Final: Sensor Fusion and Object Tracking

REVIEW | CODE REVIEW 1 | HISTORY

Meets Specifications

Splendid work Udacian! ⭐

Congratulations, you have passed the Final Sensor project by meeting all the required rubrics! 🎉🎉

You have very well understood project requirements and have done a very nice implementation of the project tasks, completed all required TODOs very nicely in filter.py, measurement.py, trackmanagement.py, association.py. Your matrices are correctly initialized and the calculations needed are rightly done.

The association and track management complex functionalities are done correctly too. Measurements.py includes camera measurements as well as needed, there are no track losses and RMSE is within rubric limits.
Nice work in preparing a write-up document with required information. 👏

Keep up with great work, I have shared some code review comments for you in Code Review section 🙌!

Important links for further study:
If you want to deepen your knowledge in sensor fusion, you can take the Sensor Fusion nanodegree:
https://www.udacity.com/course/sensor-fusion-engineer-nanodegree--nd313
Here is the PyTorch documentation, which is extensive, in case you want to learn more about PyTorch:
https://pytorch.org/docs/stable/index.html
The Wikipedia article about Extended Kalman Filters is very informative, in case you want to deepen your knowledge of EKF:
https://en.wikipedia.org/wiki/Extended_Kalman_

Tracking

✓ Step 1: `filter.py`

- EKF is implemented including appropriate system matrix F and process noise Q for constant velocity motion model.
- EKF is applied to a simple single-target scenario with lidar only.
- The mean RMSE is 0.35 or smaller. Please upload the RMSE plot as png or pdf file.

Track Management

✓ Step 2: `trackmanagement.py`

- Track initialization from unassigned measurements is implemented.
- A track score is defined and implemented.
- Track states are defined and implemented, e.g. “tentative”, “confirmed”.
- Old tracks are deleted for not updated tracks.
- The tracking works properly if you see the following results: After applying the track management to a new sequence (see instructions), the visualization shows that a new track is initialized automatically where unassigned measurements occur, the true track is confirmed quickly, and the track is deleted after it has vanished from the visible range. There is one single track without track losses in between. Please upload the RMSE plot as png or pdf file.

Data Association

✓ Step 3: `association.py`

- Nearest neighbor data association including association matrix is implemented.
- A method that returns nearest track and measurement for association is implemented.
- Gating method with chi-square-distribution is implemented to reduce complexity.
- The association works properly if you see the following results: After applying the data association to a new sequence with multiple targets, multiple tracks are updated with multiple measurements. The console output shows that each measurement is used at most once and each track is updated at most once.
- The visualization should show that there are no confirmed “ghost tracks” that do not exist in reality. There may be initialized or tentative “ghost tracks” as long as they are deleted after several frames. Please upload the RMSE plot as png or pdf file.

Great work ! ⭐
✅ chi-square-distribution is implemented and is not being called inside associate() function.

Sensor Fusion

✓ Step 4: `measurements.py`

- Camera measurements including appropriate covariance matrix R are implemented.
- Nonlinear camera measurement model h(x) is implemented. The Jacobian H is given.
- A method that checks whether an object can be seen by the camera or is outside the field of view is implemented.
- The tracking works properly if you see the following results: The tracking loop now updates all tracks with lidar measurements, then with camera measurements. The console output shows lidar updates followed by camera updates.
- The visualization shows that the tracking performs well, again no confirmed ghost tracks or track losses should occur.

Provide the output video using the `make_tracking_movie` flag as well to show your results.

Evaluation and Conclusion

✓ Overall tracking performance is evaluated using RMSE metric. RMSE plot shows at least three confirmed tracks. Two of the tracks are tracked from beginning to end of the sequence (0s - 200s) without track loss. The mean RMSE for these two tracks is below 0.25.

✅ Tracking performance is evaluated using RMSE for 3 tracks. There is no track loss for the sequence 0-200s. However the spikes are quite high , maybe due to the noise matrix.

✓
- A write-up in pdf or markdown summarizes findings by answering the questions in the starter template.
- Results are uploaded on Github with a README that contains all necessary information to install and run the code.

Excellent details in the writeup ! ⭐
✅ There is a recap of the entire project (including filters , association ,lidar fusion)
✅ Benefits of using multimodal lidar/camera data
✅ Challenges faced during the project (mainly with SNN) and other distance measures/scalability
✅ Improvised tracking and further next steps.

DOWNLOAD PROJECT

1 CODE REVIEW COMMENTS ›

RETURN TO PATH