Evaluating Motion Tracking Systems

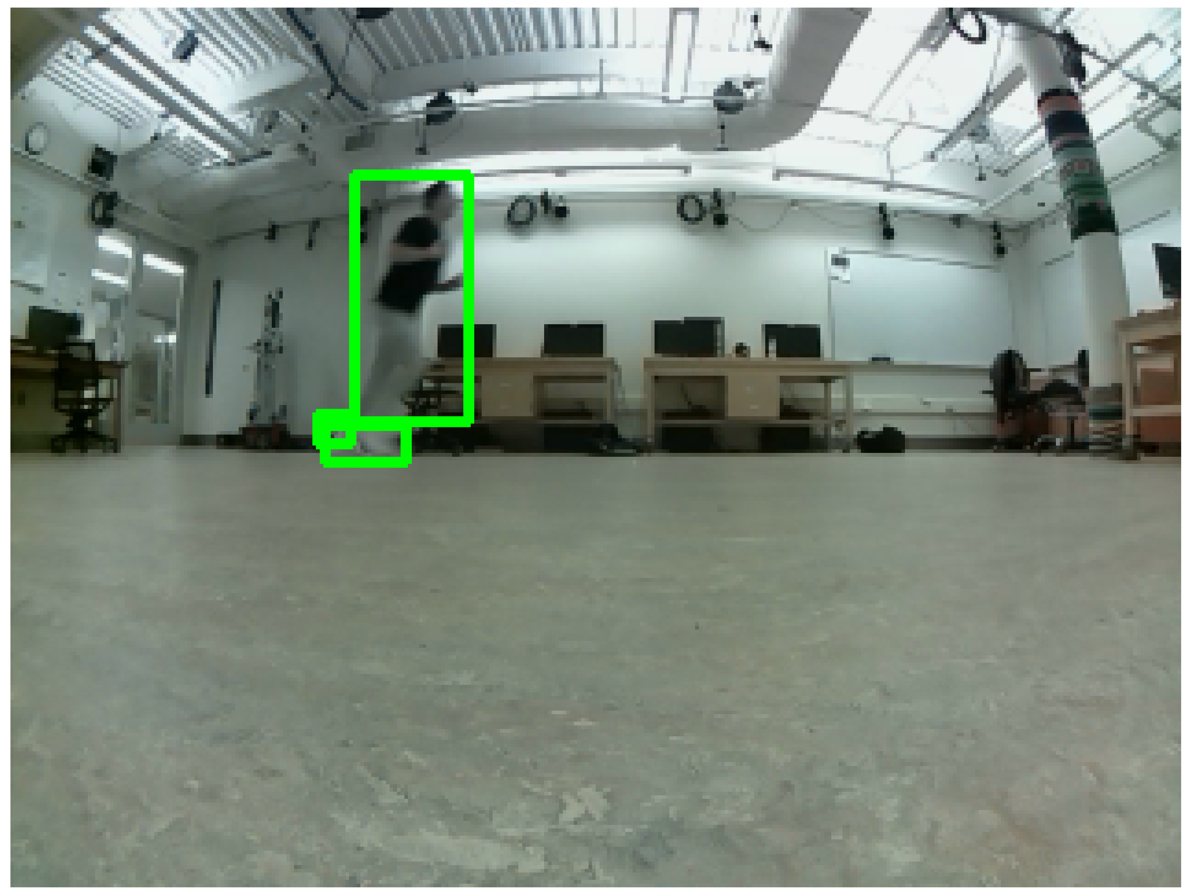
Jack Lynch Noah Sapire Aaron Cass (Advisor)
Computer Science Department, Union College

Abstract

The purpose of this project was to work on the evaluation of the data, in particular making an efficient and intuitive framework that future projects can build new detection methods around.

Introduction, Motivation, and Background

In many computer science projects, especially those involving robotics and data-driven systems, progress often depends on building upon the work of previous groups. In our ROS-based project, we were tasked with transforming the data and code left by the prior thesis group into a readable and functional state. While our respective research goals differ, we share the same immediate objective: to make the existing code and data accessible, interpretable, and ready for further experimentation. Our focus is not on generating new results but on creating a clear, organized, and executable foundation that allows others in our group to conduct meaningful evaluation and analysis.

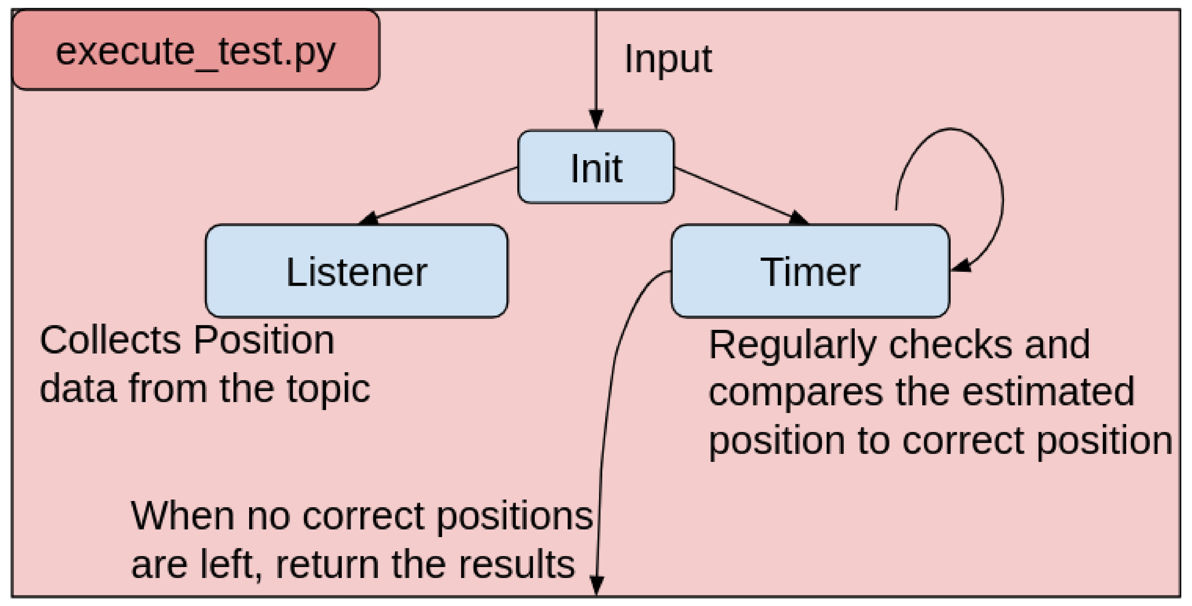


Methods

When we began our work, the codebase from the previous group was not in a functioning state. Much of the code lacked documentation, making it unclear how or why certain functions were implemented. To move forward, our goal was to make the code readable and operational so that our group could conduct evaluations effectively.
To achieve this, we took the following steps:

1. **Removed unused inputs:** Eliminated unnecessary arguments such as *group_name* which was never referenced or required for the program.
2. **Corrected callback functions:** Fixed incorrect callbacks by changing the input type from a genetic *String* to the proper custom message type *Position*. We entirely rewrote the rospy Timer callback, and
3. **Established a new evaluation plan:** Determined that the best approach for reliable evaluation would be to have the estimated position published to a topic from either a bag or an executing function, while reading the correct position from a text file. This setup mirrors what the previous group attempted but ensures a more structured and transparent method.
4. **Created an evaluate_all:** The benefit of our new evaluation is that it is run within a class object. We have made an additional wrapper function that will run all of the execute functions in order, returning the average accuracy, and storing the results into a text file.

These changes made us able to work with the code in a readable manner and gave us a structure for consistent and reproducible evaluations.

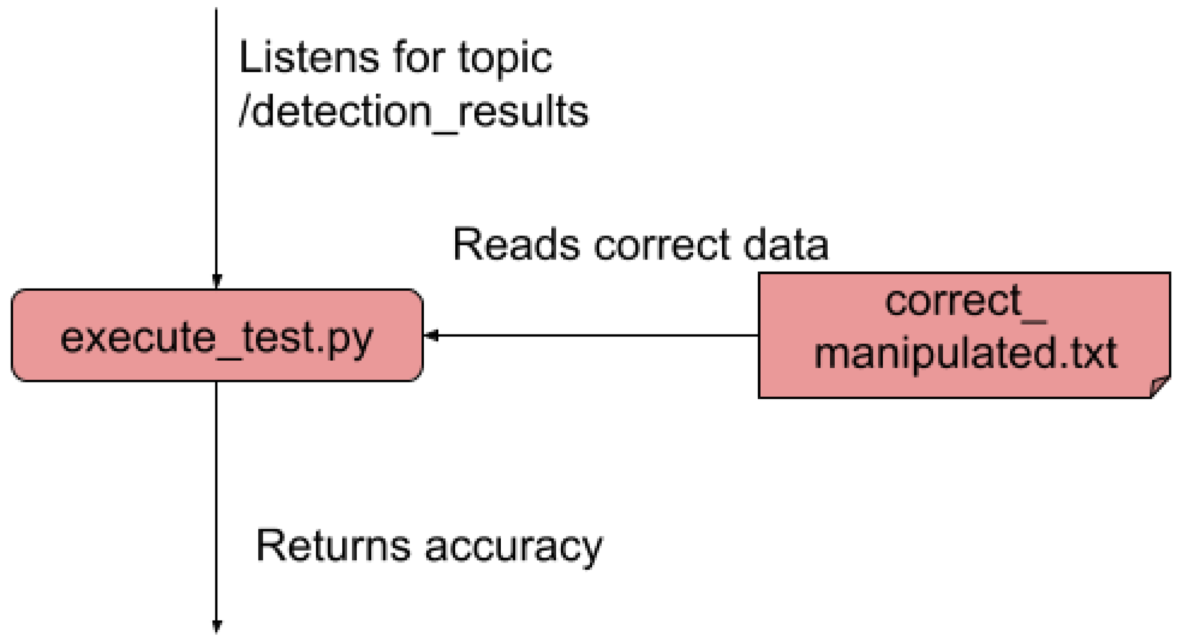


Results

After refactoring and debugging, we were able to restore core functionality to the ROS system. The corrected callbacks successfully processed data using the Position message type, and unnecessary parameters were removed, reducing confusion and potential runtime errors.
Although our task did not focus on producing final performance results, our work resulted in a stable and understandable framework. The system can now correctly read the estimated position from a bag file and compare it to a correct position loaded from a text file, allowing future researchers to easily test and validate motion tracking performance.

Analysis

Our analysis primarily focused on evaluating how the code behaved before and after refactoring. The original system often failed to execute or produced invalid results due to mismatched message types and missing documentation. After restructuring the callbacks and clarifying message usage, the system became predictable and much easier to debug.
The introduction of bag file playback for estimated positions allows for repeatable tests—an essential component of robust evaluation. Meanwhile, the decision to source the correct positions from a text file provides flexibility for generating more accurate ground truth data in the future.
Together, these improvements create a clear framework for assessing motion tracking accuracy in later stages of the project.



Conclusions

Our primary goal was to take a non-functional, undocumented ROS project and transform it into a readable, usable foundation for future evaluation. Through code cleanup, fixing message type errors, and defining a clear data flow between estimated and correct positions, we were able to achieve that goal.
We concluded that the most effective way to structure the evaluation was to play estimated positions from a bag file while reading correct positions from a text file. This design simplifies experimentation and allows future work to focus on refining the accuracy of the correct positions rather than debugging the underlying code.
In the future, we plan to enhance this setup by improving how correct positions are generated and by automating comparison metrics to quantify tracking performance. These steps will help ensure that subsequent research builds upon a solid, well-documented foundation.

Acknowledgments

Low-Cost People Detection in Low-Light Scenes by Gavin Trahan and Takumi Kojima.
Synthesizing Detection Data from Multiple Sources by Neil Datero and Hunter Gould and Shane Mullahy.

References

[1] Beach, C., Collins, J., Duong, D., Hung Duong, J. H. A. I., L'Italien, C., Mgaloblishvili, L., Nadel, E., Nguyen, T., Raj, P., Singhal, A., and Welch, K. Low-cost moving object identification. Tech. rep., Union College, 2024.

[2] Parveen, S., and Shah, J. A motion detection system in python and opencv. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (2021), pp. 1378–1382.