Localization Tests - Yabloc and NDT Comparison #3680

StepTurtle started this conversation in **Show and tell**

StepTurtle on Jul 20, 2023 Collaborator edited

This report includes the comparison results of pose estimation modules of Autoware Ndt and Yabloc.

During the tests, the modules are aimed to operate independently from each other (Link for the branch using in the tests):

- two pose estimator (one of them is Yabloc and other is Ndt)
- two pose twist fusion filter (One of them fed with Yabloc, one of them fed with Ndt)
- two twist estimator
- two pose initializer

Our test car has includes these sensors:

Sensor	Model
Camera	Lucid Vision Triton 5.4 MP 70°
GNSS/INS	SBG Ellipse-D (with RTK)
Lidar	Velodyne Alpha Prime

Tests

On the Rviz screen:

- Yellow trajectory shows pose_twist_fusion_filter (EKF) output from Yabloc and red trajectory shows pose_twist_fusion_filter (EKF) output from Ndt.
- The image with yellow lines shows lanelet lines projected onto the image plane and you can check the Yabloc accuracy.
- The image with red and black lines shows projected line segments.

- Default parameters of Yabloc and Ndt have been used during the tests.
- All of the test data has been collected from a university campus, therefore it mainly consists of single-lane roads. There is no highway driving test included in the tests.

Category

Show and tell

Labels

component:localiza...

2 participants

- During the videos, it can be seen that there are some difference between algorithms' results and GNSS and it is not about algorithms' accuracy, there are some errors between map and GNSS. Because of this reason, we have no ground truth right now. Therefore, I added the difference between Yabloc and Ndt in 6 DoF **(Shows in the table with the XYZ Error & RPY Error titles)**.

- **Images looks too small in the table, please open images on new tab.**

Test ID	Description	İmage	Link	XYZ Error	RPY Error	Outcome
1	During the test, the road mostly shines due to the sun, and the road markings are not clear. GNSS errors increase in some parts		Link			Yabloc errors in pitch axis increased sometimes. Both algorithm does not make errors and produces accurate localization results in areas where GNSS has high errors. It has been observed that Yabloc is not affected much by the sun.
2	Small road with good conditions		Link			It can be seen that Yabloc has error in axis-y and Ndt has error in axis-x
3	This test has ideal testing conditions for Yabloc. GNSS errors are		Link			Both algorithm works accurate during the test.

Test ID	Description	İmage	Link	XYZ Error	RPY Error	Outcome
	significantly low, and the road lines are well-defined. There is a roundabout is route. The only situation that could be considered as a challenge for NDT is the lack of many features around the route (builds, trees, ...).					
4	During the test, some parts of the road have distorted lane markings, and at times, the camera is significantly affected by the sun. There are not many challenges for NDT in this case. Additionally, the route includes slopes (uphill roads).		Link			It is observed that NDT closely follows the route to a high degree but Yabloc has operated with significant errors in some sections. Yabloc initially started making mistakes in the slopes, but it was able to follow the

Test ID	Description	İmage	Link	XYZ Error	RPY Error	Outcome
						correct rout in the later parts. Additionally in previous tests, it was not affected by the sun, but in this test, in a scenario with intense sunlight and a vehicle blocking its view, it lost track of the route. Furthermore despite Yabloc losing track of the route to a significant extent, it was able to regain its path later and follow the route again.

Conclusion:

- As a result of the tests, it has been observed that Yabloc initializes faster than Ndt.
- In places with high GNSS errors, both methods works well and both methods can be alternative solutions.
- After producing wrong result with high errors, Yabloc starts producing good results on its own again. In my experience, this was not the case for NDT, and it required intervention from the user. If I am mistaken, please correct me.
- Yabloc can produce incorrect results when there are vehicles in front of or around it, and similar outcomes can be expected in heavy traffic scenarios. Perhaps transitioning to a multi-camera system could resolve this issue.

I am adding some questions to developers about things I'm curious and don't know:

- We can see error of Yabloc with overlay of lanelet2. Is there a numeric error metric for Yabloc to see the instant error? If there is, could we compare with Ndt somehow.
- In my lanelet2 map there is no bicycle roads but they are seen on image. Is it affect the accuracy a lot?
- Compared to other parts, Yabloc performed poorly on slopes, what could be the reason? (especially tests in this link, there is a lot of errors in slopes, and test 4)
- Is the method used to find lanes (Line Segment Detector and Graph Segmentation) sufficient. If a CNN based lane detection model is used instead of these DNN's, will the results improve? (To detect the lanes as a single piece instead of detecting them in pieces.) (It may provide better results in scenarios where there are vehicles in the camera's field of view.) Is there a plan to change the methods?

↑ 1 ❤️ 1

2 comments

Oldest | Newest | Top

KYabuuchi on Jul 20, 2023 Collaborator

Thank you for your report! It is very insightful 😄 I will answer your questions.

> After producing wrong result with high errors, Yabloc starts producing good results on its own again. In my experience, this was not the case for NDT, and it required intervention from the user.

Indeed, it is easier for YabLoc to recover from inaccurate estimates compared to NDT. The reason lies in YabLoc's utilization of a particle filter, which allows it to maintain multiple self-position candidates. Even if almost particles temporarily converge to incorrect locations, as long as some particles remain in the correct location, the system can recover. Moreover, incorporating GNSS information also contributes to recoverance.

> Yabloc can produce incorrect results when there are vehicles in front of or around it, and similar outcomes can be expected in heavy traffic scenarios. Perhaps transitioning to a multi-camera system could resolve this issue.

Yes, YabLoc's performance may degrade when there are vehicles in front of or surrounding the ego vehicle. There are two main factors contributing to this issue.

1. The availability of road markings decreases, which negatively impacts localization performance. Employing multiple cameras might improve this situation. 👍
2. When selecting road regions based on Graph-Segmentation results, there is a possibility of mistakenly identifying the rear of vehicles as part of the road surface. This particular issue cannot be resolved solely by using multiple cameras, and improvements in graph segmentation or adopting a CNN-based road surface recognition method would be necessary.

> We can see error of Yabloc with overlay of lanelet2. Is there a numeric error metric for Yabloc to see the instant error? If there is, could we compare with Ndt somehow.

There is currently no numerical metric to quantify the misalignment observed in the overlay images. If you have any ideas on how to achieve this, please share them with me. 🙏

> In my lanelet2 map there is no bicycle roads but they are seen on image. Is it affect the accuracy a lot?

Incorporating bicycle lanes in lanelet2 could enhance accuracy. However, the absence of bicycle lanes would not necessarily cause critical errors. If you think the accuracy is insufficient, I recommend adding them to lanelet2.

> Compared to other parts, Yabloc performed poorly on slopes, what could be the reason?

YabLoc projects the road markings extracted from images onto a horizontal plane. This discrepancy between the actual slopes and the horizontal plane projection likely causes the errors observed on slopes. 😢 Please check this code

> Is the method used to find lanes (Line Segment Detector and Graph Segmentation) sufficient. If a CNN based lane detection model is used instead of these DNN's, will the results improve? (To detect the lanes as a single piece instead of detecting them in pieces.) (It may provide better results in scenarios where there are vehicles in the camera's field of view.) Is there a plan to change the methods?

The current methods used to find lanes may not be sufficient. Using a CNN-based lane detection model could potentially lead to improvements. However, I don't think anyone is actively working on incorporating cnn-based methods at this time. 🤔

↑ 1 ❤️ 1 👀 1 0 replies

StepTurtle on Jul 21, 2023 Collaborator Author edited

Thanks for your clear answers to my questions. ⭐⭐⭐

> Indeed, it is easier for YabLoc to recover from inaccurate estimates compared to NDT. The reason lies in YabLoc's utilization of a particle filter, which allows it to maintain multiple self-position candidates. Even if almost particles temporarily converge to incorrect locations, as long as some particles remain in the correct location, the system can recover. Moreover, incorporating GNSS information also contributes to recoverance.

This is really good feature. When I check the Yabloc's and the Ndt's pose estimation outputs (without EKF), Ndt looks very noisy and instead of this, Yabloc looks so smooth

> There is currently no numerical metric to quantify the misalignment observed in the overlay images. If you have any ideas on how to achieve this, please share them with me. pray

No right now 😐 , but if I have, I will definitely share with you.

> Incorporating bicycle lanes in lanelet2 could enhance accuracy. However, the absence of bicycle lanes would not necessarily cause critical errors. If you think the accuracy is insufficient, I recommend adding them to lanelet2.

I think it affect sometimes. When I check the lanelet lines projected image, sometimes The yellow lanes hold on to bicycle roads. So, it would be better if we add

> YabLoc projects the road markings extracted from images onto a horizontal plane. This discrepancy between the actual slopes and the horizontal plane projection likely causes the errors observed on slopes.

It make sense right now.

↑ 1 👍 1 0 replies