

Group name: "Juantxotl"

Carolina Avila Carrasco

Rodrigo Moreno Carrillo

Yusseff Dominguez Ruiz

Esteban Ochoa Guaraca

1. Research topic:

The main goal is to optimize the robot's behaviour to maximize map coverage while having an acceptable mapping accuracy. In order to achieve this goal we have focused our work on the following aspects:

- Development of an exploration algorithm that plans and controls the path followed by the robot.
- Improvement of the robot's pose estimation by using sensor fusion.
- Combination of two different types of obstacle avoidance behaviour: deliberative and reactive.

After the implementation of these behaviours, we carried out an experimental study to evaluate the effect of different subsystem additions on both, coverage and mapping accuracy.

2. System architecture and optimized subsystems:

A diagram of the system architecture is shown below. A scheduler function controls the execution frequency of different tasks via an Interrupt Service Routine using Timer3.

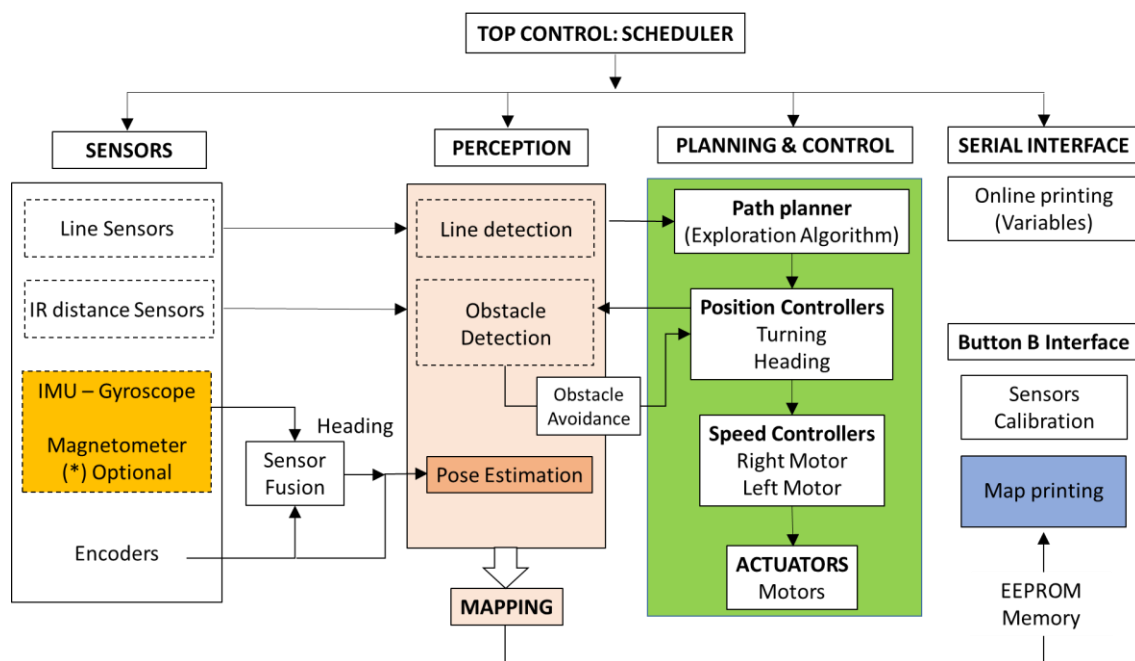


Figure 1. Diagram of the system architecture.

We have developed and optimized the following sub-systems:

1. Exploration algorithm:

The exploration algorithm relies on the mapping task, reading the map from EEPROM memory as it is generated to keep track of visited (the robot was physically in that cell) and explored cells (not visited but mapped using sensors information). The outcome of the algorithm is the next cell that the robot

should go to, giving priority to the closest non-visited cell. Figure 2 shows a work-flow diagram of the algorithm consisting of 3 main steps to obtain a new goal position each time a cell has already been explored:

- a) **Check Surround:** The algorithm retrieves information from the map about the status of surrounding cells. Of all the possible cell status (described in the mapping task section below), the algorithm only cares about OBSTACLE, VISITED, EXPLORED, and UNKNOWN in order to compute the outcome.
- b) **Check next cell available:** Based on the information from previous step, make a decision according to the following rules:
 1. Check surrounding cells in this order: North, South, East & West.
 2. **IF** the checked *CELL* is EXPLORED **THEN** *new_goal_position* = *position (CELL)* **ELSE** check next surrounding cell
 3. **IF** the checked *CELL* is UNKNOWN **THEN** *heading_toward* = *position (CELL)* **ELSE** check next surrounding cell

Could be the case that following these rules the robot gets stuck because none of the surrounding cells is available for visiting due to the presence of OBSTACLE or is already VISITED. In that case, the robot needs to “hop” to the closest available EXPLORED cell. To find that cell, we compute the Euclidean distance from the actual position (x, y) to all the cells labeled as EXPLORED and choose the one with the shortest distance.

- c) **Move:** This function sends the coordinates of the next *new_goal_position* to the Position Control task. There are two possible scenarios: Either it is adjacent to the actual position, which means it is safe to move just by a small movement; or it is far away from the actual position. In the second scenario, the obstacle avoidance reactive behaviour plays an important role to reach the next position.

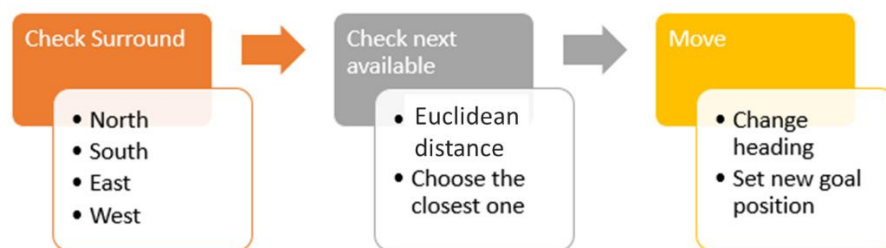


Figure 2. Process followed by the exploration algorithm for selection of next goal position.

Mapping task:

The exploration algorithm uses the mapping information to decide the next goal position as explained above. Five cell states or features were defined to track the coverage of the map with decreasing order of mapping priority:

1. OBSTACLE (O): The IR sensors detected an obstacle in their reliable range. Priority 1
2. LINE (L): The line sensors detected a line on the floor. Priority 2
3. BORDER (B): Virtual border around obstacles. Used by the exploration algorithm to reduce likelihood of clashing with the physical object. Priority 3
4. VISITED (.): The robot has been over that cell. Priority 4
5. EXPLORED (*) = The robot has detected an empty space using its IR sensors, that means that is available for visiting but it not visited yet. Priority 5
6. UNKNOWN (#) = Unexplored cell within the limits of the map. Priority 6

They have different priority to avoid overwriting already mapped cells. For instance, a VISITED cell, cannot be labelled as EXPLORED again, regardless if the sensors detect that there is an empty space, because VISITED has higher priority than EXPLORED.

2. Sensor fusion:

The performance of the exploration algorithm relies on the accuracy of the kinematics and mapping function, therefore we considered necessary to improve the pose estimation of the robot to get a better mapping accuracy. Two Kalman Filters were implemented for the heading estimation: one that merges the information coming from the encoders (angular velocity) and the gyroscope signal (gz) previously filtered, and one for estimating orientation using encoders and magnetometer. The heading estimation resulting from this sensor fusion is used to calculate the position (x,y) and update the pose of the robot. The magnetometer signal and its Kalman Filter could be optionally enabled or disabled. It is implemented in the code, but it was decided not to include the fusion of this sensor in the experimental evaluation because it was observed that the environmental conditions at the testing facilities affect the magnetometer readings significantly, therefore the information from this sensor is not considered sufficiently reliable.

3. Obstacle avoidance behaviour implemented at 2 different levels:

- a) **High level:** Cells with obstacles are mapped during exploration using the information received by the distance sensors. This information is taken into account by the exploration algorithm in the selection of the next goal position, which avoids the robot going to those cells.
- b) **Low level:** A reactive obstacle avoidance behaviour has been added through an obstacle-avoidance function using a “potential field” algorithm. When the distance sensors detect an obstacle is too close to the robot, a resultant repulsive force is generated that changes the motion direction of the robot. This force results from the balance between the attractive force generated by the goal (directly proportional to the distance) and the repulsive forces generated by the obstacle (inversely proportional to the distance). This behaviour complements the high-level obstacle avoidance dictated by the exploration algorithm and avoids the robot getting stuck when finding obstacles on its way to the next goal.

3. Experimental validation:

3.1. Goal and scope of the experiment:

The goal was to quantify the effect of different subsystem additions on the map coverage and mapping accuracy. Five different experiments were carried out on the proposed ‘Experiment-2’ map. All experiments included obstacles, bearing in mind that the aim was to evaluate the performance of the different subsystems under the ‘real’ environmental conditions proposed for the treasure hunt problem. Four tests included the exploration algorithm with their difference being the implementation or not of the sensor fusion and reactive obstacle avoidance behaviour (Table 1). These were compared with a baseline behaviour, implementing random movement with no exploration algorithm and basic obstacle avoidance using only one distance sensor in the front.

Test #	Name	Exploration Algorithm	Sensor Fusion	Low-level Obstacle avoidance
1	NoOA - NoSF	Y	N	N
2	YesOA - NoSF	Y	N	Y
3	NoOA - YesSF	Y	Y	N
4	YesOA - YesSF	Y	Y	Y
5	Baseline	N	N	N

Table 1. List of tests carried out for experimental validation according to active subsystems/behaviours.

Reactive obstacle avoidance behaviour was considered as an important factor in the implementation, since during the previous testing stage, it was observed that by enabling this feature by default, there were conflicts with the exploration algorithm, making the robot to get stuck without reaching a desired position close to a detected object and blocking further exploration. This reactive obstacle avoidance

was only useful when the robot had to move to far-away positions where the presence of obstacles along the path was uncertain. Therefore, when this type of behaviour is included, it is only enabled on far “hops” on the map. It was deemed necessary to evaluate the effect of adding or not this behaviour in our research since there was no *a priori* knowledge of the possible effects on the coverage.

3.2. Metrics:

For every experiment, coverage and mapping accuracy were evaluated using the following metrics:

Coverage:

- **Percentage of visited cells:** This information is extracted from video-tracking of the robot's path throughout the experiment. It was implemented a computer vision program that tracks and marks the robot's trajectory over the map grid and calculates the number of visited cells. This is our ground truth method for coverage evaluation and for verification of the robot's mapping accuracy.

Mapping accuracy: This is proportional to the similarity between the map printed by the robot and the true map grid, which was converted to matrix form in order to allow cell-to-cell comparison (see methods section below). Comparing both maps we have considered the following metrics:

- **Percentage of 'feature' cells correctly mapped** (from the total of cells mapped as feature).
- **Percentage of 'free' cells correctly mapped** (from the total of cells mapped as free).

In addition, the **percentage of visited cells correctly mapped** has been obtained by comparing the map printed by the robot to the ground truth exploration map resulting from the video-tracking which tells the actual visited cells (also converted to matrix form).

3.3. Materials and methods:

Materials:

- All obstacles were included in the map for all experiments.
- Romi control program: specific code version for each experiment according to Table 1.
- Sensors used (in addition to encoders):
 - 2x Line Sensors (Left and Right).
 - 4x Proximity IR sensors.
 - IMU: only gyroscope signal used.

Exceptions: RF sensor was not considered relevant for the purpose of this research. Although mounted and included as an option in the code, use of the magnetometer was discarded due to the bias observed on the readings. Baseline experiment had only line sensors and one proximity sensor active.

- Video recording of every experiment: tripod and mobile phone.

Methods:

Only one trial was recorded for each type of experiment. The condition for the end of the experiments was to complete the mapping of the whole area with the initial idea of including the time to cover the whole map in the comparative analysis. However, robot's behaviour was not robust enough to complete the coverage of the map in any of the tests due to different reasons like getting stuck at a certain position, blocked by an obstacle or simply by leaving the map boundaries. The measurements presented in the results section correspond to the maximum coverage achieved by the robot under the different test conditions, regardless of the duration of the experiment. The total duration and reasons for the end of every experiment are reported.

For every experiment, the recorded video was processed by a computer vision (CV) algorithm programmed in Python to track the full path followed by the robot. Detection of the robot was colour-based, using a range-filter detector in the HSV colour space which detected 'blue' features (robot's colour). The algorithm overlaps a 25x25 grid to the map image using a perspective transform and detects lines and obstacles edges by using a canny filter and Hough lines detector (Figure 3). Each frame

of the video was processed and a blue circle was painted in the image, moving with the robot and creating the blue lines that represented the full path of the robot. The outcomes of this video analysis are: the image of the gridded map with robot's path overlapped and the number of visited cells. This information was used as the ground truth for evaluation of the map coverage (actual number of visited cells) and mapping accuracy (comparison of actual visited cells with robot's map).

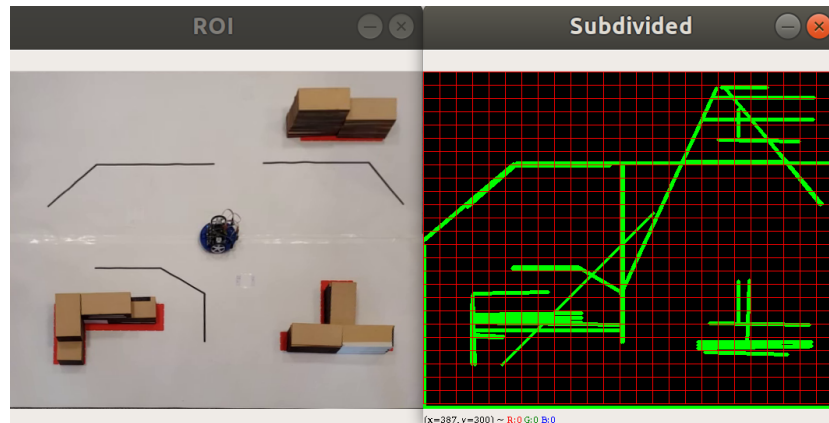


Figure 3. Map image processed by the CV algorithm showing lines and obstacle edges detection within the 25x25 grid covering the full map.

The maps printed by the robot were saved as txt files and processed in Python for results extraction (cells counting and matrix cell-to-cell comparison). For each trial, the printed map was processed and compared to the ground truth obtained by the CV program in order to verify the trajectory and the cells visited by the robot. In addition, the printed map was compared to the ground truth map matrix (described below) in order to quantify the mapping accuracy of features and free cells.

Generation of ground truth map matrix:

The image file provided with the map of the experiment was processed in Matlab to generate another image of the map with the grid on top (Figure 4). The grid has the same size defined in the robot's mapping function ($25 \times 25 = 625$ cells). From this grid image, a ground truth map file was created with Excel having the same format as the map printed from EEPROM memory by the robot, using characters 'L', 'O' and '#' to identify Lines, Obstacles and free cells.

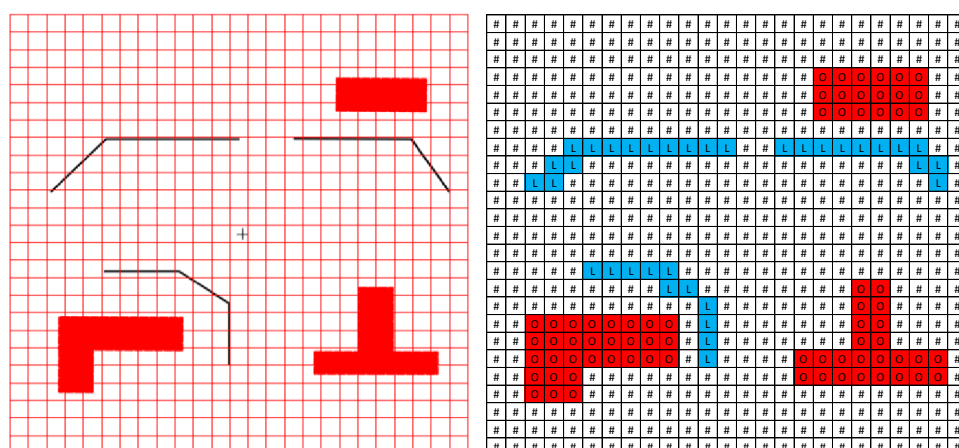


Figure 4. Left: Processed image of the true map with the grid. Right: Equivalent ground truth map matrix.

3.4. Results:

Metrics results for each experiment are summarized in Table 2, which also includes the duration of every test and the reason for the termination. None of the tests achieved a full exploration of the map; however, clear differences in coverage can be observed among the different experiments.

Map coverage evaluation:

Addition of the **exploration algorithm** only, without any additional subsystem, increased coverage with regard to the baseline behaviour (16.3% vs. 11% visited cells). Addition of **sensor fusion** to the exploration algorithm with no obstacle avoidance active (NoOA-YesSF) increased coverage more than 20% (39.2% absolute), but combined with obstacle avoidance (YesOA-YesSF), the increase in coverage was lower, around 13% compared to the plain exploration algorithm. According to these results, the biggest contribution to coverage improvement was the addition of sensor fusion for estimation of the robot's pose, with an average increase of 15% of visited cells compared to those solutions including the exploration algorithm with no sensor fusion. The exploration algorithm itself only achieved 5% improvement with regard to the baseline code but, as mentioned above, it was significantly improved when combined with sensor fusion. No clear trend can be derived from the inclusion of the **reactive obstacle avoidance behaviour** in terms of coverage. However, in those configurations where obstacle avoidance was included, the robot didn't hit any obstacle. Figure 5 shows the full trajectory followed by the robot for every test and gives a clear idea of the coverage differences just mentioned. When sensor fusion is not included, the robot mainly moves in the central region of the map, which is indicative of a large pose estimation error. In contrast, those configurations including sensor fusion show an extended map coverage where the path defined by the exploration algorithm can be more clearly distinguished.

METRICS	NoOA NoSF	NoOA YesSF	YesOA NoSF	YesOA YesSF	Baseline
COVERAGE					
Absolute number of visited cells (GT)	102	245	133	182	69
% Visited cells (GT)	16.3%	39.2%	21.3%	29.1%	11%
MAPPING ACCURACY					
Number of visited cells correctly mapped	51	135	54	98	35
% Visited cells correctly mapped	50%	55%	40.6%	53.8%	50.7%
Number of 'feature' cells correctly mapped	11	25	9	16	18
% 'Feature' cells correctly mapped	14.7%	32.9%	11.1%	22.9%	75%
Number of 'free' cells correctly mapped	232	239	224	213	54
% Free cells correctly identified	81.4%	81.8%	78%	74.7%	79.4%
% Explored cells correctly mapped	67.5%	71.7%	63.3%	67.7%	78.3%
OTHERS					
Total test duration (min:sec)	4:20	6:24	4:37	6:21	1:15
Reason for test termination	Got stuck	Blocked by obstacle	Got stuck	Got stuck	Blocked by obstacle

Table 2. Summary of results. Metrics quantifying coverage and mapping accuracy.

Mapping accuracy evaluation:

The **percentage of visited cells correctly mapped** refers to those cells identified as visited by the video ground truth which were also identified as visited by the robot. This percentage is quite similar for most of the tests and close to 50% accuracy, with the exception of the test without sensor fusion but with obstacle avoidance, which had 40% of correct cells. The difference between the actual visited cells and

those cells correctly mapped as visited by the robot can be clearly observed in Figure 6, where the correct identifications are marked on a stronger colour over the ground truth exploration map.

There is a high variability regarding the correct mapping of features among different experiments. Surprisingly, the highest **percentage of feature cells correctly mapped** is achieved by the baseline behaviour, where 75% of cells identified as features are correct. If looking at the absolute numbers, the configuration with exploration algorithm and sensor fusion (NoOA-YesSF) has the highest number of correct cells, however, it produced a large number of wrong detections resulting in a lower percentage of features correctly mapped. Results also show a trend for an improvement on feature mapping when adding sensor fusion to the exploration algorithm. Figure 7 shows these results graphically by highlighting those feature cells (lines and obstacles) correctly identified over the true grid map.

Mapping accuracy gets better with **identification of free cells** regardless of the system configuration. The percentage of free cells correctly identified is around 80% for all experiments except for the test including sensor fusion and obstacle avoidance, with a 75% of correct cells.

Last mapping accuracy percentage on Table 2 refers to the percentage of cells correctly mapped from the total number of cells explored by the robot (addition of free and feature cells). The lowest accuracy is for the configuration with obstacle avoidance and without sensor fusion whereas the highest is for the baseline test.

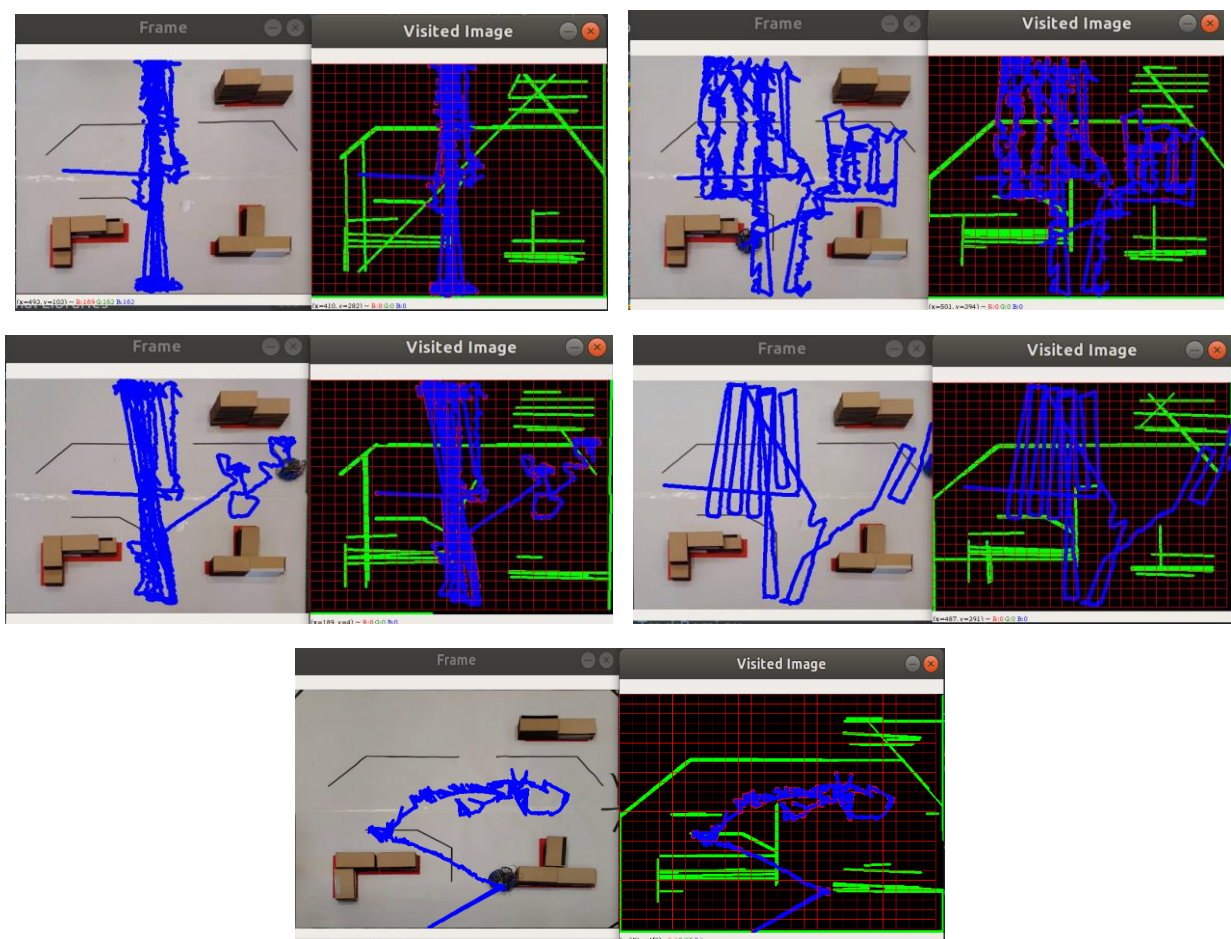


Figure 5. Full path measured by video-tracking in all experiments. Ordered from left to right and from top to bottom: NoOA-NoSF, NoOA-YesSF, YesOA-NoSF, YesOA-YesSF, Baseline.

4. Discussion:

The original idea when designing the experimentation was to include the **total exploration time** (time required to cover the whole map) as another metric for analysis so we could compare the efficiency in coverage. However, as soon as experimentation started with different system configurations we

realized it was quite an ambitious assumption, since the robot behaviour was not robust enough to ensure a full exploration of the map: it could hit or being blocked by an obstacle, it could stop at a certain position without apparent reason, etc. We could have chosen the approach of fixing the exploration time, let's say to 3 minutes, but that didn't guarantee that the technical problems mentioned above could not happen before the time considered. For this reason, we decided to evaluate the robot's coverage performance until failure, which would represent the total maximum coverage achieved by the robot when tested under real conditions.

We believe that the **ground-truth method used** is accurate enough for quantitative evaluation of the map coverage and comparison among different system configurations. A top aerial view that captures the whole map is enough to create a grid of the map and track the position of the robot within that grid. Comparison of this video-tracking map with the robot generated by the map shows a big discrepancy of the area covered by the robot, confirming the fact that the map printed by the robot cannot be used for a reliable evaluation of the coverage.

Regarding the **metric for coverage evaluation**, we could have included additional metrics to measure the actual size (e.g. maxim width) of the covered area, but we considered that visual information of the full path overlapped to the map provided by the video-tracking is a more complete way to compare the size and shape of the covered area.

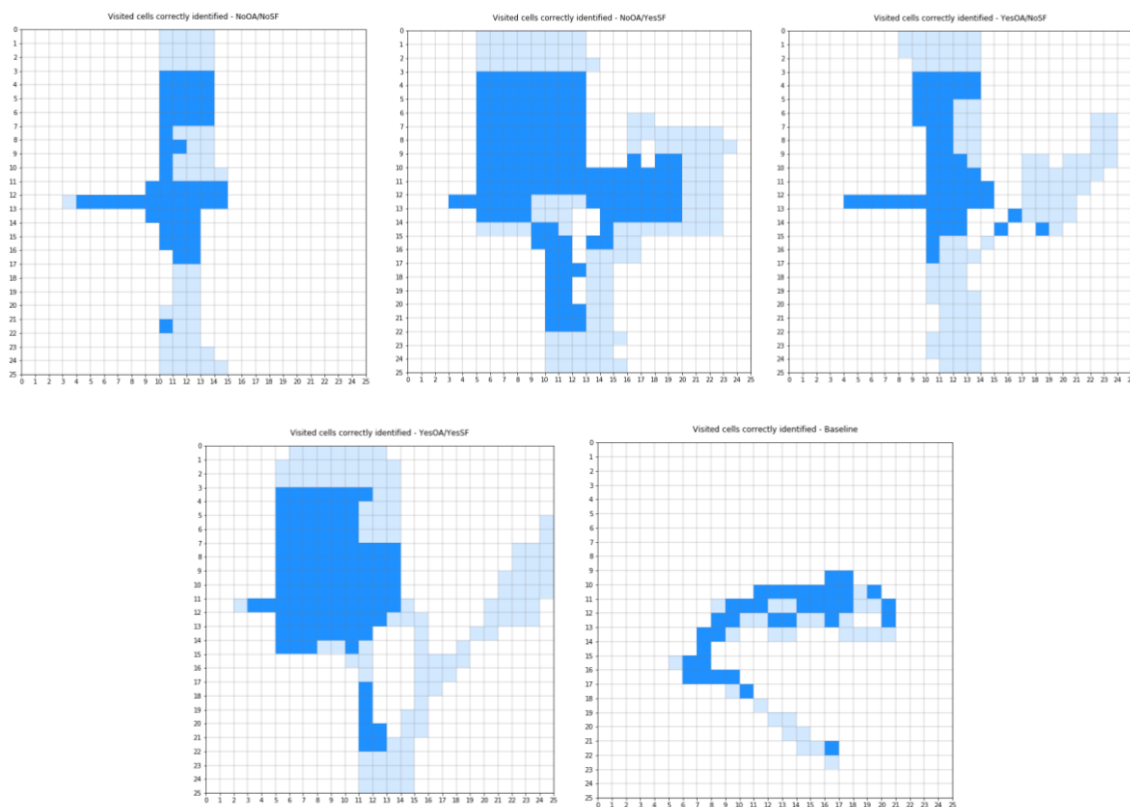


Figure 6. 'Visited cells' mapping accuracy with regard to ground truth map extracted from videos. Cells correctly mapped are marked in stronger colour over the ground truth. Ordered from left to right and from top to bottom: NoOA-NoSF, NoOA-YesSF, YesOA-NoSF, YesOA-YesSF, Baseline.

Evaluation of the **mapping accuracy** relies on the accurate quantization of the map area when building the true map grid. This was generated from the image file provided for the map; however, the original file did not cover the true map dimensions but a smaller area than the physical map. Although the file was modified to include the whole area covered by the physical map, small dimensional errors in this final map could affect the relative location of features within the grid (e.g. a line being one row up or down). Deviations of the built true map grid from the actual true map could negatively affect the

mapping accuracy but this type of error can be considered a bias that would equally affect the measurements for all of the experiments.

Regardless of the effect that small map reconstruction errors could have on the overall mapping accuracy metrics, the results of this experimental research show a big discrepancy between the map generated by the robot and the ground-truth maps. This is clearly illustrated by Figure 6 where in average, 50% of the cells which are physically visited are not considered as visited by the robot. This is particularly evident on the top and bottom regions of the map, which were traced by the robot but appear as not explored in the printed map. One **reason for this mapping error** could be that the update rate of the robot's pose was not constant or it was slower due to the workload of other tasks, producing a mismatch between actual pose and calculated pose; another reason could be an error in the kinematics due to sensor calibration errors. Consequently, this error could have affected the performance of the exploration algorithm as it relies on a precise mapping of the environment. Clearly, this is something worth of further investigation with more specific trials and more focused on the evaluation of the robot's kinematics.

Interestingly, feature mapping accuracy is much better on the baseline behaviour in terms of percentage of the explored feature cells. Due to the random nature of this test and the lack of test repetitions, we shouldn't extract conclusions from this result. The main aspect to consider here regardless of the mapping accuracy is that it is extremely likely that the baseline behaviour will always result in a premature termination of the exploration due to clashing with obstacles, and subsequently, resulting in a poor map coverage.

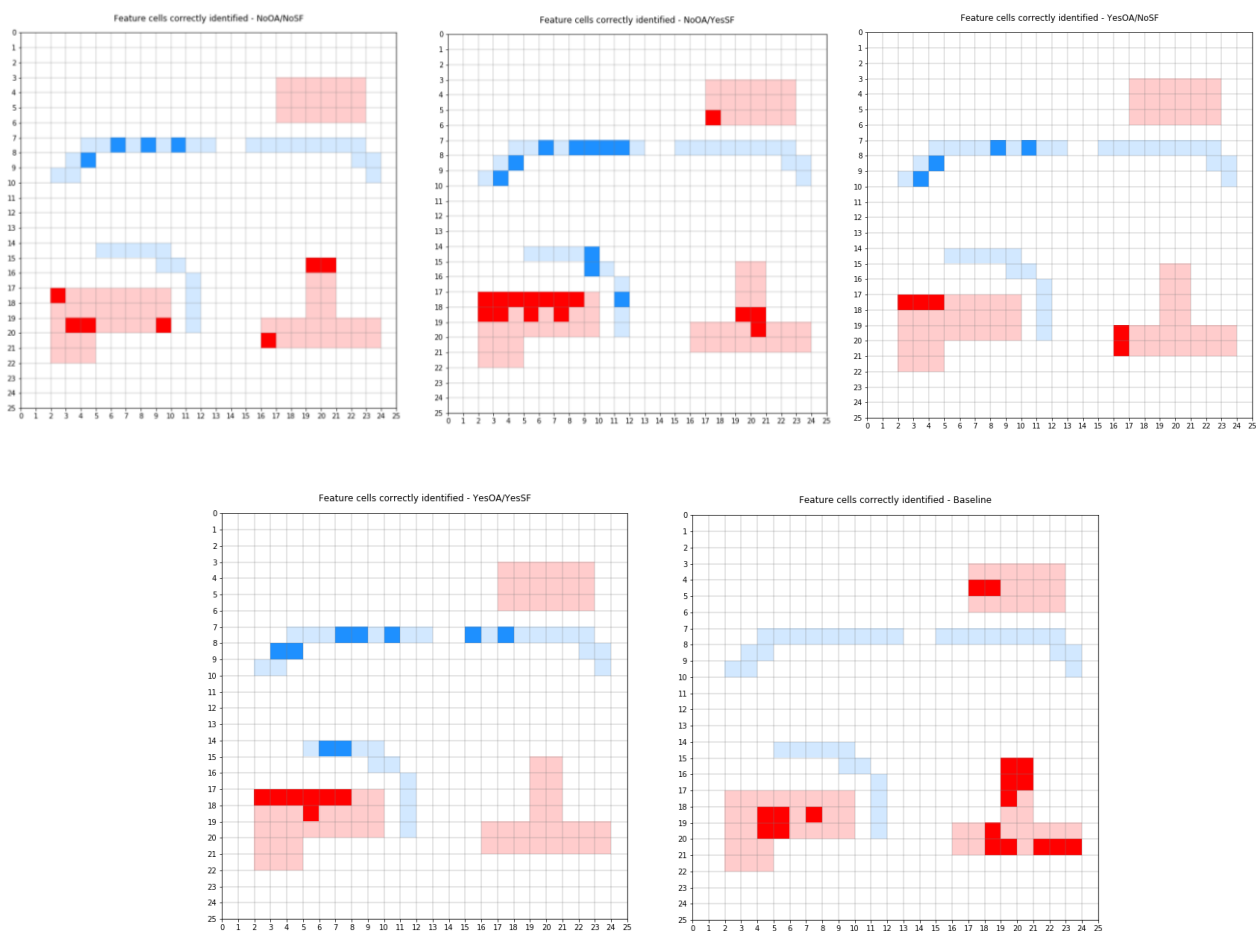


Figure 7. 'Features' mapping accuracy with regard to theoretical ground truth map. Cells correctly mapped are marked in stronger colour over the ground truth. Ordered from left to right and from top to bottom: NoOA-NoSF, NoOA-YesSF, YesOA-NoSF, YesOA-YesSF, Baseline.

The addition of a **reactive obstacle avoidance behaviour** doesn't have a clear influence on the coverage according to the results of this study (coverage increased when not using sensor fusion but decreased when using sensor fusion). It is expected that when obstacles are on the way to the next goal planned by the exploration algorithm, this obstacle avoidance behaviour will alter the path to navigate around the obstacles. However, we do not expect this path alteration having a significant influence on the overall map coverage. The **main contribution of the reactive obstacle avoidance** is to reduce the likelihood of hitting obstacles during the map exploration and this was demonstrated during the experimentation (the robot navigated around the obstacles when obstacle avoidance was active and hit an obstacle when not included). However, as mentioned in section 3.1, we couldn't make this behaviour work together with the exploration algorithm in a continuous way and chose to make it active only when the robot had to move to more distant regions of the map and presence of obstacles was uncertain. Future work should include further investigation to reduce possible conflicts between the exploration algorithm and the reactive obstacle avoidance.

In any of the experiments the **full coverage of the map** was achieved, therefore further analysis of the exploration algorithm should be performed or even other algorithms can be used to compare the impact in the coverage of the map.

As a final point for discussion, we would like to acknowledge the fact that **only one repetition** was evaluated for each of the tested configurations. The number of different configurations considered for analysis and the time constraints led us to take this decision; however, in order to extract more reliable conclusions it should be necessary to perform at least three trials per test configuration.

5. Conclusions

We have investigated the effect of the following subsystem implementations on the map coverage and mapping accuracy: exploration algorithm, sensor fusion and reactive obstacle avoidance behaviour.

The implementation of the exploration algorithm without any further improvements only increased coverage 5% with regard to the baseline behaviour.

The use of sensor fusion for estimating robot's pose had the biggest positive influence on coverage, with 15% average increase of visited cells.

The addition of the reactive obstacle avoidance didn't show a clear influence trend on the coverage but it proved to be useful to navigate around obstacles when the exploration algorithm sent the robot to distant locations on the map because the robot effectively avoided some obstacles and arrived to its destination.

Full coverage of the map couldn't be achieved with any of the tested options, which means that a revision of the algorithm implementation to achieve a more robust behaviour is needed, or even other algorithm can be used.

The experimental results show a poor mapping accuracy in all of the tests, with a big discrepancy between the map generated by the robot and the ground truth maps. In average, 50% of visited cells were correctly mapped and there were more cells wrongly identified as features than correctly identified for all the tested configurations except for the baseline. This is a strong indicative of an error in the estimation of the robot's pose.

The addition of sensor fusion to the exploration algorithm seems to improve the mapping accuracy, increasing the percentage of visited cells correctly identified as well as a higher percentage of correct identification of feature cells.

Despite the limitations of this experimentation, results show that the combination of the three developed subsystems improve the robot's performance by increasing the coverage of the exploration and reducing the likelihood of hitting obstacles. Nevertheless, further testing including more repetitions is required in order to have a more reliable quantification of this improvements¹.

¹ All the code, work and results from this investigation can be found in the following github repository:
https://github.com/biqomc/Romi_CW2