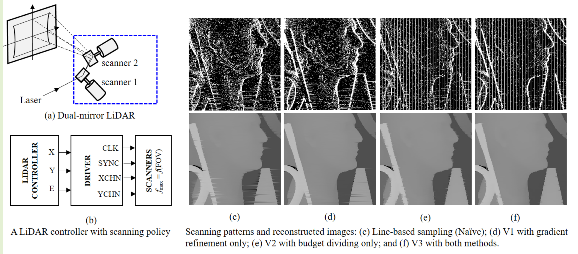
A Gradient-Aware Line Sampling Algorithm for LiDAR Scanners

Xuan Truong Nguyen, Hyun Kim, *Member, IEEE*, and Hyuk-Jae Lee, *Member, IEEE*

***Abstract*—Due to a slow acquisition, the spatial resolution of a depth-acquisition device, such as a Light Detection and Ranging (LiDAR) sensor, is limited, which strictly requires scanning a field of view (FOV) in a particular order. To accurately reconstruct a depth image from limited spatial samples, two-stage sampling has been widely used. However, from the perspective of a LiDAR, two-stage sampling requires scanning the FOV two times to build a depth map, and eventually becomes counterintuitive in practice. To address this problem, this study presents a LiDAR framework and a gradient-aware line-based sampling algorithm. Different from previous works, the proposed scanning algorithm allows a LiDAR to scan the FOV from top to bottom while simultaneously seeking sampling points along object boundary. By utilizing the information of the previous line during scanning, the proposed method maintains the conventional scanning order in a LiDAR, while efficiently predicting the sampling locations of the current line. The experimental results demonstrate that the proposed sampling outperforms grid sampling by at most 9.79 dB on the synthetic datasets. Consequently, the proposed sampling achieves reconstructed quality similar to that of optimal sampling in the previous design, while substantially reducing the computation time and memory requirements. The experimental results with the laser range data and the real data captured by the LiDAR system demonstrate that the proposed method can reduce the averaged mean-absolute-error (MAE) by 34.91%, 47.23%, 54.88%, and 57.99% for the sampling ratios of 20%, 30%, 40%, and 50%, respectively, compared to the conventional LiDAR sampling.**



(a) Dual-mirror LiDAR

(b) A LiDAR controller with scanning policy

(c) (d) (e) (f) Scanning patterns and reconstructed images: (c) Line-based sampling (Naïve); (d) V1 with gradient refinement only; (e) V2 with budget dividing only; and (f) V3 with both methods.

***Index Terms*—Compressive and non-uniform sampling, compressive sensing, depth data acquisition, light detection and ranging (LiDAR), sparse representation.**

I. Introduction

ACTIVE sensor technologies such as light detection and ranging (LiDAR) have been intensively studied in theory and widely adopted in many applications such as self-driving cars, robotics, and sensing [1]–[7]. To measure distances, a LiDAR estimates a time interval at which light photons are emitted, reflected by the object, and the reflected signals reach the detector. According to the number of emitter and detector pairs, LiDAR systems can be categorized into two types: those, which use one pair, and those, which use multiple pairs. Both types have their own advantages and disadvantages. The commercial products, Velodyne LiDARs [8], are motivated by the use of multiple emitter/detector channels in the vertical direction simultaneously. As those channels are rotated by a motor, the system can capture an image with multiple lines of data. For example, 16-, 32-, and 64-channel versions can produce 16, 32, and 64 lines of point cloud data, corresponding to 300,000, 700,000, and 1,200,000 measurements per second, respectively. 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 use of multiple channels. Moreover, increasing the number of vertical channels is trivial because the physical area and the cost increase proportionally. On the other hand, the single-channel LiDAR system is widely used in laser marking systems [9], [10] or robotic applications [11]. Because there is only one emitter and detector pair, the system is easy to control and flexible when used to scan an arbitrarily point in a field of view (FOV). Moreover, the price of this system is relatively low. Therefore, it can be used to scan various patterns and to increase the vertical resolution of the scans. However, the scanning speed and frame rate are not very high; hence, it takes a few seconds to obtain patterns in a laser marker. To this end, the single-channel LiDAR generally suffers from

Manuscript received February 6, 2020; revised April 4, 2020; accepted April 5, 2020. Date of publication April 9, 2020; date of current version July 17, 2020. This work was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology). The associate editor coordinating the review of this article and approving it for publication was Dr. Yen Kheng Tan. *(Corresponding author: Hyun Kim.)*

Xuan Truong Nguyen and Hyuk-Jae Lee are with the Inter-University Semiconductor Research Center, Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: truongnx@capp.snu.ac.kr; hyuk_jae_lee@capp.snu.ac.kr).

Hyun Kim is with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea, and also with the Research Center for Electrical and Information Technology, Seoul National University of Science and Technology, Seoul 01811, South Korea (e-mail: hyunkim@seoultech.ac.kr).

Digital Object Identifier 10.1109/JSEN.2020.2986819

a slow data acquisition, which motivates to use a sampling algorithm to find locations for measuring.

The objective of this paper is to design a scanning policy for a dual-mirror single-channel LiDAR that can actively improves the sparse-to-dense reconstruction performance. The key idea is to carefully select a sparse subset of spatial samples and to use an interpolation method for reconstructing the final dense depth map. The focus of this paper lies in the intersection of two related subjects: LiDAR sensors and compressed sensing.

A LiDAR scanning system scans an FOV, and then measures ranges for building a point-cloud or depth map. LEDDAR 16M [12] 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 [13] uses a single pair of an emitter and a detector that 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 [14] also adopts a similar rotating method, but it uses four channels; therefore, scan a vertical angle of 3.2°. Velodyne LIDAR sensors [15]–[17] also use a rotation module to extend the vertical scanning angle. It should be noted that they increase the number of measurements proportionally by increasing the number of emitter/detector channels. Interestingly, the focus of those LiDAR sensors is grid or random sampling patterns so that an FOV is presented as a fixed-resolution depth map. In [18], Nguyen *et al.* presents a resolution-adjustable LiDAR prototype by utilizing dual-mirror scanners. However, similar to other commercial products, this work only considers on grid sampling patterns. These LiDAR systems suggest that it is not straightforward to utilize the compressive sampling theory into a practical LiDAR.

Compressed sensing is a popular mathematical framework for sampling and signal recovery [19], [20]. Different from the practical LiDAR systems, theory of CS investigates the representation of natural and depth images to exhibit sparse structures in certain domains, e.g., wavelet. Motivated by the property of the wavelet transform that the relevant coefficients coincide with discontinuities, Hawe *et al.* [21] recommend that a sampling pattern should consist of sampling locations at the discontinuities or along gradients. However, this approach is not practical for two reasons. First, the gradient of a disparity map is not available prior to sampling. Therefore, all the gradient information should be inferred from the color image. Second, the gradient of a color image could be significantly different from that of the disparity map. Liu *et al.* [22] suggest using outlier elimination prior to the edge disparity estimation. Schwartz *et al.* [23], [24] propose a saliency-guided sampling approach to perform sampling in a two-stage manner. Firstly, 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. Similarly, Liu *et al.* [25] propose a two-step sampling. 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 [21]–[25] involve time-consuming rough disparity estimation. Nguyen *et al.* [26] proposes a two-stage method using k-NN expanding operator. This method directly computes a refinement sampling map from the gradient of a down-sampled image obtained in the 1st stage, which does not require a time-consuming reconstruction. Although this method is fast and significantly improves the reconstruction quality, it requires scanning an FOV twice as the conventional two-stage sampling method. Nguyen *et al.* [27] proposes an efficient sampling method by utilizing the semantic information of road, object and background areas from an RGB image.

To address the gap between a practical scanning system and a compressed sampling pattern, this study presents a framework and a sampling strategy to maintain the natural scanning order of a LiDAR and to improve the reconstruction performance. By utilizing the information of the previous line during scanning, the proposed method efficiently predicts the sampling locations of the current line. The experimental results demonstrate that the proposed sampling outperforms grid sampling by at most 9.79 dB on the synthetic datasets. As a result, the proposed sampling significantly reduces computation time and memory requirements while achieving reconstructed quality similar to that of optimal sampling. In addition, the experimental results with the laser range data and the real data captured by the LiDAR system demonstrate that the proposed method can reduce the reconstruction error by 34,91%, 47,23%, 54,88%, and 57,99% for the sampling ratios of 20%, 30%, 40%, and 50%, respectively, compared to the conventional LiDAR sampling.

The major contributions of this paper can be summarized as follows:

1) *Modeling:* The scanning procedure is described under the time and scanning order constraints in a dual-mirror LiDAR. The sampling procedure in a LiDAR is modelled and combined with a reconstruction method. Moreover, the timing constraint in a LiDAR and a sampling procedure is clearly described.

2) *Sampling:* An efficient spatial sampling strategy is proposed to maximize the reconstruction performance while maintaining the natural scanning order in a LiDAR. In particular, for a fixed sampling budget for each scanning line, a sampling pattern can be obtained by allocating random samples with probabilities in proportional to the magnitudes of depth gradients.

The rest of this paper as follows. Section II describes a LiDAR and scanning problem. The proposed sampling algorithm is presented in Section III, and the experimental results are shown in Section IV. Finally, Section V draws a conclusion.

II. LiDAR Scanning Problem

A. Background on a Dual-Mirror LiDAR

A typical structure of a dual-mirror LiDAR sensor is shown in Fig. 1 where two scanners control two motors to move their corresponding mirrors targeting a specific position in the FOV. Thanks to its simple structure, as shown in Fig. 1(a), it is

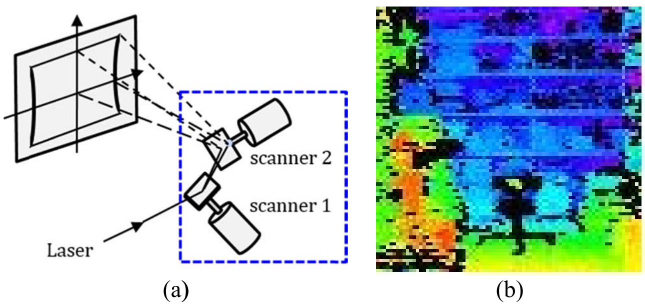


Fig. 1. Scanning an FOV with a dual-mirror LiDAR from [20]. (a) Scanning structure. (b). Generated 3D depth image.

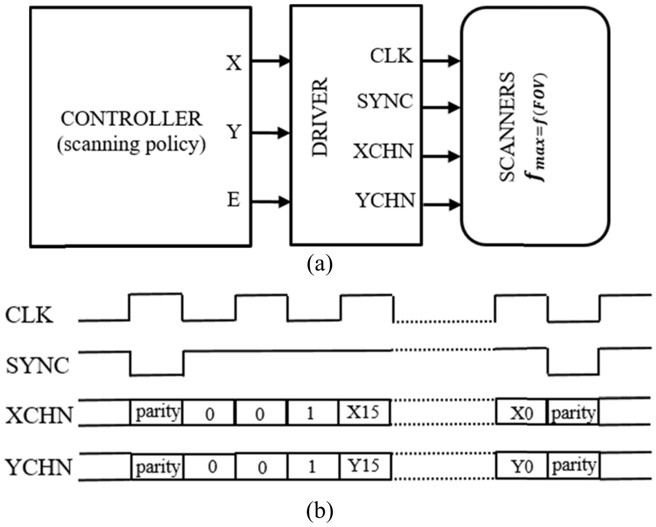


Fig. 2. Scanner controller. a) Block diagram of a LiDAR controller with scanning policy; (b) XY2 industrial protocol to communicate the controller and scanners.

utilized into a dual-mirror LiDAR system which scans an FOV in a snake scanning to build a 3D depth image in Fig. 1(b). Each mirror is responsible for scanning a direction in the FOV.

To control the scanners, 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 that of positions. Obviously, this number depends on how many new positions are sent to the scanner. Consider the case of two scanners are interfaced by an interface (i.e., XY2-100 industrial protocol [12]). An illustration of the protocol is shown in Fig. 2(a). 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 $10\mu s(= 20 \times 500ns)$. This implies that the maximum number of updates is limited by 100,000 positions.

A LiDAR usually operates by performing multiple point-wise measurements in an FOV. A block diagram of a LiDAR is illustrated in Fig. 3. 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. 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. After the mirror is aimed at the target, the laser diode in the LiDAR system emits a laser beam. 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. In the last step, the result is transmitted to the main controller. It is crucial to determine sampling locations for its controller, which is known as the sampling in LiDAR.

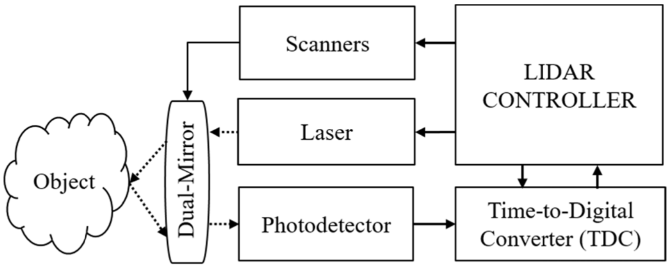


Fig. 3. Block diagram of a typical LiDAR.

B. LiDAR Scanning Problem and an Optimal Solution

Definition 1 (Scanning Problem [20]): 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, $q_1 \rightarrow q_2 \rightarrow \cdots \rightarrow q_N$, where the set $\left\{q_j\right\}_{j=1}^N$ 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 $\left\{q_j\right\}_{j=1}^N$ 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 problem 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 $q_1 \rightarrow q_2 \rightarrow \cdots \rightarrow q_N$ that minimizes the total length of the traversing path as follows:

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

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

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

C. Sampling Problem

Let $x \in R^N$ be a $N \times 1$ vector representing the depth map of an entire scene in an FOV of a capturing device such as LiDAR. For straightforwardness, x_i is normalized such that $0 \le x_i \le 1$ 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 $\{1,\dots, N\}$ denotes the set of indexes that correspond to the locations of the entire FOV, while $\{i_1,\dots, i_{\mathrm{M}}\}$ represents the set of the indexes that correspond to the sample locations among $\{1,\dots,N\}$.

Problem 2 (*Sampling problem*): The sampling problem is to derive $\{i_1,\dots, i_{\mathrm{M}}\}$ to minimize the following objective function:

$$\frac{1}{N}\sum_{j=1}^{N}\left(x_j - \tilde{x_j}\right)^2 \tag{3}$$

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

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

$$S_i = \begin{cases} 1, & \text{with probability} p_i, \\ 0, & \text{with probability} 1 - p_i, \end{cases} \tag{4}$$

where $0 \le p_i \le 1$ for $i = 1, \dots, N$ is a set of probabilities. It is obvious that finding the probabilities. Given a number of sample budget M, the constraint on the probabilities is as follows:

$$\sum_{i=1}^{N} p_i = M. \tag{5}$$

In practice, the absolute sampling budget M is usually replaced by a sampling ratio ξ under the following relationship:

$$M = N\xi. \tag{6}$$

Assume that $\{g_j\}_{j=1}^{N}$be the probabilities of sampling points and it is expected to find $\{p_j\}_{j=1}^{N}$ which has a similar distribution as that of $\{g_j\}_{j=1}^{N}$. Intuitively, p_j is likely to be proportional to g_j, and smaller or equal to one, so that it is defined as follows:

$$p_j = \min\left(\tau g_j, 1\right) \quad for\ j = 1, \dots, N. \tag{7}$$

where τ is a non-negative parameter. From (4) and (5), τ is defined as a solution of the following equation:

$$\sum_{j=1}^{N} \min(\tau g_j, 1) = M. \tag{8}$$

Let $f(\tau) = \sum_{j=1}^{N} \min(\tau g_j, 1) - M$, then the problem is to find a solution of $f(\tau) = 0$. Note that $f(\tau)$ is a piecewise linear and monotonically increasing function with $f(0) = -M < 0$ and $f(+\infty) = N - M > 0$, so that τ can be uniquely determined as the root of $g(\tau)$. Moreover, an efficient solution to derive τ is available. The derivation of (7) is presented in Appendix.

III. Single-Scan Sampling Algorithm in LiDAR

A. Line-Scanning LiDAR Sampling Algorithm

For a LiDAR, it is important to maintain the scanning order in which a LiDAR scans an FOV line by line from the top to the bottom. The challenge is that the gradient information of the depth map is not available, especially when the scanning order is required. Therefore, a predictive-update sampling process is proposed. The proposed scheme consists of two stages – an initialization stage for the first scanning line and an update and prediction stage for the remaining scanning lines. For the first line, M samples are selected using an uniformly random or grid sampling pattern. Correspondingly, a sampling location matrix S is fed to a LiDAR, and the sampled data b is obtained. Given S and b, an interpolation method is used to estimate x.

In the second stage, it performs a loop in which an estimate of previous line is used as a guide to compute the gradient. Each loop consists of six steps. At first, the gradient of the previous line is computed. By (7), this suggests that the optimal sampling probabilities is $p_j = \min\left(\tau\left[\nabla x^{r-1}\right]_j, 1\right)$. Given the computed probabilities, the sampling locations of r-th line $S^{(r)}$ are derived. After LiDAR measures and obtains the distances corresponding to $S^{(r)}$, an interpolation method is used to estimate $x^{(r)}$. The overall method is summarized in Alg. 1.

A comparison of the proposed sampling pattern and grid one is shown in Fig. 4. Setting the sampling ratio $\xi = 0.25$ (i.e., $\xi = 0.25$), the problem is to find 64 sampling locations out of 256 possible one. A Monte-Carlo simulation by repeating 32 independent trials and averaging peak signal-to-noise ratios (PSNRs) are calculated in Table I. Figs. 4(a) and (b) are a grid sampling pattern and its reconstructed image, respectively, whereas Figs. 4(c) and (d) show a line-based sampling pattern and its reconstructed image, respectively. These results indicate that for the same sampling ratio, the choice of sampling pattern has a strong impact on the reconstruction quality. In Table I, as compared to the grid sampling, a line-based sampling has about 1.58 dB improvement. In particular, the proposed sampling method greatly reduces the reconstruction artifact along object boundary by adding more sampling points on that area. As shown in Fig. 4(c), the sampling points along object boundaries are densely sampled. It should be noted that the proposed line-based sampling method maintains the natural scanning order in a LiDAR system.

Algorithm 1 Line-Scanning LiDAR Sampling Algorithm

```
Input: $W, H, \xi$.
Output: $x, S, b$.
**Initialization Stage:**
 Let $I_j = 1$ with probability $\xi$, for $j = 1, \dots, W$.
Define $S^{(1)}$ according to $I_j$.
Measure $b^{(1)}$ from $S^{(1)}$ by LiDAR.
Compute $x^{(1)}$ from $S^{(1)}$ and $b^{(1)}$ by interpolation.
**Iterative Stage:**
For $r = 2$ to $H$
    Compute $\nabla x^{(r-1)}$.
    For $j = 1, \dots, W$, define $a_j = \left[\nabla x^{(r-1)}\right]_j$.
    Compute $\tau$ such that $\sum_{j=1}^{W} \min\left(\tau a_j, 1\right) = W\xi$.
    Let $p_j = \min(\tau a_j, 1)$, for j $= 1, \dots, W$.
     Let $I_j = 1$ with probability $p_j$, for $j= 1, \dots, W$.
    Define $S^{(r)}$ according to $I_j$
    Measure $b^{(r)}$ from $S^{(r)}$ by LiDAR.
    Compute $x^{(r)}$ from $S^{(r)}$ and $b^{(r)}$ by interpolation.
```

TABLE I
COMPARISON OF THE AVERAGE PSNRS

Method	Average PSNR (dB)	PSNR difference (dB)
Grid	34.8154	-
Line-based	36.3919	+1.577

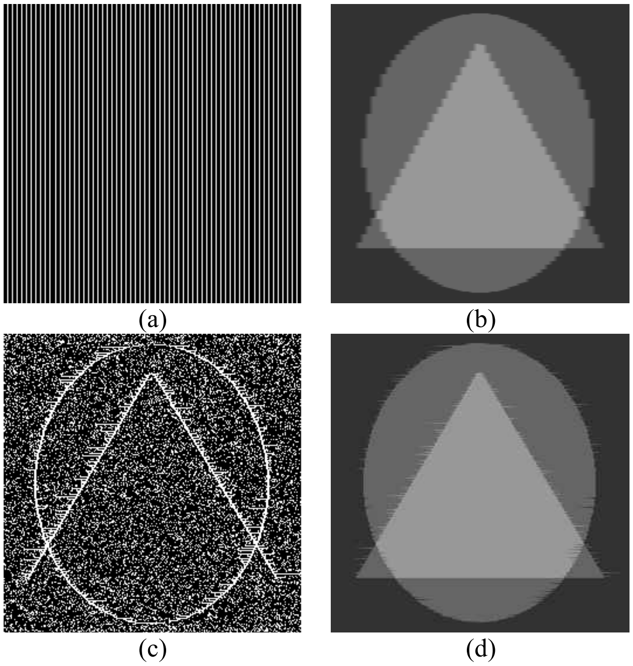

(a)
(b)
(c)
(d)

Fig. 4. Comparison between different sampling patterns: (a)-(b) a grid sampling; (c)-(d) line-based LiDAR sampling.

B. Further Improvement With Misalignment Handling

The line-based sampling algorithm can be further improved by handling the gradient misalignment along object boundary. Particularly, predicting a sampling location from the gradient of the previous line may not be accurate if two consecutive lines are not well aligned, especially on the object boundary area. To handle this phenomenon, this subsection presents two simple yet effective techniques to relax the gradient.

Algorithm 2 Line-Scanning LiDAR Sampling Algorithm

```
Input: $W, H, \xi$.
Output: $x$.
**Initialization Stage:**
 Let $I_j = 1$ with probability $\xi$, for $j = 1, \dots, W$.
Define $S^{(1)}$ according to $I_j$.
Measure $b^{(1)}$ from $S^{(1)}$ by LiDAR.
Compute $x^{(1)}$ from $S^{(1)}$ and $b^{(1)}$ by interpolation with an EP filter.
**Iterative Stage:**
For $r = 2$ to $H$
    Compute $\nabla x^{(r-1)}$.
    Let $I_j^{(1)} = 1$ with probability $\xi/2$, for $j = 1, \dots, W$.
     For $j = 1, \dots, W$,
    define
        $a_j = \begin{cases} f(\nabla x^{(r-1)}, j), & if\ I_j^{(1)} = 1, \\ 0, & if\ I_j^{(1)} = 0. \end{cases}$
Compute $\tau$ such that $\sum_{j=1}^{W} \min\left(\tau a_j, 1\right) = W\xi/2$.
Let $p_j = \min(\tau a_j, 1)$, for $j= 1, \dots, W$.
 Let $I_j^{(2)} = 1$ with probability $p_j$, for $j= 1, \dots, W$.
Define $S^{(r)}$ according to $\left\{I^{(1)}\right\}$ and $\left\{I^{(2)}\right\}$.
Measure $b^{(r)}$ from $S^{(r)}$ by LiDAR.
Compute $x^{(r)}$ from $S^{(r)}$ and $b^{(r)}$ by interpolation with an EP filter.
```

The first technique is to a gradient refinement. For each pixel of the current line, both the upper pixel of the previous line and its neighbors are considered to compute prior information. In particular, for $j = 1, \dots, W$, instead of letting $a_j = \left[\nabla x^{(r-1)}\right]_j$, the prior information is derived as follows:

$$a_j = \left[\nabla x^{(r-1)}\right]_j + \alpha \left(\left[\nabla x^{(r-1)}\right]_{j-1} + \left[\nabla x^{(r-1)}\right]_{j+1} \right) \quad (9)$$

where $\left[\nabla x^{(r-1)}\right]_0 = \left[\nabla x^{(r-1)}\right]_1, \left[\nabla x^{(r-1)}\right]_{W+1} = \left[\nabla x^{(r-1)}\right]_W$, and α is an numerical parameter to softly relax and combine the gradients from the upper neighboring pixels. In practice, α is set by 0.5.

The second technique is to limit the impact of the gradient information transferring from the previous line to the current one. For a given sampling budget, a half is used by spending a grid or random sampling, whereas the remaining half is used by spending the gradient information of the previous estimated line. Particularly, let $I_j^{(1)} = 1$ with probability $\xi/2$, for $j = 1, \dots, W$ to guarantee that for each line, a half of sampling budget is picked randomly. In order to ensure that the $W\xi/2$ samples picked at the refined stage do not overlap with those picked at the first stage, the prior information computed in (7) is further refined as follows:

$$a_j = \begin{cases} f(\nabla x^{(r-1)}, j), & if\ I_j^{(1)} = 1 \\ 0, & if\ I_j^{(1)} = 0 \end{cases} \quad (10)$$

where $f(\nabla x^{(r-1)}, j)$ is the right part of (9).

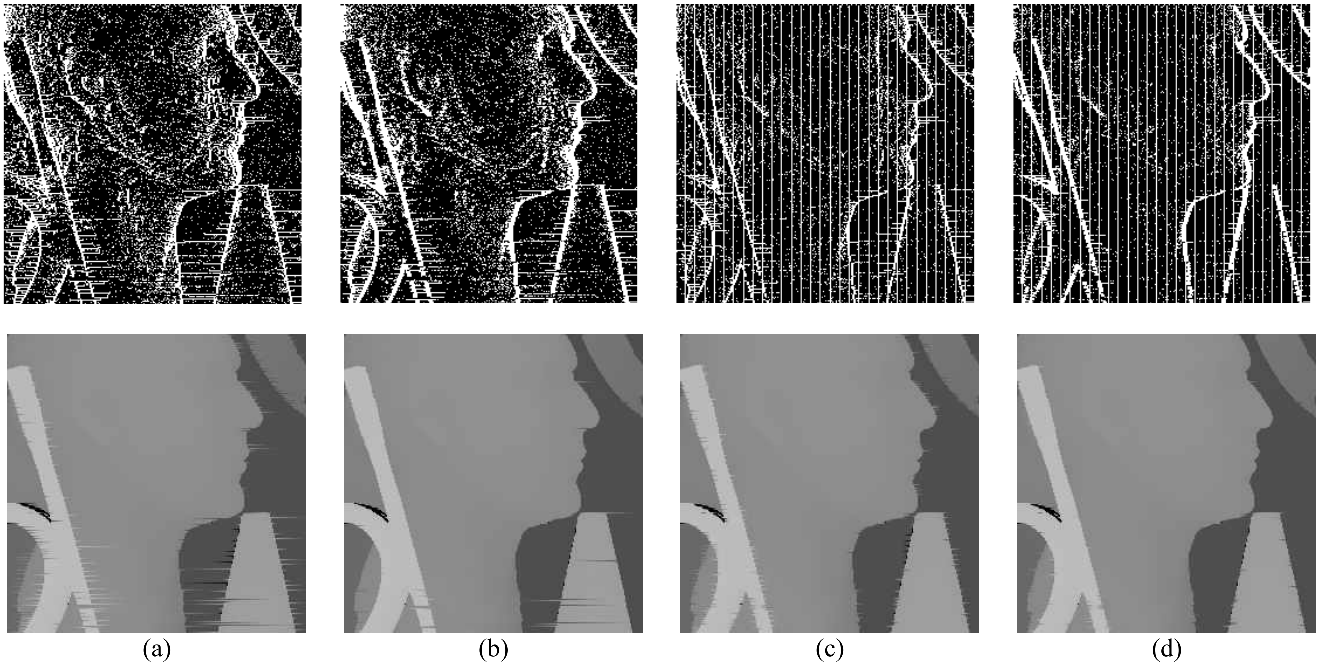


Fig. 5. Comparison between different sampling patterns: (a) Line-based sampling (Naïve); (b) V1 with gradient refinement only; (c) V2 with budget dividing only; and (d) V3 with both gradient refinement and budget dividing.

Since, $p_j = \min(\tau a_j, 1)$, (10) forces $p_j = 0$ when the j-th pixel sampling location is previously picked by random sampling. The overall method is summarized in Alg. 2.

Compared to Alg. 1, it makes four modifications. Firstly, line 11 is added to guarantee that for each line, a half of sampling budget is picked randomly. Next, line 11 of Alg. 1 is revised as line 12 in Alg. 2 to ignore the sampling locations that are picked previously. In addition, τ is computed as in line 13 because only a half of sampling locations are generated by the gradient of the previously. Finally, the random sampling and gradient-based sampling locations are merged as line 16.

A comparison between sampling patterns is shown in Fig. 5. Setting $\xi = 0.2$ (i.e., 20%), four sampling patterns generated by the proposed method are evaluated. A Monte-Carlo simulation by repeating 32 independent trials is conducted, and the averages of PSNRs are reported in Table II. The results in Fig. 5(a) are generated using the naïve line-based sampling scheme in Alg. 1, whereas the results in Figs. 5(b)-(d) are generated by using the method with gradient refinement only (V1), with sampling pattern merging only (V2), and with both techniques (V3), respectively. These results indicate that for the refinement techniques greatly improve the reconstruction quality. In Table II, as compared to the baseline (i.e., naïve line-based sampling), the method with gradient refinement has about 4.83 dB improvement. Meanwhile, the method with sampling pattern merging has about 3.88 dB improvement and can be further improved by 6.83 dB using both refinement methods. Especially, compared to the grid sampling, all four line-based sampling variations demonstrate the significant improvement of the reconstruction quality with 0.92, 6.76, 4.10, and, 7.76 dB. It is worthy to mention that the line-based sampling method does not change the natural scanning order in a LiDAR system, which is significantly different from the previous sampling methods [11]–[13]. Thanks to a prediction manner, the proposed methods can effectively find locations along gradient areas without scanning an FOV twice and building a rough depth map.

TABLE II
COMPARISON OF THE AVERAGE PSNRS

Method	Average PSNR (dB)	Improvement (dB)
Grid	31.0655	-
Line-based		
Naïve	31.9895	+0.9240
+ gradient refinement only	37.8283	+6.7583
+ pattern merging only	35.1696	+4.1041
+ gradient refinement and pattern merging	38.8224	+7.7569

C. Complex Scene Handling

In practice, a scene may become more complex because of the noise. To address this problem, the interpolation in Alg. 2 can be combined with an edge-preserving (EP) filter that has been studied in [28]–[30]. Fig. 6 shows an example of using the EP filter in [28] to filter noisy data out. The results show that the filtered data with the red line are very close to the groundtruth with the blue line thanks to the EP filter, whereas the noisy data with the black line has a large amplitude. Eventually, the gradient computation becomes more robust to the noisy. It should be noted that EP filters are efficient, especially for 1D data [28]–[30], which does not degrade the processing speed of Alg. 2.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents additional results to demonstrate the performance of the proposed method.

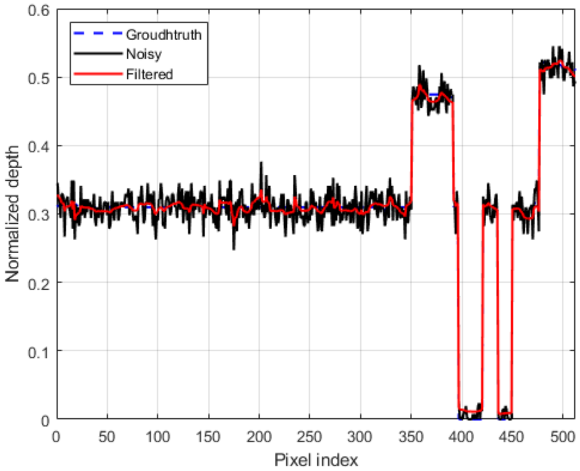


Fig. 6. Interpolation with the noise filtering by an edge-aware recursive filter in [28].

A. Synthetic Data

These experiments are conducted using the six testbeds in the Middlebury datasets[1]: Aloe, Art, Baby2, Moebius, Dolls, and Rocks [31], [32]. Table III shows the PSNR values at different sampling ratios and sampling methods. For consistency, the results with grid and two-stage sampling methods with the reconstruction method of the alternating direction method of multipliers (ADMM) (i.e., the first and second rows for each test image) are regenerated as in [25], [26]. Meanwhile, to stimulate the natural scanning order in a LiDAR, a linear interpolation is used to reconstruct line by line for the remaining scanning methods (i.e., the 3rd to 8th rows for each test image). Particularly, the two-stage sampling method [25] is modified as follows. For each line, at the first-stage, a half of sampling budget is used by using random sampling to derive sampling locations and measure their depth values. On the second stage, sampled data is used to estimate a rough dense line data, then gradient is computed and is used to find sampling locations with the remaining half of budget. In other words, each line is scanned twice as the method in [25]. The results on Table III shows that the proposed methods outperform the grid sampling and achieve similar performances as the 1D two-stage method. In many cases, the proposed method V3 even performs better than the 1D two-stage one. This phenomenon can be explained as follows. Since the sampling locations of a current line are predicted from its previous line data which are derived from M sampling points. Meanwhile, in the two-stage method, only $M/2$ sampling points are used to estimate the current line data. Consequently, the proposed sampling algorithms may predict points more accurately than the two-stage method does. Among three versions of the proposed sampling algorithms, the one combining both pattern merging and gradient refinement techniques achieves the best reconstruction quality, having about 2.68, 4.29, 5.49, and 5.78 dB improvement when being compared with the grid sampling at the sampling ratios of 10%, 15%, 20%, and 25%, respectively. Generally, two-stage methods [25], [26] usually achieve better performances than 1- D methods because they utilize depth information from both vertical and horizontal directions at the first stage to estimate a rough depth map. From the view of a LiDAR, to build a depth map, these methods require to scan the entire FOV twice: one is for estimating a rough depth in the first stage, and the other is for refining the map in the second stage. Consequently, a two-stage method becomes counterintuitive in a LiDAR that strictly requires a scanning order. Consider the LiDAR system in [20] that utilizes galvanometer scanners [9]. According to their specification, the maximum number of scanning lines is 300 per second [9]. From these results, scanning each line takes at least 3.33 milliseconds. If each line is scanned twice as the conventional two-stage method, it takes at least 6.66 milliseconds. Meanwhile, the proposed approach maintains the same scanning time of 3.33 milliseconds.

TABLE III
THE PSNR OF THE RECONSTRUCTED IMAGE

Test Images	Sampling	Recon-struction	Reconstruction performance PSNR over sampling rate (dB)			
			10%	15%	20%	25%
Aloe	Grid	ADMM	25.32	28.91	31.30	32.35
	k-NN two-stage [26]		30.75	33.75	36.54	40.91
	Two-stage [25]		27.60	31.39	36.41	38.63
	Grid	Inter-polation	27.12	28.98	30.07	31.18
	1D Two-stage [25]		27.79	30.66	33.94	36.37
	V1		28.40	31.12	32.53	33.79
	V2		27.25	29.59	31.50	33.12
	V3		29.04	32.17	34.38	35.88
Art	Grid	ADMM	27.52	28.95	32.52	33.71
	k-NN two-stage [26]		31.97	35.71	38.18	43.38
	Two-stage [25]		30.87	34.15	42.97	48.00
	Grid	Inter-polation	27.21	29.24	30.38	31.23
	1D Two-stage [25]		27.15	29.29	31.61	34.71
	V1		28.02	31.11	32.88	34.48
	V2		27.70	32.13	34.77	37.54
	V3		28.94	32.49	35.49	38.79
Baby	Grid	ADMM	34.44	36.80	39.05	40.07
	k-NN two-stage [26]		39.99	44.73	48.08	53.58
	Two-stage [25]		39.70	44.90	52.50	52.00
	Grid	Inter-polation	33.60	34.75	35.30	36.84
	1D Two-stage [25]		36.42	41.42	45.54	49.09
	V1		37.93	40.18	41.68	42.53
	V2		41.22	42.51	43.56	44.12
	V3		40.48	44.39	47.03	46.63
Dolls	Grid	ADMM	28.49	29.05	30.81	31.67
	k-NN two-stage [26]		31.37	33.48	35.72	37.32
	Two-stage [25]		29.51	32.53	36.27	37.65
	Grid	Inter-polation	28.54	29.54	30.27	31.20
	1D Two-stage [25]		28.95	30.88	32.69	33.98
	V1		29.78	32.16	33.86	34.89
	V2		29.35	32.64	34.87	34.35
	V3		30.03	32.86	34.64	36.33
Moebius	Grid	ADMM	27.69	28.72	31.17	32.24
	k-NN two-stage [26]		32.16	35.05	37.55	41.1
	Two-stage [25]		31.07	35.11	39.92	41.89
	Grid	Inter-polation	29.65	30.92	31.95	32.89
	1D Two-stage [25]		30.37	32.32	34.70	37.29
	V1		31.28	32.70	33.57	35.02
	V2		30.67	32.82	32.35	34.55
	V3		31.24	32.93	34.53	36.13
Rocks	Grid	ADMM	27.69	28.72	31.17	32.24
	k-NN two-stage [26]		32.76	35.99	40.15	46.74
	Two-stage [25]		31.07	35.11	39.92	41.89
	Grid	Inter-polation	28.61	30.18	31.49	32.48
	1D Two-stage [25]		30.04	33.83	36.82	37.48
	V1		30.59	32.29	33.71	34.31
	V2		31.12	33.30	34.38	35.83
	V3		31.05	34.52	36.33	36.77

Note: V1 (w/ pattern merging), V2 (w/ gradient refinement), V3 (w/ both methods)

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

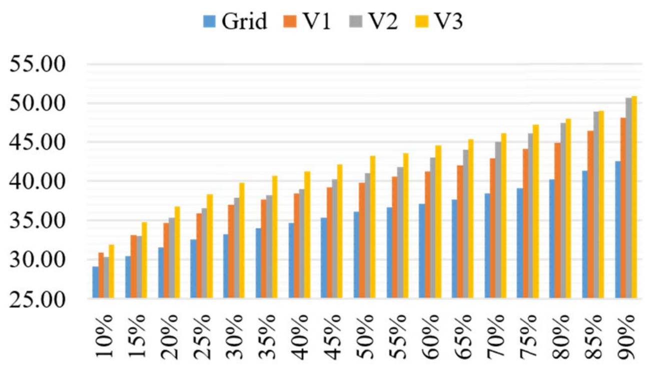


Fig. 7. Comparison of the reconstruction performance with the various sampling ratios.

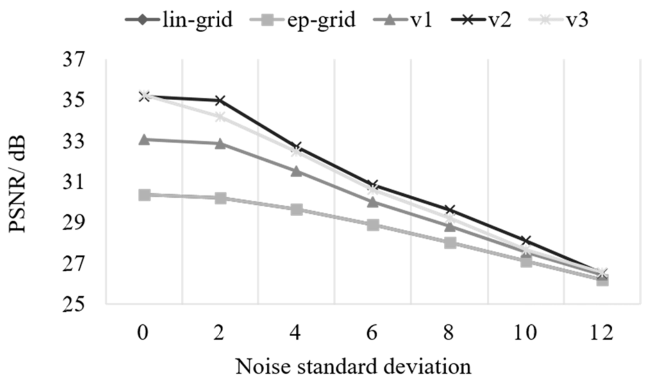
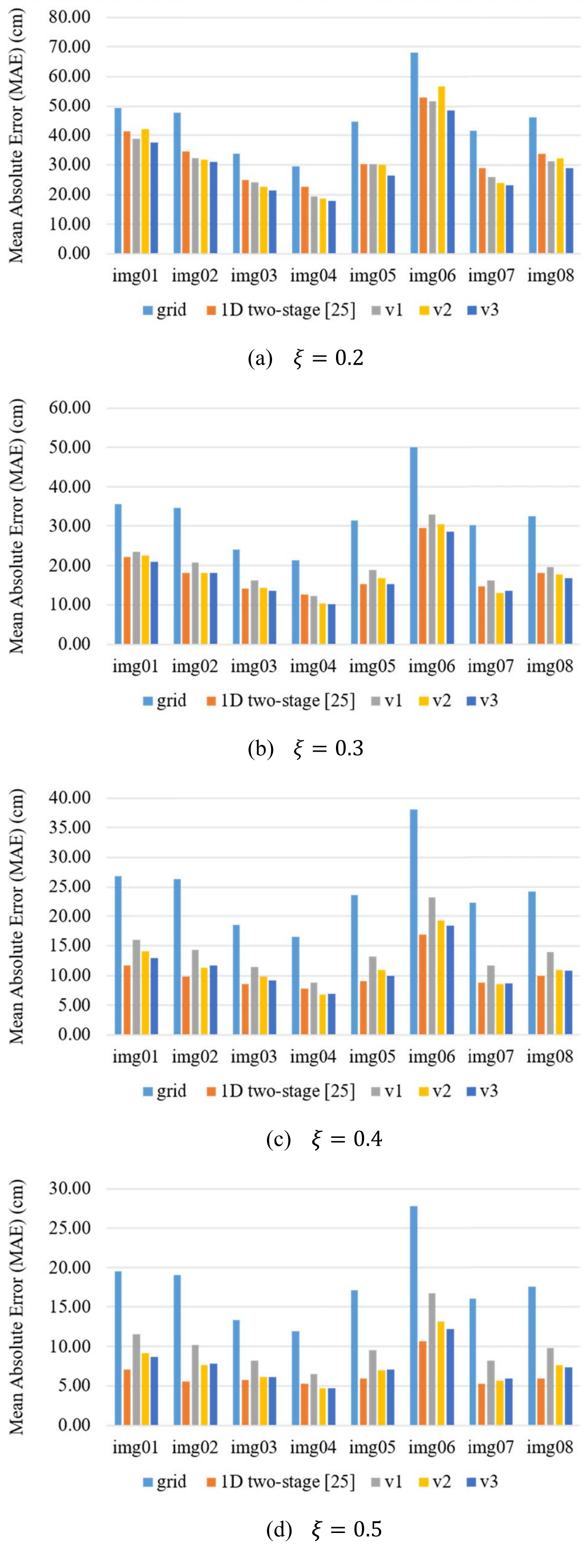


Fig. 8. Comparison of the reconstruction performance with noise samples. The test image is “Art”, and the sampling rate is 0.2. “lingrid” refers to the grid sampling with linear interpolation, while “epgrid” refers to the grid sampling with linear interpolation combined with an EP filter.

The experimenal results with the sampling ratios from 10% to 90% are reported in Fig. 7. The PSNR results are obtained by averaging each PSNR of the same six test images as in Table III. The results demonstrate that the proposed algorithm (*i.e*, V3) consistently outperforms the baseline (*i.e.*, Grid). As discussed in [25], depth maps are sparser than natural images. Therefore, it is possible to apply the sampling with a relatively small sampling ratio (*e.g.*, 0.2). In addition, it is obvious that the frame rate becomes higher as the sampling rate lowers.

In addition, Fig. 8 shows the performance of the proposed algorithm towards additive independent and identically distributed (i.i.d.) Gaussian noise. The goal of this experiment is to demonstrate the sensitivy and robustness of the proposed algorithm in the presence of noise. Actually, the noise in the depth estimation is not i.i.d. Gaussian, but the results presented here are used as a reference for the algorithm’s performance. Using “Grid” as the baseline, the propopsed “v3” enhances the PSNR up to 4.87 dB, while the propopsed “v1” enhances the PSNR up to 2.67 dB. This provides a good indicator of the robustness of the proposed algorithm.



(a) $\xi = 0.2$

(b) $\xi = 0.3$

(c) $\xi = 0.4$

(d) $\xi = 0.5$

Fig. 9. Comparison of the reconstruction performance between different sampling patterns on the Brown laser range dataset. The sampling ratios are set to 20%, 30%, 40% and 50% corresponding to Figs. (a), (b), (c) and (d), respectively.

B. Laser Range Data

The experiments are also performed with indoor and outdoor laser range data sets from Brown University range image database [33]. The laser range data in the Brown database are acquired using the Riegl LMS-Z210 laser range-finding apparatus with a rotating mirror. Each range image consists of 444 × 1440 measurements (each measurement based on time of flight of laser beam) with an angular separation of

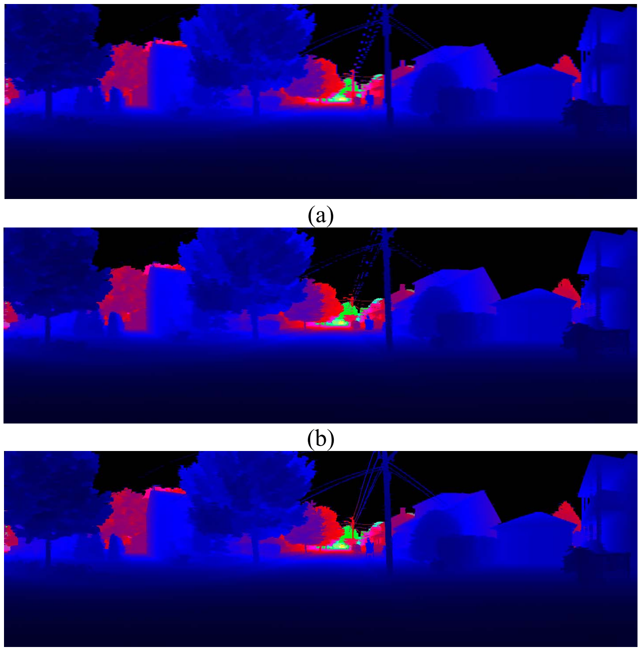


Fig. 10. Reconstruction results of sampling patterns at the sampling ratio of 20%: (a) grid; (b) 1D two-stage [25]; and (c) Proposed (V3).

0.18 degrees, resulting in an FOV of 80 degrees vertically and 259 degrees horizontally. The operational range is around 2-200 m.

The proposed sampling patterns are compared with the grid and 1D two-stage sampling patterns [25] as described in subsection V-A. Since the laser range data covers a much large FOV than the synthetic data, the sampling ratios are set to 20%, 30%, 40%, and 50%. The mean-absolute-error (MAE) results in centimeter are reported in Fig. 9. The experimental results demonstrate that all three proposed sampling patterns consistently outperforms the grid sampling in all cases, while achieving the reconstructed quality similar to the 1D two-stage sampling method [25]. Furthermore, the proposed sampling pattern of V3 achieves the best performance in most cases. In particular, on average, it achieves the MAE reduction by 34.91%, 47,23%, 54,88%, and 57,99% for the sampling ratios of 20%, 30%, 40%, and 50%, respectively, compared to the grid sampling.

Fig. 10 shows an example of reconstructed images by three sampling patterns: grid, 1D two-stage, and the proposed V3 for the sampling ratio of 20%. It is clearly shown that both 1D two-stage method and the proposed V3 method are able to capture the object boundary areas much better than a grid sampling, for example, the roof areas or electric wire areas. Note that the proposed method only scans each line (or each FOV) once, while the 1D two-stage method has to scan each line twice. It means that the proposed method retains the merit of a gradient-aware sampling method while maintaining the conventional scan order in a LiDAR.

V. Conclusion

This study addresses the scanning problem in a LiDAR system by carefully selecting the samples along object gradients in the on-the-flight manner to guarantee both the required timing issue and the natural scanning order in a LIDAR. The proposed scheme achieves high-quality reconstruction results at a specified sampling budget, and more importantly, it is remarkably rapid in the complexity of linear time. This study also provides a better understanding of the structures of depth images captured by the LiDAR system in the context of sampling. The simple yet effective method in this study is applicable to numerous depth-data-processing tasks for data acquisition, compression, and enhancements applications.

Appendix

Followed by [25], this section presents the former derivation of (7). To be more precise, let us define the average gradient computed by all N samples as

$$\mu = \frac{1}{N}\sum\nolimits_{j=1}^{N} g_j \tag{11}$$

Similarly, let us define the average gradient computed from a random subset of $N\xi$ samples as

$$Y = \frac{1}{N}\sum\nolimits_{j=1}^{N} \frac{g_j}{p_j} I_j \tag{12}$$

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

From (9) and (10), minimizing the difference between Y and μ can be achieved by minimizing the variance $E[(Y-\mu)^2]$. Moreover, it is observed that

$$\left[(Y-\mu)^2\right] = \frac{1}{N}\sum_{j=1}^{N}\frac{g_j^2}{p_j^2}Var\left[I_j\right] = \frac{1}{N}\sum_{j=1}^{N} g_j^2\left(\frac{1-p_j}{p_j}\right).$$

where the last holds because $Var\left[I_j\right] = p_j(1-p_j)$. Thus, the optimal probabilities $\{p_j\}_j^N$ can be computed by solving the optimization problem:

$$\underset{p_1,\ldots,p_N}{\text{minimize}} \frac{1}{N}\sum_{j=1}^{N}\frac{g_j^2}{p_j}. \tag{13}$$

subject to $\sum_{j=1}^{N}\frac{g_j^2}{p_j} = N\xi$, and $0 \le p_j \le 1$.

The solution of (13) is derived in [34] as follows:

$$p_j = \min\left(\tau g_j, 1\right) \quad for\ j = 1, \ldots, N. \tag{14}$$

where τ is the root of the following equation:

$$\sum_{j=1}^{N}\min(\tau g_j, 1) = N\xi. \tag{15}$$

Thus, (7) and (8) are derived.

References

[1] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," *J. Robotic Syst.*, vol. 23, no. 9, pp. 661–692, 2006.

[2] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[3] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.

[4] E. Ackerman, "Lidar that will make self-driving cars affordable [news]," *IEEE Spectr.*, vol. 53, no. 10, p. 14, Oct. 2016.

[5] M. D. Adams, "Coaxial range measurement–current trends for mobile robotic applications," *IEEE Sensors J.*, vol. 2, no. 1, pp. 2–13, 2002.

[6] H. Deilamsalehy and T. C. Havens, "Sensor fused three-dimensional localization using IMU, camera and LiDAR," in *Proc. IEEE SENSORS*, Oct. 2016, pp. 1–3.

[7] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, USA: MIT Press, 2004, pp. 181–256.

[8] D. S. Hall, "High definition LiDAR system," U.S. Patent 7 969 558 B2, Jun. 28, 2011.

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

[10] Newson Engineering, Dendermonde, Belgium. (2007). *rhothorTM X7 I/O Configuration: XY2-100 Technical Datasheet, NEWSON Nv, 9200*. [Online]. Available: http://www.newson.be/doc.php?id=XY2-100

[11] G. F. Marshall, *Handbook of Optical and Laser Scanning*. New York, NY, USA: Marcel Dekker, 2004.

[12] LeddarTech Inc, Leddar, LeddarTech HQ, Quebec City, QC, Canada. (May 2017). *LEDDAR M16: Multi-Element Sensor Module*. [Online]. Available: http://leddartech.com/modules/m16-multi-element-sensor-module/

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

[14] J. Scholz, V. Willhoeft, R. Schulz, T. Kluge, *ALASCA User Manual*, Ibeo Automobile Sensor GmbH, Hamburg, Germany, 2006. [Online]. Available: http://www.raginbot.com/wiki/images/d/d8/Manual_ALASCA.pdf

[15] Velodyne LiDAR Inc., San Jose, CA, USA. (2017). *Velodyne LiDAR PuckTM: Real-time 3D LiDAR Senso*. [Online]. Available: http://velodynelidar.com/vlp-16.html

[16] Velodyne LiDAR Inc., San Jose, CA, USA. (2017). *Velodyne LiDAR HDL-32E: High Definition Real-time 3D LiDAR Sensor*. [Online]. Available: http://velodynelidar.com/hdl-32e.html

[17] Velodyne LiDAR Inc., San Jose, CA, USA. (2017). *Velodyne LiDAR HDL-64E S3: High Definition Real-time 3D LiDAR Sensor*. [Online]. Available: http://velodynelidar.com/hdl-64e.html

[18] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[19] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[20] 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 J.*, vol. 18, no. 2, pp. 559–568, Jan. 2018.

[21] S. Hawe, M. Kleinsteuber, and K. Diepold, "Dense disparity maps from sparse disparity measurements," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2126–2133.

[22] L.-K. Liu, Z. Lee, and T. Nguyen, "Sharp disparity reconstruction using sparse disparity measurement and color information," in *Proc. IVMSP*, Jun. 2013, pp. 1–4.

[23] S. Schwartz, A. Wong, and D. A. Clausi, "Multi-scale saliency-guided compressive sensing approach to efficient robotic laser range measurements," in *Proc. 9th Conf. Comput. Robot Vis.*, May 2012, pp. 1–8.

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

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

[26] X. T. Nguyen, H. Kim, and H.-J. Lee, "An efficient sampling algorithm with a K-NN expanding operator for depth data acquisition in a LiDAR system," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jan. 1, 2020, doi: 10.1109/TCSVT.2019.2963448.

[27] X. T. Nguyen, K.-T. Nguyen, H.-J. Lee, and H. Kim, "ROI-based LiDAR sampling algorithm in on-road environment for autonomous driving," *IEEE Access*, vol. 7, pp. 90243–90253, 2019.

[28] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, p. 69, Jul. 2011.

[29] Q. Yang, "Recursive bilateral filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 399–413.

[30] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5638–5653, Dec. 2014.

[31] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[32] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[33] T. B. U. Pattern Theory Group. *Brown Range Image Database*. Accessed: Mar. 31, 2019. [Online]. Available: http://www.dam.brown.edu/ptg/brid/range/index.html

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

Xuan Truong Nguyen received the B.S. degree in electrical engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2011, and the M.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2015 and 2019, respectively. He is currently a Postdoctoral Fellow of the Department of Electrical and Computer Engineering, Seoul National University. His research interests include the areas of algorithms, and SoC design for low-complexity computer vision and multimedia applications.

Hyun Kim (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2009, 2011, and 2015, respectively. From 2015 to 2018, he was an Assistant Professor of BK21 Creative Research Engineer Development for IT, Seoul National University. In 2018, he joined the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, where he is currently working as an Assistant Professor. His research interests are in the areas of algorithms, computer architecture, memory, and SoC design for low-complexity multimedia applications and deep neural networks.

Hyuk-Jae Lee (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1996. From 1996 to 1998, he was with the Faculty of the Department of Computer Science, Louisiana Tech University, Ruston, LS, USA. From 1998 to 2001, he was with the Server and Workstation Chipset Division, Intel Corporation, Hillsboro, OR, USA, as a Senior Component Design Engineer. In 2001, he joined the School of Electrical Engineering and Computer Science, Seoul National University, where he is currently a Professor. He is the Founder of Mamurian Design, Inc., a fabless SoC design house for multimedia applications. His research interests are in the areas of computer architecture and SoC design for multimedia applications.