Report
Representation and relative positioning from visual information

Submitted by:
ASMA BRAZI

Supervised by:
CÉDRIC HERPSON

Laboratory of Computer Sciences, Paris 6
Sorbonne University - Faculty of Sciences and Engineering

June - July 2019

Contents

Chapter 1

Abstract

Chapter 2

Introduction

The internship at Laboratory of Computer Sciences, Paris 6 (LIP6) was mainly considered as a continuation of works that we carried out during a university project in first Master's degree. These previous works presented a naive approach of object recognition and an exploration strategy that allows an autonomous robot with a camera to roughly reconstruct its environment.

During this internship, we focused the most on the object recognition, because it was the processing which took the most time to execute. To remedy this situation, we rely on a learning approach where the robot becomes able to recognize an object based on its knowledge. Unlike our previous work where the robot was trying to match the detected object in its environment with all objects in the database, hoping to recognize it.

We summarize in this report our work and the results obtained.

Chapter 3

Reviews of Existing Work

The main purpose of our works is to improve the functionalities developed previously. These functionalities allowed the autonomous robot to move in its environment and explore it. The visual information brought by the exploration is used to build a virtual 3D representation of this environment, and to recognize a target object. The target object is an object from the knowledge database. It is specified by the user, at the launching of the application.

In short, our previous strategy can be summarized as follows:

- At each move, the autonomous robot took a picture and analyzed it.
- The analysis of the picture brought new knowledge to the autonomous robot.
- With this new knowledge, The autonomous robot was able to represent its environment and to recognize the target object.

However, we assumed some hypothesis that facilitated the process of the exploration. For example, we placed some objects in each corner of the environment and every time the autonomous robot met these objects, it concluded that this was a wall end. So the autonomous robot relied on these objects to decide whether or not it finished exploring the current wall.

To no longer depend on this hypothesis, we present in our works another manner to construct the environment. Firstly, the autonomous robot does not try to detect the walls anymore by detecting corners. Instead, it takes a picture, analyzes it to extract the contours of the environment. With these contours, the robot decides if it is in front of a face, or if it still has a way to go...etc.
By this way, the autonomous robot gathers the analysis's results to estimate the dimensions of the walls.

Regarding the lengths and heights of walls, we globally estimate them in the same way as in the previous works. Nevertheless, the algorithm behind the calculations that we will develop later, is a little bit different.

Chapter 4

Realization

We suppose that the robot is situated in a closed environment, and it has been informed about the target object to search.

4.1 Exploration strategy

4.1.1 Image processing

The first task of the autonomous robot is to represent roughly the environment, using visual information. For walls detection, we distinguish three types of segments on the picture taken by the autonomous robot:

- Vertical segment which makes 90 degrees with the X-axis.
- Horizontal segment which makes 0 degrees with the X-axis.
- The rest of segments with the other angles.

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

Corners-based detection

We have begun with a strategy based on corners detection. Where we kept the same exploration strategy as in our previous works. But, we replaced only the part of corners detection. So instead of detecting a specific objects substituting the corners, we tried two methods. We developed an approach based on *segments intersection* and we used a pre-defined method from OpenCV for *corners detection*. These two methods were used under a certain constraints, to decide whether it is a corner or not.

Defining these constraints is not that easy. For example, it is getting arduous if we have a textured floor or walls. So the method consider all the intersection points of the walls and 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 situated in most cases on the bottom of the image, this assumption becomes incorrect if the robot is so far from the corner. In this case, the corner is situated on the top of the image. The figure below explicits that:

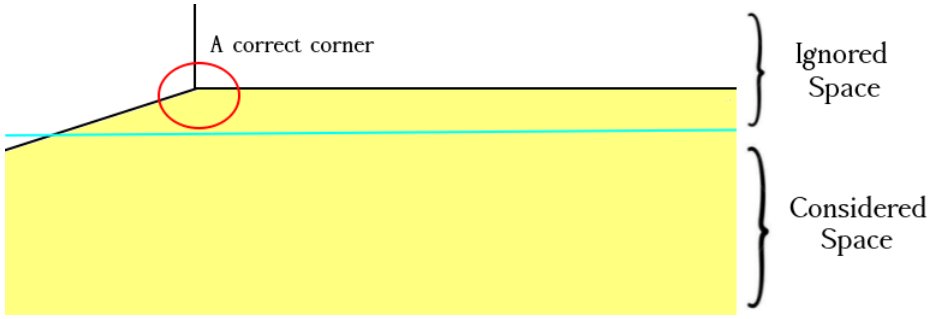


Figure 4.1: An ignored corner because of the corners placement constraint

So, this corner will be ignored as it is not situated on the bottom of the image.

These particular pictures taken by the autonomous robot make the strategy not really captivating, because it is sensitive to the autonomous robot's position.

Non corners-based detection

After encounting some difficulties to detect the right corners, we decided to ditch this strategy because it was impractical.

The next idea is 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). However the diagonal segments detected 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 algorithm: *Line Segment Detector* (LSD) to detect straight contours on the image. 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.

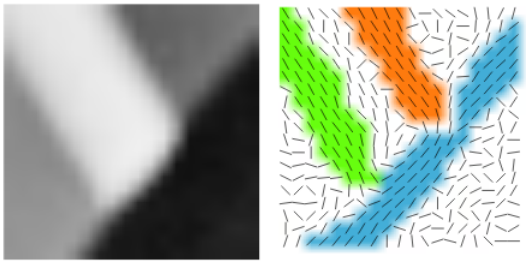


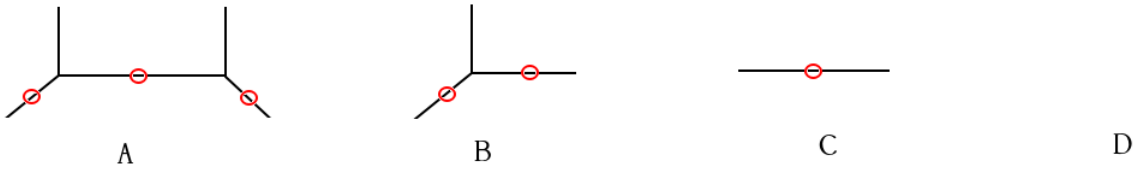
Figure 4.2: Line Support Regions. Adapted from "Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall, LSD: a Line Segment Detector, Image Processing On Line, 2 (2012)"

So, on each picture taken by the autonomous robot, we apply LSD to obtain segments. After that, the segments are classified into groups: the verticals, the horizontals and the diagonals, up to a certain tolerance ε. Each group of segments is used for a certain purpose for the navigation.

4.1.2 Indoor navigation

We remind that our goal is to build a rough 3D representation of the environment and look for the target object. We represent the environement by defining the walls composing it. The detection of walls is based on LSD's segments.

Depending on the autonomous robot's position, the picture obtained may in one of the general cases showed in the picture below.



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: 2 types of segments (horizontal and diagonal) sharing an endpoint are detected.
- Case C: An horizontal segment is detected.
- Case D: Any segment is detected.

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 on the X and Z axis.

4.2 3D Object recognition

Chapter 5

Testing

Chapter 6

Conclusion, Limitation and Future work

Chapter 7

References

Appendix A

List of components

A.1 Thymio II

Description

Thymio II is a mobile robot dedicated to the education field. It has many sensors for different purposes. These sensors covered:infrared receiver,proximity, 3 axis accelerometer, ground sensors for line following...etc.

Data sheet

`https://www.generationrobots.com/fr/401213-robot-mobile-thymio-2.htmlURL`

A.2 Raspberry-Pi

Description

The Raspberry-Pi is a s single-board computer with wireless LAN and Bluetooth connectivity. It needs a micro USB power supply (2.1 A) in order to be plugged into a power-bank. It has 1GB RAM, 4 USB 2 ports, Full size HDMI, 100 base Ethernet and including a quad core 1.2GHz Broadcom BCM2837 64bit CPU.

Data sheet

https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

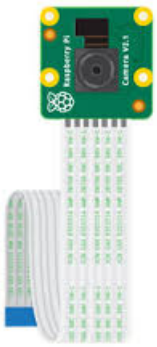
A.3 Raspberry-Pi Camera

Description

The Raspberry-Pi Camera delivers a 5MP resolution image, and 1080p HD video recording at 30 frame/second. It plugs into the Camera Serial Interface connector on the Raspberry-Pi.

Data sheet

https://uk.pi-supply.com/products/raspberry-pi-camera-board-v1-3-5mp-1080p?lang=fr



A.4 Power-bank

Description

For testing, we used RAVPower powerbank to power the Raspberry-Pi. It has 2A input which can charge the 6700 mAh portable charger. this feature guarantees that the Raspberry-Pi's services work correctly.

Data sheet

https://www.ravpower.com/p/Ravpower-6700mAh-Portable-Charger.html