autowarefoundation / **autoware** Public

Code · Issues 42 · **Pull requests 13** · Discussions · Actions · Security · ···

Feature/open planner dev #1400

New issue

Merged

hatem-darweesh merged 21 commits into `develop` from `feature/OpenPlanner-dev` on Aug 23, 2018

Conversation 51 · Commits 21 · Checks 0 · Files changed

hatem-darweesh commented on Jul 26, 2018 • edited

Status

** DEVELOPMENT**

Description

Add OpenPlanner new Modifications and Tools to Autoware develop branch.

- OpenPlanner - Global Planner
- OpenPlanner - Local Planner
- OpenPlanner - Utilities
- OpenPlanner - Simulator
- lidar_kf_contour_track
  Remove Old OpenPlanner (dp_planner, way_planner)
  Add readme for each module

**For More Details about the updated OpenPlanner, interface description and how to test **
OpenPlanner Nodes.pdf

Steps to Test or Reproduce

- Should produce same behaviour as previous (way_planner, dp_planner).
- Should show same output as Demo Video
- Basic Functions To Test:

1. OpenPlanner - Global planner should load map successfully and generate global path from start to goal points specified by rviz or from .csv file.
2. OpenPlanner - Local Planner should generate local trajectories along the global path, follow other

Reviewers

kfunaoka

kitsukawa ✓

Assignees

No one assigned

Labels

version:autoware-ai

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

5 participants

simulated vehicles, avoid obstacles, stop for stop lines, and stop in traffic lights if light is red.

3. OpenPlanner - Utilities should be able to play rosbag file including localization, pause, play backwards, and publish TF for ndt_pose
4. OpenPlanner - Simulator should simulate a vehicle moving with specific speed from start to goal position, simulate simple intersection traffic lights, and simulate perception by generating cloud_clusters for simulate vehicles.
5. lidar_kf_contour_track should track detected objects, find heading and velocity for objects detected by lidar_euclidean_cluster or op_perception_simulator.

Follow README files and Demo Video

hatem-darweesh added 12 commits 6 years ago

- `Update OpenPlanner libraries (op_planner, op_utitity, op_ros_helpers)` … 81556d7
- `Update op_utility files for csv files loading` … 537dedd
- `Fix Vector Map parser problem, tested with three different maps` … 429a119
- `Modify Map loading for OpenPlanner, now it reads from Autoware vector…` … 5d3169b
- `Test Simulated Vehicles` … a426f7a
- `Add ReadMe to lidar_kf_contour_track node` 526720e
- `Update README.md` 0d22a39
- `Add README files for OpenPlanner packages` 1c011af
- `Merge branch 'feature/OpenPlanner-dev' of https://github.com/CPFL/Aut…` … e64bced
- `Update README.md` 485e51a
- `Update README.md` ee9e360
- `Merge branch 'develop' into feature/OpenPlanner-dev` 3d9acbc

hatem-darweesh assigned **kitsukawa** on Jul 26, 2018

kfunaoka mentioned this pull request on Aug 1, 2018

[fix] PascalCase messages #1408 Merged

kfunaoka assigned **aohsato** on Aug 1, 2018

hatem-darweesh and others added 3 commits
6 years ago

```
Fix Indigo build issues                         0955298
fix dependencies to make                        112006d
industrial_ci test pass
Merge branch 'develop'                          1dec531
into feature/OpenPlanner-
dev
```

kitsukawa suggested changes on Aug 4, 2018 — View reviewed changes

ros/src/util/packages/runtime_manager/scripts/computing.yaml Outdated

```
609 +        desc : Simulate traffic Sig
610 +        cmd  : roslaunch op_simulat
611 +        param: op_signs_simulator
612 +      - name : op_car_simulator_1
```

kitsukawa on Aug 4, 2018

@hatem-darweesh There are 5 checkboxes for op_car_simulator. This should be only one here.

kfunaoka on Aug 15, 2018

@hatem-darweesh How about @kitsukawa's comment?

hatem-darweesh on Aug 19, 2018 Author

@kitsukawa @kfunaoka Only 1 check box for simulated car remaining, I will add multi car simulation option in "Simulation" tab later.

kfunaoka reviewed on Aug 15, 2018 — View reviewed changes

kfunaoka left a comment • edited

@hatem-darweesh I've added some comments, but it is difficult to review whole because the diff is very wide range. Would you list reasons why these changes are necessary for each items with comparing with the old implementation?

- OpenPlanner - Global planner
- OpenPlanner - Local Planner
- OpenPlanner - Utilities
- OpenPlanner - Simulator
- lidar_kf_contour_track

.../computing/perception/detection/lidar_tracker/packages/lidar_kf_contour_track/CMakeLists.txt

```
11   12       )
12   13
13          -
14          - set(CMAKE_CXX_FLAGS "-O2 -Wall ${CM
```

kfunaoka on Aug 15, 2018

Is this deletion intended?

hatem-darweesh on Aug 15, 2018 Author

I saw a PR which adds common compiler flags to all cmake files "autoware_build_flags". so I deleted the special ones

kfunaoka on Aug 15, 2018 • edited ▾

Thanks! I should accomplish autoware_build_flags because it sets only -std=c++11 now.

hatem-darweesh on Aug 19, 2018 Author

The other flags are needed only locally for debugging and performance test so I don't want to push them to the server

...erception/detection/lidar_tracker/packages/lidar_kf_contour_track/include/PolygonGenerator.h

```
...  ...    @@ -0,0 +1,95 @@
       1  +
```

kfunaoka on Aug 15, 2018

Is it possible to distribute this file under BSD 3-Clause "New" or "Revised" License?
If yes, please insert the license terms.

hatem-darweesh on Aug 15, 2018 Author

can you give me an example to follow.

kfunaoka on Aug 15, 2018

This is the license.
https://github.com/CPFL/Autoware/blob/master/LICENSE

An example is here. The year 2015 in it should be replaced with 2018 this year.
https://github.com/CPFL/Autoware/blob/develop/ros/src/computing/perception/detection/vision_detector/packages/vision_ssd_detect/src/vision_ssd_detect.cpp

...erception/detection/lidar_tracker/packages/lidar_kf_contour_track/include/PolygonGenerator.h

```
+         PlannerHNS::WayPoint max_fr
+         bool bFirst;
+
+         QuarterView(const int& min_a
```

kfunaoka on Aug 15, 2018

It is better to use int or const int instead of const int&.

hatem-darweesh on Aug 15, 2018 Author

it is less confusing to use one parameter format for both objects and standard types.
and it will be really dangerous and will take plenty of time and testing to change it through the whole OpenPlanner library.

kfunaoka on Aug 15, 2018

Thanks. I think so too in the point of view where it takes much time and dangerous. I think it is better to be as is in this pull request.

But using const int & instead of int simply incurs piled-up overhead. I think C++ programmers do not confuse about it because they should know C and C++ basically employs pass by value.

...erception/detection/lidar_tracker/packages/lidar_kf_contour_track/include/PolygonGenerator.h

```
+
+         bool UpdateQuarterView(cons
+         {
+                 if(v.pos.a <= min_a
```

kfunaoka on Aug 15, 2018

It seems that clang-format was not used. Did you use clang-format? If no, please use clang-format to all the files you modified. (https://github.com/CPFL/Autoware/wiki/Contribution-Rules#ros-c-coding)

hatem-darweesh on Aug 15, 2018 Author

I will apply it later with different PR, after merging this one.

kfunaoka on Aug 15, 2018

@hatem-darweesh Your choice is the best method.

```
...g/perception/detection/lidar_tracker/packages/lidar_kf_contour_track/include/SimpleTracker.h
```

```
+ namespace ContourTrackerNS
+ {
+
+ #define DEBUG_TRACKER 0
```

kfunaoka on Aug 15, 2018

It's better to use constexpr instead of #define. (https://en.cppreference.com/w/cpp/language/constexpr)

hatem-darweesh on Aug 15, 2018 Author

Can we add this as ToDo, I need to check for each #define whether it is a compiler directive or a constant. so it will take long time.

kfunaoka on Aug 15, 2018

@hatem-darweesh NP :)

2 hidden conversations
Load more...

```
...computing/planning/common/lib/openplanner/op_planner/include/op_planner/BehaviorPrediction.h
```

```
+                               f.pTrack->U
+
+                               bool bFound
+                               for(unsigned
```

kfunaoka on Aug 15, 2018

It's better to use standard functions. It seems that this loop can be replaced by std::find.

hatem-darweesh on Aug 15, 2018 Author

personal preference

kfunaoka on Aug 15, 2018

@hatem-darweesh NP. But std::find can reduce the number of lines 1/9 (one ninth) where we develop, review, and maintain. If there are ten loops finding an element, it can reduce 81 (= 90 - 9) lines in the Autoware code base. If you like it, please use it :)

- The number of lines of this loop is 9 (from Line 688 to 695).
- The number of lines of std::find is 1.

...ar_tracker/packages/lidar_kf_contour_track/nodes/lidar_kf_contour_track/PolygonGenerator.cpp

```
+
+ std::vector<PlannerHNS::GPSPoint> Po
+ {
+         for(unsigned int i=0; i < m_
```

kfunaoka on Aug 15, 2018

You can use range-for. https://en.cppreference.com/w/cpp/language/range-for

hatem-darweesh on Aug 15, 2018 Author

personal preference

kfunaoka on Aug 15, 2018

Same as #1400 (comment)

...rc/computing/planning/common/lib/openplanner/op_planner/include/op_planner/PlanningHelpers.h Outdated

```

  #ifndef PLANNINGHELPERS_H_
  #define PLANNINGHELPERS_H_

– #include <math.h>
+ //#include <math.h>
```

kfunaoka on Aug 15, 2018

Should be deleted. Unnecessary comments will be junk.

ros/src/computing/planning/common/lib/openplanner/op_planner/src/BehaviorPrediction.cpp

```
+           }
+ }
+
+ int BehaviorPrediction::FromIndicat
```

kfunaoka on Aug 15, 2018

Why are FromIndicatorToNumber and FromNumbertoIndicator necessary? Is it impossible to use enum value itself?

hatem-darweesh on Aug 15, 2018 • edited ▾ Author

the compiler give me an error casting from int to the enumerator, so I create a function to convert from one to another and the vice versa , just to be consistent

kfunaoka on Aug 15, 2018

@hatem-darweesh Is it just a design problem or an unavoidable problem? Basically, it seems that it is not a good idea to have the same type of data (Particle::indicator and the data returned from BehaviorPrediction::FromNumbertoIndicator) with different types (int and lannerHNS::LIGHT_INDICATOR, respectively).

kfunaoka on Aug 16, 2018

I see. Particle::indicator is defined as int in autoware_msgs::DetectedObject. It is difficult to avoid.

ros/src/computing/planning/common/lib/openplanner/op_planner/src/BehaviorPrediction.cpp

```
+ double BehaviorPrediction::CalcIndi
+ {
+         if((obj_ind == PlannerHNS::I
+                 return 0.99;
```

kfunaoka on Aug 15, 2018

Please remove magic numbers as much as possible.

hatem-darweesh on Aug 15, 2018 Author

This part (Prediction) is under development (research work) , all numbers will be removed or added as parameters soon.

👍 1

kfunaoka commented on Aug 15, 2018

@hatem-darweesh My comments are general principles. Would you try to apply the principles to the other places?

kfunaoka reviewed on Aug 15, 2018

View reviewed changes

ros/src/computing/planning/mission/packages/op_global_planner/package.xml Outdated

```
+   <!--   <run_depend>message_runtime
+   <!-- Use test_depend for packages
+   <!--   <test_depend>gtest</test_de
+   <buildtool_depend>catkin</buildtoo
```

kfunaoka on Aug 15, 2018

Please add buildtool_depend of autoware_build_flags. https://github.com/CPFL/Autoware/wiki/Quality-Control

ros/src/computing/planning/mission/packages/op_global_planner/package.xml Outdated

```
+   <version>0.0.1</version>
+   <description>The OpenPlanner Globa
+
+   <!-- One maintainer tag required,
```

kfunaoka on Aug 15, 2018

Please delete unnecessary comments.

ros/src/computing/planning/mission/packages/op_global_planner/CMakeLists.txt Outdated

```
+ project(op_global_planner)
+
+ find_package(catkin REQUIRED COMPONE
+   geometry_msgs
```

kfunaoka on Aug 15, 2018

Please add autoware_build_flags.
https://github.com/CPFL/Autoware/wiki/Quality-Control

kfunaoka closed this on Aug 15, 2018

kfunaoka reopened this on Aug 15, 2018

kfunaoka commented on Aug 15, 2018

Sorry for mistake.

It's better to use clang format after all codes are reviewed because it make the history unreadable.

hatem-darweesh added 3 commits 6 years ago

```
Merge branch 'develop' into feature/OpenPlanner-dev    e42c128
Update BehaviorPrediction.h …                          4476edb
Update PlanningHelpers.h …                             2b5b952
```

kfunaoka previously requested changes on Aug 16, 2018

View reviewed changes

kfunaoka left a comment • edited

@hatem-darweesh Though I've made a lot of comments, my mandatory requests now are:

- Inserting license information into *.{h,cpp} files as mentioned at Feature/open planner dev #1400 (comment).
- Inserting autoware_build_flags into package.xml and CMakeLists.txt as shown at Feature/open planner dev #1400 (review)
- Updating the headline of this pull request to show what are different from before and why they are necessary. Your slide and descriptions will help us to understand your work.

kfunaoka commented on Aug 16, 2018

@hatem-darweesh Your work is great. To approve this pull request, please be reviewed by aohsato from other aspects. I think I am not the best person who reviews this pull request.

In your future work, it's better to make pull requests as accordingly-small as possible. Stepwise pull requests will help reviewers. Enough review will improve your work more and more. I think small feedback loop is better than large feedback loop for you and for reviews.

kfunaoka commented on Aug 16, 2018

I've added the following item into my review comment (#1400 (review)).

- Inserting autoware_build_flags into package.xml and CMakeLists.txt as shown at Feature/open planner dev #1400 (review)

hatem-darweesh added 3 commits 6 years ago

`Support old behavior of insert static object for obstacle avoidance t…` … 01f2ba7

`Merge branch 'develop' into feature/OpenPlanner-dev` 5d1f7ac

`Merge develop into OpenPlanner-dev` 0261672

✓ **kitsukawa** approved these changes on Aug 21, 2018 View reviewed changes

hatem-darweesh commented on Aug 21, 2018 Author

@kfunaoka can you check your review comments and current changes.

kfunaoka reviewed on Aug 22, 2018 View reviewed changes

kfunaoka left a comment

@hatem-darweesh I've confirmed that all my mandatory request are OK. I might find an item that you want to change.

```
.../computing/perception/detection/lidar_tra
cker/packages/lidar_kf_contour_track/CMakeLi
```

sts.txt

```
11  12       )
12  13
13       -
14       - set(CMAKE_CXX_FLAGS "-O2 -Wall ${CM
    14   + find_package(OpenCV REQUIRED)
15  15
```

kfunaoka on Aug 22, 2018

```
- set(CMAKE_CXX_FLAGS "-O2 -Wall ${C
```

Should we revert this line because you said

> The other flags are needed only locally for debugging and performance test so I don't want to push them to the server

hatem-darweesh on Aug 23, 2018 Author

@kfunaoka I will add to different PR if necessary

✕ **kfunaoka** dismissed their stale review 6 years ago

All my mandatory request are OK

hatem-darweesh merged commit **875625a** into `develop` on Aug 23, 2018

hatem-darweesh deleted the `feature/OpenPlanner-dev` branch 6 years ago

kfunaoka mentioned this pull request on Sep 6, 2018

Feature/beyond pixel tracker #1473 Merged

2 tasks

hatem-darweesh mentioned this pull request on Dec 4, 2018

[Discussion] Planning Architecture Closed

autowarefoundation/autoware_ai#419

gbiggs unassigned **aohsato** and **kitsukawa** on Mar 24, 2019

anubhavashok pushed a commit to NuronLabs/autoware.ai that referenced this pull request on Sep 7, 2021

```
Merge pull request autowarefoundation#1400 from          f64d0c6
CPFL/feature/OpenPlan…  …
```

mitsudome-r added the version:autoware-ai label on Jun 14, 2022