sampling approaches are not suitable for a practical LiDAR system. Therefore, this study presents a novel sampling method to efficiently perform non-uniform random sampling. The proposed method performs uniform sampling at the pilot stage and non-uniform sampling at the refinement stage. However, unlike in the previous methods, the proposed technique efficiently derives non-uniform sampling based on the gradient of the downsampled image. Consequently, the proposed method follows implicit global properties notwithstanding decision-making using a greedy approach. More critically, the proposed method for computing the gradient and refinement-sampling map is substantially faster than other methods because it does not require intermediate reconstruction. Consequently, it is computationally efficient with $\mathrm{O}(n)$ complexity for n image pixels. In addition, the proposed method reduces the requirement of additional memory (or bandwidth) to store (or transmit) the sampling pattern. To this end, the proposed method outperforms grid sampling by at most 5.92 dB. As a result, the proposed sampling achieves a reconstructed quality that is similar to the optimal sampling in the previous design, while substantially reducing the computation time and memory requirements.

Thirdly, this study addresses the LiDAR sampling problem when an RGB image and regions of interest are provided. A systematical framework of depth acquisition in an on-road environment is presented. Unlike previous approach, our proposed detects the objects in a road and segments a scene into background, ROI, and object areas. From the segmentation result, the approach distributes samples across the segmented areas. An ROI-based sampling problem is proposed to optimize a depth sensing system in a LiDAR for an on-road environment. The optimization problem has an optical solution, which effectively addresses the two problems of the LiDAR discussed above. Experimental results demonstrate that the proposed approach significantly reduces the mean-absolute-error (MAE) in the object area by at most 52.8%. Moreover, it achieves robust reconstruction quality at a very low sampling rate of 1%. In addition, the proposed sampling is remarkably fast (i.e., within a few milliseconds), rendering it applicable to a real-time acquisition system.

References

[1] "The Economist Special Report," [Online]. Available: https://www.economist.com/news/special-report/21737425-foreseen-and-unforeseen-consequences-self-driving-cars-will-profoundly-change-way. (Accessed on 11.05.2018).

[2] S. Thrun, et al., "Stanley: The robot that won the DARPA Grand Challenge: Research Articles," *Journal of Robotic Systems,* vol. 23, no. 9, pp. 661-692, 2006.

[3] C. Urmson, et al., "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robots*, vol. 25, no 8, pp. 425-466, 2008.

[4] J. Levinson, et al., "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium* (*IV*), 2011.

[5] U. Lee, J. Jung, S. Shin, Y. Jeong, K. Park, D. H. Shim, and I.-S. Kweon, "EureCar turbo: A self-driving car that can handle adverse weather conditions," in *International Conference of Intelligent Robots and Systems* (*IROS*), 2016.

[6] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: review and future perspectives," *IEEE Intelligent Transportation Systems Magazine,* vol. 6, p. 6–22, 2014.

[7] E. Ackerman, "Lidar that will make self-driving cars affordable [News]," *IEEE Spectrum*, vol. 53, no. 10, pp. 14-14, 2016.

[8] H. Deilamsalehy, and T. C. Havens, "Sensor fused three-dimensional localization using IMU, camera and LiDAR," in *IEEE Sensors 2016 Proceedings,* 2016.

[9] A. Geiger, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2012.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Robotics Research,* vol. 32, no. 11, pp. 1231–1237, 2013.

[11] D. Scharstein, et. al, “Learning conditional random fields for stereo,” in *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*)*,* Jun. 2007.

[12] H. Hirschmüller, et. al, “Evaluation of cost functions for stereo matching,” in *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*)*,* Jun. 2007.

[13] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, “On building an accurate stereo matching system on graphics hardware,” in *2011 IEEE International Conference on Computer Vision Workshops* (*ICCV’11*), Nov. 2011, pp. 467–474.

[14] B. Schwarz, "Lidar: Mapping the world in 3D," *Nature Photon.*, vol. 4, pp. 429-430, 2010.

[15] D. S. Hall., “High definition LiDAR system,” *U.S. Patents 7 969 558 B2*, issued date: June 28, 2011.

[16] J. Scholz, V. Willhoeft, Dr. R. Schulz, T. Kluge, (2006) “ALASCA User Manual,” *Ibeo Automobile Sensor GmbH*, Hamburg. [Online] Available: http://www.raginbot.com/wiki/images/d/d8/Manual_ALASCA.pdf (Accessed on 18.04.2017).

[17] LeddarTech Inc. Leddar® (2017, May), "LEDDAR M16: Multi-element sensor module," *LeddarTech HQ*, Quebec, Canada. [Online] Available: http://leddartech.com/modules/m16-multi-element-sensor-module/ (Accessed on 18.04.2017).

[18] RIEGL USA Inc. (2017), "VUX-1UAV: Unmaned Laser Scanners for UAS/UAV-Based Laser Scanning," *RIEGL*, USA Inc. Florida. [Online] Available: http://products.rieglusa.com/product/all-categories-unmanned-scanners/vux-1uav-survey-grade-unmanned-laser-scanners (Accessed on 18.04.2017).

[19] Velodyne LiDAR Inc. (2017), "Velodyne LiDAR Puck™: Real-time 3D LiDAR Sensor," Velodyne LiDAR, Inc. CA. [Online] Available: http://velodynelidar.com/vlp-16.html (Accessed on 18.04.2017).

[20] Velodyne LiDAR Inc. (2017), "Velodyne LiDAR HDL-32E: High Definition Real-time 3D LiDAR Sensor," Velodyne LiDAR, Inc. CA. Available: http://velodynelidar.com/hdl-32e.html (Accessed on 18.04.2017).

[21] Velodyne LiDAR Inc. (2017), "Velodyne LiDAR HDL-64E S3: High Definition Real-time 3D LiDAR Sensor," Velodyne LiDAR, Inc. CA. [Online] Available: http://velodynelidar.com/hdl-64e.html (Accessed on 18.04.2017).

[22] SCANLAB (2006), "The RTC 4 PC inteface board for real time control of scan heads and lasers," SCANLAB AG, Puchheim, Germany. [Online] Available: rlab.org.uk/mediawiki/images/3/3e/RTC4.pdf (Accessed on 18.04.2017).

[23] M.-F. Chen, Y.-P. Chen, W.-T. Hsiao, S.-Y. Wu, C.-W. Hu, and Z.- P. Gu, "A scribing laser marking system using DSP controller," *Optics and Lasers in Engineering*, vol. 46, no. 5, pp. 410–418, 2008.

[24] W. Dongyun, and Y. Xinpiao, "An embedded laser marking controller based on ARM and FPGA processors," *The Scientific World Journal*, 2014 pp.1–7, 2014.

[25] D. Wang, Q. Yu, and Y. Zhang, "Research on Laser Marking Speed Optimization by Using Genetic Algorithm," *PLoS ONE*, vol. 10, no. 5, 2015.

[26] S. Hawe, M. Kleinsteuber, and K. Diepold, "Dense disparity maps from sparse disparity measurements," in *2011 International Conference on Computer Vision* (*ICCV'11*), Nov. 2011, pp. 2126–2133.

[27] L.-K. Liu, Z. Lee, and T. Q. Nguyen, "Sharp disparity reconstruction using sparse disparity measurement and color information," in *IVMSP Workshop*, Sep. 2013.

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

[29] S. Schwartz, A. Wong, and D. A. Clausi, "Multi-scale saliency guided compressive sensing approach to efficient robotic laser range measurement," in *2012 Ninth Conference on Computer and Robot Vision,* 2012, pp. 1–8.

[30] S. Schwartz, A. Wong, and D. A. Clausi, "Saliency-guided compressive sensing approach to efficient laser range measurement," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 160–170, 2013.

[31] L.-K. Liu, and T. Q. Nguyen, “A Framework for Depth Video Reconstruction from a Subset of Samples and Its Applications,” *IEEE Transactions on Image Processing*, vol. 25, no. 10, Oct. 2016.

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

[33] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[34] E. J. Cand`es, and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[35] A. Beck, and M. Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183-202, 2009.

[36] Zhai, G. and X. Yang, "Image reconstruction from random samples with multiscale hybrid parametric and nonparametric modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 11, pp. 1554-1563, 2012.

[37] G. A. Kordelas, D. S. Alexiadis, P. Daras, and E. Izquierdo, “Contentbased guided image filtering, weighted semi-global optimization, and efficient disparity refinement for fast and accurate disparity estimation,” *IEEE Transactions on Multimedia*, vol. 18, no. 2, pp. 155–170, Feb. 2016.

[38] F. Ma, L. Carlone, U. Ayaz, and S. Karaman, “Sparse sensing for resource constrained depth reconstruction,” in *IEEE International Conference on Intelligent Robots and Systems* (*IROS*), 2016, pp. 96–103.

[39] F. Ma, L. Carlone, U. Ayaz, and S. Karaman, “Sparse depth sensing for resource-constrained robots,” *Arxiv*. [Online] Available: https://arxiv.org/abs/1703.01398. (Accessed on 31.10.2017).

[40] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox and A. Geiger, “Sparsity Invariant CNNs,” in *International Conference on 3D Vision* (3DV) 2017.

[41] N. Chodosh, C. Wang, and S. Lucey,”Deep convolutional compressed sensing for lidar depth completion,” *arXiv.* [Online] Available preprint arXiv:1803.08949, 2018. (Accessed on 31.10.2018).

[42] M. Jaritz, R. Charette, E. Wirbel, X. Perrotton and F. Nashashibi, “Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation,” in *International Conference on 3D Vision* (*3DV*) 2018, 2018.

[43] F. Ma et. *al* “Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image”, in *International Conference on Robotics and Automation* (*ICRA*), 2018.

[44] SINO-GALVANOMETER Inc. (2015), “SG8220 Galvanometer Scanner,” Sino-Galvanometer Technology Co., Ltd, Beijng, China. [Online] Available: http://www.sino-galvo.com/ypros-show-124- 488.aspx.

[45] Newson Engineering (2007), “rhothor™ X7 I/O configuration: XY2-100 Technical Datasheet,” Newson NV, 9200 Dendermonde, Belgium. [Online] Available: “http://www.newson.be/doc.php?id=XY2-100

[46] R. Graham and P. Hell, “On the History of the Minimum Spanning Tree Problem,” *Annual History of Computing*, vol. 7, no. 1, pp. 43-57, 1985.

[47] R. C., Prim,"Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol.36, no.6, pp. 1389-1401, 1957.

[48] Raspberry Pi board, [Online] Available: https://www.raspberrypi.org/

[49] LOGI-PI FPGA board, [Online] Available: http://valentfx.com/logi-pi/

[50] L. Parziale, D.T. Britt, C. Davis, J. Forrester, W. Liu, C. Matthews, and N. Rosselot, “Application structure and programming interfaces,” *TCP/IP Tutorial and Technical Overview*, Redbooks, 8th edition, Dec. 2006, pp. 407-423.

[51] TI Inc. (2017), “TDC7200: Time-to-Digital for Time-of-Flight Applications in LiDAR, Magnetostrictive and Flow Meters,” TI Inc., Texas. [Online] Available: http://www.ti.com/product/TDC7200

[52] APD Avalanche Photodetector, [Online] Available: https:// www.thorlabs.com/newgrouppage9.cfm?objectgroup_ID=4047

[53] T. Fersch, et al. "A CDMA Modulation Technique for Automotive Time-of-Flight LiDAR Systems," *IEEE Sensors Journal,* vol.17, no. 11, pp. 3507-3516, 2017.

[54] D. Perenzoni, et al. “ Depth-range extension with folding technique for SPAD-based TOF LIDAR systems,” In *IEEE Sensors Proceedings*, 2014.

[55] The Brown University pattern theory group. Brown range image database. <http://www.dam.brown.edu/ptg/brid/range/index.html>

[56] X. T. Nguyen, V. L. Dinh, H.-J. Lee, and H. Kim, "A High-Definition LIDAR System Based on Two-Mirror Deflection Scanners," *IEEE Sensors Journal,* vol. 18, no. 2, pp. 559–568, Jan. 2018.

[57] T. K. Nguyen, X. T. Nguyen, and H-J. Lee, "An ROI-Based Lidar Sampling Algorithm in the Road Environment," in *Asia Pacific Conference on Information Communication Technology* (*APCICT*), Jul. 2018.

[58] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016.

[60] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016.

[61] J. Redmon, and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2017.

[62] G. Oliveira, W. Burgard, and T. Brox, "Efficient Deep Methods for Monocular Road Segmentation," in *International Conference of Intelligent Robots and Systems* (*IROS*) , 2016.

[63] Z. Chen, and Z. Chen, "RBNet: A Deep Neural Network for Unified Road and Road Boundary Detection," in *International Conference on Neural Information Processing* (*NIPS*), 2017.

[64] "KITTI road benchmark," KITTI road benchmark, [Online] Available: http://www.cvlibs.net/datasets/kitti/eval_road.php (Accessed on Oct. 31, 2017).

초록

최근 LIDAR (light detection and ranging)와 같은 능동적 센서 기술은 이론적으로도 집중적으로 연구되었고, 자율주행차, 로봇, 센싱 등 다양한 응용 분야에 널리 사용되고 있다. 일반적으로 LiDAR 센서와 같은 심도 측정장치는 느린 속도 때문에 공간적 해상도가 제한된다. 제한된 공간적 해상도로부터 심도 이미지를 정확하게 재구성하기 위해서 2단계 샘플링 방법이 널리 사용되고 있다. 그러나 2단계 샘플링은 불규칙적인 샘플링 패턴으로 샘플링을 하기 때문에, 재구성 과정에 많은 양의 연산이 필요하다. LiDAR 시스템을 수학적인 모델을 사용하여 분석하였을 때, 기존의 2단계 샘플링은 실용적으로 사용되기 위한 타이밍 제약 조건을 만족하지 못함을 확인하였다. 따라서 효율적인 샘플링 알고리즘을 사용하는 LiDAR 시스템을 설계하는 것은 중요한 기술적 과제이다.

첫째, 본 논문은 최신의 레이저 마킹 시스템을 dual-mirror 스캐너에 적용하여 고해상도 LiDAR 시스템을 만드는 문제를 다룬다. Galvanometer 스캐너 컨트롤러와 잘 알려진 래스터 스캔 방법에 기초하여 Galvanometer 스캐너를 모델링, 매개변수화한다. 그리고 물리적인 스캐닝 움직임과 최소 신장 트리를 고려하여 스캐닝 방법을 모델링하고 분석한다. 분석으로부터 원하는 FOV (field of view)로 캡쳐된 이미지의 품질과 스캐닝 속도 사이의 관계를 밝혔다. 또한 획득된 이미지가 FOV를 완전히 표현하며, 캡쳐된 object들이 특정 프레임 레이트에서 잘 정렬됨을 나타내는 충분조건을 유도하였다. 마지막으로 제안된 개념을 확인하기

위해 샘플 LIDAR 시스템을 개발하였다.

둘째, 2단계 샘플링의 단점을 극복하기 위해, 다운 샘플 데이터에서 샘플링 패턴의 최적 표현을 생성함으로써 연산 복잡도와 메모리 요구량을 줄일 수 있는 새로운 샘플링 방법을 제안한다. 샘플링 패턴은 다운 샘플된 표현의 k-NN 확장 연산으로부터 도출된다. 제안된 방법은 물체 경계 또는 복잡한 텍스처에 한해서 확장연산을 수행함으로써 물체 경계를 보존하도록 설계되었다. 또한 제안하는 방법은 선형적인 시간 복잡도로 동작하며 다운 샘플링 비율을 이용하여 메모리 요구량을 줄인다. Middlebury 데이터셋과 Brown laser-range 데이터셋을 사용한 실험 결과가 제시된다.

셋째, 2단계 샘플링과 같은 최신의 적응적 방법들은 비교적 낮은 샘플링 레이트로 실내의 복잡하지 않은 장면들을 처리하는 데 매우 효과적이다. 그러나 복잡한 도로 환경에서는, 특히 측정 장비의 샘플링 레이트가 낮은 경우에, 해당 방법들의 성능이 상대적으로 떨어진다. 이 문제를 해결하기 위해 본 논문은 자율주행을 위한 도로 환경에서의 ROI (region-of-interest) 기반 샘플링 알고리즘을 제안한다. 제안된 샘플링 알고리즘은 CNN (convolutional neural network) 기반의 빠르고 정확한 도로 및 물체 감지 알고리즘을 사용하여, semantic 정보를 활용하고 도로, 물체, 배경 영역에 샘플들을 효과적으로 분배한다. KITTI 데이터셋을 사용한 실험 결과가 제시된다.

주요어 : 라이다, 스캐닝 샘플링 패턴, 샘플링, 재구성.
학 번 : 2015-30752