Project Notes

Robot:

Teleop-twist-joy:

A teleop node to control the motion of the car via Xbox controller

Structure:
It takes both Xbox controller input and the command from the autoware
Subscribe to /joy from joy node, /ndt_monitor/ndt_status and /twist_cmd from autoware
Publish to /cmd_vel for the motor control node in the Arduino

Functions:
Convert the messages from the joystick to twist message format
Depends on the input from joy (the enable button) to change the mode (auto or manual mode)
Depends on the mode, send the corresponding twist message to Arduino for motor control

Features:
Auto mode is allowed only when the ndt_status is ok

- auto mode will stop immediately when the car is not localized
- it will automatically resume the auto mode once the car is localized

When the car is not localized, it will stop

- it acts as a reminder for the driver to know the car is not localized
- it attempts to wait to see if the car can localize again itself
- the driver can overwrite this by resetting the joystick to zero, then the driver can control the car again

Example:

> When the car is localized and driving by the driver in manual mode, if the car suddenly not localized, it will stop to alert the pilot that it is not localized. The pilot can choose to wait for it to localized again (since usually stopping immediately can help the car to localize again) or choose to control it again. To control the car again, the pilot needs to reset the joystick to zero position to acknowledge that he/she already knows the car is not localized. Then moving the joystick again allows the pilot to control the car again

Remarks:
This node relies on the topic /clock
If a bag is playing in autoware, set the rosparam of use_sim_time to false by typing:

```
rosparam set use_sim_time false
```

Check the readme in Github for information of parameters
The parameters can be changed easily in the header file

Fake GPS:

The fake_gps_node is to help the robot to do localization
Due to the fact that the robot has to initialize every time in the initial position if it lost in the middle, it is quite troublesome to go back to the initial position every time.
The fake_gps_node takes the last localized position as the GPS input for the ndt_matching node

Structure:
Require the ndt_matching and ndt_matching_monitor node
Subscribe to /ndt_monitor/ndt_status for checking if the robot is localized
Require vel_pose_connect node to get the current position
Publish to the topic /gnss_pose for the ndt_matchin
Publish to the topic /tf for visualization

Remarks:
The node must be off before the robot is initialized at the start
The sequence is as follows:

1. Turn on TF, map, and voxel_grid_filter
2. Turn on ndt_matching, ndt_matching_monitor, and vel_pose_connect
3. Wait for the robot to be localized properly at the initial position
4. Run rosrun fake_gps fake_gps_node after the robot is localized properly

If the robot is not localized or localized in a wrong position, the fake_gps output will be wrong
This makes the ndt_matching keep rematching in the wrong position
To fix this, turn off fake_gps node first, then restart ndt_matching and ndt_matching_monitor

BRANCH: click-to-localize
Can implement one more stuff to enable localising with clicked points
Subscribe to the topic /clicked_points and publish it to gnss_pose
This hope to make the robot to be able to localize at any point on the map

Arduino Node:

Take twist message as input and send a command to the motor driver

Structure:
Subscribe to /cmd_vel from teleop_twist_joy and send a command to the motor driver via the driver library - as saved in the google drive

Some libraries are required to compile this node
Including the ROS library for Arduino and the Sabertooth library for the motor driver board
This link is the place stored the library

- Download the library with in zip files
- In the Arduino IDE, there is an option to input library by zip file
- Use that function to input the libraries, compile and upload to the arduino board

Run the node by the command rosrun serial node and specify the port connected
The name of the port can be checked in the Arduino IDE
It is the same name when the code is flashed to the board

There is a watchdog in the node, if after a certain period of time no message is received from the robot
Then this node will stop the robot

Points to note from some customized nodes:

IMU

- Change the output topic to /imu_raw

CAMERA

- Originally it will publish to a topic called /camera_info
- Where the calibration publisher publishes camera distortion information to /camer_info also
- Changed the topic from camera node to another name, like /camera_info

ImageViewerPlugin in Rviz - check below in the detection section

Autoware

Localization

Mapping

Using the ndt_mapping node under computing tab in autoware
Turn on Rviz and open the ndt_mapping configuration

- File -> open config -> go to autoware/ros/src/.config folder and find ndt_mapping

Enable the ndt_mapping node, Rviz will display the map

- Beware that the place you start the node will be the origin position of the map
- This largely affects how your localization later

Slowly move around the space to be mapped, keep track on how the map looks like using Rviz

- Be careful when turning around corners, do turn slowly
- The runtime manager terminal will show how many points are recorded and processing
- If it stops or does not show up means there are some problems with the ndt_mapping node

When finish, while the map looks good, unplug the power of the lidar to stop the node from still accumulating points, then press OUTPUT PCD to output the points to a pcd file

- Output in original gives the most accurate map
- I do suggest outputting the original map as you can filter the map later using software like CloudCompare
- The default filter size is 0.2, where the larger the size, the more simple the map is

CloudCompare

- Can use CloudCompare to edit the pcd file in customizable ways
- Filter the points by specifying the parameters, this clean the noise of the map but this reduced the amount of information for autoware to process
- Remove the ceiling and floor of the map using the cross section function
- You can cut and crop the map by that cross section tool, then output the new layer back to pcd format
- But I am not sure if the floor and ceiling is actually confusing or helping the ndt_matching

Recording bags

Using rosbag to record the topics for later simulation and testing
Click the rosbag button next to the Rviz button

- A window will popup, click the refresh button to get all the current running topics
- Select the raw topic, which are the topics that output sensors data
- No need to record all the topics
- /image_raw, /points_raw, /imu_raw are enough

Reminder:

- Do make sure you have enough space in the computer before recording bags, usually, it takes around 2 to 3 gigabytes, if the disk is full, there will be alert in the runtime manager terminal
- Do start recording at the initial position and wait for a few seconds before you move, otherwise, you will not be able to localize the robot when playback the bag

When recording, there should be a file with the name.bag.active - this means the bag is recording
When finish recording a bag, just press stop, it will save the file in the format of bag
The runtime manager terminal will show the data are saved to the bag

Playback a bag
A bag may be corrupted or unable to play
When you open the bag file in the simulation tab of the runtime manager, there should be some information about the bag, including the duration and the topics it subscribed
If the bags cannot be played in autoware, it might be autoware problem
Try running: *rosrun rqt_bag rqt_bag* and use this default tool to play the bag
If it still cannot be played, the bags should be corrupted

Localization in simulator

The sequence of localization is important
Some nodes must be started before the other

Playing the bag is to enable the simulation clock
TF is to set up the display and to transfer the data of the Velodyne and the map

Simulation

- Press Play then press Pause
- (this step is to enable the simulation time, which is used by TF)

Setup

- Press TF
- Choose a correct model then press Vehicle Model (not necessary to turn on Vehicle Model)

Map

- Select the point cloud then press Point Cloud
- Select the vector map then press Vector Map (if have)
- Press TF

At this point, the RQT should be two separate tree World -> map -> mobility

Sensing

- Select voxedl_grid_filter node with topic : /points_raw

Computing

- Select nmea2tfpose (if have GPS, i.e. running the demo bag)
- At this point, the TF Tree should have map pointing to both mobility and GPS

Computing

- enable ndt_matching and ndt_matching_monitor (choose init if start in init position, else choose GNSS if have GPS)
- Turn on RViz and in the RQT tree, the map should be pointing to base_link

Localization in real-time

Pretty much the same as localization using the simulator
Differences are no need to manually play the bag and must start in the initial position

Localization with wheelchair (setup)

- Turn on Velodyne to base_link (Setup tab, TF)
- Turn on the point cloud (import map from desktop)
- Turn on TF in map tab, with path "~/.autoware/data/tf/tf.launch"
- Turn on sensing tab, voxel_grid_filter

In this phase, the tf tree should be two separate trees with:
If the RQT button does not work, try the following command in the terminal
`rosrun rqt_tf_tree rqt_tf_tree`

World → Map → Mobility
Base_link → Velodyne

Localization with wheelchair (localizing)

- In computing tab, app of ndt_mapping, check Initial Pos instead of GNSS
- Check the get height box, leave others as default
- Leave ndt_matching_monitor settings as default
- Turn on ndt_matching and ndt_matching_monitor under lidar_localizer

In the Rviz, it should have all five global frames, choose world
In the view, choose ThirdPersonFollower (Rviz) and target the frame to Velodyne
Adjust the distance and the angle

Uncheck and check the points map and points raw to refresh, they should be mapping and localizing
Try to move the wheelchair to match and wait for the heading to change from:
"NDT MONITOR - WARNING - NO GNSS AVAILABLE" - orange, to:
"NDT MONITOR - OK - NO GNSS AVAILABLE" - green

If the title turns red, the ndt_matching stops, try to restart by going back to Runtime Manager
Uncheck and check both the ndt_matching and ndt_matching monitor two nodes

Also, if you turn on the fake GPS node, there should show GNSS AVAILABLE
Turning the GPS node after the robot is localized is better
As the last GNSS pose may confuse the ndt_matching node - more information in the fake GPS node section

Detection

Calibration

Link for reference:
Chinese Instructions
Autoware Wiki

Record a bag with only image and Velodyne <- follow the instructions in the above two links

The exact position of the cardboard is not that important, but the more is the better

- Do try to move the board up and down so that it appears any place on the frame
- Both close and far away from the camera is important
- No need to be perfect for every position

However, there is no need to tilt the board that much

- The software may not be able to detect if it tilts too much
- An open space is better for moving the board around
- Also, less obstacle makes selecting the plane later easier

Remember to input the correct values in the calibration toolkit
The unit for the size of the squares are in metres
The ratio is counting the number of intersections (not the number of squares)
It should be the number of squares minus one

The hardest part is to locate the plane
The red dots must be aligned with the grid and located in the middle of the board, if it is not, select the plane again until the position looks good

One of the methods to find the plane
Only arrows can change the viewpoints, use arrows to move in and out to change the viewpoint
, and . is only used to zoom in and out of the viewpoint but doesn't change the viewpoint
And 'q', 'w', 'e', 'a', 's', 'd' is only tilting the viewpoint
Use arrows to find a place not blocked first, then use others to zoom and tilt to plot

Calibration toolkit:
Use the scroll on the mouse to adjust the distance changing for each button press
Scrolling down means a button press displace a little bit
Scrolling up means a button press will displace larger

Calibration - save as a yaml file (need to type .yaml yourself)
No need to save all the data - the results should be less than 1MB

Camera Node (read this if /points2image node died due to segmentation fault)

Problem: when using a real-time camera, the /points2image node died due to segmentation fault

- /calibration_publisher take the yaml file then output distortion array to /camera_info
- However, the real-time /uvc_camera node also publish to topic /camera_info with 0 as data
- They publish to the same topic causes the points2image node to confuse and crash
- Changed uvc_camera to publish to topic /camera_info_raw, problem solved
- Reference File - replace the camera.cpp file in the src folder

Installing Cuda and Yolov3

Download yolov3 weights - GDrive or HERE
Put into the darknet/data folder
As long as the Cuda version is correct, it will work (up till Cuda 9.2 for autoware v1.10)

For Cuda 9, can check this webpage
You need to load the Nvidia driver and download the correct Cuda version

Remember to specify the version when doing the installation

```
sudo apt-get install cuda=9.0.176-1
```

Otherwise, the default version will be the newest version

You can check if the driver is installed correctly through the command *nvidia-smi*
This will show the information of the Nvidia driver and the Cuda version
If the Cuda version is not the one you installed, it showed the latest version like 10.1
Use the following command to check the version:

```
cat /usr/local/cuda/version.txt   OR        nvcc --version
```

If the version is the one you installed, then it should be ok
The Cuda folder linked to in the path /usr/local is the one the computer using like there might be both cuda-9 and cuda-10-1 two folder in the path /usr/local. Then there is a folder link created with the folder named as cuda, if this folder link to the cuda-9 folder then it is using version 9

Overlaying lidar points on the camera frame

Turn off the camera first, i.e. kill the /uvc_camera node
Turn on calibration publisher in the sensing tab with the selected .yaml file
Turn on the points2image node for outputting the points on the image
Turn on the camera node again

The camera must be off when the calibration publisher is starting

If the ImageViewerPlugin does not work, try the camera panel provided by Rviz
There should be points on the image also

You might need to increase the size of the points raw in the display panel in order to view the points on the image in the camera panel

ImageViewerPlugin Problem

(read this if Rviz died due to segmentation fault when subscribed to points2image)

The problem is, the resolution of the camera changed accidentally

Causing the display being wrong

- What I encounter is when I do the calibration, the video is in resolution 320 * 240
- However, the resolution changed back to 640 * 480
- The plugin cannot deal with a resolution change
- Where it cannot display it or remap it to the full-frame

Solution

- Force the calculation to be done in the resolution of 320 * 240
- Then rescale the result buy the ration between the new resolution and the old resolution
- Scale the output by that ratio then is ok

If calibration is done again, do need to change the resolution of calculation

Also, some points may be too far from the centre are recorded, causing the calculation loop too long

This causes a segmentation fault

- Limit the furthest point to be a certain distance
- If it is too far away, skip that point and do not consider it

Edit the changes in the draw_points.cpp in the source files of ImageViewerPlugin

Reference File

Fusion and reprojection

Reprojection:

- It is used to project the image from the camera to the point cloud
- Overlay the camera image to the point cloud

/pixel_cloud_fusion

- Need calibration publisher and image rectifier
- Display /points_fused point cloud in Rviz, set the size to 0.08 - 0.1

Range fusion

- Fused the detection from lidar and camera
- Output bounding box and points cluster

/range_vision_fusion

- Need lidar_cluster and yolo
- Choose the subscribe topics in the app correctly
- Lidar: detection/lidar_detector/objects
- Image: detection/image_detector/objects
- Display the fusion_tools/objects market_array

Reference Video for how to setup reprojection and range fusion

Path planning

Waypoints Saver and Loader

Waypoints saver is used to saving the path the robot passed as waypoints
Remember this node requires the vel_pose_connect node to get the position of the robot
Use the node waypoints_saver in the computing tab

- Open app to select the path to save to with the file type as CSV
- The smaller the interval, the closer the waypoints are
- If you require the robots to turn around corners and walk through narrow paths
- Then a smaller interval is better, like 0.2 to 0.5
- Recording the velocity or not does not affect the speed of the robot when following the pathway

Starting the node will start recording the path

- There should already be a CSV file in the path of the CSV file
- As the node continues to record the points, there will be more points saved to the CSV file
- If there is no CSV file after you start the node, it means the node cannot start properly and the node is not recording anything

Waypoint Loader
Input the waypoints CSV file using the app

Auto Mode

Nodes require for auto mode (7)

- Vel_pose_connect
- Lane_rule and lane_select
- Obstacle_avoid and velocity_set
- Use waypoint_loader to load the pre-generated waypoints
- Pure_pursuit and twist_filter

There should be a path displayed on Rviz, which are the waypoints
A green ball should be on the plane, to be the simulated obstacle
There should be a white arc on the car, which is the twist trajectory mark

The twist_filter publish to the topic /twist_cmd which is subscribed by the teleop node

The car can automatically adjust its path and go back to waypoints, even moving backwards
The car can turn by following the waypoints

There is no detection and object avoidance
And remember the car still need to start at the initial position

The speed cannot be fast, as it may make the robot shaky or turn too fast
The robot may be lost and not localized

Rviz display in auto mode

Next target mark
- /next_target_mark
- Green ball on the waypoint, the car is targeting that green ball

Next waypoint mark
- /next_waypoint_mark
- Blue ball next to the green ball, not sure the function

Obstacle
- /obstacle
- A red box display, located at one of the obstacles in front of the car
- Also creating a red wall in front of the car, on the same plane of the red box

Detection Range
- /detection_range
- The detection range along with the waypoints, forming a green circle on each waypoint
- Larger the range larger the green circle

PP Trajectory mark
- /trajectory_circle_mark
- A while arc line showing the twist of the car

Search Circle Mark
- /search_circle_mark
- A red circle around the car published by pure_pursuit

Fusion Markers
- /detection/fusion_tools/object_markers
- The green markers that create bounding polygon around obstacles, the blue dot is the centroid of the polygon, marked with the distance from the car
- Similar to the object marker output from euclidean cluster detect
- But this already take the output from both lidar detector and image detector
- Published by range_vision_fusion node

Reprojection
- /points_fused
- Set the size to 0.08 to 0.1 to view the reprojection
- Reprojecting the image from the camera to the point cloud
- Showing image on point cloud

Other Misc Notes

Cannot start runtime manager

Check the error from the runtime manager terminal
When you start the runtime manager by the run script, two terminals will be started automatically
One of the terminals will be the roscore, if the master is already running, then it will close automatically
The other terminal is the terminal for all the process from the runtime manager
Information from the launch file will be shown when you enable a node

If you cannot start the runtime manager, do look close to the runtime manager terminal
Check where does the command stop before the terminal dies
Usually, at the start, it will load all the parameters from .yaml file

For example, if the terminal stops after the line loading param.yaml file
Go to autoware source file and find the scripts, check the param.yaml file
It might be the param.yaml file is accidentally deleted or empty causing the runtime manager stopped

Launching autoware simultaneously on two computers

Connect two computers via a LAN cable

- Check the IP address with the command *ifconfig*
- Choose one of the computers to run the ROS master
- Export the master address and the local IP address in .bashrc, then reboot

Run the runtime manager

- Run the runtime manager as usual
- If it cannot communicate with the master, check the master by running rostopic list or rosnode list, if there is an error then the connection is not setup

Separate the workload

- Ndt_matching and Yolo detection are two processes that require more computational power
- Separating these two nodes on two computers may make the process smoother
- Displaying in Rviz also takes resources
- Be aware that the camera should connect to the computer that is running Yolo
- Nodes publish or subscribe to large transmission topics should be on the same computer, like the camera is connected to computer A, then uvc_camera_node, Yolo, and other fusion nodes should better run on computer A