Localization Sensor Fusion Library

Sensors	**Odometries**
• Encoder • IMU • GPS • Camera • Lidar	• Encoder Odometry • IMU Odometry • GPS Odometry • Visual Odometry • Lidar Odometry

Input / Output Data Types (as IDL)

- nav_msgs/Odometry
- sensor_msgs/Imu
- geometry_msgs/PoseWithCovarianceStamped

Fusion Algorithm

- Extended Kalman Filter
- Unscented Kalman Filter
- … more can be added if needed

Relevant Links

- Robot Localization Package
  - https://github.com/cra-ros-pkg/robot_localization
  - https://github.com/methylDragon/ros-sensor-fusion-tutorial/
- Generic Sensor Fusion Package
  - https://github.com/tuw-cpsg/sf-pkg
- Robot Pose EKF Package
  - https://github.com/udacity/robot_pose_ekf
- Error-State Kalman Filter Package
  - https://github.com/EliaTarasov/ESKF
- Python Robotics Localization Guide
  - https://pythonrobotics.readthedocs.io/en/latest/modules/localization.html

Preliminary Plan (4-months)

Phases & Tasks		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
1.	**Related Work Review**	🟩	🟩	🟩															
a)	Checking available implementations	🟩																	
b)	Checking previous publications		🟩																
c)	▸ Report the outcomes with justifications			🟩															
2.	**Initial Library Pipeline**			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩							
a)	Selecting one input and output datatypes			🟩															
b)	Selecting one fusion algorithm			🟩															
c)	▸ Implementing initial library pipeline				🟩	🟩	🟩	🟩	🟩	🟩									
d)	Testing the pipeline over a dataset										🟩								
e)	▸ Report the outcomes with discussions											🟩							
f)	▸ Generate a README / documentation											🟩							
3.	**Update Library Pipeline**												🟩	🟩	🟩	🟩	🟩	🟩	🟩
a)	Allow various input / output datatypes												🟩						
b)	Allow various fusion algorithms												🟩						
c)	▸ Implementing updated library pipeline													🟩	🟩	🟩	🟩		
d)	Testing the pipeline over a dataset																	🟩	
e)	▸ Report the outcomes with discussions																		🟩
f)	▸ Update a README / documentation																		🟩

Tasks Descriptions

--

1.a) Checking available implementations

There are few packages available online, which worth checking as first step before proceeding with implementation, as a sense of inspiration and possible citation.

1.b) Checking previous publications

Not all published work has open-source packages, accordingly, checking publication over the last 5-years in the topic would give a better understanding of the problem, the current advances and a guide on where to go.

1.c) ▸ Report the outcomes with justifications

Deliverable report on the outcomes of the previous steps, with justifications of the decisions to be made for the next steps.

--

2.a) Selecting one input and output datatypes

Recommendation to start with PoseWithCovarianceStamped as the initial datatype for both inputs and output, however, if there is a proper justification for another choice, it will be welcomed and appreciated. Any selection should be decoded to an internal format, for the ease of modularity in the future.

2.b) Selecting one fusion algorithm

Recommendation to start with Extended Kalman Filter as the fusion algorithm, however, if there is a proper justification for another choice, it will be welcomed and appreciated. Any selection should be implemented using template classes, for the ease of modularity in the future.

2.c) ▸ Implementing initial library pipeline

Deliverable package with the implementation of the selected algorithm to fuse localizations with selected datatypes.

2.d) Testing the pipeline over a dataset

Testing must be done over a dataset with reference localization.

2.e) ▸ Report the outcomes with discussions

Based on selected evaluation metrics, the obtained results are compared to the reference one, in addition to discussion / comments on the behavior as validation to the initial hypothesis that fusion leads to better outcome, and in case of otherwise, justification is required.

2.f) ▸ Generate a README / documentation

Writing a full README for the package with documentation on how to use

--

3.a) Allow various input / output datatypes

Through a configuration file, allow the possibility for more datatypes for the inputs and outputs, which should be decoded to the same internal format.

3.b) Allow various fusion algorithms

Through a configuration file, allow the possibility for more algorithms for the fusion, which should be using the same template classes.

3.c) ▸ Implementing updated library pipeline

Deliverable update of the package with the implementation of the extension of various algorithms to fuse localizations with various datatypes.

3.d) Testing the pipeline over a dataset

Testing must be done over the same dataset with reference localization.

3.e) ▸ Report the outcomes with discussions

Updated the report with the findings based on the same evaluation metrics

3.f) ▸ Update a README / documentation

Update the README for the package with documentation on how to use.