2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

D2SR: Decentralized Detection, De-Synchronization, and Recovery of LiDAR Interference

Darshana Rathnayake[1], Hemanth Sabbella[1], Meera Radhakrishnan[2], and Archan Misra[1]
[1]Singapore Management University, [2]University of Technology Sydney
{darshanakg.2021, hemanthrs}@smu.edu.sg, meeralakshmi.radhakrishnan@uts.edu.au, archanm@smu.edu.sg

***Abstract*— We address the challenge of multi-LiDAR interference, an issue of growing importance as LiDAR sensors are embedded in a growing set of pervasive devices. We introduce a novel approach named *D2SR*, enabling decentralized interference detection, mitigation, and recovery without explicit coordination among nearby LiDAR devices. *D2SR* comprises three stages: (a) Detection, which identifies interfered frames, (b) Mitigation, which performs time-shifting of a LiDAR's active period to reduce interference, and (c) Recovery, which corrects or reconstructs the depth values in interfered regions of a depth frame. Key contributions include a lightweight interference detection algorithm achieving an F1-score of 92%, a simple yet effective decentralized de-synchronization mechanism, and a lightweight depth recovery pipeline that preserves high throughput processing on edge devices. Evaluation on Nvidia Jetson devices demonstrates *D2SR*'s efficacy: under static settings, *D2SR* accurately detects interference in 93% of cases (recall=82%) and reduces the depth estimation error by 27% (RMSE= 38.7 cm, compared to RMSE= 60.6 cm for a baseline without *D2SR*). Furthermore, *D2SR* is able to reduce the fraction of interfered frames by 75.1% and reduce the depth estimation error (for interfered frames) by 24.9% even for a moving robot scenario.**

***Index Terms*— LiDAR Interference; Multi-Robot Systems**

I. Introduction

LiDAR (Light Detection and Ranging) sensors have witnessed a remarkable surge in its applications across various fields, such as for autonomous vehicles [1], robotic navigation [2], indoor mapping [3] and environmental monitoring [4]. When multiple LiDAR-equipped devices coexist and operate in close proximity (such as multiple object picking robots in a warehouse), they can generate mutual interference–very specifically, a receiver may incorrectly receive and process the light pulse emitted by another device. As we shall quantify, such interference can lead to significant errors in depth estimation, thereby undermining the reliability of LiDAR-based 3D sensing especially in indoor environments.

While some of the consequent incorrect depth readings manifest as outliers and can be mitigated partially through preprocessing techniques (e.g., median filtering [5]), we shall see that such interference often generates erroneous readings over substantial parts of one or more captured depth frames. This can significantly degrade tasks like object detection, particularly for smaller objects (e.g., chairs, laptops) commonly found in indoor environments. Various DNN-based strategies have been proposed to reconstruct erroneous depth

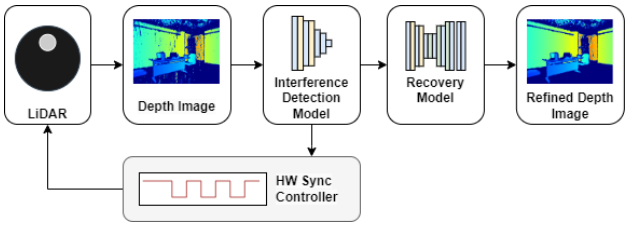


Fig. 1: *D2SR's* High-level Architecture

estimation, using collocated RGB image data and "clean" depth map values. However, a key challenge in applying such DNN-based reconstruction is to separate such *interference-generated* errors from the *random* errors that inevitably occur due to environmental artifacts (e.g., reflectivity variations) and sensor imperfections (e.g., noise).

In this work, we tackle this progressively-important problem of multi-LiDAR interference in indoor environments and present a novel approach, called *D2SR*, that enables decentralized (without any explicit coordination or signaling among nearby scanning LiDAR devices) interference detection, mitigation, and recovery. *D2SR* conceptually involves three distinct stages, illustrated in Figure 1: (a) the *Detection* stage first identifies the individual frames that are subject to (unpredictable) interference-generated degradation; (b) the *Mitigation* stage then uses a simple, but effective, dynamic duty cycle phase adjustment strategy to partially de-synchronize multiple emitter-receiver pairs, thereby reducing (but not completely eliminating) the fraction of interfered frames; and (c) the *Recovery* stage operates on each individual interference-afflicted frame to first *identify* the sub-regions (masks) distorted due to interference and then *estimate/generate* the depth values for such regions, thereby significantly reducing the overall depth estimation error. *D2SR's* design is notable not just for its ability to tackle this problem of interference holistically, for a varying number of collocated LiDARs, but also it is lightweight enough to permit real-time, *on device* execution of the entire pipeline on resource-constrained pervasive/edge devices.

We make the following **key contributions**:

- *Lightweight Approach for Interference Detection*: We propose a lightweight interference detection algorithm, based on a convolutional neural network (CNN) architecture, that effectively identifies the subset of interfered frames that a LiDAR sensor experiences because of possible interference with concurrent proximate LiDARs. We shall show

that this approach can accurately discriminate interfered depth frames from genuine LiDAR data (F1-score= 92%), enabling us to activate the subsequent stages of *D2SR* as needed.

- *Simple but Effective Decentralized LiDAR De-synchronization*: To minimize the interference among the LiDARs, we introduce a decentralized technique that attempts to de-synchronize the active periods of different LiDAR sensors without requiring any explicit signaling among LiDAR devices (or even knowledge of the number of interfering devices). The de-synchronization mechanism effectively adjusts both a LiDAR device's duty cycle, as well as time-shifts the active period, to reduce the likelihood of collision with other nearby LiDARs. We shall show that this approach significantly reduces the number of interfered frames (70.3% reduction when N=2 LiDARs, operating at half their frame rate), although its efficacy understandably diminishes when the number of interfering LiDARs increases (the reduction is only 3.5% for N=5 LiDARs).
- *Lightweight Depth Recovery Technique:* For depth frames that continue to suffer from interference, we present a recovery mechanism that uses a multi-stage unified DNN model to compute an interference mask (sub-regions whose values were corrupted by interference) and then recovers (or regenerates) the depth values for pixels within each interference mask. Using a prototype implementation, we show that this recovery pipeline is lightweight enough to sustain high throughput processing on multiple edge-class devices (163.9 fps on a Jetson AGX Orin, and 14.7 fps on a much more constrained Jetson Nano).
- *Demonstrate Real-World Benefits of* D2SR: We evaluate *D2SR's* performance on multiple Nvidia Jetson devices, under varying levels of interference caused by stationary LiDARs (N=$\{1,2,\ldots,5\}$ devices), in three different indoor environments. We also tested *D2SR's* performance with a LiDAR device mounted on a moving robotic platform. Our studies show that, under static settings, (a) *D2SR* can detect interference in 93% cases with a high precision of 99% and recall of 82% even under dynamically changing environments, and (b) *D2SR* can reduce the depth estimation error by 27% (RMSE= 38.7 cm, compared to RMSE= 60.6 cm for a baseline without *D2SR*). These gains persist even for the moving robot, where *D2SR* can reduce the fraction of interfered frames by 75.1% and reduce the depth estimation error (for interfered frames) by 24.9%.

Overall, we contribute to addressing the challenge of interference that LiDAR-enabled devices will increasingly face in smart factories, warehouses, and other environments where multiple robots operate concurrently, as well as develop a lightweight, integrated approach to mitigate and recover from the impact of such interference.

II. Related Work

We discuss prior works focusing on LiDAR interference detection, characterization, and mitigation approaches.

Interference Detection & Characterization Studies: Multiple prior works [6]–[8] have studied the effect of mutual interference among LiDARs operating in close proximity. Martin et al. [6] examined the correlation of interference with the relative sensor pose in commercial RGB-D sensors (such as Kinect and Xtion RGB-D sensors). Their experiments revealed that up to 95% of depth measurements in the interference image region may disappear when two RGB-D sensors interfere. Eom et al. [8] presented various mutual interference scenarios involving off-the-shelf 2D LiDARs, including direct interference between LiDAR scanners, indirect interference caused by reflection from other objects, and potential high density of interference sources. Curto et al. [9] conducted a comparative assessment of depth measurements, and the impact of random errors, using three RealSense cameras (D415, SR305, and L515) for various transparent and translucent objects. However, the impact of cross-LiDAR interference was not explored. A recent study [10] examined interference between different 360° LiDAR sensors.

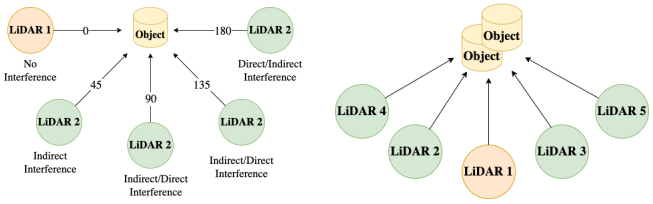
Interference Mitigation Studies: Interference mitigation in multi-LiDAR systems can draw from various wireless scheduling techniques such as SDMA, FDMA, WDMA, TDMA, and CDMA [11]. However, their direct application is hindered by the challenge of centrally coordinating all sensors, particularly in uncontrolled environments. Among direct ToF LiDARs, prior studies have explored hardware/sensor-level modifications such as interference suppression schemes by adding a variable delay between the laser emissions [11], [12], modulated pulse scanning to add information to distinguish self–vs.-other laser pulses [13], and a chaotic laser-based architecture [14] that is resilient to interference. Simple software-level solutions involve (a) filtering methods to distinguish between direct and scattered interference [15] and (b) interference cancellation techniques [16]. Additionally, crosstalk effects between 3D spinning LiDAR devices have been addressed using data-driven spatiotemporal filtering [17]. However, these approaches do not perform well as the number of LiDAR sensors grows and are also unable to respond rapidly to changing interference patterns, especially in dynamic and diverse indoor environments. There is also little prior work on *recovery mechanisms* that are able to correct corrupted or missing LiDAR data caused by interference.

III. Impact of LiDAR Interference

Mutual interference arises when a LiDAR receiver receives a foreign signal (laser pulses) generated by the emitter component of other LiDARs [18]. In future scenarios with multiple LiDAR-equipped robots or wearable devices, such interference poses a significant challenge. Consider a bustling restaurant where robot waiters, each equipped with a LiDAR for navigation, converge near tables or cross paths while serving customers. This interference may cause navigation errors, service delays, or even robot collisions, disrupting restaurant operations.

There are two types of mutual interference: (a) *Direct Interference*, occurring when two LiDAR sensors face each

other, coupling one sensor's signal to the other's receiver, and (b) *Indirect/Scattering Interference*, where a LiDAR receiver captures scattered/reflected signals from another LiDAR's emitter. Additionally, LiDAR sensors employing Avalanche Photodiodes (APDs) often exhibit random errors [19]. These errors cause fluctuations in depth image sequences captured by stationary cameras, particularly in static environments. The extent of these errors depends on the reflective properties of encountered surfaces. Further details on this issue will be discussed in Section V-C.1.

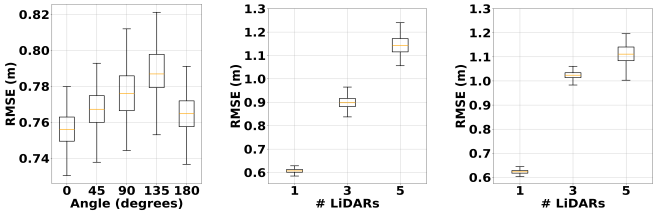


(a) Angle between two LiDARs (b) Varying number of LiDARs

Fig. 2: The experimental setups to study the impact of interference of a scanning LiDAR

We first investigate the effects of two distinct types of interference on a *scanning* LiDAR system, selected due to their superior precision in depth estimation and heightened susceptibility compared to other LiDAR systems (e.g., flash LiDAR). This evaluation is conducted within a controlled experimental setup, wherein two LiDAR devices are situated in a static environment featuring a singular object (a chair) positioned in their field of view. As depicted in Figure 2a, one LiDAR (LiDAR 1) remains stationary while the other (LiDAR 2) is systematically adjusted to vary the angle between their lines of sight. The Root Mean Squared Error (RMSE) between reference depth images and interfered depth images captured by LiDAR 1 is used to quantify interference impact. Reference depth images are generated by averaging pixel values across 300 consecutive depth images (dampening the effect of random errors) acquired over a 10-second *baseline* interference-free period. Subsequently, sequences of 300 depth images are captured at four distinct angles: {45, 90, 135, 180} degrees. RMSE values, depicted in Figure 3a, peak at 135°, where both types of interference are most pronounced. Also, the error due to scattering interference is observed to be higher compared to direct interference, as indicated by the lower RMSE values observed at 180°, where the scattering interference is minimal.

Next, we study the influence of scattering interference across a varying number of spatially and temporally overlapping LiDAR systems. Our setup, depicted in Figure 2b, closely mirrors the methodology employed in the previous study. We obtained sequences of 300 depth images for three distinct scenarios: utilizing 1, 3, and 5 concurrently operating LiDARs. Each scenario is assessed under two conditions: one with a singular object (chair) and the other with four objects (chairs) present in the environment. RMSE variations, shown in Figures 3b and 3c, increase with the number of interfering LiDARs and environmental clutter. When there



(a) Different angles (b) Single object (c) Multiple objects

Fig. 3: Comparison of RMSE for Different LiDAR Configurations and Environmental Complexity

is only a single LiDAR operating, the RMSE with the reference frame is about 65 cm which signifies the influence of the random error. For instance, with 3 concurrent LiDARs, RMSE rises by 13.3% from 90 cm (for one object) to 102 cm with increased clutter (for three objects), emphasizing interference's dependency on spatial and temporal overlap of LiDAR units and environmental complexity.

IV. System Design

We now outline the design goals and the high-level architecture of our system.

A. Design Goals

Low Computational Complexity: We aim to develop a solution that can be bundled as a middleware component on the pervasive device containing the embedded LiDAR sensor. Accordingly, we require the computational complexity of *D2SR* to be low enough to permit execution on resource-constrained hardware (e.g., Jetson AGX Orin), without causing any drop in the the LiDAR frame rate.

Decentralized Operation: *D2SR* must operate autonomously and independently across multiple LiDAR devices in a decentralized manner–i.e., it cannot utilize any centralized or inter-device communication for coordinating the sensing behavior of individual LiDAR devices.

Robust Recovery Mechanism: The recovery mechanism of *D2SR* should be robust enough to reduce the interference-induced error in the depth frames, across a variety of environmental context (e.g., varying number of interfering LiDARs, stationary vs. moving operation).

Precision in Interference Detection: Our solution functions as a closed-loop control system, strategically engineered to identify and reduce interference-inducted errors. Maintaining high precision in interference detection is critical to prevent unwarranted false positives that could unnecessarily trigger the recovery pipeline, adding undue processing latency and also introducing errors in the resulting depth images.

B. System Architecture

For efficient concurrent operation, it is essential to ensure that the operating schedules of different sensors are sufficiently de-conflicted. However, this is challenging without direct communication. Even if communication was possible, localizing sensors relative to each other remains challenging without techniques such as point cloud registration. Additionally, broadcasting sensor depth images to neighboring LiDAR devices will require significant wireless bandwidth.

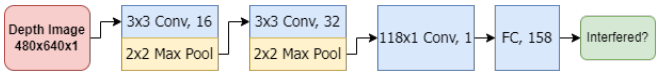


Fig. 4: Interference Detection Model

To address these challenges, we propose *D2SR*, a closed-loop control system. It autonomously detects interference and optimizes resource sharing among LiDAR units, eliminating the need for inter-device coordination or communication. *D2SR* aims to consistently deliver low-error depth images while minimizing interference. Illustrated in Figure 1, our solution comprises three key stages.

1) Interference Classification: Initially, the interference classification model identifies any interference present within a given depth image.
2) Hardware Sync Control: The second stage involves the hardware synchronization controller, which dynamically adjusts the duty cycle's period while maintaining a constant pulse width. This ensures efficient resource allocation, with reduced conflicts, across multiple sensors.
3) Recovery Model: The final stage encompasses the recovery process, which serves a dual purpose: (a) identifying interfered regions within the depth image, and (b) refining the depth values for these identified interfered regions.

V. *D2SR's* Key Components

In this section, we delve into the technical details of the key components comprising *D2SR*.

A. Interference Detection Model

The interference detection model classifies individual depth frames as either "interfered" or "clean". It is a pivotal component of *D2SR* as it not only triggers the *recovery* stage for "interfered" images but also impacts the duty cycle adjustments performed by the *de-synchronization* stage.

Illustrated in Figure 4, our neural network-based model comprises convolutional layers, each followed by Leaky ReLU activation, with the final layer utilizing a sigmoid function for activation. Interference pattern detection involves scanning the entire image to identify potential interference noise patterns, as it is hard to differentiate random vs. interference noise within smaller patches. While increasing the receptive field through additional convolutional layers could enhance performance, it would concurrently increase computational demands and diverge from our on-device, high throughput design objective. Consequently, we have opted to modify the filter size to an asymmetric shape of (118, 1) in the final convolutional layer, which improved model performance without significantly increasing computational complexity.

To train the model, we compiled a dataset containing frames from multiple Intel L515 LiDARs, This dataset comprises 7,945 images acquired from three distinct environments: office, meeting room, and a university lab space, with people moving around naturally. Within these environments, four LiDARs remained stationary while one LiDAR moved to capture depth images, both with and without interfering with the others. We partitioned the dataset into 80% for training and 20% for testing, using *BinaryCrossEntropyLoss* as the loss function for training.

We evaluate the model's performance using F1-score, precision, and recall. The model should exhibit higher precision, as it functions as a gating mechanism within the pipeline. Conversely, enhancing recall may lead to more frequent invocation of the recovery stage, thereby potentially increasing the computational overhead of *D2SR*. Our model achieves high performance on both the training and test sets, with F1-scores of 90% and 89%, precision of 99% and 98%, and recall of 83% and 82% respectively.

B. Decentralized De-Synchronization

Our proposed decentralized de-synchronization approach seeks to reduce (but not eliminate) the interference between LiDAR sensors, thereby increasing the fidelity of individual depth frames. However, such increased fidelity will come at the expense of a lower frame rate, as an individual LiDAR will be actively emitting pulses less often.

Our approach involves dynamically adjusting the sensor's duty cycle *phase* by monitoring the fraction of interfered frames over a predefined *observation* period. In *D2SR*, we configure this period to last one second. During this period, we carefully analyze the output of the interference detection model to facilitate adjustments to the duty cycle, with the goal of minimizing the occurrence of interfered frames.

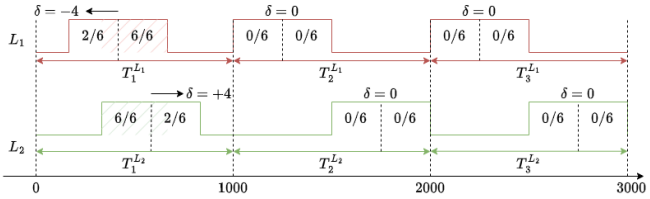


Fig. 5: An Example of Dynamic Duty Cycle Algorithm

Algorithm 1 Dynamic Duty Cycle

$n \leftarrow 12$
$total\_frames \leftarrow 24$
$t_{start} \leftarrow random(0, total\_frames - n)$
loop
  $\delta = \sum_{i=1}^{n/2} buffer[i] - \sum_{i=n/2}^{n} buffer[i]$
  $t_{start} = max(0, min(n, t_{start_t} + \delta)$
  $t_{end} = max(0, total\_frames - t_{start} - n)$
  $sleep(t_{start})$
  $laser\_on()$
  $sleep(n)$
  $laser\_off()$
  $wait(t_{end})$
end loop

Let $total\_frames$ be the maximum number of frames the sensor can capture per second, and n be the number of frames captured during the active period (i.e., predetermined target frame rate). A $buffer$ will store the output of the interference detection model for these n frames. Following each

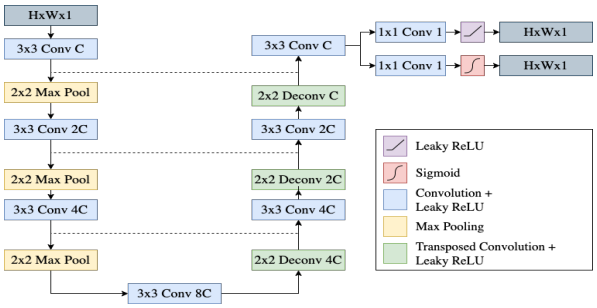


Fig. 6: Architecture of Recovery Model

observation period, the duty cycle is adjusted by δ, calculated as per Algorithm 1. This δ represents the difference in the number of interfered frames between the first and second halves of an active period. In the initial observation period, the active period starts after a random offset.

The dynamic duty cycle algorithm (Algorithm 1) is illustrated in Figure 5 and seeks to *de-conflict LiDAR devices by essentially time-shifting the active period for each individual device*. Let's consider a scenario involving two LiDAR units ($L1$, $L2$) that overlap spatially and temporally. We set n to 12, corresponding to a frame rate of 12 frames per second. For simplicity, let's assume that the observation periods of both LiDARs are synchronized, although achieving such alignment may not be feasible in real-world scenarios.

In the initial observation period, both LiDARs encounter a combined total of 8 interfered frames, with $L1$ and $L2$ experiencing 6 interfered frames, respectively, in the first and second halves of their active period. As a result, $L1$'s duty cycle will be adjusted backward by 4 frames, while $L2$'s duty cycle will be adjusted forward by 4 frames for the subsequent observation period, thereby deconflicting their active periods. Consequently, no further adjustments will be made to the duty cycles of either LiDAR unit. In implementing *D2SR*, we leverage a feature of the Intel L515 sensor allowing laser control via an external trigger[1], enabling laser activation and deactivation as needed.

C. Recovery of Depth Images

Refining interfered depth images involves a two-step process: (a) identifying the interference-affected regions and (b) estimating depth values for these areas using available data. The effectiveness of our recovery models depends heavily on the interference classification model's performance. Erroneous triggers (false positives, generated due to similarity between random and interference noise patterns) can degrade overall recovery performance. Since the recovery model is only trained to recover the interference error, false positives can lead to inaccurate depth value reconstruction for such regions, thereby increasing the overall reconstruction RMSE. Hence, precise interference classification is crucial for ensuring recovery process reliability.

As depicted in Figure 6, the recovery component uses a streamlined version of the UNet model [20]. Our approach

[1]https://dev.intelrealsense.com/docs/lidar-camera-l515-multi-camera-setup

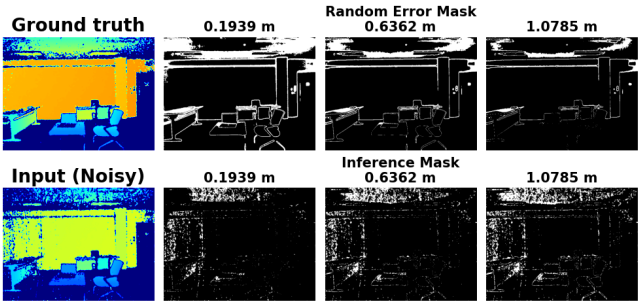


Fig. 7: Thresholds Comparison for Interference/Random Error Mask Estimation

integrates two heads that serve two functions: segmenting interference masks and regressing missing or inaccurate depth estimations within those masks. Consequently, our model functions as a unified system, performing both classification and regression tasks within a single backbone architecture.

Let x_i and y_i represent the input (noisy) and the ground truth depth images, m_i and $\hat{m}_i$ denote the ground truth and estimated interference masks, and $\hat{x}_i$ represent the regression output of the recovery model. The output depth image $\hat{y}_i$ is computed using the formula $\hat{y}_i = \hat{x}_i * \hat{m}_i + x_i * (1 - \hat{m}_i)$. *SmoothL1Loss* function minimizes the error between y_i and $\hat{y}_i$ and *BinaryCrossEntropyLoss* minimizes the loss between m_i and $\hat{m}_i$.

1) Ground Truth Interference Mask: The key challenge in the recovery task stems from generating accurate ground truth interference mask, especially in small regions where they resemble other noise patterns. To tackle this challenge, we employ a two-step approach for generating the ground truth.

We collect data in a stationary environment utilizing a fixed set of LiDARs. Each LiDAR captures a sequence lasting approximately 20 seconds, consisting of interference-free depth images. Following this, we capture multiple sequences of depth images while activating combinations of 2, 3, 4, and 5 LiDAR sensors simultaneously. This enables us to gather a varied set of interference patterns. We repeat this procedure across six unique layouts, each featuring variations in furniture arrangement and LiDAR placement within a static environment. In total, we gathered a dataset comprising 78,000 interfered depth images.

Masks for random errors are initially created, comprising pixels with higher depth variation under interference-free operation, for each stationary LiDAR position. To create such masks, the median depth image for a fixed field of view is computed by finding the pixel-wise median depth value across the sequence. Subsequently, the mean pixel-wise error is calculated between the median frame and each frame within the sequence, and outliers are identified from the distribution of these errors by applying different standard deviation (σ) based thresholds. Figure 7 depicts the variation of the random error mask based on different thresholds $\{\mu, \mu + \sigma, \mu + 2\sigma\}$ of the pixel-wise error distribution.

Next, we compute the error between the same median frame and the interfered frames from other sequences cap-

tured for the same field of view. Utilizing identical thresholds, an error mask is generated, encompassing both random error and interference error. Then, the interference mask for each image is obtained by removing the intersection of the random error mask and the error mask calculated for each depth image. The final output for different thresholds is shown in the last row of Figure 7.

2) Ground Truth Depth Images: For each noisy depth image, the ground truth depth image is sampled from its corresponding interference-free sequence. However, as random errors fluctuate across frames, zero values can inflate the final computed RMSE. To address this issue, we employ a technique where zero-valued pixels outside the interference mask in the noisy depth image are matched with the corresponding locations in its ground truth image, and vice-versa, ensuring consistency during the RMSE calculation.

D. Training and Evaluation

As seen in Figure 6, the number of filters (C) in the recovery model doubles in each subsequent convolution layer of the encoder, while it correspondingly reduces in successive blocks of the decoder. We consider three recovery models with different numbers of first-layer filters, evaluating them based on the F1-score for interference mask estimation and RMSE (Table I) between reconstructed and ground truth depth images.

Model	F1 Score (%)		RMSE (cm)	
	Train	Test	Train	Test
Identity	-	-	80.73	80.74
UNet32	0.86	0.86	39.60	39.78
UNet16	0.84	0.84	42.93	43.06
UNet8	0.80	0.80	47.89	48.07

TABLE I: Recovery models' performance

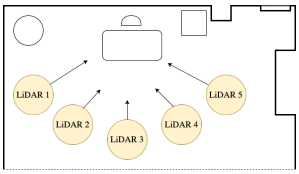
The influence of the random error has been removed from the dataset. Consequently, the reported RMSE reflects the error from the interference. The *Identity* model refers to the error between the ground truth depth image and the input (interfered) depth image. Increasing the input channels of the initial convolution layer reduces the RMSE by 40.5%, 46.7%, and 50.7% for 8, 16, and 32 channels, respectively.

VI. Evaluation

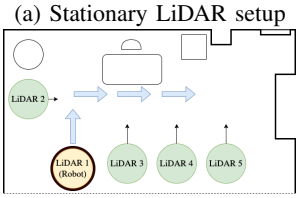
This section outlines our experimental setup, the dataset collected, and the suite of evaluation metrics utilized for a comprehensive performance evaluation of *D2SR*.

Experiment Setup: To evaluate the performance of *D2SR*, we conduct multiple experiments in two different settings.

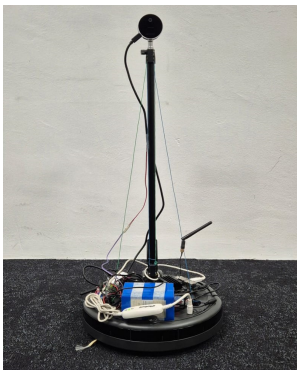
- **Stationary LiDARs:** We place five static LiDARs (at an elevation of 90 cm from the ground) in an arc (as shown in Figure 8a). Each LiDAR spatially overlaps with every other LiDAR.
- **Moving Robot's LiDAR:** A robot platform with a LiDAR mounted at a height of 80 cm (see Figures 8b and 8c) moves around the test area. An additional four stationary LiDAR units are strategically positioned in the vicinity to interfere with the LiDAR on this robot.



(a) Stationary LiDAR setup



(b) Setup with Moving robot



(c) The robot platform

Fig. 8: Experiment Setup: Stationary & Moving Robot LiDARs

Data Collection: For the stationary LiDAR experiment, we adopt an approach similar to that used to train the recovery model. To obtain the ground truth interference masks and depth images, we first capture a sequence of depth images without interference. Subsequently, we record two sequences for different numbers of spatially overlapped LiDARs (e.g., 2, 3, 4, and 5) and at five distinct frame rates (e.g., 25, 20, 15, 10, and 6), resulting in a total of 45 sequences.

In the second scenario with the moving robot, data collection occurs in two phases to independently evaluate the de-synchronization and recovery models. Sequences are recorded with different spatially overlapped LiDAR configurations (e.g., 1 to 5 LiDARs) and three frame rates (30, 15, and 6), totaling 12 sequences. Only the moving LiDAR's stream is utilized for evaluation in each combination, lasting around 30 seconds along a predetermined trajectory. Obtaining ground truth interference masks and depth images for the moving LiDAR is essential for assessing the recovery process. However, collecting this data while the LiDAR is in motion and synchronized with other LiDARs poses challenges. Therefore, the robot pauses at four different locations along the trajectory, capturing one interference-free sequence and two sequences involving varying numbers of spatially overlapped LiDARs (e.g., 2, 3, 4, and 5) at 30 fps at each stop.

Evaluation Metrics: We establish the following metrics to assess the performance of our system.

1) *Interference Ratio*: This metric quantifies the proportion of interfered frames relative to the total number of frames per sequence.

2) *Frame Rate*: Defined as the total number of frames divided by the sequence duration.

3) *F1-Score*: Utilized to assess the performance of interference mask segmentation during the recovery phase.

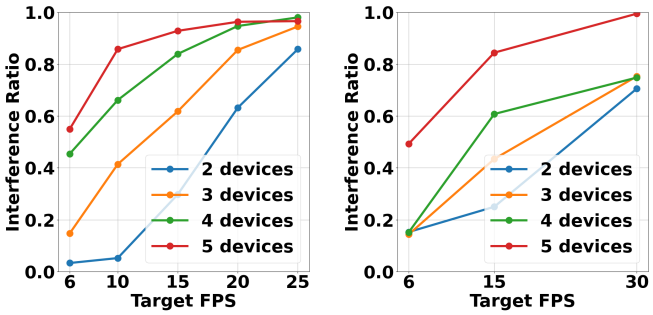
4) *Root Mean Squared Error (RMSE)*: Measures the disparity between the predicted and ground truth depth images.

A. Dynamic Duty Cycle Algorithm's Performance

We illustrate the performance of the dynamic duty cycle algorithm by evaluating the interference ratio (fraction of interfered frames) under different frame rates. The trade-off between interference ratio and frame rate is shown for

stationary LiDARs and LiDAR setups on a moving robot in Figure 9. While our system operates without assuming the number of LiDARs, optimal duty cycle adjustment (i.e., timesharing) requires knowledge of the LiDAR counts. For instance, with two spatially overlapped LiDARs at a frame rate of 15, the interference ratio is 29.7%, including duty cycle adjustment duration. As both the frame rate and number of LiDARs increase, the interference ratio escalates due to greater temporal overlap among the duty cycles of multiple devices.

Both the stationary LiDARs and the LiDAR on the moving robot exhibit comparable performance. However, with certain static LiDAR combinations and moving LiDAR, the spatial overlap may not be consistent throughout the trajectory. Hence, the interference ratio might not consistently reach 1 at the highest frame rate, often dropping below stationary LiDAR setups. This mimics real-world scenarios where continuous spatial and temporal overlap among moving LiDARs is not guaranteed.



(a) Stationary LiDARs (b) Moving robot's LiDAR

Fig. 9: Interference Ratio vs. (Frame Rates, No. of LiDARs)

B. Recovery Performance

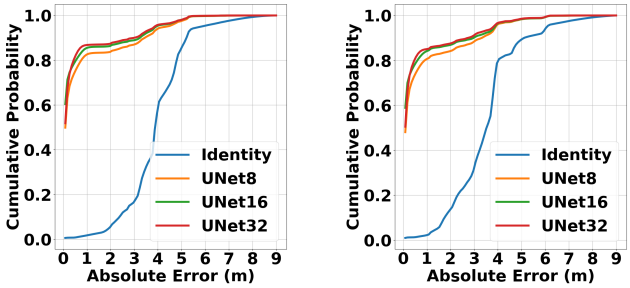
Table II presents the F1-score for interference mask segmentation and the RMSE difference in the stationary LiDAR setup. A minor performance decline is observed compared to the training dataset, attributed to its limited size. The RMSE improvement ranges from 31.6% to 36.0% (max reduction ≈21.85 cm) compared to the identity model, indicating overfitting of the large model. Similarly, the recovery performance, varying between 11.5%–24.9%, diminishes (compared to the stationary training set) for the moving robot's LiDAR. This decline is primarily due to reduced F1-score in interference mask segmentation, evident in Figure 10's cumulative distribution plot for pixel-wise absolute error. Notably, approximately 80% of interfered pixels are rectified by the recovery models. However, false positives in the estimated mask have increased the overall RMSE. It's important to note that the reconstruction process currently relies solely on a single depth image; further improvements, potentially using a sequence of such images and additional RGB images may be explored in the future.

C. Latency

We present the latency of interference detection and recovery models on diverse pervasive devices, such as Jetson Nano 4GB, Jetson Xavier NX 6GB, and Jetson AGX Orin 32GB. Table III depicts the inference times for each model.

	Stationary LiDARs		Moving Robot's LiDAR	
Model	**F1-score**	**RMSE (cm)**	**F1-score**	**RMSE (cm)**
Identity		60.64		53.89
UNet8	0.75	38.79	0.65	40.45
UNet16	0.74	40.72	0.63	43.78
UNet32	0.73	41.47	0.59	47.66

TABLE II: Performance of the recovery Models for stationary and moving robot's LiDARs setups



(a) Stationary LiDARs (b) Moving Robot's LiDAR

Fig. 10: Cumulative Distribution of RMSE for Recovery Model in Stationary and Moving Robot Setup

Our interference detection models achieve impressive processing rates: the Jetson Nano 4GB, Jetson Xavier NX 6GB, and Jetson AGX Orin 32GB devices reach 52, 170, and 379 fps, respectively, exceeding the camera's real-time operation of 30 frames per second and facilitating faster decision-making for the duty cycle algorithm. Additionally, accelerated frame rates are feasible across all recovery model variants, with the Jetson AGX Orin achieving rates ranging from 63 to 164 fps. However, the Jetson Nano devices achieve lower frame rates of 9 and 14 fps for UNet8 and UNet16 variants, respectively, due to memory constraints.

Model	AGX Orin 32GB	Xavier NX 6GB	Nano 4GB
UNet8	6.1	15.2	67.9
UNet16	8.8	25.0	117.2
UNet32	15.7	60.2	-
Interference Detection	2.6	5.9	19.1

TABLE III: Inference Time (msec) of the recovery model variants and the interference detection model

Our results show that we are able to achieve to efficient interference mitigation across a variety of pervasive devices via a judicious, resource-aware selection of interference detection and recovery models. In resource-rich environments, advanced models achieve accelerated frame rates and robust interference suppression, while resource-constrained settings benefit from lower-demand models for optimal performance.

VII. Discussion

Recovery of missing depth images: To optimize dynamic duty cycling efficiency, configuring the frame rate below the sensor's maximum threshold is crucial. We can extend the UNet model to reconstruct missing depth images by fusing features from a prior depth image and the current color image. However, performance degradation is likely when the

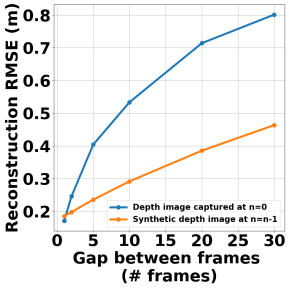


Fig. 11: Reconstruction RMSE: Synthetic vs. Prior Captured depth image

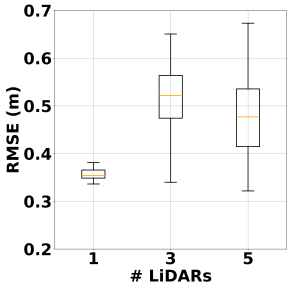


Fig. 12: Impact of Interference on ArduCAM ToF LiDARs

LiDAR deviates from the FoV associated with the prior depth image. Figure 11 illustrates an increase in reconstruction RMSE with expanding frame intervals. We see that RMSE is reduced by using the synthetic (generated) depth image from the preceding step as an input for the current frame.

Evaluating interference impact on a Flash LiDAR: While *D2SR* is tailored for scanning LiDARs, we investigated interference effects on a low-cost flash LiDAR, ArduCAM ToF[2]. The flash LiDARs are susceptible to interference when their flashes synchronize, although the flash duration is shorter than scanning time. Using a scene with five differently sized boxes on a tabletop, we evaluated RMSE (following previous methods) shown in Figure 12 for configurations with 1, 3, and 5 LiDARs simultaneously. We see that the performance degradation due to interference occurs even in flash LiDARs.

VIII. Conclusion

We have demonstrated the feasibility and performance of *D2SR*, addressing key challenges in shared LiDAR environments. Through rigorous evaluation, we have demonstrated the adaptability and effectiveness of our approach across diverse scenarios. Specifically, experimental results reveal that *D2SR* achieves: (a) a reduction of 70.3% in interfered frames with two LiDAR units operating at half their frame rate, (b) a significant improvement in depth estimation accuracy by 27%, and (c) a 75.1% reduction in interfered frames and a 24.9% decrease in depth estimation error for a moving robotic platform. These findings underscore *D2SR's* capability to mitigate interference and recover interfered frames among multiple LiDAR units, both in stationary and mobile environments. Overall, our study highlights how decentralized techniques can enhance the robustness and reliability of multi-LiDAR systems in real-world applications.

Acknowledgment

This work was supported by National Research Foundation, Prime Minister's Office, Singapore, both under its NRF Investigatorship grant (NRF-NRFI05-2019-0007), and its Campus for Research Excellence and Technological Enterprise (CREATE) program. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore MIT Alliance for Research and Technology (SMART) centre. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

[2]https://www.arducam.com/time-of-flight-camera-raspberry-pi

References

[1] M. Elhousni and X. Huang, "A survey on 3d lidar localization for autonomous vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1879–1884.

[2] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot slam system for 3d lidars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1004–1011.

[3] T. H. Chan, H. Hesse, and S. G. Ho, "Lidar-based 3d slam for indoor mapping," in *2021 7th International Conference on Control, Automation and Robotics (ICCAR)*, 2021, pp. 285–289.

[4] T. Hu, Y. Sun, W. Jia, D. Li, M. Zou, and M. Zhang, "Study on the estimation of forest volume based on multi-source data," *Sensors*, vol. 21, no. 23, p. 7796, 2021.

[5] B. I. Justusson, *Median Filtering: Statistical Properties*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 161–196.

[6] R. M. Martín, M. Lorbach, and O. Brock, "Deterioration of depth measurements due to interference of multiple rgb-d sensors," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4205–4212.

[7] G. Kim, J. Eom, and Y. Park, "An experiment of mutual interference between automotive lidar scanners," in *2015 12th International Conference on Information Technology-New Generations*. IEEE, 2015.

[8] J. Eom, G. Kim, S. Hur, and Y. Park, "Assessment of mutual interference potential and impact with off-the-shelf mobile lidar," in *Bragg Gratings, Photosensitivity and Poling in Glass Waveguides and Materials*. Optica Publishing Group, 2018, pp. JTu2A–66.

[9] E. Curto and H. Araujo, "An experimental assessment of depth estimation in transparent and translucent scenes for intel realsense d415, sr305 and l515," *Sensors*, vol. 22, no. 19, p. 7378, 2022.

[10] J. Robinson, A. Venturi, R. Dudley, M. Bevilacqua, and V. Donzella, "Methodology to investigate interference using off-the-shelf lidars," in *2022 2nd International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*. IEEE, 2022, pp. 124–130.

[11] S. Grollius, S. Grosse, M. Ligges, and A. Grabmaier, "Optimized interference suppression for tcspc lidar," *IEEE Sensors Journal*, vol. 22, no. 24, pp. 24 094–24 101, 2022.

[12] L. Carrara and E. Fiergolski, "An optical interference suppression scheme for tcspc flash lidar imagers," *Applied Sciences*, vol. 9, 2019.

[13] G. Kim, J. Eom, and Y. Park, "Design of pulsed scanning lidar without mutual interferences," in *Smart Photonic and Optoelectronic Integrated Circuits XX*, vol. 10536. SPIE, 2018, pp. 168–173.

[14] J.-D. Chen, K.-W. Wu, H.-L. Ho, C.-T. Lee, and F.-Y. Lin, "3-d multi-input multi-output (mimo) pulsed chaos lidar based on time-division multiplexing," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 28, no. 5: Lidars and Photonic Radars, pp. 1–9, 2022.

[15] G. B. Popko, T. K. Gaylord, and C. R. Valenta, "Interference measurements between single-beam, mechanical scanning, time-of-flight lidars," *Optical Engineering*, vol. 59, no. 5, 2020.

[16] Y. Hai, Z. Yan, A. Pang, C. Liu, and A. Dang, "Immunity to mutual interference in coded phase-shift lidar," *Journal of Lightwave Technology*, vol. 41, no. 2, pp. 570–577, 2022.

[17] A. L. Diehm, M. Hammer, M. Hebel, and M. Arens, "Mitigation of crosstalk effects in multi-lidar configurations," in *Electro-Optical Remote Sensing XII*, vol. 10796. SPIE, 2018, pp. 13–24.

[18] G. Kim, J. Eom, S. Park, and Y. Park, "Occurrence and characteristics of mutual interference between lidar scanners," in *Photon Counting Applications 2015*, vol. 9504. SPIE, 2015, pp. 76–84.

[19] Z. Liu, W. Hunt, M. Vaughan, C. Hostetler, M. McGill, K. Powell, D. Winker, and Y. Hu, "Estimating random errors due to shot noise in backscatter lidar observations," *Applied optics*, vol. 45, no. 18, pp. 4437–4447, 2006.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, proceedings, part III 18*. Springer, 2015.