Report
Representation and relative positioning from visual information

Submitted by:
ASMA BRAZI

Supervised by:
CÉDRIC HERPSON

LIP6 Computer Science laboratory
Sorbonne University - Faculty of Sciences and Engineering

June - July 2019

Abstract

We carried out our internship at the LIP6, the computer science laboratory of Sorbonne University, Paris, France under the supervision of Dr. Cédric Herpson, from June 3rd, 2019 until July 26th, 2019. The internship was mainly considered as a continuation of works that we carried out during a university project (PANDROIDE) in our first year of Master's degree.

Our previous works presented a naive approach of object recognition and exploration strategy that allowed an autonomous robot with a camera to roughly build its environment. The environment is closed and composed of walls where the robot was able to locate itself. In addition, we could ask the robot to search for a specific object, and the robot was able to accomplish that task. Nevertheless, the conducted works assumed some hypotheses which did not make the exploration strategy generic.

During this internship, we focused the most on improving the exploration strategy to no longer depend on certain hypotheses. For example, in our previous works, we had some specific objects placed on corners to indicate to the robot that there were a corner at this spot. This hypothesis allowed the robot not to perform image processing to detect corners.

Our new strategy does not depends on these objects anymore.It exploits some visual information as contours. Also, the information provided by the robot's IR sensors is taken into account. We thus developed a strategy where no prior knowledge of the environment is necessary, and no hypotheses helping the robot building its environment are considered.

Acknowledgements

I would like to deeply thank my supervisor, Mr. Cédric Herpson, for his guidance and help, his presence and his trust in me to lead the initiative in my work.

I can say that I have been truly lucky to have Mr. Cédric Herpson as a supervisor who cared so much about my work, although he had a lot of work ahead of him.

Contents

Introduction

When an autonomous robot navigates in a structured or unstructured environment, it should be able to build a map representing this environment, to localize itself in it, and to define a safe path to move from a place to another one. These tasks are traditionally called Simultaneous Localization And Mapping (SLAM).

Applications to autonomous robot navigation are numerous : surveillance, cleaning, transportation tasks...etc. Paired with the progresses in term of computational embedded capabilities, the number of contributions of researchers in this field keeps going up [2].

Our project aims to enable a Thymio robot[1] with a monocular camera to roughly build its environment. We assume that the environment is closed and it is composed of walls where the autonomous robot is able to locate itself. Note that even if we focus on a specific platform for the experimentation, our proposal intends to be generic.

Unlike our previous work where we had some landmarks helping the autonomous robot in the detection of corners, we develop throughout this internship a more generic strategy. The autonomous robot only uses its sensors (IR and camera) to build its own representation of the environment.

So with this strategy, the robot can manage to build its entire environment autonomously.

Outline of this report

This document is organized as follows:

- Chapter 1: We realize a brief overview of the state of the art related to SLAM.
- Chapter 2: We detail the proposed approach.
- Chapter 3: We present our experiments and discuss the associated results.

Finally we conclude this work and highlight its current limitations before suggesting potential solutions and possible future works.

[1] An educational programmable robot

Chapter 1

Reviews of Existing Work

In this chapter, we first propose a brief overview of the literature before presenting the work done by our predecessors.

1.1 Brief overview of the literature

The exploration method we propose is based on the article **Exploration of an Indoor-Environment by an Autonomous Mobile Robot** [4], written by Ewald von Puttkamer and Thomas Edlinger, at the University of Kaiserslautern and published in 1994. This paper presents an algorithm for exploring an indoor-environment. The robot used is provided with an optical range finder, which is considered as the principal sensor for perception.

However, we note some differences between our two methods. Indeed, the robot presented in the article uses its 16 infrared sensors, a camera on pan/tilt-unit and a laser-scanner. Whereas our robot uses only its own 7 sensors and a 5-megapixels Raspberry-camera. Briefly, their robot is equipped with a stronger sensors unlike our robot. Therefore our robot needs to stop at regular short intervals to take a picture and retrieve visual information.

MonoSLAM [3] is another paper which interests us a lot because it uses only a camera as a source of information. MonoSLAM is a real-time algorithm based on SLAM [1] for 3D localization and mapping. The main difference with this approach is that it is performed in real-time, and it is especially designed for powerful humanoid robot and augmented reality. Unlike our strategy as explained above, the robot takes a picture, stops moving while analyzing it and than decides for the next movement. So, the mapping and the movement are sequential.

1.2 State of the project

The main purpose of our works is to improve the functionalities developed previously. These functionalities allowed one autonomous robot to move in a rectangular environment and to explore it.
The visual information brought by the exploration is used to build a virtual 3D representation of the environment, and to recognize a target object that should be found on one of the walls. The target object is

[1] A problem of simultaneous mapping and localization of an agent in an unknown environment

assumed to be available within the agent knowledge database. It is specified by the user at the launching of the application. In short, the previous strategy can be summarized as follows:

1. The autonomous robot takes a picture and analyses it.
2. The analysis generates new knowledge. The robot populates its 3D representation accordingly.
3. The robot is then in one of the following situations:
    - The target is detected and the exploration ends.
    - The four walls are explored and the target was not detected. So, the exploration ends.
    - A corner is detected. As a result, the autonomous robot goes on the next wall.
    - Beside the wall distance and size, nothing is acquired, the autonomous is ready to move to the next part of the same wall.
4. Execute next move until the exploration ends.

As it can be seen, we assumed some hypotheses which facilitated the exploration process. For example, we placed some landmarks in each corner of the environment and every time the robot met these objects, it concluded that this was a wall end. So the robot relied on these objects to decide whether or not it finished exploring the current wall. Otherwise seen, as long as the autonomous robot did not meet a landmark, then it continued moving forward assuming it is the same wall.

Another hypothesis is when the robot did not detect anything, and it assumed that it was the continuation of the current wall. Such a hypothesis may be hard to omit if we want to adjust the strategy to be usable in a more complex environment. A complex environment may be a room with corridors. In that case, a corridor may be ignored, because the robot only took into account the presence of objects for the decision and not the structure of the environment.

To no longer depend on these hypotheses, we decided to change the way we build the environment. Firstly, we do not anymore assume having a rectangular environment and propose a strategy which may work independently of the environment structure.
Secondly, we will not suppose the presence of a wall by detecting one of its corners. Instead, we will extract the contours of the environment. From there, the agent will have to decide if it is facing a close/far wall, if there is a wall behind them...etc. We will also use images analysis's results to estimate the dimensions of the walls. We will present these steps in detail in the next chapter.

Chapter 2

Realization

We suppose that the autonomous robot is situated in a closed environment, and it has been informed about the target to be looking for.

2.1 Exploration strategy

2.1.1 Image processing

The first task for the robot is to roughly represent the environment using visual information. For that we rely on segment recognition, a lightweight image process. [6] For walls detection, we distinguish three types of segments on the picture taken by the autonomous robot:

- Vertical segment which makes almost 90 degrees with the X-axis.
- Horizontal segment which makes almost 0 degree with the X-axis.
- Diagonal segment which makes an acute angle with the X-axis.

In the sections to come, we present the different strategies we have developed and the one we have particularly kept.

Corners-based detection

We begun with a strategy based on corners detection. We kept the same exploration strategy as in our previous works and replaced the part on corners detection. Thus, instead of detecting a specific objects substituting the corners, we tried two methods.

- A hand-made approach based on *segments intersection*
- A method for *corners detection* available in OpenCV [1].

These two methods require to fix a threshold to decide whether a corner has been detected or not. Defining this separating limit is not that easy. For example, it is getting arduous if we have a textured floor or walls. So the method considers all the intersection points situated on the walls and on the floor as corners.

Also, if the floor and the walls have almost the same color, the segment separating the wall from the floor may not be detected. As a result, these methods detect a lot of features which are not all correct

corners.
On the other hand, if we assume that corners are in most cases situated at the bottom of the image, this assumption becomes incorrect when the autonomous robot is far from the corner. In this latter case, the corner is situated on the top of the image. The figure 2.1 below illustrates this situation. As a result, this

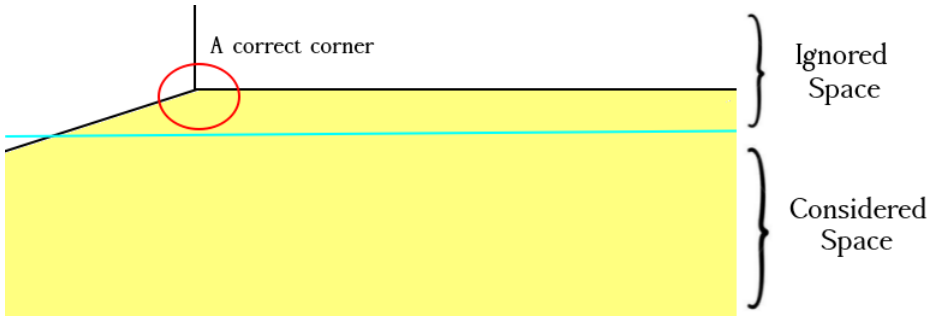

Figure 2.1: An ignored corner because of the corners placement constraint

corner will be ignored, because it is not situated on the bottom of the image.

The previous case makes the strategy not really captivating, because it is sensitive to the autonomous robot's position.

Non corners-based detection

Facing the difficulty to detect the right corners, we decided to ditch this strategy because it was impractical and to focus on the segment orientation.

The next idea is thus to consider the segments on the 3 axis X,Y and Z. The vertical (resp. horizontal) segments are used to estimate the walls height (resp. width). Diagonal segments indicate that there is a depth. Knowing that a certain depth exists on the image, guarantees to the autonomous robot that there is still space between them and the wall in front. So, it can always move forward.

We have used the *Line Segment Detector* algorithm (LSD) to detect straight contours on the image [5]. At each pixel of the image, LSD estimates the angle of the gradient. After that, the pixels sharing nearly the same angle of gradient are merged into regions. The connected regions are called *line support regions*. Then, the algorithm keeps some of these *line support regions* to be a *line segment*.

The picture below shows an image on which the LSD was applied and as a result, the *line support regions* that were detected.

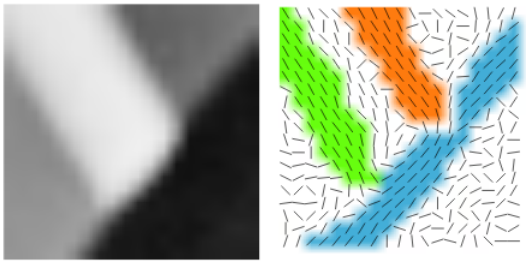

Figure 2.2: Line Support Regions. Adapted from "Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall, LSD: a Line Segment Detector, Image Processing On Line, 2 (2012)"

So, on each picture taken by the autonomous robot, we apply LSD to obtain segments. After that, the segments are classified into groups: Verticals, horizontals and diagonals, up to a certain tolerance ε. Then, in each group, we merge the closet segments up to a certain tolerance λ, in order to get a longer segments, we keep only the longest one from each group at the end. Each group of segments is used for a certain purpose for the navigation.

The horizontal segments which are situated on the top of the image are ignored. We assume that theses segments represent a ceiling. We do not risk anything by ignoring theses segments because, even if the segments represent a frontier between the floor and far wall. We develop a strategy where the autonomous robot does not build a wall unless it is close to it.

2.1.2 3D map building

We remind that our goal is to build a rough 3D representation of the environment and to look for the target object. We represent the environment by defining the walls composing it. The detection of walls is based on LSD's segments. Depending on the robot's position, the picture obtained may be in one of the general cases showed in Fig. 2.3 below.

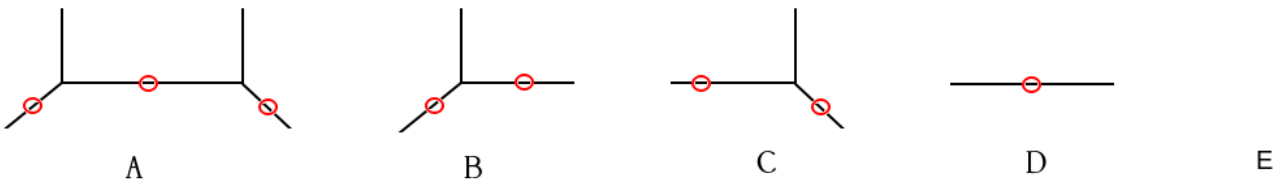

Figure 2.3: Walls detection depending on the robot's position

The useful information that we can retrieve from each case is as follows:

- Case A: 3 types of segments are detected, an horizontal one and 2 diagonals on 2 opposing walls.
- Case B and C: 2 types of segments (horizontal and diagonal) sharing an endpoint are detected.
- Case D: An horizontal segment is detected.
- Case E: Any segment is detected.

We mention that the vertical segments are used only to set the height of walls, they are not used in the exploration strategy because the autonomous robot is moving only along the X and Z axis.

2.1.3 Navigation

Navigation is the determination of an adequate and secure path for the autonomous robot, to move from a starting point to a targeted one [2]. On the basis of visual information, the autonomous robot is able to define its next decision. The decision may be to:

- Stop if the target has been found or if the robot has explored the entire environment, without spotting the target.
- Avoid obstacles.
- Move back, move forward and rotate to continue exploring.

Obstacle avoidance

In our previous work, when the autonomous robot decided to move forward, it did not check progressively while moving whether there is an obstacle or not. It checked for the occurrence of an obstacle after it stopped moving. The drawback of such a decision is that the autonomous robot was not able to distinguish between its last position at the end of the movement and its position at the first moment when it met the obstacle. As a result, the autonomous robot thought that it traveled the entire distance.

In our new strategy, the robot stops as soon as it detects an obstacle. Its sensors may detect the presence of an obstacle at almost 10 centimeters. When the autonomous robot stops, the real distance traveled is returned. If there were no obstacles, the distance would be entirely covered.

Strategy

As there are a lot of different decisions to be taken depending on the visual information, we summarize these cases using an activity diagram (see Fig.2.4). In this diagram, the green boxes represent a physical movement done by the autonomous robot, and the blue ones represent treatments.

We assume that the user has already mentioned the name of the object to find, and that the robot is situated in a closed environment. If we start with the initial point of the diagram, we can see that the autonomous robot begins by taking a picture and analyzes it. After that, it extracts the visual information and sends them to the computer, where the orders of movements are provided.

From the collected information, if the target is found, the robot stops. Otherwise, if there are any segments detected, each of these segments is represented by a wall and added to the 3D map. At the

end of the treatment, a movement is ordered to the autonomous robot (rotate, move back...etc). Then, it regains from the beginning until it stops.

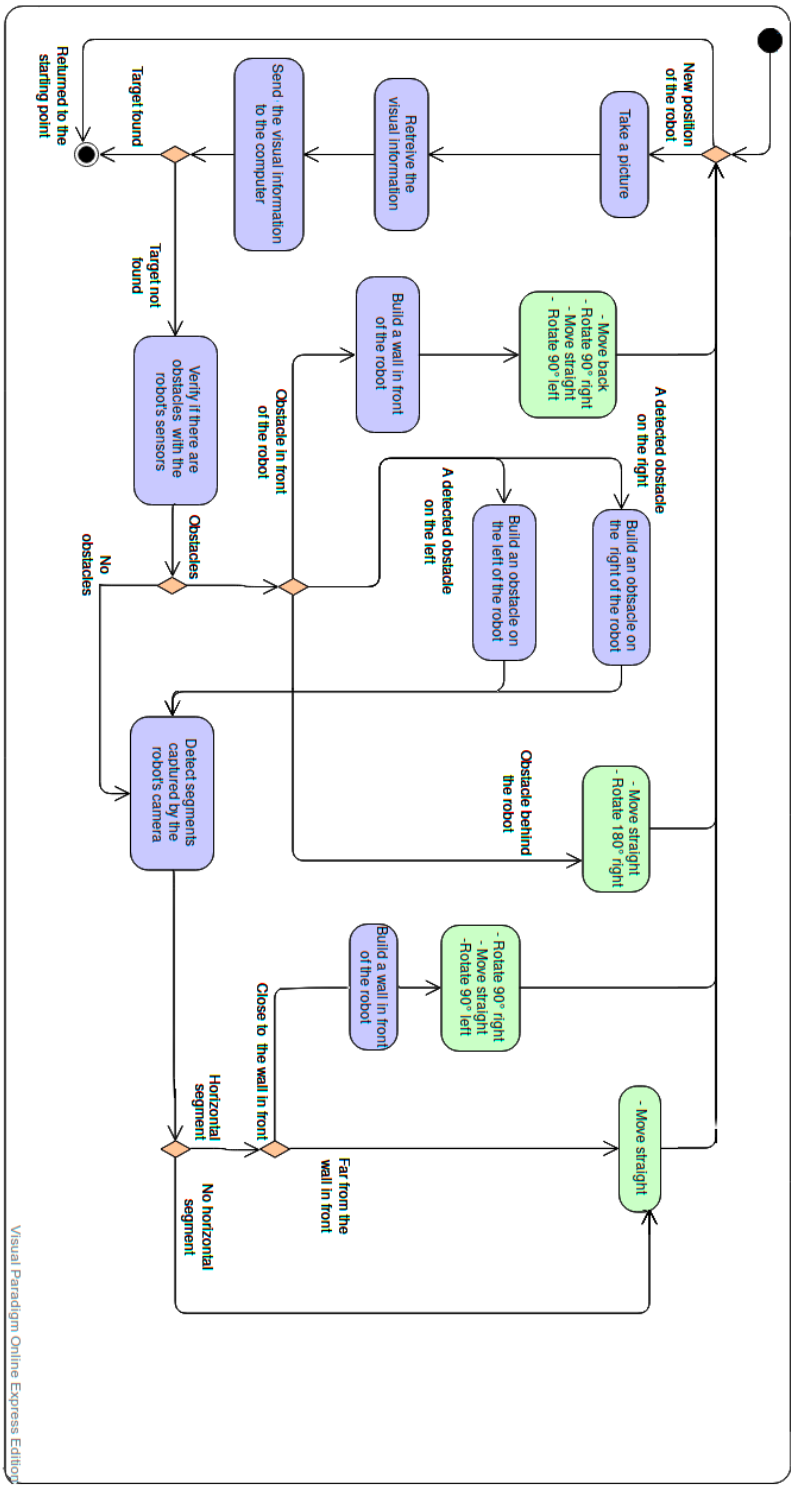

Figure 2.4: Activity diagram of the exploration strategy

Chapter 3

Testing

3.1 Environment plan

We chose an environment which contained corridors, obstacles, textured walls, different level of light and different types of door. Such an environment is interesting to reveal the robustness of the strategy.

When launching the application, the target object is filled in. Then, the autonomous robot is going to build progressively its environment while searching the target object. The picture below represents the environment plan.

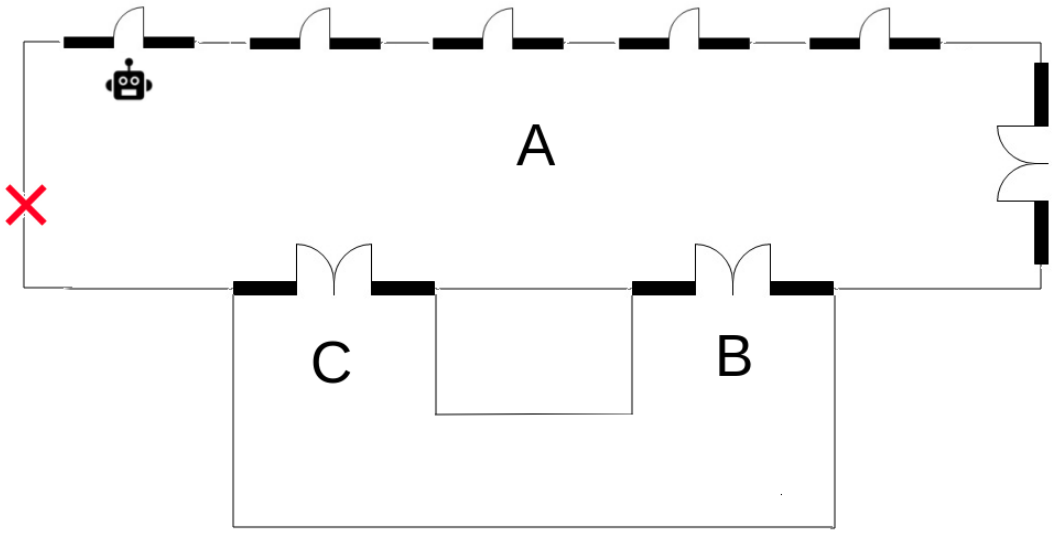

Figure 3.1: The environment plan

We can see that the plan is composed of a principal corridor (A) with 5 simple doors and 3 double-doors. There are two double-doors in front of the simple ones which lead to 2 other corridors (B and C). Than, the autonomous robot is represented by a robot symbol and it is in front of a simple door in the

principal corridor. The autonomous robot will try to find the target object which is represented by a red cross.

3.2 Image processing

In this section, we present the environment structure detection in different spots. We precise that the vertical segments are green, the horizontal ones are red and the diagonal ones are blue.

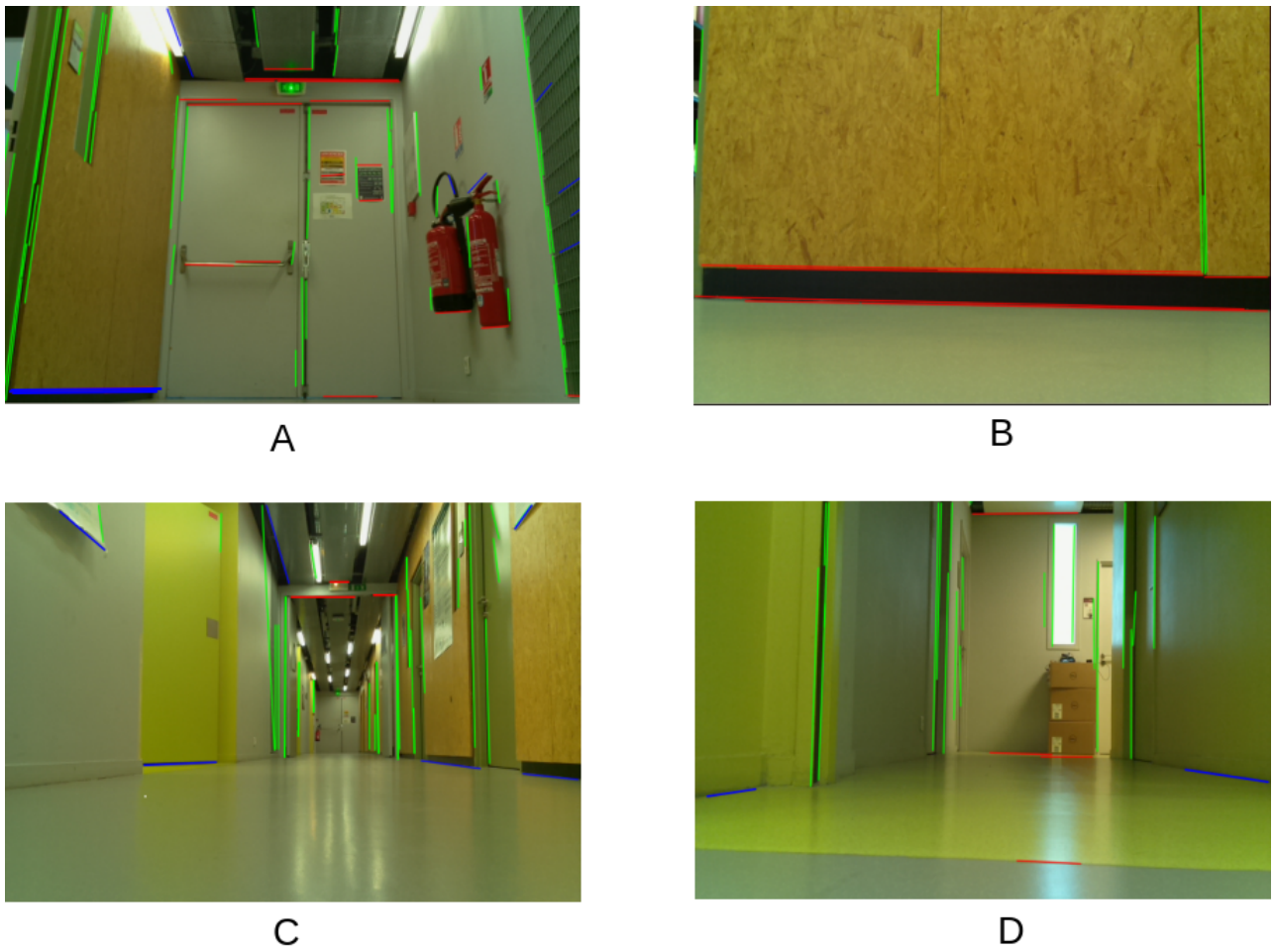

Figure 3.2: Segments detection testing

The segments shown in the figure above are not all considered. For example for those which are on the top of the image (On the ceiling), they are ignored by the algorithm later. Also, we remind that we keep only the longest segment from each category (Horizontal, vertical and diagonal).

The weak point of this algorithm is when the floor and the wall have almost the same color, we can see that in the figure C. However the algorithm encounters any difficulties while detecting segments on the brown wall. But, it fails at the detection of the frontier between the gray wall and the floor on the left.

Another weak point which may be noticed is when there is not enough lights in the room, as in the figure D. The wall on the left is nearly dark. Moreover both the wall and the floor are gray.

Beyond these weak points, the algorithm is able to extract the most important contours from the image, which remains sufficient for the decision.

3.3 3D map building

The picture below represents a 3D map representation of the environment at the end of the exploration. We precise that the environment has been built at a scale of 1/100 centimeters, hence the eventual impression that the walls are very offbeat.

At the beginning and as shown on the plan, the autonomous robot is in front of the door 5. As mentioned in the strategy that the autonomous robot goes always right while exploring the wall. Consequently, the autonomous robot explores almost all the environment before finding the target object.

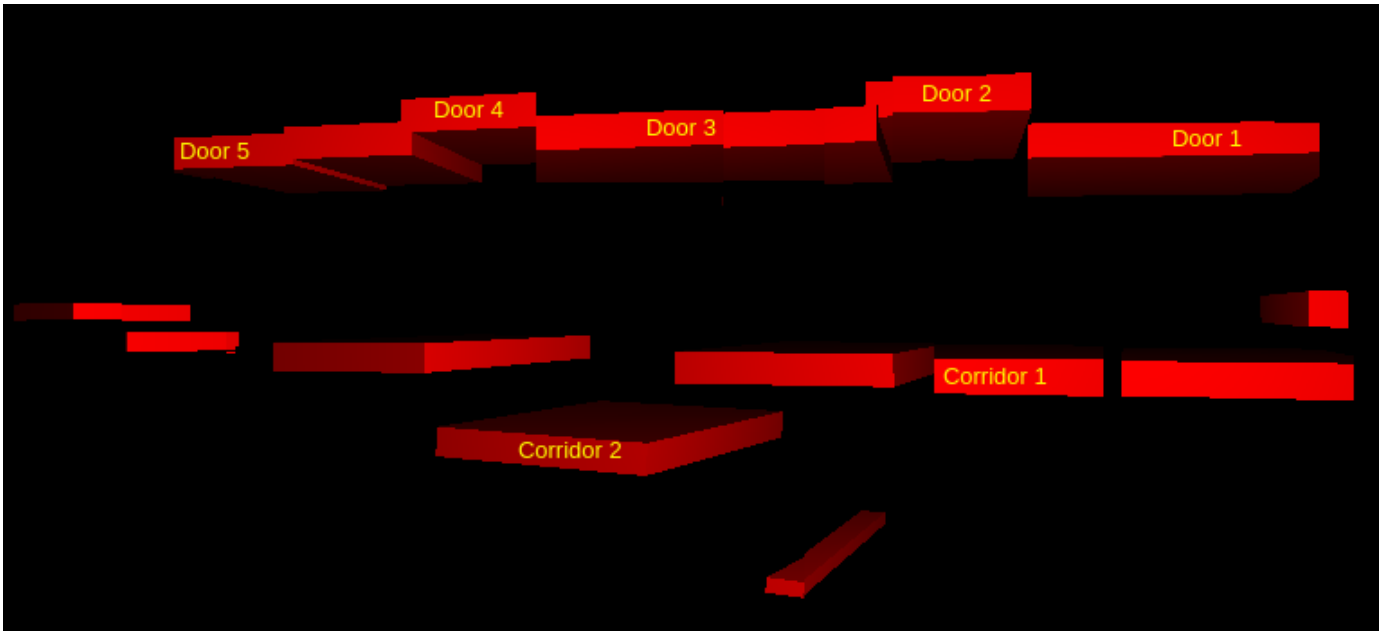

Figure 3.3: 3D map building of the environment

Now, we detail the 3D representation and we explain some details about it:

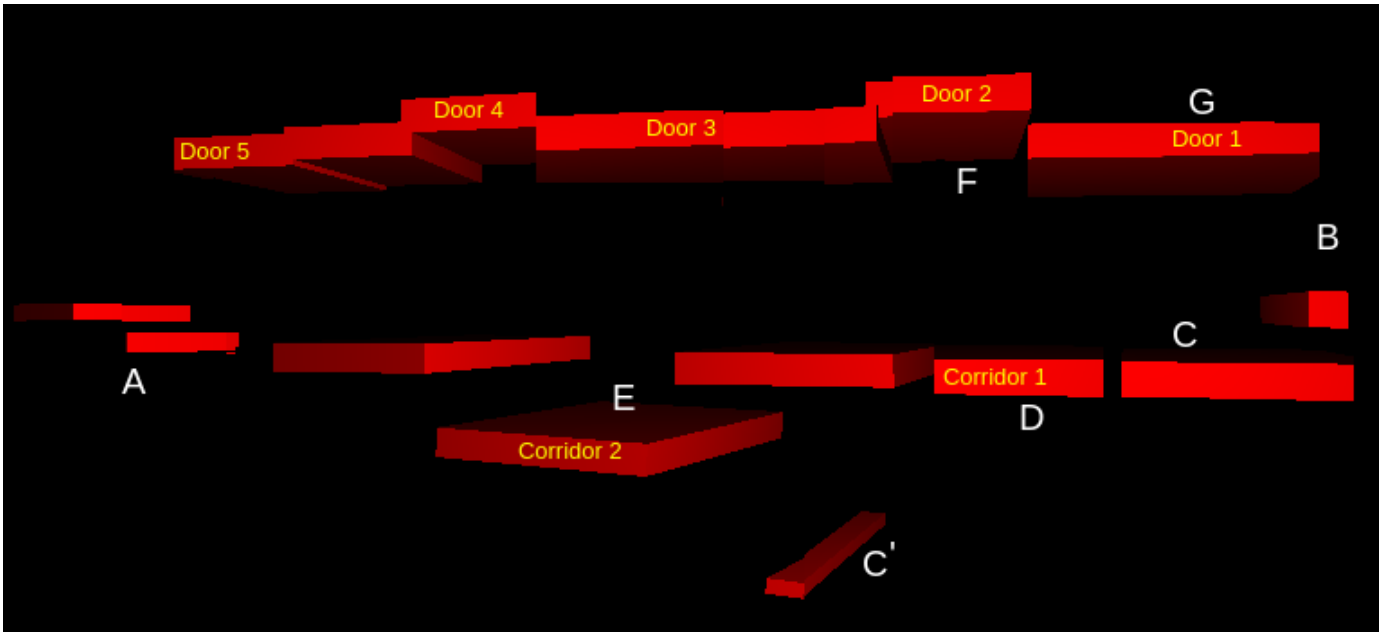

Figure 3.4: Discussion of the environment representation

- Case A: The autonomous robot is close to the opposite wall and it travels a short distance to explore a new part of the same wall. So, the recovery phenomenon can take place because a part of the explored wall at the first spot remains in the field of view of the autonomous robot at the second spot. As a result, we get two bunk walls.
- Case B: There is a gap between the double-door and the floor. As a result, the segment separating the wall and the double-door may sometimes not be detected. According to the strategy, the autonomous robot moves forward and meets an obstacle (The double-door). Consequently, it draws a wall with a small width and continues exploring the rest of the environment. The picture below shows that specific case.

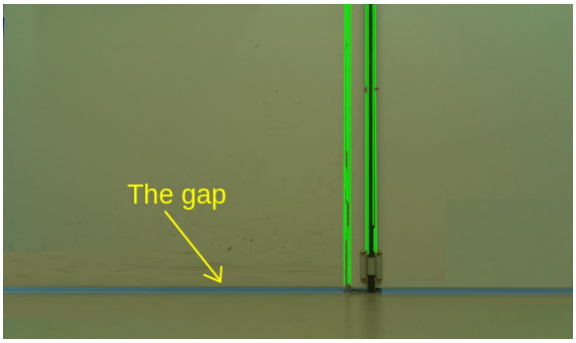

Figure 3.5: The grap between the double-door and the floor case

We do not consider the vertical segment (In green) and conclude that it is a wall, because the height does not guarantee the presence of a close wall. It may represent the height of an entire wall which is very far from the autonomous robot.

- Case C: At that spot, there is a big ventilation grid (fig. below). The autonomous robot detects one of the horizontal segment of the ventilation and considers it as a width of the wall in front. Unfortunately, the ventilation grid is lifted off the ground. Thus, The autonomous robot concludes that it's a border between the ground and a wall away from them, and places a far wall at C'.

Figure 3.6: Ventilation grid obstacle

- Case D: We turn off the lights of the corridor 1 to see how the autonomous robot behaves. Naturally, the corridor is dark on the picture at this spot. As a result, any segment of the corridor 1 is detected. But, the autonomous robot is located in the principal corridor where the lights is on. And the floor of the principal corridor and the corridor 1 had different colors. Consequently, The limit between the two floors is detected as an horizontal segment and the autonomous robot believes that it is a wall.

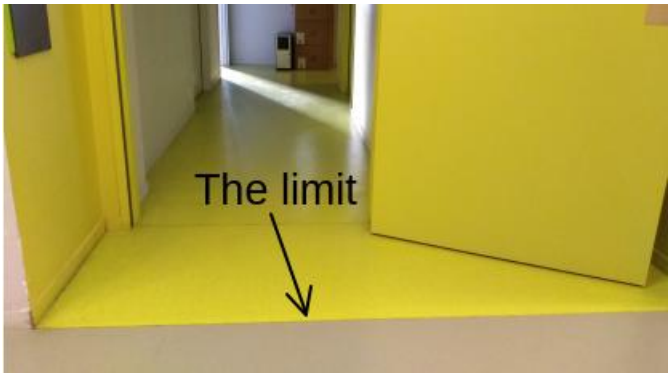

Figure 3.7: Two floors with different colors case

- Case E: We keep the lights on in the corridor 2. We notice that the far wall of the corridor 2 is detected. Normally, the autonomous robot should move forward, because it detects too the diagonal segments which guarantee the presence of a depth. Nevertheless, we have omitted from our strategy the consideration of diagonals in decision. Because, a shadow was detected once as a diagonal in a case where there were any depth, and this misleads the autonomous robot.

- Cases F and G: Bearing in mind the margin of error in rotation and movement, the closed doors 2 and 4 are detected properly. However the doors 1, 4 and 5 may be missed and considered as a walls.

After testing the strategy in a complex environment contrary to a simple rectangular room, we can judge that the strategy remains more generic and does not depend anymore of the environment structure, a specific objects substituting corners...etc. Also, we arrived at a precision at a scale of 1/100 centimeters. And the same exploration strategy runs at a scale of centimeters would have a smoother representation.

Conclusion, Limitation and Future work

During our internship, we focused the most on how to make the exploration strategy more generic. We can be somewhat satisfied with the work that has been done so far, comparing to our previous works and with respect to the internship's duration.

Nevertheless, we suggest some ideas of how our works can be improved:

- Multi-agent exploration: Instead of exploring the environment with only one autonomous robot, we can imagine having a group of autonomous robots doing so. In this case, the communication between the autonomous robots is to develop and the tasks to be shared.
- 3D object recognition: The current database contains only a 2D models. Works could be done in that field to manipulate a 3D models instead of 2D ones. This facilitates the recognition of the object at any views. Because now, the autonomous robot should be in front of the object to be able to recognize it.
- 2D/3D object detection with Deep Learning: If the database becomes large, the current object recognition may take a long time to analyze the picture. Because it tries to match the picture with each element of the database.
- Smooth environment representation: In our exploration strategy, a huge wall may be explored progressively. As a result, a parts of that wall are built side by side instead of building that wall as one piece. An improvement would be to merge that blocks of walls into one wall.

To conclude, having an internship at LIP6 and being guided by an amazing supervisor provide us with unique opportunities for personal growth. I did not expect at my first day that I will learn as much knowledge at the end. Well, we come out of our internship further strengthened.

Bibliography

[1] Line Segments Detector — OpenCV 3.0.0-dev documentation.

[2] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual Navigation for Mobile Robots: A Survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296, November 2008.

[3] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[4] T. Edlinger and E. von Puttkamer. Exploration of an indoor-environment by an autonomous mobile robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 2, pages 1278–1284. IEEE.

[5] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: a Line Segment Detector. *Image Processing On Line*, 2:35–55, March 2012.

[6] Payam S. Rahmdel, Richard Comley, Daming Shi, and Siobhan McElduff. A Review of Hough Transform and Line Segment Detection Approaches:. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, pages 411–418, Berlin, Germany, 2015. SCITEPRESS - Science and and Technology Publications.