

Autoware Perception Performance Question #5041

 Closed

 Answered by [liuXinGangChina](#)

[masierralomby](#) asked this question in Q&A



[masierralomby](#) on Jul 28

I am running Autoware 2024.01 Release in a computer with the following configuration:

32 GB RAM

12th Gen Intel Core i7 x 20

NVIDIA RTX 4000 SFF Ada

I am considering this system for a mobile robot operating in the city and thus some scenarios might vary from the usual ones found in a car.

I already have all the setup for sensors, vehicle interface, etc. I have been operating the robot using only the first nodes for Obstacle Segmentation and Occupancy Grid Map without any problem. This approach worked for several situations but then I reached a point where I needed to detect, classify, track ,predict...so I started using the other parts of Autoware Perception Stack.

I first tried a 3D Lidar only approach but I was having problem with false positive pedestrian detection, when some trees along the road were misclassified as humans. I decided to train the Centerpoint algorithm with trees and other structures that could be misclassified, but the data collection, annotation, etc takes quite some time.

I tried other approaches while I am advancing on the training of the model: Use the Camera-Lidar Fusion that was already included in AW and could detect trees more efficiently that only the 3d Lidar. I was able to use this approach and do some tests with it but now I am reaching a limit in performance.

When using the Camera-Lidar fusion approach, the centerpoint algorithm, TensorX Node and Trafficlight Node were all being computed in the GPU (confirmed with nvidia-smi) and it was only using around 20%.

However, the other nodes were using almost 98-99% of the CPU power. This made that other calculations like localization or even the publish of /perception/object_recognition/objects being delayed and it end up being unstable.

Do you have any advice or know if there is a way to improve the performance using the Camera-Lidar fusion approach? I looked at the current available solutions but they were usually using another machine at the same time (Jetson).

Category



Q&A

Labels

None yet

2 participants





Answered by **liuXinGangChina** on Jul 29

hi **@masierralomby** ,regarding lidar-camera fusion pipeline, there are several ways to optimaize cpu usage

1. fine tune euclidean cluster parameters, like voxel filter size -- less point less calculation

[View full answer](#) ↓

1 comment · 3 replies

Oldest

Newest

Top



liuXinGangChina on Jul 29

Collaborator

hi **@masierralomby** ,regarding lidar-camera fusion pipeline, there are several ways to optimaize cpu usage

1. fine tune euclidean cluster parameters, like voxel filter size -- less point less calculation
2. try to use cuda-based algorithm to accelarate the pre-process and post-process part of lidar-centerpoint

In my opinion . first solution maybe enough for you

happy hacking!



Marked as answer

↑ 2

👍 1

3 replies



masierralomby on Jul 29

Author

edited ▼

Thank you so much for the reply **@liuXinGangChina** ! I will take these 2 points into account!

Effectively the CPU usage is getting better!



liuXinGangChina on Sep 3

Collaborator

hi **@masierralomby** if no further question in terms of "fusion performance" please close this discussion and mark it as answered

thank you

happy hacking



liuXinGangChina on Sep 3

Collaborator

Since issue solved and no further question related to this rasied, i will close this discussion and mark it answered

Answer selected by **liuXinGangChina**