Master of Science Thesis

A Semantic-Aware ROI-Based LiDAR Sampling Algorithm

August 2018

Graduate School of Engineering

Seoul National University

Electrical and Computer Engineering Major

Nguyen Khac Thai

A Semantic-Aware ROI-Based LiDAR Sampling Algorithm

Supervisor: Prof. Hyuk-Jae Lee

This work is submitted as a Master of Science thesis

August 2018

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

Nguyen Khac Thai

Confirming the master's thesis written by
Nguyen Khac Thai
August 2018

Chair _____

Vice Chair _____

Examiner _____

Abstract

The acquisition of laser range measurements (i.e. light detection and ranging sensor (LiDAR)) can be a time-consuming process if a high spatial resolution is required. Hence, designing an effective sampling acquisition is of high importance for many laser range applications. Previous approaches such as two-step sampling are useful in general cases in indoor and less complex scene. However, they show a deficiency in the real-world outdoor complex situation, especially in the condition of a very low sampling rate. To address this problem, this paper proposes a semantic-aware ROI-based sampling algorithm. Taking the merit of existing road and object detection algorithms, i.e. YOLO, our method utilizes the semantic information; and then effectively distribute the sample budgets to maximize the reconstruction quality, especially in object areas. Experimental results show that our proposed method significantly reduces the mean-absolute-error (MAE) in the object area and overall area compared to two-step sampling by 85.6% and 32.9%, respectively. In addition, it achieves robust reconstruction quality with an extremely low sampling rate (i.e. 1% in our experiment).

Keyword: ADAS, Advanced driver-assistance system, Light Detection and Ranging sensor, LiDAR, sampling algorithm, sampling strategy

Student Number: 2016-24658

Table of Contents

List of Figures

List of Table

CHAPTER 1 - INTRODUCTION

1.1. Overview

The advanced driver-assistance systems, or ADAS, is considered the next logical step of the car industry. They are considered as the new revolution that changes the way people live and transport, not just grants more personal freedom, allow people to free their hands and mind while driving. Self-driving cars are also expected to vastly reduce road accidents, congestion, pollution and eliminate the huge cost of owning personal vehicles when integrating with blooming share-driving services [1]. There are five driving levels, starting with level 0 of no automation to level 5 of complete automation where drivers simply just enter the destination and let the systems do the rest of work. While the ultimate goal is a safe and fully autonomous driving system, current commercial systems now can constantly sense the surrounding environments and immediately alert the drivers of danger, traffic situation and road condition. Being considered a key to the future, ADAS has been actively developed in recent year [2] [3] not just in academia but many companies are competing to release the first commercial system.

In the stream of those developments, object detection algorithms have been experiencing rapid developments. Their accuracy and speed have been improved significantly over time. Among them, YOLO is a real-time model

which is fast, efficient and optimized end-to-end and thus provides better efficiency with respect to the computational cost [4]. YOLOv3, which could have frame rate up to 40-50fps, is much faster than the typical frame rate of a LiDAR system. Besides, the road detection, which could be considered as an easier problem has also been solved effectively. Many road detection approaches submitted to KITTI road detection benchmarks have the precision of more than 96% while the runtime is just about 0.06s [5].

1.2. LiDAR and LiDAR sampling

In ADAS, the sensor system is the critical part which let the car monitor surrounding environment. For the task of detecting obstacles on the road, cameras and light detection and ranging sensors (LiDAR), such as Velodyne LiDAR, have been proved to be valuable in these systems. LiDAR is the range sensing sensor that use a laser pulse to estimate distances to surrounding objects by measuring the time of flight of the laser beam. The quality of the measurements is affected by reflection property and angle of the reflecting surface. Many measurements are missing because the laser beam does not return. It is common for a LiDAR system to have more than 1 laser unit, ie. the Velodyne HDL-64E has 64 lasers that can measures up to more than 2.2 million points per second [6]. In our lab, a LiDAR model has been also developed [7]. Having the same problem with other systems, our LiDAR is

limited with 100.000 measurements per second. However, having been equipped with two-mirror deflection scanner, it has the flexibility and capability to sample any position in the Field of View by any given order. This improvement opens an opportunity for developing a sampling algorithm that could exploit the sampling budget in a non-uniform scanning pattern to enhance reconstruction result.

While being able to produce high-quality depth information, the scanning rate of LiDAR system is relatively slow, typically 10fps [6]. In other words, the sampling budget for a given duration is limited. An efficient sampling technique could help to overcome this limitation. Reducing the requirement for spatial resolution could increase the scanning rate or lower the cost of the system. This thesis focuses on developing an effective LiDAR sampling algorithm for real-world situation using semantics cues.

1.3. Thesis outline

The rest of this thesis is organized as follows. Some background and related works are described in Chapter 2. Workflow of our system and the sampling algorithm is presented in Chapter 3. The evaluation method and the experimental results are presented in Chapter 4. Chapter 5 summarizes and concludes the thesis.

CHAPTER 2 - BACKGROUND AND RELATED WORKS

In this thesis, various terms related to our work are presented. This section gives a brief introduction to them along with previous works of our research. In the first part, Sensor systems in ADAS are presented including camera and LiDAR used in our system, then methods to calibrate these two sensors together are presented. After that, the current status of object and road detection algorithms is summarized. The second part shows previous researches which consist of depth gradient guided sampling and color image-guided sampling approaches.

2.1. Background

2.1.1 Sensor systems in ADAS

The car is a highly complex system with hundreds of processors and variety types of sensors. These sensor systems give the car the ability to monitor its internal conditions to detect malfunctions. They are also used to sense the external environment, which is the key component to the success of an ADAS system. Active sensors like light detection and ranging (LiDAR) and radio detection and ranging (radar) or passive sensors like cameras are used in experimental and commercial systems.

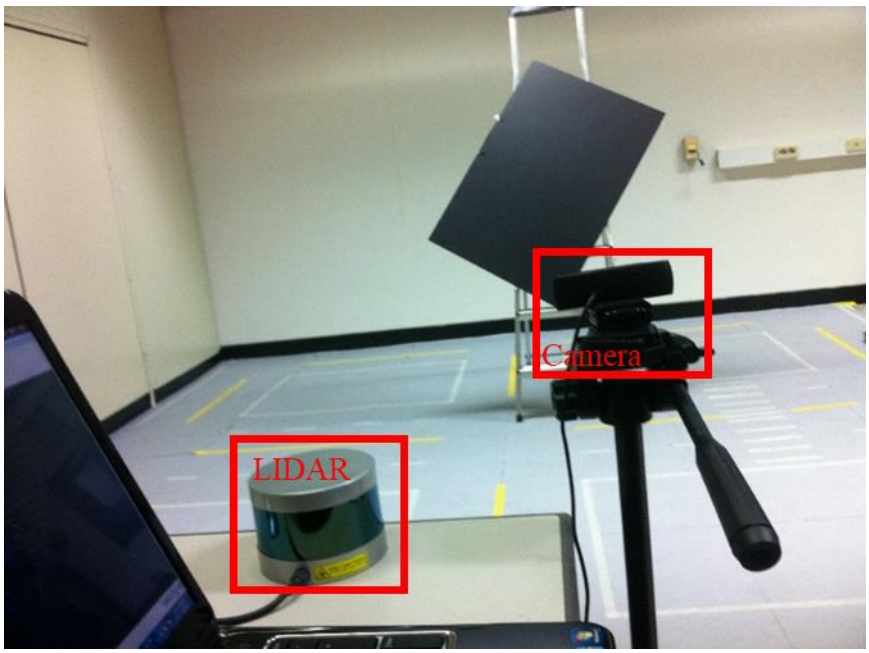


Figure 1. A test system with a Lidar and a camera in our lab

For object detection, camera and LiDAR are well supplementary to each other. The LiDAR has advantages of high-precision range measurement being independent of the distance, unaffected by night condition and light intensity change, however, its resolution is very limited. While cameras are cheap, much higher resolution and frame rate but have the drawbacks of sensitivity to intensity shift and inability to work during night time. Therefore, fusion techniques to combine them together are developed. Calibration is necessary to fuse these data together, which specifies how to convert measurements

from a LiDAR sensor to an image coordination. The next part describes in detail about this calibration step.

2.1.2 Calibration between LiDAR and Camera

Many calibration algorithms use checkerboards as the target. However, checkerboards are often observed with range discrepancies in the LiDAR [8] and that may affect the final estimated outputs [9]. To tackle this problem, Alismail uses a circular target in [9], while it is straightforward to detect the circle's center in the image, finding its 3D correspondence point becomes less reliable in sparse data. In [8], Park uses a monochromatic polygonal board, and estimates its edges and intersects them to find the board's vertices. This method improves accuracy for a low-resolution LiDAR. However, with the monochromatic board, specifying the correct board segment is more challenging in both 2D and 3D range domain. In [8], this task requires manual interventions.

Following, the background subtraction technique is applied to automate that work. First, two samples with and without the board are taken. Then, the background subtraction is performed on both 2D and 3D data while the board area is the largest segment of the subtraction output.

First, the background image and board image are processed as follows. First, these two are converted to grayscale and then subtracted to generate a

binary image by comparing absolute subtraction value and a threshold. Second, the board area is detected as the largest region in the subtraction image. Third, that area is grown to the full board segment using the region growing technique. Finally, its edges are estimated and their intersections are determined as the board's vertices.

Then, for 3D data, both scans are first converted to a range image. Then, the subtraction and board segment detection are carried out as in normal images. The final set of 3D measurements belong to the board is that consists of points corresponding to the board segment on the range image.

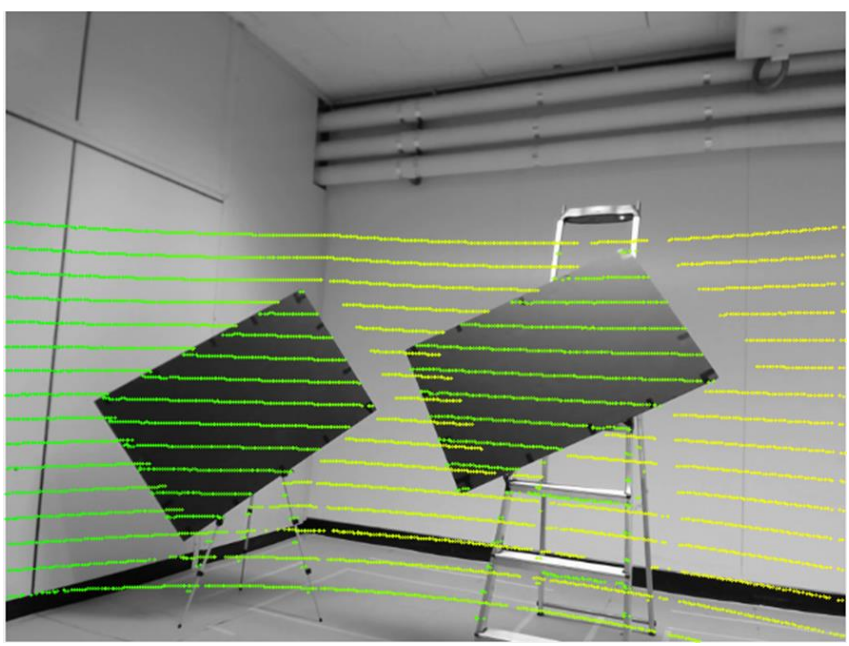


Figure 2. Calibration result. Color indicates distance. Occlusion is observed due to a different viewpoint.

Finally, after extracting corner points on the 2D image and board data from LiDAR, the calibration process is performed as described in [8]. In this work, the perpendicularity property between two adjacent edges of the rectangle is also enforced. That improves the estimation results when there are little points on edges.

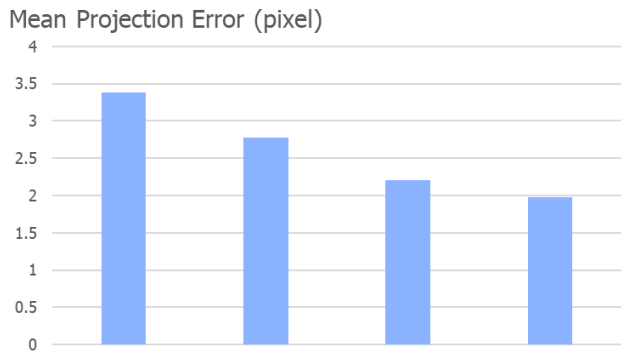


Figure 3. Mean projection error according to the number of samples used for optimization

The experiment is conducted by using a Velodyne VLP16 and a camera. The target is a rectangle slanted to the ground and placed 3-5 meter away from the LiDAR to ensure there are about 10 data lines. The mean projection error in a pixel is provided in Figure 3, which gradually declines when increasing the number of samples used for optimization. Figure 2 depicts the calibration result. Color indicates distance. Occlusion is observed due to a different viewpoint between the camera and LiDAR.

2.1.3 Object detection algorithms

Object detection algorithms have been experiencing rapid developments. In 2012, the first CNN network outperforms human in ImageNet challenge, since then it has become the gold standard in object recognition. Their accuracy and speed have been improved significantly over time. Among them, YOLO is a real-time model which is fast, efficient and optimized end-to-end and thus provides better efficiency with respect to the computational cost [4]. YOLOv3, which could have frame rate up to 40-50fps, is much faster than the typical frame rate of LiDAR system. Besides, the road detection, which could be considered as an easier problem has also been solved effectively. Many road detection approaches submitted to KITTI road detection benchmarks have the precision of more than 96% while runtime is just about 0.06s [5]. CNN networks are also the gold standard in this area.

2.1.4 Iterative Closest Point algorithms

In this section, the definition of Iterative Closest Point (ICP) algorithm will be given. In our system, ICP is used to register between two successive LiDAR scans, the result indicates the 6D movement of the LiDAR sensor, which could be converted to the movement and velocity of the car.

Given two sets of corresponding point X1 and X2, ICP algorithm finds the translation vector t and rotation matrix R that minimize the sum of squared error between these two sets:

$$E(R,t) = \frac{1}{N}\sum_{i=1}^{N}(X1i - R * X2i - t)^2$$

Where $X1_i$ and $X2_i$ are the corresponding points from two sets. This minimization problem is solved in an iterative manner, the gradient of error from the previous iteration is used to set up variables for the next iteration. Above error term is called point to point error. This term is effective for the cases when the number of samples is dense enough. However, with LiDAR the sample density is very sparse, hence, the point to plane error is much more effective and stable.

In more complex systems, visual cues are used to increase the odometry performance and avoiding ICP to fall to a local minimum. [10] (V-LOAM), which is the top performer in KITTI odometry dataset at the moment of my thesis is one of those follow this direction. Visual odometry is used for ego-motion estimation then refinement step is carried out using lidar odometry (LOAM) [11], which itself stands at the second position on the KITTI odometry benchmark. This proves that LiDAR measurements are very important for the odometry problem.

2.2.Related works

This section begins with an introduction to the sampling problem. Then, previous approaches and their limitations are subsequently discussed.

Considering measure a scene consists of MxN sampling location with a total budget of N points per scan. Define a sampling mask that consists of binary value 0 and 1, where a location with the value of 1 indicates the location would be sampled and a location with the value of 0 would not. Depend on each objective, the above sampling mask is optimized with different criteria. In this work, our goal is to minimize the reconstruction error inside the ROI area. There are some sampling techniques can carry out this task. Uniform or grid sampling is the most straightforward and widely used in the real systems. However, they could not exploit additional information to build a better sampling strategy. Several non-uniform sampling strategies [12] [13] [14] [15] have been proposed. Non-uniform sampling techniques for LiDAR can be divided into two categories.

2.2.1 Depth-gradient-guided sampling

The first category using properties of the depth image by assuming that the depth image is already available or performing sampling to acquire a part of it. Simon [12], proposed that sampling along the depth gradient could give

the optimal result. He was able to recover the depth map in Middlebury dataset with only 5% of the sample. However, this approach was not practical since the depth gradient is not available prior to sampling. Therefore, two-step sampling [13] was designed to apply gradient-based sampling in a more practical manner. At the first step, uniform sampling was carried out with a half budget. Then he reconstructed depth from these samples to find the gradient of the reconstructed map. At the second step, the sampling was performed along the gradient of the reconstructed map from the first step. Drawbacks of this method introduce a delay between two-step for the reconstruction of the depth image, which requires very heavy computation. Second, when the sampling rate becomes too low, there are not enough samples to conduct a good reconstruction from the first step so this affects the overall results.

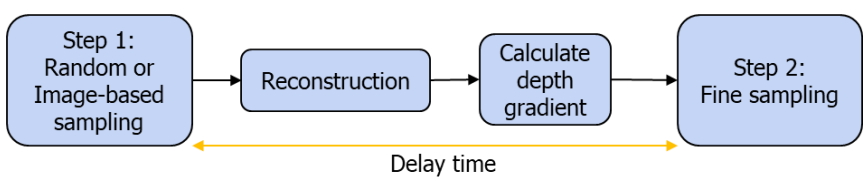


Figure 4. Workflow of two-step sampling

2.2.2 Image-guided sampling

The second category uses an image to guide the sampling process. In [9], Lee uses stereo images to compute a depth disparity map, then samples along

the edge of this raw estimation. However, the depth computation from stereo images is complex and time-consuming, and large occlusion is usually found in the estimated depth. A sampling method which uses saliency information from the color image to select sampling position is proposed in [15]. This method has an advantage that the computation of saliency map is cheaper compared to methods mentioned above, however, it is noisier with many outliers remaining in the saliency map.

Despite different types and categories, all previous methods were introduced and tested in the indoor scenes where the gradient and details of both color image and depth image are much simpler than the outdoor situation. Targeting the real-world environment of ADAS system, our method is not only designed for outdoor scenes but also for a very low sampling rate with a budget about 1000 samples per scan (equivalent to about 1% sampling rate). To achieve this, semantics information detected from cameras is used.

Although this also requires complex computation, we argue that in most existing ADAS systems, object detection is a must-be-done task and some algorithms such as YOLO is real-time and very efficient. Therefore, our target is to inherit results from this step to improve the LiDAR sampling efficiency based on the observation that the road area is much more simpler and can be constructed with very few samples and the object area is more semantically

important. Hence, sampling at the object areas is conducted denser while the sampling rate of road area is kept just enough to a decent reconstruction result. Practically, to enforce this system, a synchronization method between camera and lidar need to be developed. However, this task is left for a future work and not included in this thesis.

CHAPTER 3 - SAMPLING ALGORITHM

In this section, the system architecture and configuration are first described then I will analyze each area inside ROI and finally explain the algorithm to distribute sampling budget between areas.

4.1.System Workflow

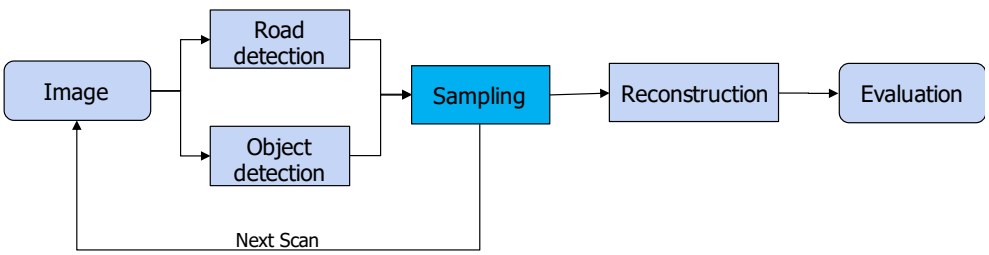


Figure 5. Workflow of our system

The workflow of our system is described as in figure 5. First, an image is captured using a camera. Then, the object and road detection are carried out simultaneously. The result of this step is used to guide our sampling algorithm. At this step, the camera can take the next image immediately instead of waiting for the sampling process to be finished. That way, there will be no delay time in the system as it has in two-step sampling, in two-step sampling the second stage cannot be initiated before the reconstruction and gradient depth calculation of the first step get done. The reconstruction and evaluation step are used for evaluating the sampling algorithm only. They could be performed offline at any time if the samples are stored. The steps require most

computing power in our system is road and object detection. They are usually implemented by a neuron network. The state-of-the-art algorithms can run at about 40-50fps as we mentioned at the background part. This frame rate is already much higher than LiDAR frame rate.

4.2. System configuration

Figure 6. Road and objects mask generated from KITTI dataset

As described in figure 5, our system includes road and object detection step. This thesis only proposes and evaluates the sampling phase so ground truth data is used for those steps from KITTI. The KITTI object detection dataset also contains unrelated objects and cars are not on the road so they must be filtered out manually. On the road, KITTI road dataset only includes

the road in the current direction. The road in the opposite direction and the pavement are not included. In the two-way street, all lanes are included [16]. This is a very reasonable setup since it is the part of the road where almost all traffic situation related to current car happens. There are many available methods for reconstruction step. Recently, there is a domination of the CNN-based methods in this area, while some conventional image processing methods still have good performance [17]. However, in our application, linear interpolation is used for evaluation, in details it is the griddata function in Matlab. That is because we believe that it is a classical method, not specifically modeled for any sampling method, so it is able to give a fair comparison. The evaluation method used is the KITTI which provides evaluation code. In this study, mean absolute error (MAE) is used as the criteria for all comparisons.

$$\mathbf{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

4.3. Characteristics of each area in ROI

Given the total budget of N samples in ROI area, the problem is to specify the budget for road area Ns and budget for object areas No (Ns + No = N), which minimizes the reconstruction error in ROI area. Inside each area, random sampling is used with a minor modification that the closer area

is sampled sparser than further area. The reconstruction error can only be calculated after sampling. Therefore, the MAE result has to be predicted prior to sampling to be able to optimize final output. Which model has better prediction will result in better performance.

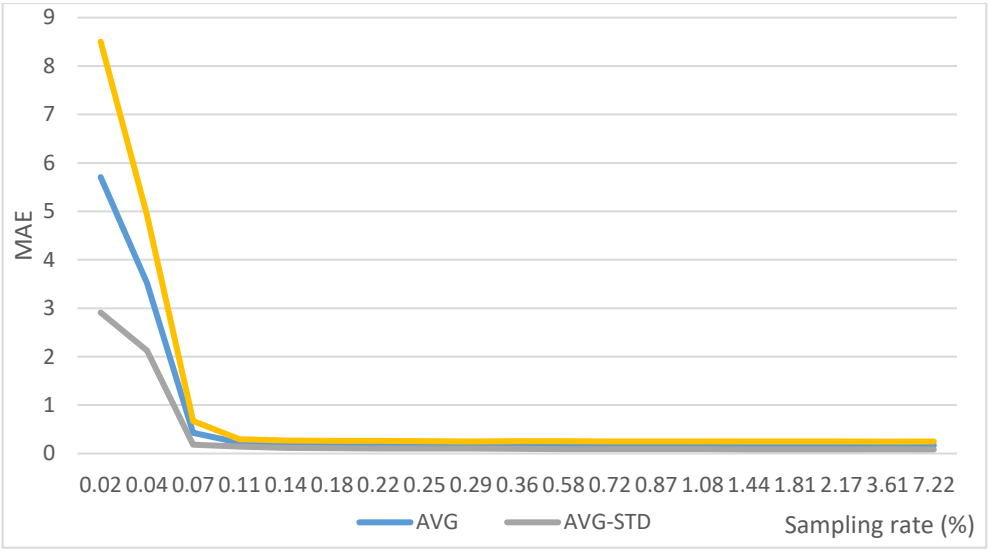


Figure 7. Reconstruction characteristic of the road area

Figure 7 examines the MAE output of road area based on the sampling rate. Three lines are demonstrated: AVG as the mean MAE results of all image, AVG-STD is the mean results minus the standard deviation and AVG+STD is the mean result plus standard deviation. It is clear that the road has much smaller errors and it converges very fast, starting from 0.11%. At this point, the small standard deviation indicates that results among images

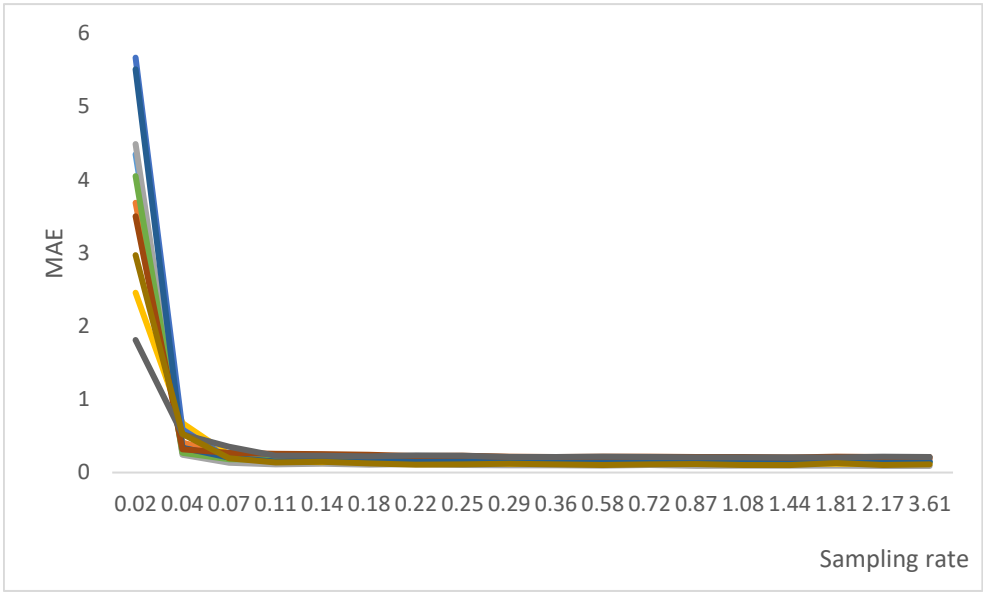


Figure 8. MAE after reconstruction of 10 random scans

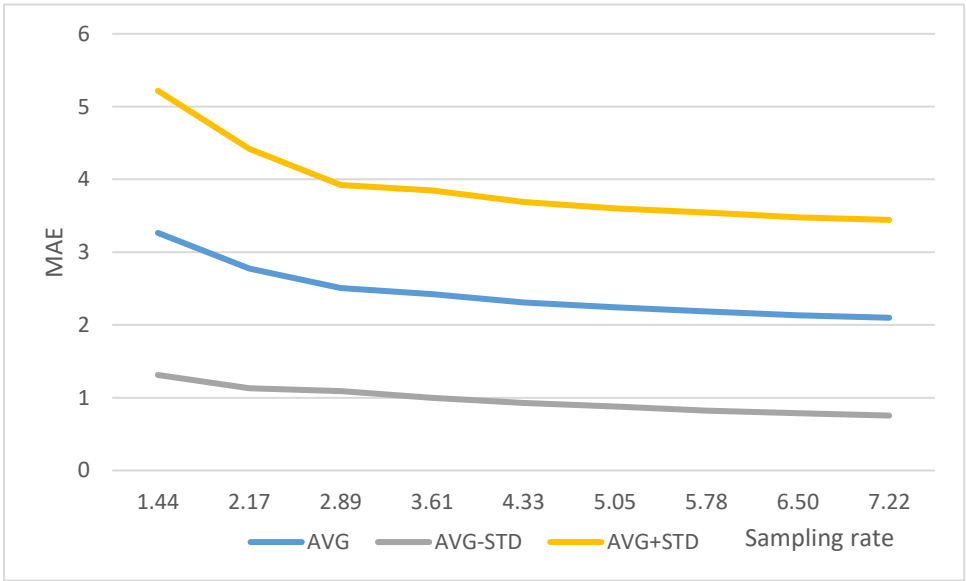


Figure 9. Reconstruction characteristics of objects area

are consistent. This is demonstrated again in figure 8, where MAE of the reconstruction of 10 random scans is shown. The same result holds when they

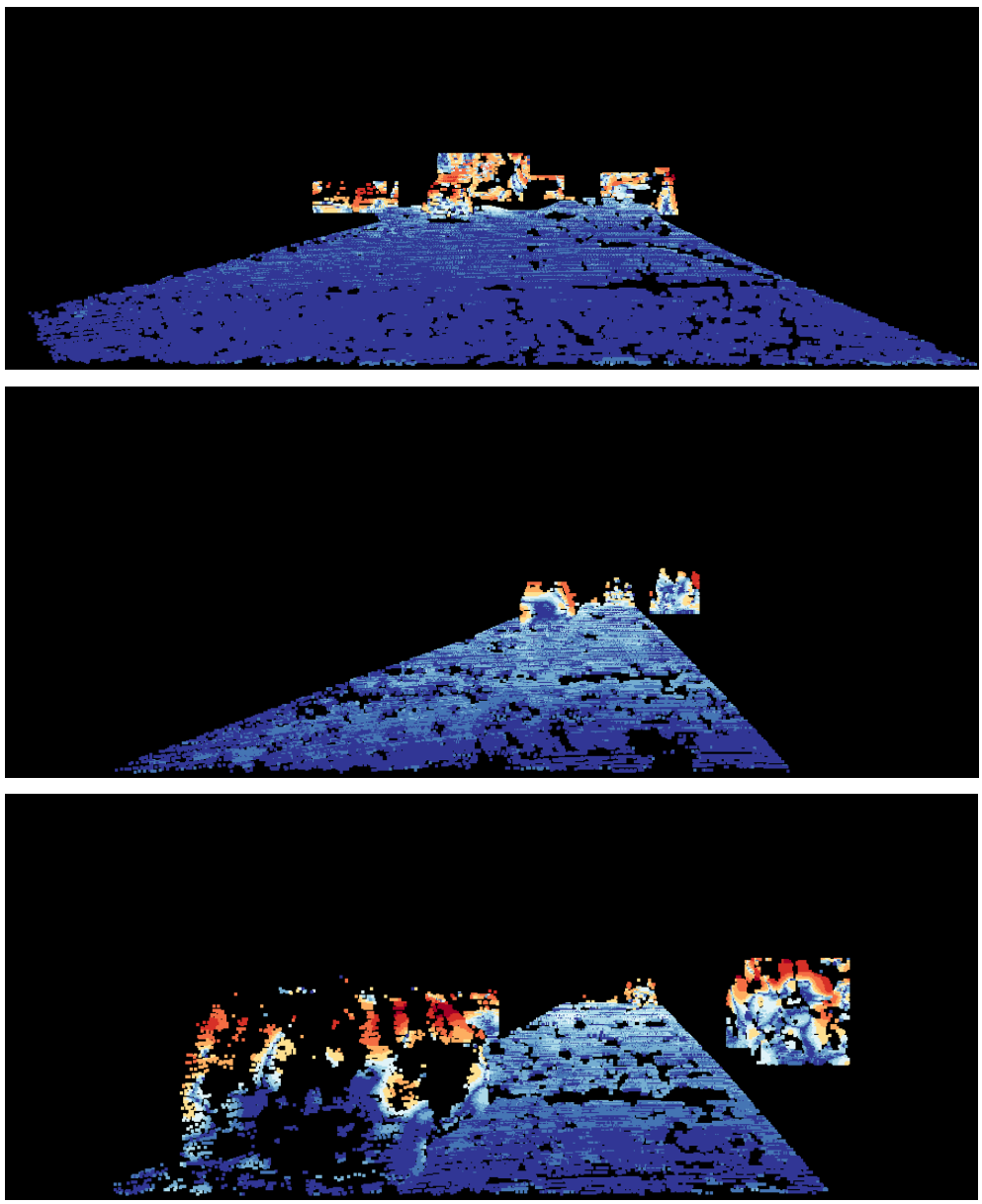


Figure 8. Error image after reconstruction. Blue indicates small error. More red color means higher error.

start to converge at a sampling rate of 0.11%.

The reconstruction characteristics of objects area are shown in figure 9. As we can see, the MAE of objects is worse and no convergence was found. The standard deviation stays high along the sampling rate axis. This means that we have to keep the sampling rate of the objects area as high as possible. Fortunately, the decrease in MAE of this area starts to slow down at 4.33% mean that they begin to become more stable at this point. In figure 10, we can have a comparison of this difference. In all three images, the error in objects area is much higher than in road area. The second image also shows that in road area, the error in the further area is higher than closer area. In the third image, although the object is close, its error is still high. In this particular image, object area is larger than the other two.

As mention before, to distribute sample budget, the MAE result of road and objects area must be predicted. Base on above observation, the characteristics of MAE of road and objects area are modeled into a ration function with three parameters to control its behavior as follows:

$$\mathrm{F}(x) = \mathrm{a} + \frac{b}{x-c} \quad (1)$$

Where a = 0.1627, b = 0.0074, c = 0.0255 for road and a = 1.831, b = 2.011, c = 0.03956 for objects. The SSE of these fitting are 0.0000057 and 0.002,

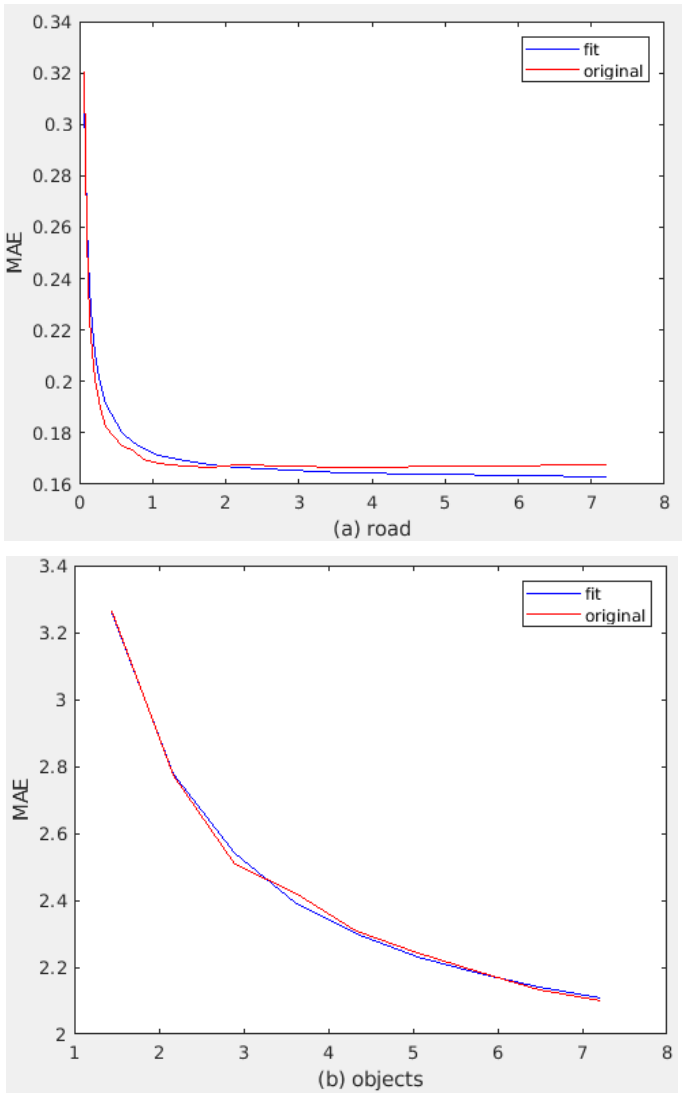


Figure 9. Fitting result of MAE of road and objects area

respectively. We can see the result of these fitting in figure 11. As analyzed as above, the fitting of road area is very reliable since MAE results of all images converge. While the MAE results vary across scenes, hence, we must

fit with mean results of all images, so it may not be good when considering a single case. However, for all test images, this method still gives a good result.

4.4. Distribution budget between road and object area

Based on the model achieved from above section, our algorithm is stated as follow: Given budget N, the area of road Sr and object So in pixels, the problem is to find the budget for the road Nr and budget for object areas No (Nr + No = N) by optimizing following criteria:

$$\text{Minimize } \frac{S_r \times F_1(N_r/S_r)+S_o \times F_2(N_o/S_o)}{S_r + S_o} \quad (2)$$

With F1and F2 is the fitted function for road and object areas in (1), respectively. Sampling budget is finite and integer, so this minimization could be solved numerically effectively.

In KITTI dataset, the road area is about 4-5 times larger than object area. So, when sampling budget is set at 1000 points per scan, the criteria usually converge with about 300-400 samples for road area, the rest of the budget for object area. When the objects area/road area ratio increases this number decrease to reserve more samples for objects area. There are some cases where there are no valid objects, hence all samples are used for the road.

4.5.Distribution budget between ROI and non-ROI area

Ideally, when the objective is increasing the reconstruction results as much as possible, then if all budget is distrusted to the ROI area, the reconstruction results inside ROI will be best. In real ADAS system, although it can be argued that only samples insides ROI will give the semantics meaning and be helpful for tasks like obstacle detection, detecting distance from the car to objects. However, samples in non-ROI area are also helpful for tasks like odometry, detecting motion and movement of the car. When testing odometry using LiDAR samples by Iterative Closest Point (ICP) algorithm, we detected a huge decrease in performance when sampling is performed inside ROI region only. The figures are described in table 1:

Table 1. Odometry errors when sampling in ROI area only vs other setups

	translation driff (m)	rotation driff (degree)
raw depth	0.0385	0.004
random	0.14607	0.01421
ROIonly	0.71255	0.03817

In table 1, we have translation and rotation drift of raw depth (all samples available in LiDAR raw scan), random sampling in the entire image with a budget of 2000 samples and our method when performing sampling inside ROI area only with the same budget of 2000 samples. The odometry result of raw depth is very reliable, the translation drift is less than 4 centimeters, very small and insignificant because the accuracy of ground truth is 10 centimeters.

The result of random sampling is about 4 times higher than that of raw depth, this result is acceptable when considering the number of sample in this set up is just about one tenth of the raw depth. However, when using our algorithm inside the ROI area only, the odometry result is very bad. With the same budget of random sampling, the output is 5 time worse. This shows that to get better odometry result, we also need samples outside ROI area. Our main objective is getting better reconstruction result; therefore, we keep the sampling density inside ROI area denser than that in non-ROI area. The budget for non-ROI will be set only to make its odometry result keep up with that of random sampling.

Given N total budget for entire FOV, call N1 and N2 are budget for ROI and non-ROI area (N1+N2=N), then N1 and N2 are determined as follow:

$$\frac{N1}{S1} = \propto \frac{N2}{S2} \quad (3)$$

Where S1 and S2 are the area of ROI and non-ROI area. When α increases, the reconstruction result inside ROI increases while the odometry result decreases. When α decreases, the opposite is true.

CHAPTER 4 - EXPERIMENTAL RESULTS

This section explains our experiments. First, the KITTI dataset is introduced with adjustments we made to fit our scenarios. Then the objective and a quantitative result are shown and explained in detail.

5.1. Dataset

Raw scan

Ground truth

Figure 10. Raw scan and ground truth depth from KITTI

KITTI dataset is used for our experiments, the raw depth map as input and ground truth data for evaluation. Figure 12 shows an example of the raw and ground truth depth data in KITTI after mapping to its corresponding color image. The raw depth data is taken using Velodyne HDE64 sensor [18]. The

ground truth is generated by combining 11 raw scans together and discarding measurements that are not consistent among those scans [19]. One drawback of this method is that the measurements on moving objects are usually not consistent and being removed, leading to a sparser ground truth on these important areas, whereas the static object areas have denser sample density. This can be seen in figure 13, the cyclist even has more samples in raw depth than in ground truth while technically the ground truth is combined with 11 raw depth images. In some other areas, the ground truth is very dense, some others do not have any ground truth data at. Therefore, in our experiment, even the ground truth has much more data than the raw depth image, the raw depth images give better reconstruction result than ground truth after

Figure 11. Raw depth and ground truth in moving object area

sampling.

We also use ground truth of both object detection and road detection benchmarks in our experiment. Therefore, images that could be used in our

tests must exist in three benchmarks: depth completion, road detection, and object detection. Only a few images, exactly 106 images, match these requirements. For our sampling problem, we believe this is enough to develop our algorithm and give a comprehensive evaluation. Inside ROI, raw depth has the sampling rate of about 7-8%. In table 2, in our dataset, the minimum sampling rate of raw depth data is 3.65%, the maximum is 8%, while the average number is 7.19%. Between road and objects areas, on average the road is 4 times larger than the objects area. However, in some exceptional cases, the objects area is 2-3 times larger than road area. In these cases, the objects are very close to the sensors. About the number of samples inside each area, on average the road areas have 6.75 times more samples than the objects area. These number shows that the current sampling strategy is not efficient. As mention above, the road need smaller sampling rate than objects area, while in this case, we have the opposite.

Table 2. Sampling rate inside ROI and ratio between road and objects area

	Area Object/Road	N. sample Object/Road	Sampling rate inside ROI
Min	0.00344115	0.000524109	0.036568
Max	2.703898226	1.728091529	0.0807
Mean	0.232687667	0.148326604	0.071916

A sampling the budget of 1000 samples per scan is used in our experiment. In table 3, we can see the average number of samples is 7202, this means that we use about 13.89% of available samples. The reconstruction with all samples in raw depth is used as upper-bound performance because that is all measurements available. The random sampling is used as base performance and two-step sampling for comparison. Grid sampling is not possible in our experiments, because the raw depth does not have regular sample grid so make grid sampling from it is not possible.

Table 3. Number of samples inside road and objects area

	Number of samples		
	Total	Road	Object
Min	3527	3348	3
Max	12722	10806	7099
Mean	7202.388	6397.369	805.0194

5.2. Objective evaluation

Figure 14 shows sampling mask of raw depth, random sampling, two-step sampling and our algorithm. There is no specific pattern for random sampling. While two-step sampling is a little denser at object area, our algorithm yields a very significant difference between two areas.

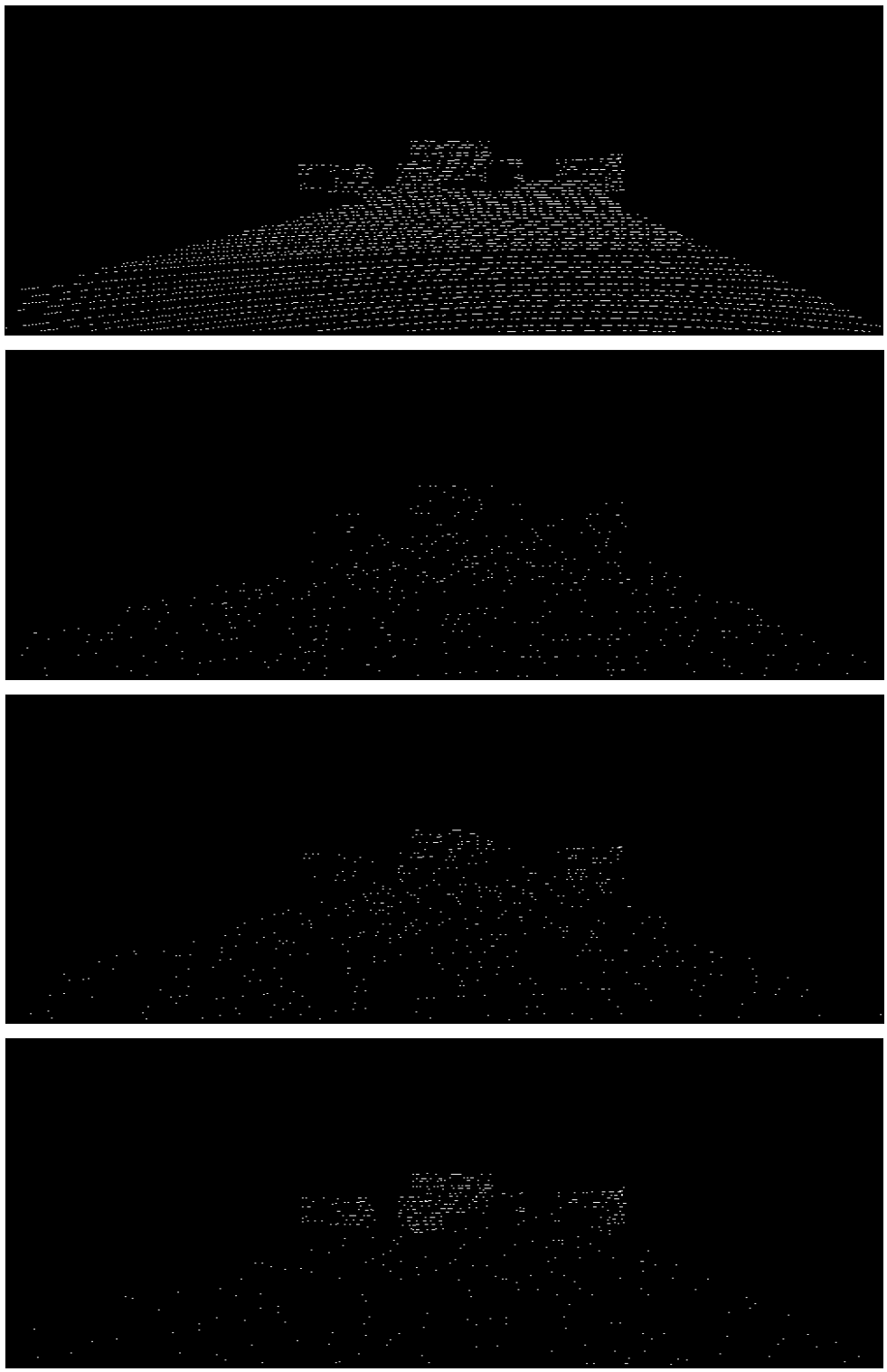
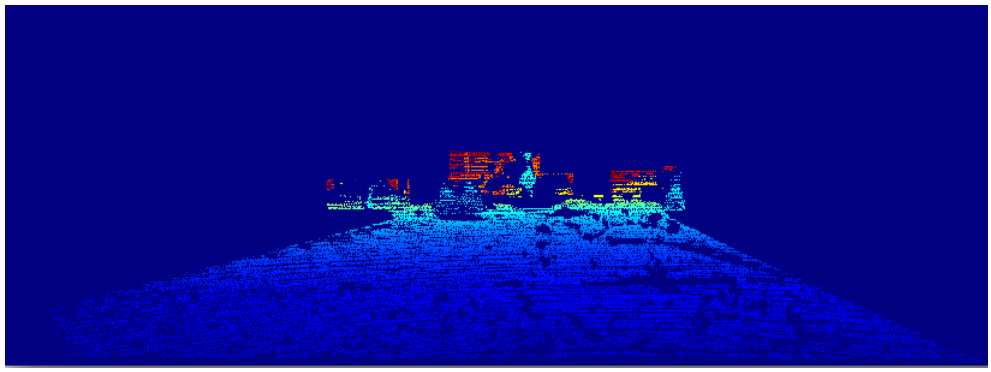
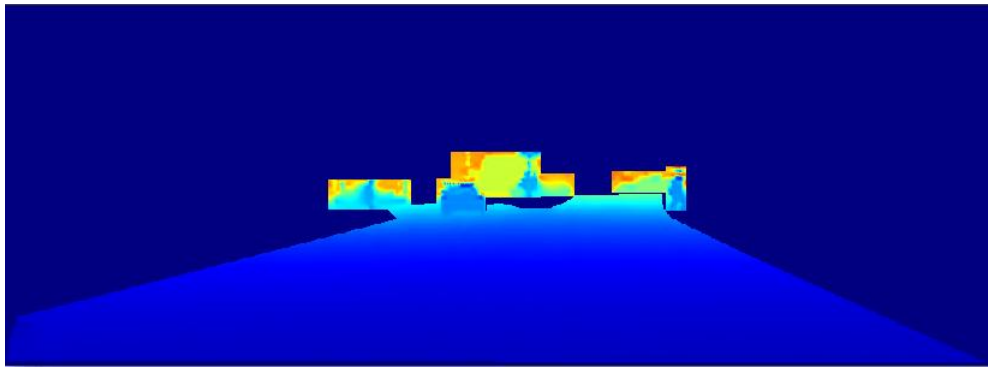
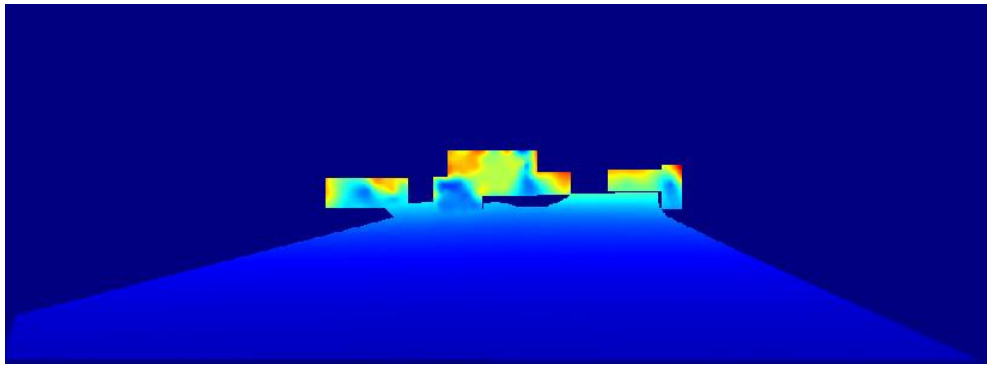
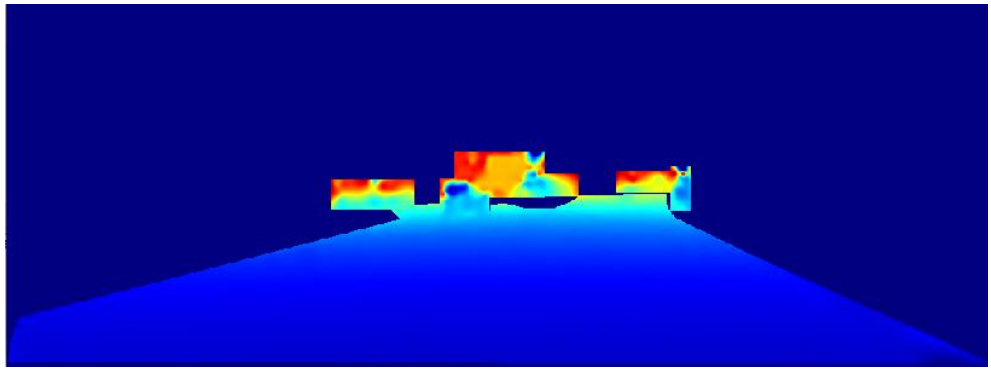


Figure 12. From top to bottom: mask of raw depth and sampling mask of random sampling, two-step sampling and our algorithm









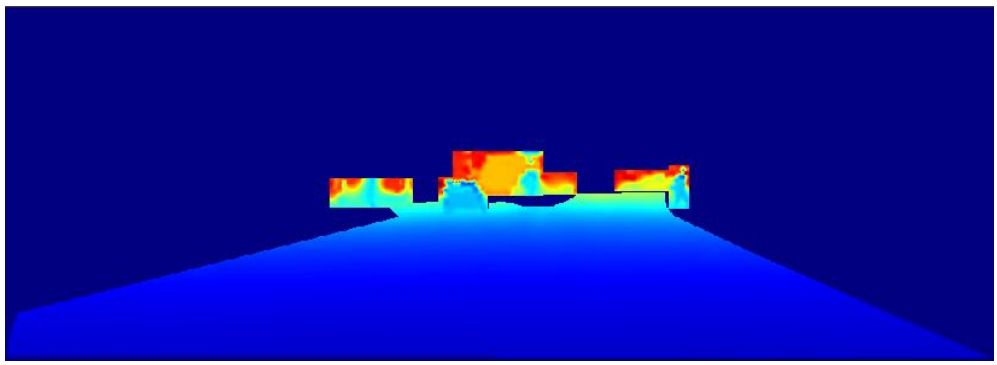


Figure 13. depth ground truth and the reconstruction result from raw depth, random sampling, two-step sampling and our algorithm

The final reconstructed image is shown in figure 15 and crop out of objects area is depicted in figure 16. As being demonstrated, the road area is well reconstructed in all for all three methods. Objectively, we could see no differences between these images. However, when considering the objects area, there are significant differences. In the reconstructed image of raw depth, we can see the shape of the car and pedestrian recognizable. In the reconstructed image of random sampling, we could recognize nothing. The car starts looming in the reconstructed image of two-step sampling and in our reconstructed image, it becomes clearer. In our reconstructed image, it is not as clear as in the reconstructed image of raw depth, however, they are still recognizable. Hence, our algorithm has the closest output compared to upper-bound performance and the random sampling does worst. Above analysis shows the superiority of our method, the differences between these three algorithms are significant in both sampling masks and reconstructed images.

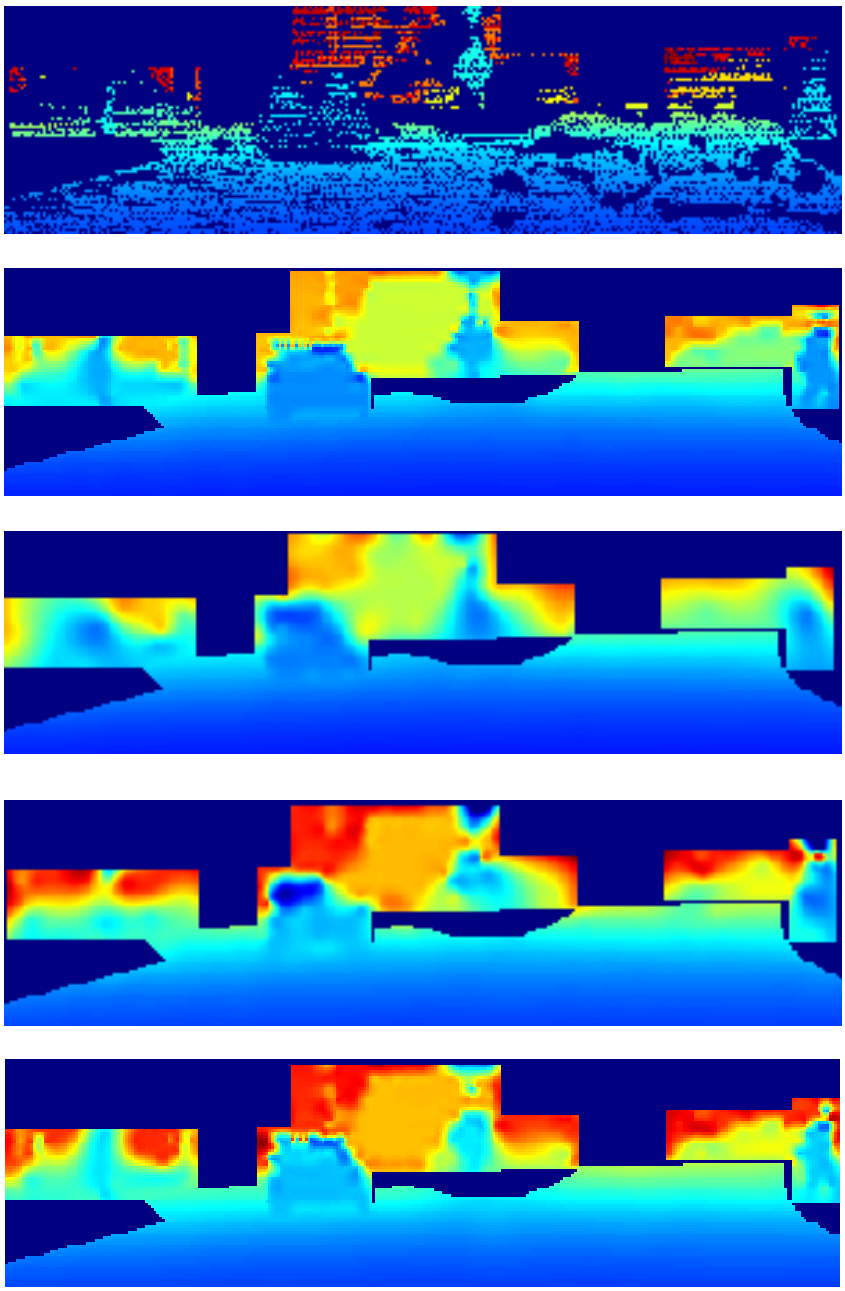


Figure 14. Zoomed out of objects' area in depth ground truth and the reconstruction result from raw depth, random sampling, two-step sampling and our algorithm

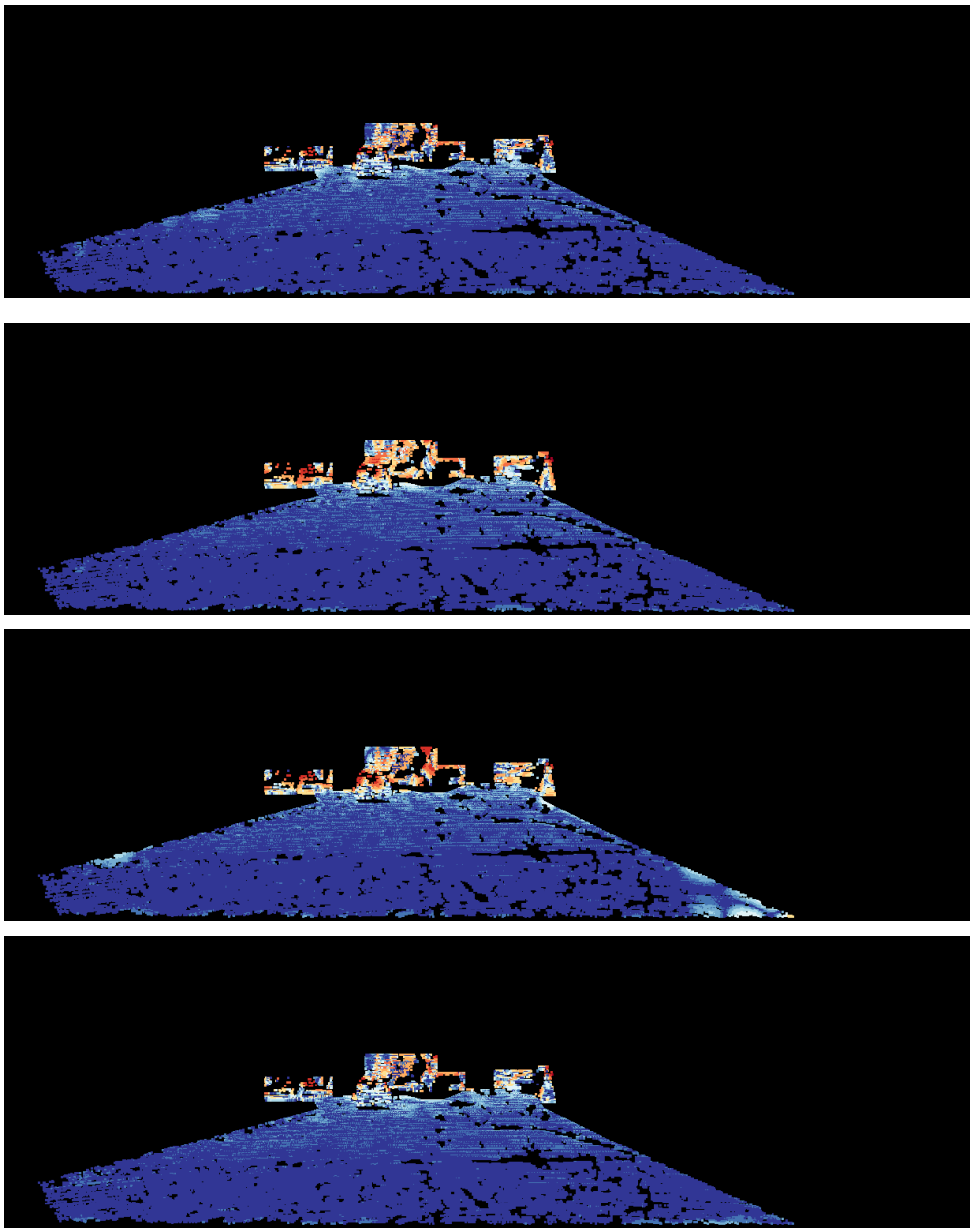


Figure 15. Error images from reconstruction result from raw depth, random sampling, two-step sampling and our algorithm

From error images, we have the same conclusion, we have the same conclusion. Random sampling has more errors than any other, while our

methods are most close to raw depth.

5.3. Quantitative evaluation

Table 4. Mean Absolute Error comparison between methods

N	Method	MAE		
		Road	Object	Overall
Maximum		0.16696	2.1082	0.23508
1000	Random	0.16985	3.4527	0.30366
	Two-step	0.16528	2.9457	0.27984
	Ours	0.18114	2.2286	0.26508

The quantitative result is illustrated in table 4. The same result still holds as above objective evaluations. In road area, Two-step sampling does best, following by random sampling and our algorithm has the largest difference compared to the upper-bound performance. However, this gap is only less than 2cm. This is in range of Velodyne LiDAR accuracy and therefore the sacrifice is acceptable. In road area, the random sampling does worst and our algorithm has a very good result with the difference in reconstructed depth from the upper-bound performance is only about 10cm while two-step sampling yield more than 80cm difference. Objects area is semantically more important than road area, so this is an important improvement.

5.4. Distribution between ROI and non-ROI area

To perform this evaluation, we use KITTI depth reconstruction benchmark as above tests and the KITTI odometry benchmark. The benchmark consists of 22 sequences, 11 sequences for training and 11 sequences for testing. The top performer of this benchmark is currently being the Visual-lidar odometry and mapping (V-LOAM) that uses visual data to make an estimate and then uses LiDAR data to refine the result. The purpose of our test is to evaluate the effect of different sampling algorithm to odometry result so we only use LiDAR data using ICP algorithm. Because the LiDAR data is very sparse, point to plane metrics is used and performs significantly better than the point to point metrics.

To evaluate our algorithm with this dataset, object detection, and road detection is annotated manually. There are only 20 images prepared for the tests. The results of the various methods are shown as the following table:

Table 5. Odometry results of various setups

	translation drift (m)	rotation drift (degree)
raw depth	0.0385	0.004
random	0.14607	0.01421
two-step	0.09205	0.01275
Our-ROIonly	0.71255	0.03817
Our-SameRate	0.10723	0.01455
Our-ROIPriority	0.13511	0.01752

In the table 5, the first result is from raw depth with all available samples. The rest of them is taken when sampling budget is set to 2000. The second row shows the result of random sampling, the next row is from two-step sampling. The last three rows demonstrate results from our algorithm with three different setups. First, the result when sampling is performed inside ROI region only is shown, the second result is archived when we set the same sampling rate inside and outside ROI area. The last result is got when we do sampling inside ROI area denser than non-ROI area, in equation 4, the parameter α is set to 1.2.

As demonstrated in the table, the output of raw depth is very good with the translation drift only about 4 centimeters. Our algorithm when being performed in inside ROI region only performs worst, the error is much higher than any other methods. However, when being performed on entire Field of View, the result of our method is satisfactory with the translation drift only about 10 centimeters, this is approximately the same as the accuracy of the odometry dataset. When increasing the sampling rate inside ROI area in order to achieve higher reconstruction result, the odometry result gets worse. At this setup, our algorithm still gets a better result than random sampling. Two-step sampling performs best with odometry, about 1.5 centimeters better than our algorithm at the best setup.

Considering the reconstruction result of above setups, their result is imprinted in following table:

Table 6. Reconstruction results of various setups

	road	object	ROI	background	overall
raw depth	0.131448	0.94143	0.239427	0.46965	0.339022
random	0.142748	1.97609	0.366456	0.98562	0.636075
Two-step	0.139731	2.08233	0.378107	0.90666	0.619105
Our-ROIonly	0.170837	1.02971	0.287913		
Our-SameRate	0.196628	1.26516	0.340591	0.97286	0.627206
Our-ROIPriority	0.200792	1.20714	0.328167	1.13332	0.675041

Comparing the reconstruction results of ROI area, our algorithm performs best. All three setups obtain better results than both random sampling and two-step sampling. The result when doing sampling inside ROI area only obviously gives the best result, followed by the result when setting the sampling rate inside ROI higher than non-ROI area. When the sampling rate of ROI area is equal to that of non-ROI, which is the same as random and two-step sampling, our algorithm also performs better. Comparing to the reconstruction result in the background area and overall result, two-step, and random sampling are better. However, ROI is our priority so the results are reasonable.

CHAPTER 5 - CONCLUSION

In this thesis, we introduce an idea of using results from object and road detection step in ADAS system to better guide the LiDAR sampling process, specifically sampling on the ROI. The focus is distributing the sampling budget between object and road areas in the ROI. By observation that road area is simpler and easier to construct, our algorithm distributes to this area just enough points to keep a decent result, while the rest of sample budget is reserved to object areas. In comparison, two-step sampling is more general than our approach and when being applied to our application, its gradient-based approach also results in sparser sampling in road area and denser sampling in objects area. However, our algorithm is highly optimized for this typical ADAS application, our method yields better results in both objective and quantitative evaluation. The experiment shows our method has a good compromise, significantly improves reconstruction result in objects area by 85.6% and entire image by 32.9% while just scarifying little in road area.

REFERENCES

[1] "The Economist Special Report," [Online]. Available: https://www.economist.com/news/special-report/21737425-foreseen-and-unforeseen-consequences-self-driving-cars-will-profoundly-change-way. [Accessed 11 05 2018].

[2] U. J. J. S. S. J. Y. P. K. S. D. H. S. K. Lee, "EureCar turbo: A self-driving car that can handle adverse weather conditions," in *The International Conference of Intelligent Robots and Systems (IROS)*, 2016.

[3] K. D. K. F. B. M. M. S. C. W. H. Bengler, "Three decades of driver assistance systems: review and future perspectives," *IEEE Intell. Trans. Syst. Mag*, vol. 6, p. 6–22, 2014.

[4] D. S. G. R. e. a. Redmon J, "You Only Look Once: Unified, Real-Time Object Detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] "KITTI road benchmark," KITTI road benchmark, [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_road.php. [Accessed 12 05 2018].

[6] "Velodyne LiDAR," Velodyne LiDAR, [Online]. Available: http://velodynelidar.com/hdl-64e.html. [Accessed 10 05 2018].

[7] D. V. L. H. K. a. H.-J. L. Xuan Truong Nguyen, "A High-Definition LIDAR System Based on Two-Mirror Deflection Scanners," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 559-568, Jan. 2018.

[8] Y. S. W. C. C. K. U. K. a. S. S. Park Y, "Calibration between color camera and 3D LIDAR instruments with a polygonal planar board," *Sensors (Basel),* 2014.

[9] L. D. B. a. B. B. H. Alismail, "Automatic Calibration of a Range Sensor and Camera System," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, Zurich, 2012.

[10] J. Z. a. S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots,* vol. 41, no. 2, pp. 401-416, 2017.

[11] J. Z. a. S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems Conference*, July, 2014.

[12] M. K. a. K. D. S. Hawe, "Dense disparity maps from sparse disparity measurements," in *IEEE Int. Conf. Computer Vision (ICCV'11)*, Nov. 2011.

[13] S. C. a. T. N. L.-K Liu, "Depth reconstruction from sparse samples: Representation, algorithm, and sampling," vol. 26, no. 6, p. 1983–1996, Jun. 2015.

[14] Z. L. T. N. Lee-Kang Liu, "Sharp disparity reconstruction using sparse disparity measurement and color information," IVMSP Workshop, Sep. 2013.

[15]A. W. a. D. A. C. S. Schwartz, "Saliency-guided compressive sensing approach to efficient laser range measurement," *J. Vis. Commun. Image R.*, vol. 24, no. 2, p. 160–170, 2013.

[16]J. F. a. T. K. a. A. Geiger, "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[17]J. a. H. A. a. W. S. L. Ku, "In Defense of Classical Image Processing: Fast Depth Completion on the CPU," *arXiv preprint arXiv:1802.00036*, 2018.

[18]A. G. a. P. L. a. R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19]A. G. a. P. L. a. C. S. a. R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.