

# STEREO CAMERA AND SOLID STATE LIDAR OBJECT DETECTION ON SMALL DELIVERY ROBOT

Jiarun Wei<sup>1,†</sup>, Ding Zhao<sup>2,†</sup>

<sup>1</sup>Carnegie Mellon University, Pittsburgh, PA

<sup>2</sup>Carnegie Mellon University, Pittsburgh, PA

## ABSTRACT

*Stereo camera is popular in object detection due to its dense data. It works by matching the key points of the left and right images, using the geometric constraints to calculate the depth map, and then feed these into a neural network. Due to the geometric constraints, it only works for a certain range of objects and is unstable for some objects. Here we present a demo to stabilize the detection using solid state lidars, which is a very cost efficient solution especially for small and slow robots.*

**Keywords:** Object detection, Computer Vision, Solid-state Lidar, Autonomous Delivery

## 1. INTRODUCTION

In the industry of autonomous delivery, the velodyne lidar is the most important sensor. It is used to detect objects in the environment with multiple threads. However, due to the demand of high speed of real vehicles, the density of its threads are required to be very high. For small delivery robots, many researchers directly deploys the velodyne lidar, which might not be necessary given the speed and height of those small vehicles. In this paper, we propose a novel method to utilize a very cost efficient solid state lidar combined with stereo camera to stabilize the detection of objects on small delivery robots.

## 2. SYSTEM DESIGN

As is shown in figure 3, the delivery robot can be divided into 6 parts: stereo camera, router, battery, solid-state lidar and the processor. The part list is shown in table 1.

**Stereo camera** The stereo camera is used for the detection of dynamic obstacles such as human and vehicles. Though the depth obtained by the stereo camera is not as accurate as the lidar, its object detection ability is a lot more better due to the RGB triple input channels.

**Solid-state lidar** The lidar is used in multiple tasks: mapping, localization and object detection. During the mapping task, the lidar scans around a certain area and reconstructs a 3D point

ID	Category	Part	Qty
SAFEAI001	Joystick Controllers	Mecanum Wheel Controller	1
SAFEAI002	Joystick Controllers	Regular Wheel Controller	1
SAFEAI003	LiDARs	Solid-state Livox Mid-70 Lidars	3
SAFEAI004	LiDARs	SLAMTec Lidars	2
SAFEAI005	Cameras	ZED Camera	2
SAFEAI006	Power Supplies	Krisdonia Power Bank	1
SAFEAI007	Power Supplies	MAXOAK Power Bank	2
SAFEAI008	Computers	AGX Jetson	2
SAFEAI009	Internet	Router communications	2

TABLE 1: PART LIST

cloud map of the surroundings. In the localization task, the lidar scans forward and locate the robot using a Normal Distribution Transform matching method, which matches the distribution of point cloud of previous map and the current scanning. In the object detection task, the lidar gets the semantics of the current scanning.

**Processor** The processor is used to process the data obtained by the lidar and the stereo camera. The processor uses a neural network that can process the data and generate the semantics of the current scanning.

**Battery** There are 2 types of batteries, one is the chassis battery which is used to power the chassis and move the whole vehicle. Another is the power banks which are used to power the processor,

<sup>†</sup>Joint first authors

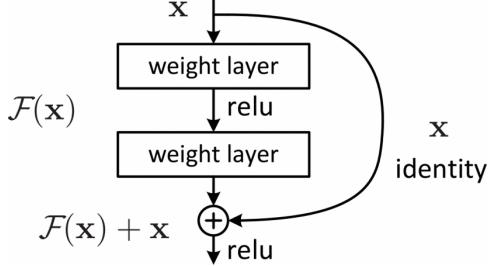


FIGURE 1: RESNET BLOCK

lidars, and the router. The rated power of the chassis and the power banks are different so they have to be separated.

**Router** The router is used to connect the lidar and the processor. It plays an important role in the synchronization process between the lidar and the processor.

### 3. METHODS

Stereo R-CNN based 3D Object Detection [1] is deployed in the system. It can be decomposed into several parts: feature extraction, region proposal, region of intersection alignment, key-point prediction and 3D bounding box estimation, which involves deep neural networks such as ResNet and binocular geometric constraints. In the following experiment, the lidar point clouds are used to verify the detection.

**Feature extraction** The ResNet [2] is one of the most commonly used feature extraction methods. Its architecture is shown in figure 1. Unlike the regular sequential CNNs, the ResNet adds the forward propagation from the input of each blocks directly to their output.

**Region proposal** The Faster R-CNN network[3] is used to propose the regions of interest. It can be divided into 3 steps: anchor generation, foreground-background classification and bounding box regression. The anchors are generated in the feature space of the output of the feature extraction backbone. They are mapped back to the image regions according to the aspect ratios and scales, which are all trainable parameters.

**Region of intersection alignment** Given the ground truth bounding boxes, the candidate boxes are further classified into foreground or background using the overlapping ratio with the ground truth as criterion. The ground truth is a heavy work which requires a lot of human resources.

**Keypoint prediction and 3D bounding box estimation** After the region of intersection is calculated, the network can do a regression to estimate the position and size of the bounding boxes using backpropagation.

### 4. EXPERIMENT

The overall work flow is shown in figure 2. Before the actual experiment begins, the stereo camera is calibrated using

the intrinsic matrix  $K$  and the distortion coefficients  $DC$ , which are defined as:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$DC = [ k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3 ] \quad (2)$$

The corresponding parameters' values are shown in table 2.

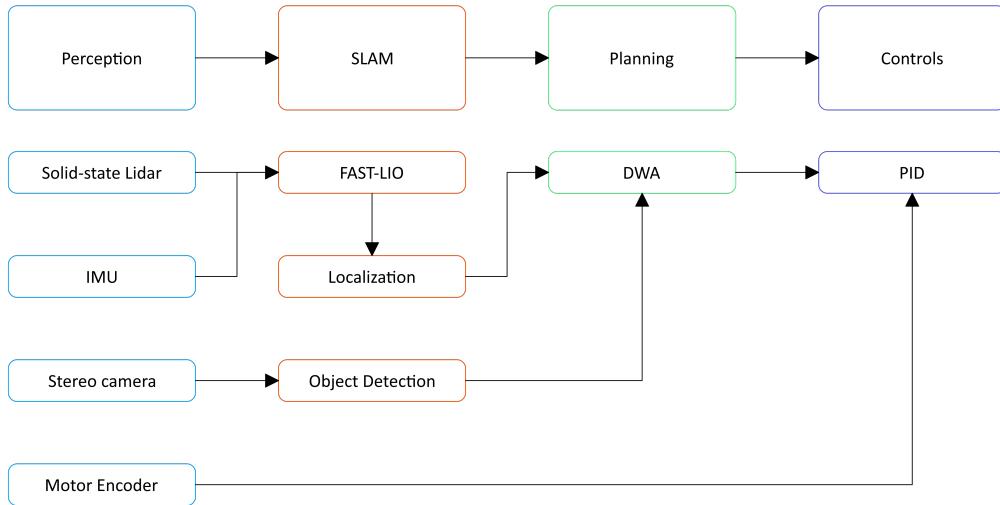
	Left lence	Right lence
$f_x$	525.575	528.400
$f_y$	525.1050	527.8600
$c_x$	355.7350	627.9750
$c_y$	350.3825	370.8590
$k_1$	-0.0408	-0.0453
$k_2$	0.0081	0.0140
$p_1$	0.0000	0.0000
$p_2$	0.0000	0.0000
$k_3$	-0.0041	-0.0041

TABLE 2: CAMERA INTRINSIC PARAMETERS

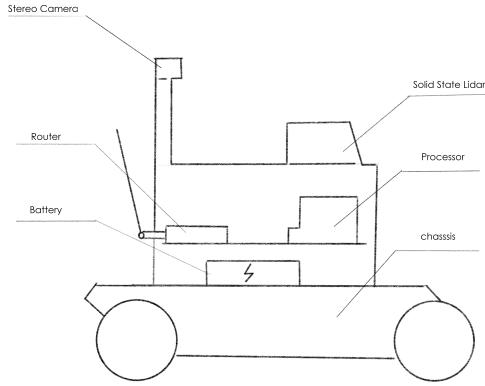
**Perception** Besides the Solid-state lidar and the stereo camera, which is introduced in the previous section, there are 2 more sensors: the IMU and the motor encoder. The IMU is used to measure the dynamic properties of the robot. For example the acceleration and the angular acceleration. The velocity and the estimated position can be obtained by integrating the acceleration. It should be noticed that the position obtained here can not be used directly because there is drifts due to the intrinsic noise of the IMU, hence the point cloud matching localization of the lidar is necessary. The motor encoder is used to measure the rotational speed of the wheels so that the commanded speed can be maintained using a PID control policy.

**Simultaneously Localization and Mapping** There are 2 sensors in charge of the localization process: the IMU and the lidar. The IMU is stable as long as there is no strong magnetic field around. It will never be influenced by the surrounding obstacles. However, it will have large drift upon small errors of the accelerometer. On the other hand, the lidar point cloud matching method will correct itself when more points are detected. However, the lidar is very sensitive with respect to the environment. Given the pros and cons of these 2 sensors. The final localization is done by merging these 2 sensors' data using the Kalmen filter. Figure 5 shows the cost map of the route between CMU's Hunt library and gym.

**Planning** The planning is divided into global planning and local planning. The global planning is done by using the A\* algorithm and the local planning is done by DWA algorithm. The A\* algorithm is a breath first search based method with weights added to each steps of the exploration. The Dynamic Window Approach is a local planning method that searches the robot control space



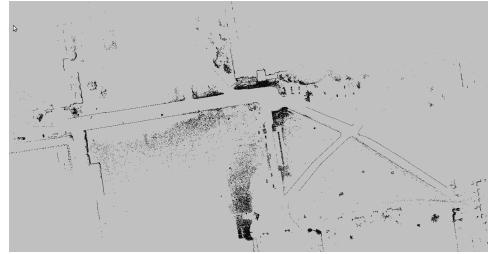
**FIGURE 2: FLOW CHART**



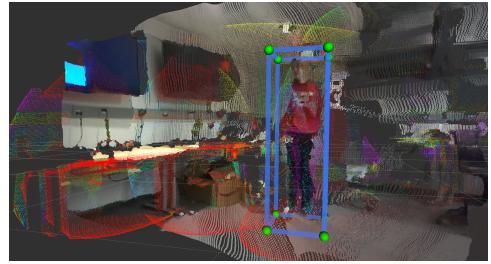
**FIGURE 3: SYSTEM DESIGN**



**FIGURE 4: COST MAP**



**FIGURE 5: COST MAP FOR ANOTHER ROUTE**



**FIGURE 6: OBJECT DETECTION VERIFIED BY LIDAR**

$u$ , where:

$$u = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (3)$$

**Control** The control is done by using the PID control policy. The PID control policy is a feedback control policy that is used to maintain the speed of the robot.

## 5. RESULTS

Figure 6 shows 2 views of the stereo camera object detection result of a person. The results are shown in blue bounding box with green vertices. The strips in the figures are the lidar point cloud. It is shown that the person in the bounding box is closely surrounded by the strips, which shows that the lidar point cloud overlaps with the camera object detection. Figure 4 shows the planning result of the vehicle. The green line is the global planning path, the red line is the local planning path and the blue line is the localization.

According to the figures, the stereo camera's object detection



FIGURE 7: COMPARISON WITH GOOGLE MAP

is relatively accurate even under indoor environment. The lidar's point cloud is also shown to have a very good alignment with the depth map given by the stereo camera.

## ACKNOWLEDGMENTS

Thanks to Professor Ding and all members in the team of Safe AI Scout at Safe AI Lab, the system of our prototype is developed in a very smooth way. We are very grateful to the members of the team not only for their hard work and dedication but also for their suggestions and feedbacks.

## REFERENCES

- [1] Li, Peiliang, Chen, Xiaozhi and Shen, Shaojie. "Stereo R-CNN based 3D Object Detection for Autonomous Driving." *CoRR* Vol. abs/1902.09738. URL [1902.09738](https://arxiv.org/abs/1902.09738), URL <http://arxiv.org/abs/1902.09738>.
- [2] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing and Sun, Jian. "Deep Residual Learning for Image Recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: pp. 770–778. 2016. DOI [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [3] Ren, Shaoqing, He, Kaiming, Girshick, Ross B. and Sun, Jian. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *CoRR* Vol. abs/1506.01497. URL [1506.01497](https://arxiv.org/abs/1506.01497), URL <http://arxiv.org/abs/1506.01497>.

## APPENDIX A. CODES FOR CAMERA LIDAR CALIBRATION

```

1 import pyzed.sl as sl # Import the stereo
   ↵ camera's library
2 import cv2 # Import OpenCV
3 init_params = sl.InitParameters() # Create a
   ↵ sl.InitParameters object
4 init_params.camera_resolution =
   ↵ sl.RESOLUTION.HD2K # Use HD2K resolution
5 zed = sl.Camera() # Create a camera object
6 zed.open(init_params) # Open the camera
7 if not zed.is_opened(): # Check if camera opened
   ↵ successfully
8     exit()
9 print("Opening ZED Camera...") # Remind the user
   ↵ that the camera is opening
10 zed.grab(init_params) # Grab a new image
11 image = sl.Mat() # Create a sl.Mat object
12
13 zed.retrieve_image(image) # Retrieve the left
   ↵ image
14 cv2.imwrite("test.png", image.get_data()) # Save
   ↵ the image

```