Appendices
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

A List of components 2
A.1 Thymio II 2
A.2 Raspberry-Pi 2
A.3 Raspberry-Pi Camera 3
A.4 Power-bank 3

B Conversion between pixels and centimeters 5
B.0.1 Nonlinear Regression 5
B.0.2 Multiple Polynomial Regression 6

C 3D Object recognition 7

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/

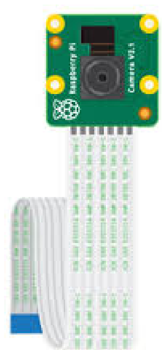
A.3 Raspberry-Pi Camera

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

Appendix B

Conversion between pixels and centimeters

In this section, we explain how we built the conversion system which permitted us to use the distances in pixels in the reality.

Firstly, we performed some measurements to know the dimensions of some objects in pixels and in centimeters depending on a certain distance from the camera.

B.0.1 Nonlinear Regression

The first table contains the equivalence between the real distance in centimeters and the one in pixels, between an object and the autonomous robot. The distance in pixels is the column X and the one in centimeters is in the column Y.

	x	y	Best Function Calculated y	Best Function Error
1.	392	2	18.62263621	16.62263621
2.	380	5	20.66766475	15.66766475
3.	365	10	23.91535548	13.91535548
4.	347	20	29.32422069	9.324220689
5.	328	30	38.04548798	8.045487977
6.	317	40	45.50036461	5.50036461
7.	302	60	60.74989973	$7.498997335 \cdot 10^{-1}$
8.	290	80	80.45851629	$4.585162872 \cdot 10^{-1}$
9.	283	100	97.35643493	2.643565069
10.	277	120	116.8711326	3.128867438
11.	268	140	159.986603	19.98660301
12.	265	160	179.80583	19.80582999
13.	263	180	195.0688499	15.0688499
14.	261	200	212.2498379	12.24983786
15.	260	220	221.6429632	1.642963214
16.	259	240	231.6189761	8.381023901
17.	257	260	253.4709531	6.529046932
18.	256	280	265.4244513	14.57554866
19.	254	300	291.5789532	8.42104681
20.	253	320	305.8522072	14.14779279
21.	251	340	336.9413971	3.058602898
22.	250	360	353.7965108	6.203489198
23.	249	380	371.5329225	8.467077498
24.	247	400	409.5732798	9.573279766
25.	245	476	450.6655805	25.3344195
26.	240	523.5	558.9172161	35.41721612
27.	230	659.5	650.899337	8.600663007

Figure B.1: Equivalence pixels-centimeters for the distance between the robot and the object

We performed a Nonlinear Regression on the data to obtain an equation with the distance in pixels as entry and the distance in centimeters as a result.

We can see in the column Best function calculated Y the result of the equation, knowing the distance in pixels. And in the Best function error, the difference between the distance in centimeters we have measured and the one estimated by the equation. The biggest error is estimated to 14.15 with X=253 and Y=320. The error remains acceptable because it is in centimeters and we remind that our goal is only to build roughly the environment. So, we do not need that precision of estimation.

B.0.2 Multiple Polynomial Regression

The second table contains

	x_1	x_2	y	Calculated y	Error
1.	12	291.57	6	5.941085609	$5.891439107 \cdot 10^{-2}$
2.	17	493.90	34	34.01596939	$1.596938913 \cdot 10^{-2}$
3.	30	321	22	22.03124041	$3.124041279 \cdot 10^{-2}$
4.	41	493.90	11	10.98618923	$1.381077155 \cdot 10^{-2}$
5.	43	103.25	14.5	14.67331914	$1.733191432 \cdot 10^{-1}$
6.	80	27.84	21	18.03244112	2.967558875
7.	100	24.68	11	15.15052281	4.150522812
8.	180	33.40	21	20.63091866	$3.690813367 \cdot 10^{-1}$
9.	145	25.78	14.5	13.91776739	$5.822326054 \cdot 10^{-1}$
10.	195	18.78	14.5	13.16978524	1.330214761
11.	241	21.65	21	21.96089411	$9.608941076 \cdot 10^{-1}$
12.	347	71.11	55	55.00197246	$1.972455392 \cdot 10^{-3}$
13.	582	21.05	51	50.98789442	$1.21055796 \cdot 10^{-2}$

Appendix C

3D Object recognition

This project may improved for 3D Object Recognition. Currently, the objects to detect are only in 2D. Obviously, it will be more interesting to make the autonomous robot able to detect 3D object, and to integrate them to the 3D virtual map.

In this case, we have begun to work on 3D object recognition, and we have thought about a solution where Deep Learning is used. For that,