

What we did this week

At this point, since we integrated our own exit-finder algorithm (now dubbed RoomExit, for compliance with the original name) into the simulator's code, from now on, we'll consider the simulator side of the project and the RoomExit algorithm side of the project as a single combined unit. Yes, this does mean the integration was successful. As expected, (and mentioned in previous reports), we did run into several problems, one of them was especially annoying as we had to spend hours upon hours debugging it. The problem could be considered a simple typo, it was a difference check that was supposed to be an equality check (in the code, replacing "!=" with "=="). However, the time spent on debugging this problem didn't go to waste with the solution as in our attempts to solve the problem, we started refactoring and logging the code, this helped us find some other (relatively small) issues. The two main issues we found were:

- 1) The hyperparameters we use need some tuning, especially the ones that have to do with ransac, dbscan, and retries.
- 2) Scalability problems: as described in the previous report, when we try to make a complete scan of the room, sometimes, ORB-SLAM "loses track", the solutions we implemented beforehand seem to no longer suffice. To combat this, we split the retry part to a "soft" retry (retry the last scan x times), and a "hard" retry (retry the last y scans z times). This fix works well but not perfect, we figure that by modifying x , y , z and the angle difference between each scan, we'll reach a sweet spot (those are the hyperparameters regarding the retries mentioned in the previous section).

As we mentioned in the previous report, aside from the main algorithm and code and integrating it to the simulator we have written several small utilities for our project:

- 1) Bash scripts to make working in command line with the simulator more fluid and easier (the scripts can be found in our github), for even easier use we added appropriate aliases.
- 2) A trivial algorithm to find exit points (which simply returns the furthest points from drone) for comparison with our more precise algorithm.
- 3) An option for a random start.

As mentioned above, during our debugging process we started refactoring out code, and removing stuff we no longer need, we're not quite done but to quote Bon Jovi, "we're (roughly) halfway there".

As it looks, aside from creating visualization for our project (which should come along with the simulator), it seems that we managed to do our predefined goals for this week!

Plans for next week:

Last week we laid out a general plan, seeing how we managed to follow those steps, a good course of action would be to follow with the plan. This means that this week we'll focus on further refactoring, adding visualization, and making a final report that includes details we dismissed from the weekly reports.

Seeing how after Monday we won't have any tests left to do (unless the northern front changes things), we might add some little features to our project, though we're not sure yet if we'll have the time or what features to add.