

MINISTRY OF HIGHER EDUCATION AND RESEARCH

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES

NATIONAL INSTITUTE OF APPLIED SCIENCES

TOULOUSE, FRANCE



INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
TOULOUSE



Department of electrical and computer science engineering

INTERNSHIP REPORT

Development of a non-contact waist circumference measurement system

Biomedical Signal Processing Laboratory
National Electronics and Computer Technology Center (NECTEC)
Pathum Thani, THAILAND



Author:
Vincent Maire

Supervisor:
Dr. Dusadee Treeumnuuk

September 2018

Abstract

As a master's student in Electronics and Electrical Engineering at the National Institute of Applied Sciences in Toulouse, France, I ran an internship for 20 weeks (90 business days) in the Biomedical Signal Processing Laboratory (BSPL) at the National Electronics and Computer Technology Center (NECTEC), located near Bangkok in Thailand.

The internship's goals were mainly to prove my ability to apply my technical and communication skills in a professional environment, but also to improve them. This document is a report of this internship and its purpose is to show my main activities, works, scientific results and its contribution to my professional project. Moreover, you will find attached a shorter summary of this present report written in French language only. A PDF version of the document with hyperlinks can be found at: www.github.com/Vincema/report_internship_nectec_VM.

Acknowledgment

I would like to take this opportunity to thank all the people who made possible for me to carry out this internship.

First of all, I would like to express my deep gratitude to my training supervisor Dr. Dusadee Treeumnuuk for giving me the chance to show my abilities in his laboratory, the Biomedical Signal Processing Laboratory, and also for his patient guidance, useful critiques and suggestions during the development of this research work throughout my internship.

Moreover, I would like to express my deep gratitude to Dr. Decho Surangsirat, researcher at the Biomedical Signal Processing Laboratory, who has believed in my technical abilities and offered me the opportunity to contribute on another project in parallel of my main activities.

I especially must acknowledge the willingness of Mr. Dirat Tangwattanakom, Human Resource Management Officer at NECTEC, to make himself available to answer my numerous questions and helped me with the heavy paperwork.

I am also very grateful to my French teachers and supervisors, Prof. Daniele Fournier who kindly offered me her support, as well as Assistant Prof. Pascal Acco who would have been ready to provide me with his help with his well-known enthusiasm.

Finally, I want to express my kind thoughts to Mrs. Séverine Michon who took care of my presentation's organization, despite of my numerous requests about it.

Contents

Acronyms	6
Glossary	7
Introduction	8
1 Projects and objectives	9
1.1 Automatic waist circumference checkup system	9
1.1.1 Context	9
1.1.2 Review of similar existing systems	10
1.1.3 Specifications of the developed system	10
1.1.4 Hardware design	10
1.1.5 Software requirements	12
1.1.6 Internship objectives and scope	13
1.2 Orientation measurement system	14
1.2.1 Context	14
1.2.2 Project's specifications	15
1.2.3 Objectives	15
2 Automatic health checkup system - Scanning and recovering data points	16
2.1 System's architecture and hardware implementation	16
2.1.1 Raspberry Pi	16
2.1.2 LIDARs	16
2.1.3 Servomotors	17
2.2 Finding the user navel's height	17
2.2.1 Detecting the navel's height with a body scan	17
2.2.2 Estimating the navel's height	19
2.3 Scanning procedure	21
2.3.1 Notations	21
2.3.2 Scanning procedure	22
2.3.3 Running a scan	23
2.4 Raw data to 3D point cloud	23
2.5 Detection of outlying points	25
2.5.1 Outlier detection via minimum spanning tree	25
2.5.2 Outlier detection via Gabriel's graph	26
2.6 Source code	28
3 Automatic health checkup system - Reconstructing a curve and computing the waist circumference	29
3.1 Point cloud generator	29
3.2 Research and choice of curve reconstruction methods	29

3.3	Curve reconstruction by LOESS regression	31
3.3.1	Concept	31
3.3.2	Implementation	31
3.3.3	Results	32
3.3.4	Discussion	34
3.4	Curve reconstruction by B-spline via sequence joining method	34
3.4.1	Concept	35
3.4.2	Results	35
3.5	Curve reconstruction by B-spline via Curvature Based Squared Distance Minimization	36
3.5.1	Concept	36
3.5.2	Sum up on B-spline	36
3.5.3	Method's steps	37
3.5.4	Implementation	37
3.5.5	Results	43
3.6	Computation of the waist circumference	47
4	Automatic health checkup system - Validation	48
5	Orientation measurement system	49
5.1	System's architecture	49
5.2	Raw data recovery	49
5.3	Software's structure	50
5.4	Magnetometer values' calibration	50
5.4.1	Origins of the magnetometer biases	50
5.4.2	Proposed solution	51
5.4.3	Results	51
5.5	Orientation processing	52
5.5.1	Mahony's filter	52
5.5.2	Usage of the quaternion representation	52
5.6	Software's features	53
5.6.1	Serial communication	53
5.6.2	Orientation processing	53
5.6.3	Magnetometer's calibration	53
5.6.4	Displaying the orientation	54
5.7	Project's conclusion	54
5.8	Source code	55
6	Conclusions & perspectives	56
6.1	Technical contributions	56
6.2	Improvements that could have been made on my work	56
6.3	Personal feedback on my internship	57
Appendix A	NECTEC's presentation and board	58
Appendix B	Scanning procedure algorithm	65
Appendix C	Equations for circle fitting	66
Appendix D	Magnetometer calibration algorithm	67

Acronyms

BMI Body Mass Index. 9, 70

BSPL Biomedical Signal Processing Laboratory. 2, 3, 8, 58

DOF Degrees Of Freedom. 15, 49, 52

IMU Inertial Measurement Unit. 14, 15, 49, 51–54, 71

LOESS LOcally weighted Scatterplot Smoother. 30, 31, 33, 38, 44, 46, 70

MST Minimum Spanning Tree. 25–28, 70

NCD Non-Communicable Diseases. 9, 10

NECTEC National Electronics and Computer Technology Center. 2, 8, 13, 58

NSTDA National Science and Technology Development Agency. 58, 71

SDM Squared Distance Minimization. 30, 36, 44, 46, 70

WHO World Health Organization. 9, 17

Glossary

complementary filter Filter that fuses data from different sources (accelerometer, gyroscope, magnetometer) and outputs the orientation of an . 52

gimbal lock Loss of a degree of freedom in a three dimensional gyroscope mechanism appearing when the axis of 2 gimbals are parallel. 52–54

hard iron distortion Distortion of a magnetic field sensor which exhibit a constant additive magnetic field to the Earth's magnetic field. 51

inertial measurement unit Combination of accelerometers, gyroscopes and magnetometers integrated in an chip that measures the external forces, the angular rate and the surrounding magnetic field. 7, 8

LIDAR Laser/Light Detection And Ranging. Sensor or technology that measures distances by sending a laser beam and detecting its reflection on surfaces. It can be used to map a 3D space. 11–13, 16–19, 21–25, 70

perceptron Supervised machine learning algorithm used to binary classify a data by optimizing weights between the inputs and outputs. 10

quaternion Number system extending the complex numbers, where each object is a combination of 4 scalar variables. It can be used to represent 3D rotations. 52

soft iron distortion Distortion of a magnetic field sensor which wrongly scales the different axis of the surrounding magnetic field. 51

Introduction

For a period of 20 weeks (90 business days), I ran an internship in the Biomedical Signal Processing Laboratory at NECTEC. NECTEC is a Thai governmental research institution that clusters many different research units in various science fields of interests. Its purpose is "taking, supporting and promoting the research and development of electronics and computer technologies". A detailed presentation of the institution can be found in the appendix A.

My supervisor, Dr. Dusadee Treeumnuuk and others researchers at the BSPL submitted a research proposal, which consists basically in developing an Automatic Health Check-Up System. Dr. Treeumnuuk gave me the opportunity to develop autonomously one part of the system. Thus, my internship could be labelled as: "Development of a non-parametric regression algorithm for non contact waist circumference measurement and system's implementation". My work was intended to demonstrate the ability of the system to measure the waist circumference of an user, not designing a final product for commercial release. The results are then supposed to be published in an article. I had the chance to be let free into the development of this system, starting from zero, and I could make my own solutions proposal after a review of the literature. This research covers various technical fields that will be explained all along this report, such as optimization techniques and algorithms, geometry, software programming and basic hardware integration.

In parallel of the work I described above, another topic was offered to me by Dr. Decho Surangsirat. This could be labelled as "Data retrieving, processing and displaying from an inertial measurement unit". It consists mainly in a system able to assist a surgeon in a specific task where precision could be improved with the use of such a system. This project was much lighter than the previous one but still needs to be presented. Contrary to the waist circumference measurement machine, this system had more restrictive specifications (size, user interface's style, features...). The system needs then to be reviewed and tested by a surgeon to improve it before further investigation about its effectiveness.

Of course, this report will cover my work on both of the projects I described above. First of all, I will describe more sharply the projects, what they consist to, their purpose and the phases of development. This second chapter also explains what were the objectives given to me. Then, a second chapter, less verbose than the others, will show the solution and choices I have made and explaining why and how. It will also display the results, compare them to the expectations and conclude about how they could be improved. Finally, a last chapter shares my personal opinion about my work, this internship, and the results I have had, while showing how it changed my way to work and my methodology.

Chapter 1

Projects and objectives

During this internship, I had the opportunity to work on 2 different research projects that will be described in this chapter. It also talks about the work objectives that were given to me.

1.1 Automatic waist circumference checkup system

The main topic of my internship was: “Development of a non-parametric regression algorithm for non-contact waist circumference measurement”. This first section relates about the context in which this study has born, the project in which it will be implemented to, the characteristics of the system to develop and my goals.

1.1.1 Context

Non-Communicable Diseases (NCD) kill more than 38 million people around the world each year even though they cannot be passed from person to person. Therefore, World Health Organization (WHO) has addressed these diseases as an emerging problem that needs control and prevention. To avoid the risk of NCD, basic health checkup machines like weight scales can be used for obesity checks. However, WHO has recommended that waist circumference should also be taken into account as weight measurement alone may not be enough. Recently, the national health security of Thailand has launched a number of automatic health checkup systems for promoting prevention and control of NCD as pictured in figure 1.1.



Figure 1.1: Weight scale for BMI measurement in Thailand

In this context, Dr. Treeumnuuk and his colleagues proposed a research that aims to study and implement a cost-effective, high performance, non-contact machine that can automatically measure waist circumference in order to elevate a higher level of NCD prevention.

1.1.2 Review of similar existing systems

Apparently, it appears that the waist circumference measurement is a missing feature of almost all automatic health checkup systems even though some machines having the waist circumference measurement capability are available on the market. However, they are not dedicated to this only purpose and thus, are very expensive while providing the user with many anthropomorphic measurements based on a 3D body scan. Thus, they are not suitable to be used as automatic health checkup systems. For instance, the Treleaven et al.'s paper [1] presents a 3D body scan technology where numerous different human parts can be visualized and measured, using various techniques like laser-based scans or radio wave scanning. Although this system is very accurate and powerful, it uses too expensive and complex technologies. Another method can be found in the Li et al.'s paper [2], which introduces an image-based approach that can be used for anthropomorphic measurements. Taking into account the high cost of the system, it also requires a spinning base on which the user will have to stand still during the measurement, which is inconvenient and misfit with our expected purpose.

1.1.3 Specifications of the developed system

This project focuses on developing:

- An innovative technique of waist circumference measurement at the navel's level based on an hardware design which does not require high investments, heavy infrastructures or user inconveniences
- A new/existing process able to estimate the waist circumference accurately from a set of measurements

1.1.4 Hardware design

As explained in the section 1.1.3, the system has to be low cost enough to be integrated into an auto health checkup system for high scale deployment. Thus, no expensive sensor should be used and the size of the machine matters and must fit in a small room. Finally, it should be able to perform a measurement automatically, which means that no technician or medical assistant to tune the machine between each measurement. To lie in those specifications, several solutions have been explored:

Array of distance sensors

A first solution is to use a linear rack of many distance sensor (can be infrared or ultrasonic based sensors) as shown in the figure 1.2.

The user needs to stand still on the base while the system retrieves the distance for each sensor. Then, a part of the user's body can be reconstructed as a curve, which can lead to a machine learning based process in order to extrapolate the entire body contour and circumference. Another approach could be to use the inputs as entry points of a perceptron or multilayered perceptron for instance, which would give whether the patient is healthy or not. In any case, and as we cannot recover the whole body contour information, we will need a machine learning based procedure.

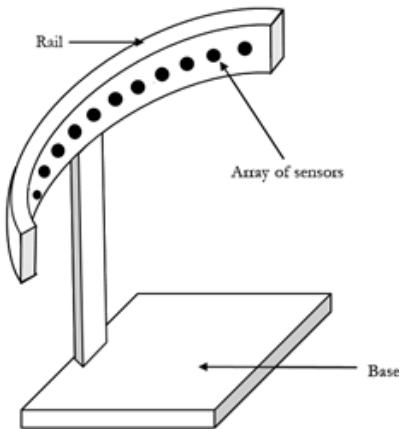


Figure 1.2: Schematic of the solution made of an array of distance sensors

This architecture answers quite well our expectations as:

- It is low cost
- It does not require a lot of space

However, it brings up some issues that may be hard to overpass:

- The algorithm to develop should be powerful enough to extrapolate more than the half of the user's waist contour
- As it requires a machine learning based process, many samples of patients are required to train the algorithm, which can be cost and time expensive for its development
- The sensor rack needs to adjust its inclination or height to perform measurement on people with different sizes (incorrect rack positioning can cause measurement errors)

For the reasons cited above, I have not worked on this system's architecture.

Contour measurement with LIDARs

Another solution imagined by Dr. Treeumnuk consists in a set of 3 LIDARs (Light/Laser Detection And Ranging) surrounding the patient. The general design of this solution is presented in figure 1.3.

Basically, the idea is to gather a point cloud which represents the user's body contour. A curve can then be reconstructed from the point cloud, and the circumference can be extracted from it. To be able to measure the waist circumference at the navel's level of patients with different statures, the LIDARs are mounted on a servomotor which is itself on the top of a pole. As shown in figure 1.4, the sensors tilt up or down, extending the range of reachable heights.

Here are the benefits brought by such a design:

- High precision sensors and actuators which makes the data much less noisy
- Wide range of measurable heights brought by the servomotors that tilt the LIDARs
- Low cost architecture (even if higher than the previous solution)
- No more complex machine learning algorithms required as the whole body contour is measured

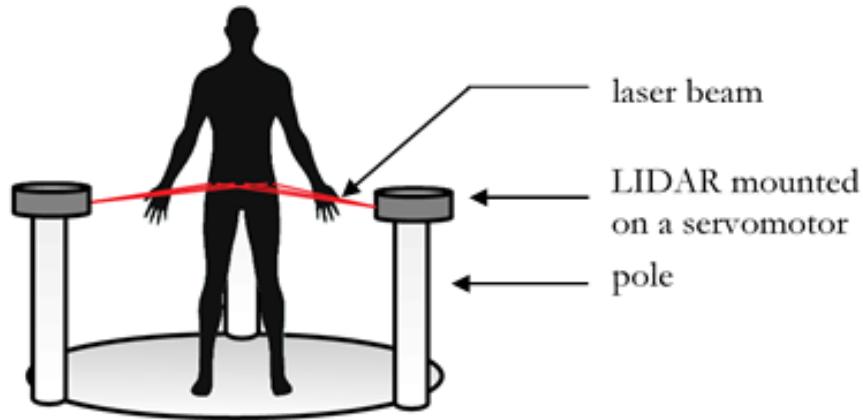


Figure 1.3: Schematic of the solution made of set of 3 LIDARs

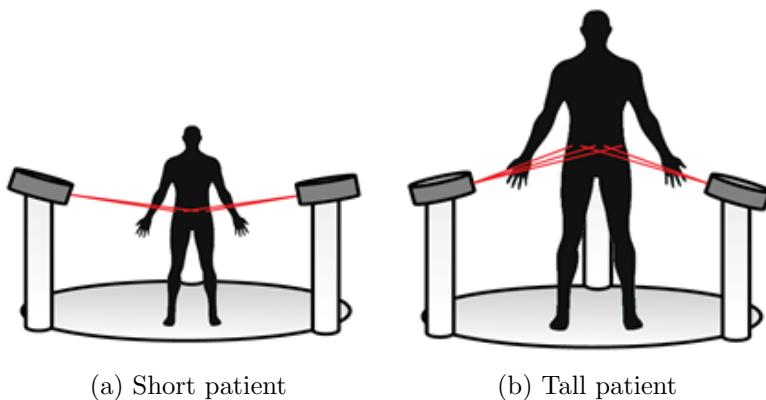


Figure 1.4: Illustration of the LIDARs tilting for patients with different statures

Nevertheless, I have noticed some limitations to this system:

- The base's width could be very big and may be cumbersome for several reasons:
 - most common LIDARs cannot measure very close objects (15-20 cm)
 - it is round shaped
 - it may be inaccurate to yield the body contour of high or low stature patients as the LIDARs must tilt a lot, thus, the greater the distance, the better the accuracy
- Because of the use of 3 sensors that aim at the same target, the precision of the three poles in the 3D space matters as a shift of few millimeters of one of the LIDAR or servomotor will change the final reconstructed body shape

This solution has been kept for me to work on it as it shows better chance of success than the previous one. A picture of the prototype of the top of a single pole is shown in the figure 1.5, where we can see a LIDAR mounted on a servomotor.

1.1.5 Software requirements

Considering the mechanical part of the system built-up and functional, a big part of the project consisted in developing a program that controls the sensors and actuators to perform a scan, process the raw data and find the circumference of the user's waist. Let's explore deeply the expectations of the software. First, the software must give a reliable value of the circumference



Figure 1.5: Prototype of a pole with a LIDAR mounted on a servomotor

and thus, should be robust. By robust, I mean accurate even with noisy data points sensed, but also to the human behavior and its various anthropomorphic sizes and colors. Indeed, during a scan that can last seconds, the user is likely to move more or less and, in the case he uses the machine while wearing clothes, it brings even more uncertainty. As a fully automatic software, it should be able to detect unexpected measurements or conditions. Furthermore, the processing should be fast enough to not be inconvenient. NECTEC being a governmental institution, the modules used must be open-source in order to deploy the system regardless to their license or owner. Last but not least, the cost remains significant, so the controller used for processing must be as cheap as possible.

1.1.6 Internship objectives and scope

The main goal of my work was **to prove the ability of such a system design to recover the user's waist circumference** with the hardware as described in the section 1.1.4 and with the specifications given in the sections 1.1.3 and 1.1.5. To do so, a scope of duties regarding this project was given to me as follows:

- Being able to control the servomotors and the LIDARs
- Scanning the body of a patient and recovering the data (body shape in space)
- Studying the literature about the shape extraction and reconstruction in order to compute the waist circumference
- Implementing such an algorithm
- Transferring the results to a computer through a serial communication
- Optionally, detecting the navel's position in order to measure the waist circumference at its level
- Writing a technical report for skill and knowledge transmission

The chart in figure 1.6 is derived from the previous scope of work. It shows the successive steps that I followed to make this project. This report will describe my work on the steps *A* to *E* in the chapter 2 while the steps *F* and *G* will be discussed in the chapter 3. The part *H* and *I* are not relevant and won't be discussed in this report (even though the internal technical report can be found on the Internet at the address given in the section 2.6).

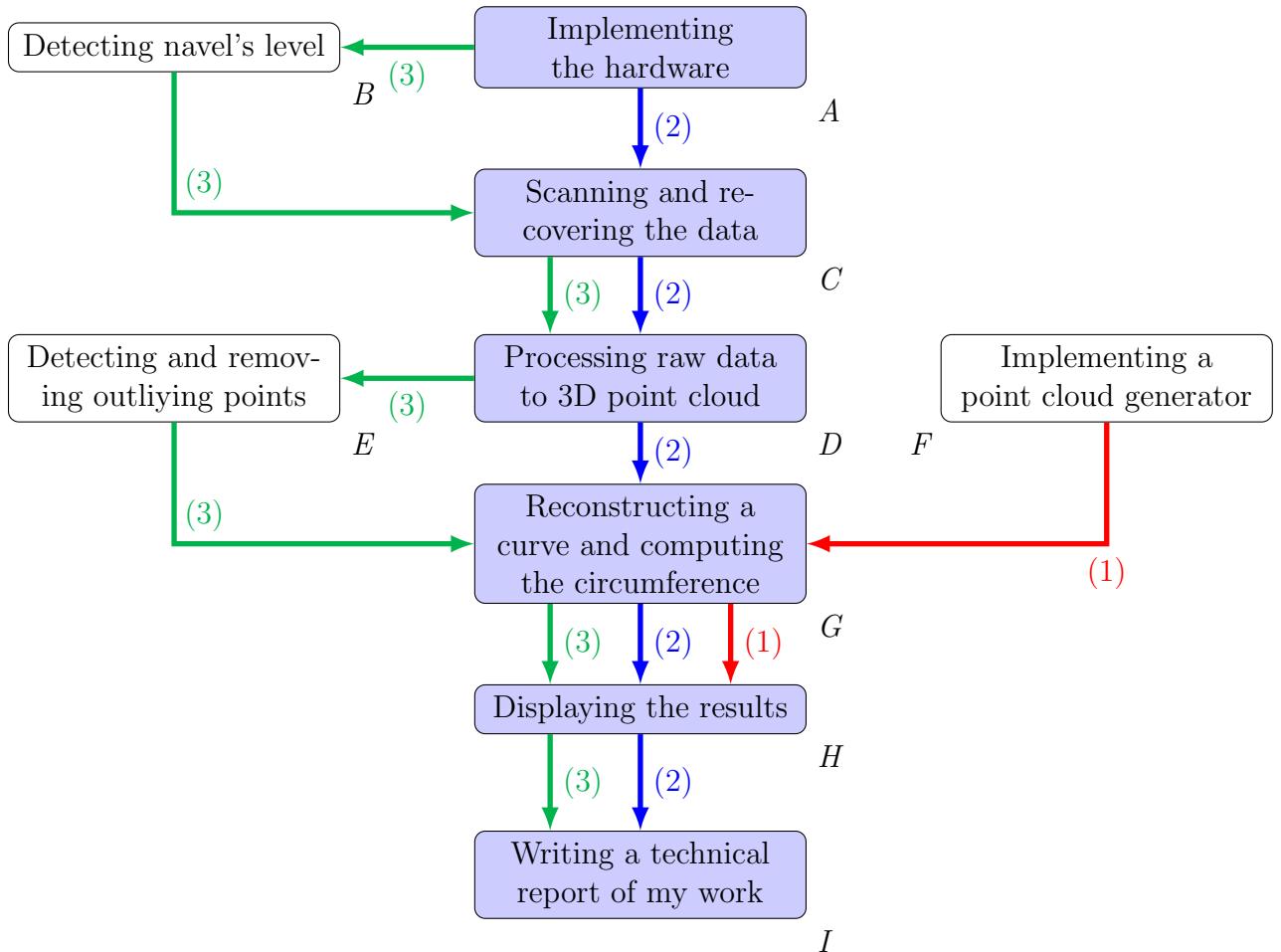


Figure 1.6: Flow chart of my internship's tasks on the automatic health checkup project

The main steps required to tick my internship as done are shown in **blue** while those optional are displayed in **white**. Moreover, the three coloured paths indicate the order of accomplishment of each task. **(1)** is the minimum requirement of my internship and was used while waiting the prototype of the system to be ready. **(2)** follows the scope of work given to me, and **(3)** follows the scope of work including several optional improvements for robustness and usage convenience.

1.2 Orientation measurement system

This section relates about a project which aimed to retrieve and display the orientation of an IMU in order to improve precision for surgery purpose, especially in the case of screws implantation. It describes the specifications of the project and the work that was assigned to me.

1.2.1 Context

Many surgical treatments require a thoracic, lumbar or sacral screw implantation. This requires high precision as the angle of insertion must be exactly as wanted after studying the x-ray pictures. Nowadays, this is performed manually by the surgeon, after verification with measurements tools, and he has to keep the same orientation all along the perforation. Jost et al. describe in a paper [3], an IMU assisted method to improve the precision compared to the free-hand technique. Based on this study, Dr. Surangsirat would like to test it and maybe improve it with the help of a surgeon.

1.2.2 Project's specifications

The first step is to implement a system that could replicate the one presented by Jost [3]. It consists in an IMU, positioned onto of a screwdriver as shown in figure 1.7, that transmits the values of its sensor to a software, which is able to compute the attitude of the sensor from the raw data (acceleration, angular speed and magnetic field's intensity) to an human-readable form, typically Euler Angles. Then, the attitude must be displayed to an user interface.



Figure 1.7: Common screwdrivers for surgery purpose

Moreover, the system, being a prototype, has to be cheap, easy to replicate and easy to use. Many orientation sensors with magnetic field distortion and temperature compensation, power consumption optimization, high precision attitude processing and other fancy features, are available on the market. However, those solutions are very expensive (at least 5000 USD) and do not match with our expected usage. Thus, a simple *MPU9250*, which is a non-expensive 10 Degrees Of Freedom (DOF) integrated chip has been used.

1.2.3 Objectives

The scope of work that was given to me can be described as follows:

- Recovering the raw data from the IMU
- Transmitting the data to a computer
- Computing the raw data into a human-readable form
- Programming an user interface to display the orientation and to manage the communication
- Writing a report of the work done

Implicitly, a review of the literature was needed to improve the orientation processing.

Chapter 2

Automatic health checkup system - Scanning and recovering data points

This section will describe the system architecture and the hardware implementation, the body scan procedure, the work done to detect the user's navel position, the transformation from raw LIDARs measurement to a point cloud, and finally the method used to remove outlying points.

2.1 System's architecture and hardware implementation

This project uses a Raspberry Pi v3, 3 RpLidar A1, and 3 AX12 servomotors as described in the figure 2.1.

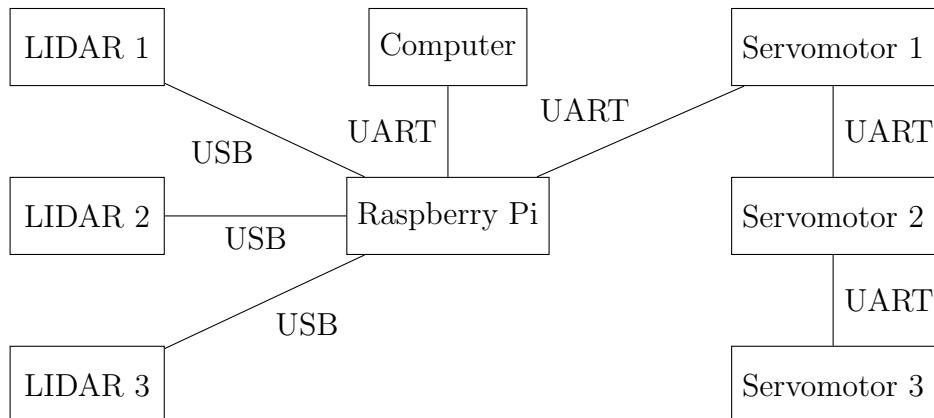


Figure 2.1: Basic architecture of the developed system

2.1.1 Raspberry Pi

A Raspberry Pi is used as the main core of the system as it recover the sensors' data, controls the actuators, runq the algorithm described in the chapter 3 and send the results to a computer. This microcontroller has been chosen for obvious reason of cost, computation speed, simplicity (many GPIO, communication ports, remote GUI through SSH...). All the code of this project is written in Python 3.6.

2.1.2 LIDARs

The open source library: <https://github.com/Roboticia/RPLidar> is used to control the LIDARs. While scanning, the 3 LIDARs gather the data alternately. They just need to be

plugged to the Raspberry via USB.

2.1.3 Servomotors

The project uses the following open source library to control the servomotors: <https://pypi.org/project/pyax12/>. The AX12 servomotors can be used in a daisy chain configuration. They have only 3 pins (2 for POWER and 1 for UART), which means that the RX/TX signals have to be multiplexed in order to establish the communication. Thus, a simple circuit was designed as shown on the figure 2.2. Here, a 74LS240 tri-state buffer is used. It allows to control the direction of the communication with a single GPIO from the Raspi. When it is high, the Raspi can send data, when it is low, it receives data.

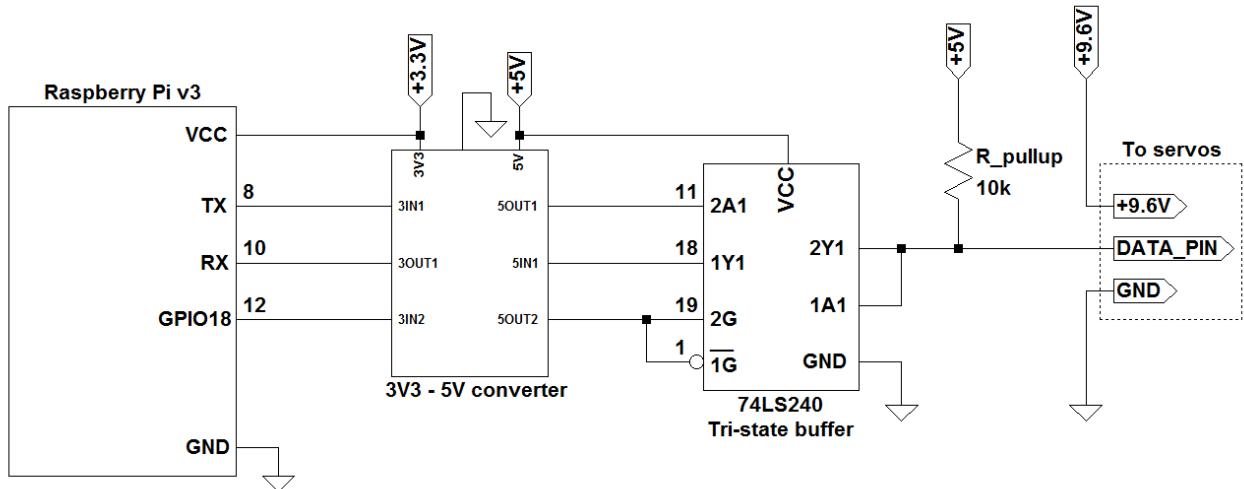


Figure 2.2: Electronic circuit diagram for the servomotors' control

This kind of servomotors is quite powerful as many parameters can be written in their EEPROM in order to control them safely and easily. I have written a program able to communicate with them, to configure them (setting their baudrate via brute-force, their identification number, their maximum and minimum angles of rotations...) and to test them.

2.2 Finding the user navel's height

As the WHO said, the waist circumference must be measured at the navel's level. Two different approach could be used to find the height of the user's navel: scanning the whole or a part of his body and detecting it, or estimating it through a statistical analysis.

2.2.1 Detecting the navel's height with a body scan

Before implementing the detection, we need to know the degree of accuracy of our sensors and actuators. First, the datasheets of the LIDARs and servomotors give:

- Servomotors' resolution: $\frac{330}{1024}^\circ = 0.322^\circ$
- LIDARs' distance resolution: $< 0.5 \text{ mm}$
- LIDARs' angular resolution: $\leq 1^\circ$

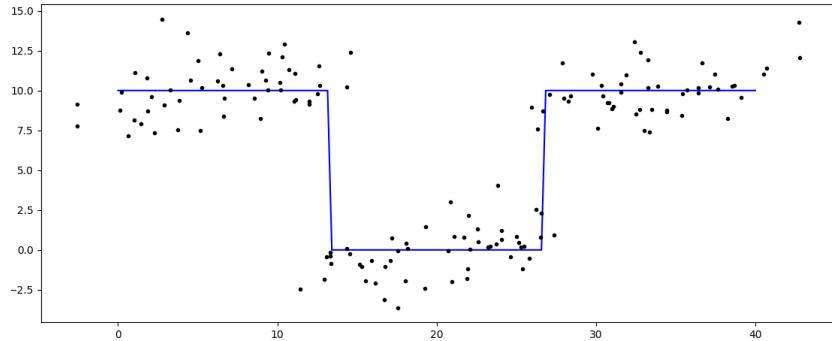
We will consider the uncertainty of measurement of the servomotors' position and the LIDARs' distance as negligible. However, let's compute the uncertainty u of the LIDARs' angle measurement:

$$u = \frac{\text{angular_resolution} \times 0.5}{\sqrt{3}} \text{ deg} = \frac{0.5}{\sqrt{3}} \text{ deg} \quad (2.1)$$

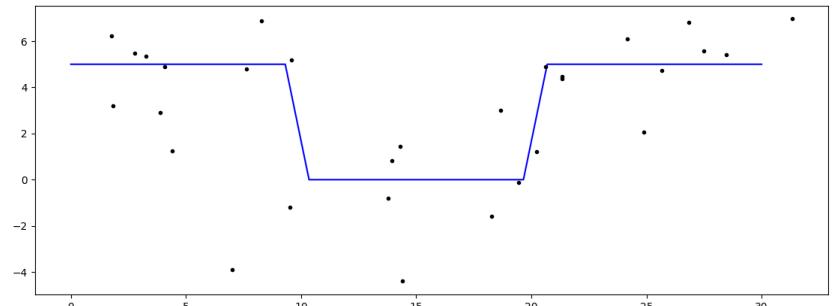
We choose to divide the half of the angular resolution by $\sqrt{3}$ as the measurement is likely made by a rotary encoder, giving measurements following an uniform probability distribution. Let $v = [m, 0]$ be the measured vector in 2D space with m the measured distance by the LIDARs, the uncertainty of position u_p is given by:

$$u_p = \sqrt{[m - m.\cos(u)]^2 + [-m.\sin(u)]^2} \quad (2.2)$$

Then, the expanded uncertainty $U_p = k.u_p$ with k the coverage factor. $k = 1.96$ for a level of confidence of 95%. Assuming for example, that the user is 500mm away from the LIDAR ($m = 500$), then $U_p = 5.0\text{mm}$. Simulation have been made and are shown on the figure 2.3.



(a) Favourable conditions (width=10mm, depth=10mm, 150 data points)



(b) Bad conditions (width=5mm, depth=5mm, 30 data points)

Figure 2.3: Simulations of the user's navel detection
In blue, the simulated vertical slice of an user's navel.

The main problems with detecting the users' navel are cited below:

- The characteristics of each navel are highly different for each person
- If the navel is too small or not deep enough, the data become too noisy (refer to figure 2.3b)
- It requires a lot of time to scan a part of the body with LIDARs with the scanning procedure described in section 2.3

- If the patient is short or tall compared to the pole's height, then the navel is not aligned horizontally with the LIDARs
- It can cause some user inconvenience because he must not move at all and must not inflate his waist while breathing during the detection

For these reasons, the detection algorithm has not been implemented.

2.2.2 Estimating the navel's height

The other approach is to estimate the navel's height using statistics, based on other anthropomorphic parameters such as stature, weight and age. Two open databases have been used for this purpose: one for children from the *United States Consumer Product Safety Commission* [4] and another from the *United States Army* [5] for adults. They both gathered data from both genders. The figure 2.4 shows the thousands of measurement and the height of the navel depending of the patients' stature (height), their weight and age. Other anthropomorphic data could be used for such an estimation but they are hard to measure by the automatic health checkup system specified by our project.

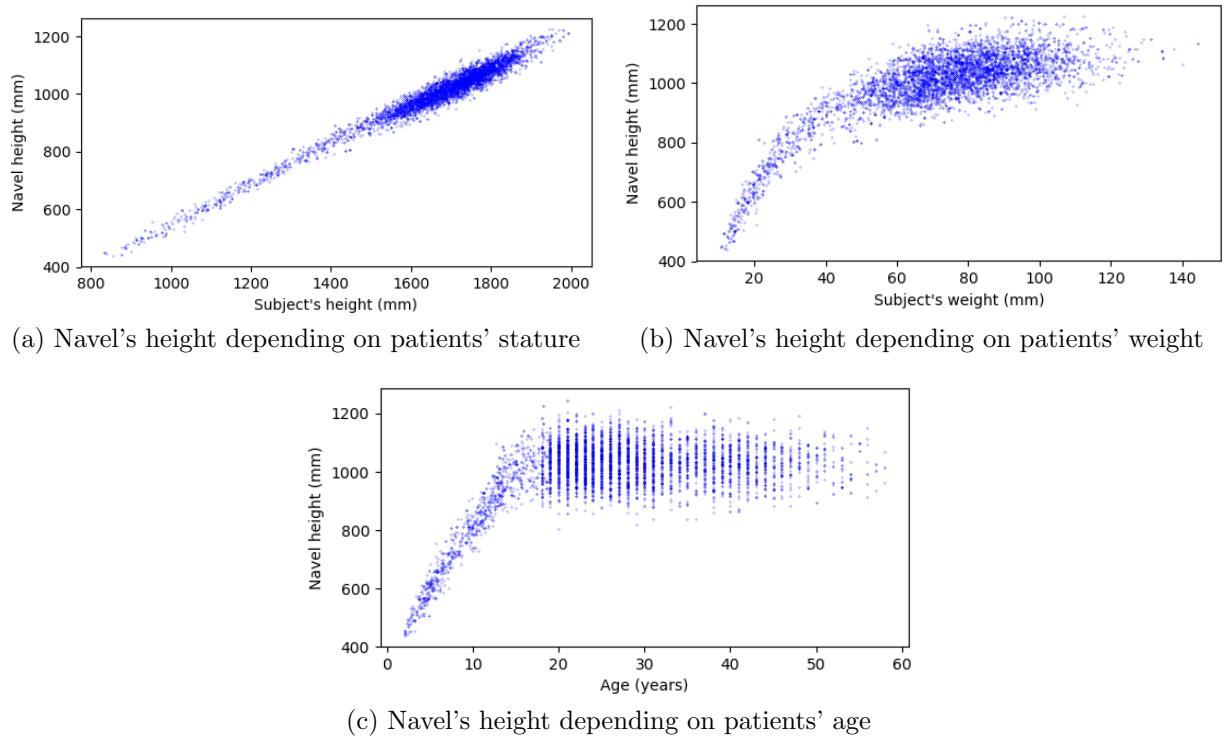


Figure 2.4: Patient navel's height depending on their stature, weight and age

Intuitively, we can notice a correlation between the navel's height and each of the parameters. If we compute the correlation coefficient r described in the equation (2.3), we get the table 2.1.

$$r = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (2.3)$$

X and Y are the variables and σ_X and σ_Y their standard deviation. Moreover $r \in [-1, 1]$. As the 4 variables are all pretty well dependant to each other (r close to 1), a multi-layered neural network has been tested (provided by the *sckit-learn* library) to estimate the navel's height. Moreover, as the correlation coefficients of the navel's height with the weight and stature

Correlation (%)	Stature	Weight	Age
Navel's height	98%	79%	50%
Stature	/	84%	54%
Weight	/	/	61%

Table 2.1: Correlation of various anthropomorphic data among each others

is higher, a 2D linear regression has been implemented. Finally, a simple linear regression between the navel's height and stature has been tested. By splitting the set of data into two groups, we can use a first part to train the regression while the other part is used to test the regression. The error $\epsilon\%$ of the estimated navel's height for each test data is computed as follows:

$$\epsilon\% = \frac{|\text{estimated} - \text{true}|}{\text{stature}} \cdot 100 \quad (2.4)$$

The error is thus given as a fraction of the patient's stature. The errors found for each 3 methods are given below:

- Multi-layer perceptron: 1.21%
- 2D linear regression stature/weight: 1.04%
- Linear regression on the stature: 0.94%

As the simple linear regression shows the best results, I used this method. The error terms of a set of test are shown in figure 2.5 which is a distribution graph.

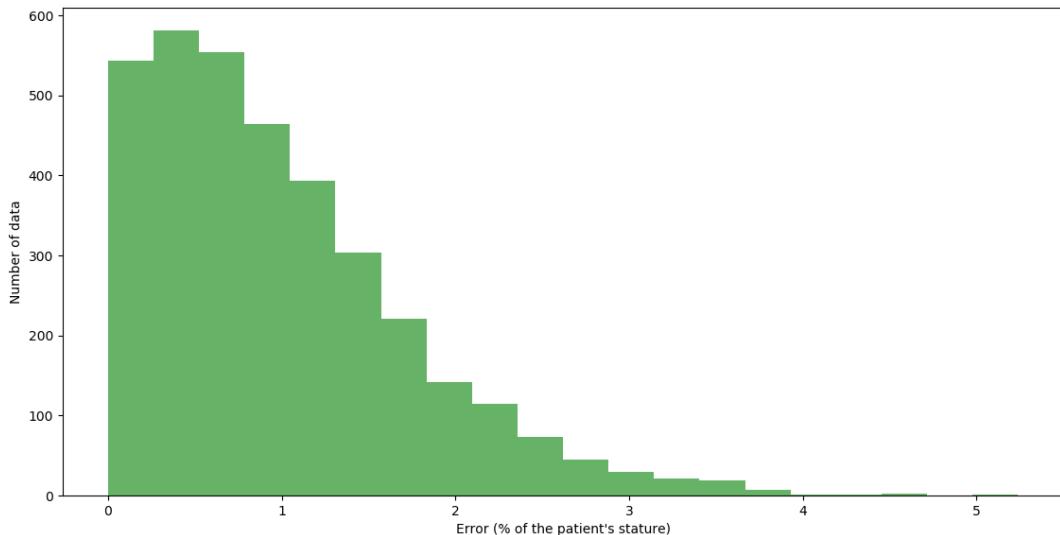


Figure 2.5: Distribution of the estimated navel height's error

All the results are presented in the table 2.2 and are interpolated for people 140cm, 160cm and 180cm tall. With this table, we can say that this estimation gives quite good results as for most of the patients, the estimated value won't be more than few centimeters away from the truth. Note that the databases used are not sampling a good representation of the population (soldiers and American children). Some other sets of data more adapted to the global population might be found.

	Fraction of the stature	140cm tall	160cm tall	180cm tall
Mean	0.94%	13.16mm	15.04mm	16.92mm
Median	0.83%	11.62mm	13.28mm	14.94mm
75-percentile	1.39%	19.46mm	22.24mm	25.02mm
95-percentile	2.40%	38.40mm	38.40mm	43.20mm
Max error	5.24%	73.36mm	83.84mm	94.32mm

Table 2.2: Error of the estimated navel's height
(in fraction of the stature and for patients with different statures)

2.3 Scanning procedure

Let's describe the procedure used to recover a set of point cloud representing the body contour.

2.3.1 Notations

Since now and until the end of the section 2.3, the notations described below will be used.

The figure 2.6 is a top view of the system. The origin of the system frame in **blue** is the center of the base, represented by the black circle. Since the LIDARs lie in a circle with D as radius, the angle $\{\beta_1, \beta_2, \beta_3\}$ are the angles of the LIDARs from the origin vector X . In our case:

$$\begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \end{bmatrix} = \begin{bmatrix} -\frac{\pi}{6} & -\frac{5\pi}{6} & -\frac{9\pi}{6} \end{bmatrix}$$

Each lidar has its own local frame represented in **gold**, and α is the angle between the Y vector of the local frame (oriented to the center of the circle) and the measured laser beam. m is the distance between the laser beam source and the impact. The red zone inside the circle is a non-measurable area. Indeed, the LIDARs need a minimum distance to be able to perform a measurement. Moreover, the more this red zone is large, the wider the range of reachable scan heights. d is the radius of the non-measurable area. Note that in this case, $\alpha < 0$ and $\beta < 0$.

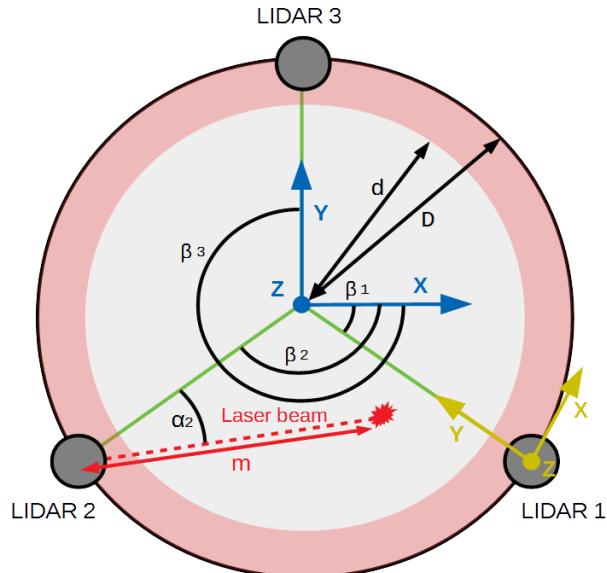


Figure 2.6: Top view of the system

The figure 2.7 shows a side view of the system. h is the height of the lidar relative to the base. It is important because the origin of the altitude is at the base level. We can see that a

servomotor is tilting the LIDAR and the angle between the zeroing (LIDAR in flat position) and the tilting is called θ . Note that again in this case, $\theta < 0$.

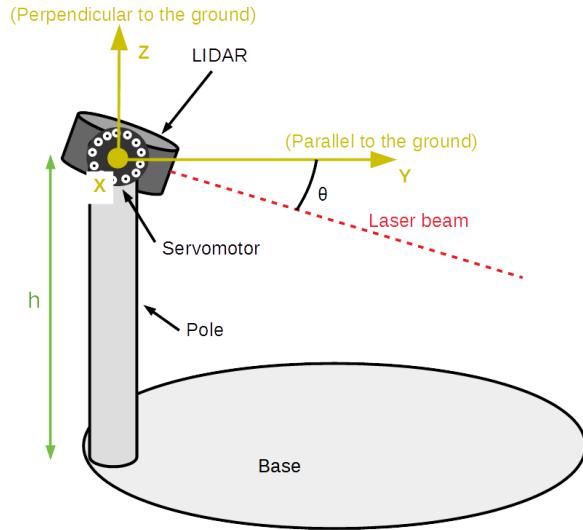


Figure 2.7: Side view of the system

2.3.2 Scanning procedure

Let's assume in this section that we want to perform a scan at an height \tilde{h} at the navel's level found with the procedure described in the section 2.2.2. The figure 2.8 intuitively shows the main problem that occurs when the LIDAR is above or below the desired part of the body to scan. If we assume the human body as a cylinder, we can see that the LIDAR will measure the body at different heights instead of the one desired. To overpass this issue, a protocol has been implemented. It consists basically into tilting the lidar at many different angles, and to flatten the data measured close to the height \tilde{h} .

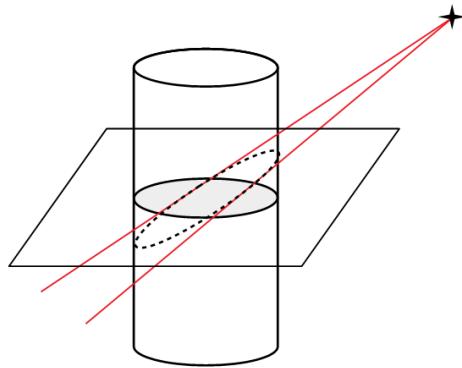


Figure 2.8: Cylinder cut from an higher source

The figure 2.9 is a side view of the system, showing the LIDAR mounted on the pole on the left and the base at the bottom. The algorithm 1 in the appendix B describes the different steps of the scanning process. The figure 2.9 shows a tolerance zone around the desired height measurement \tilde{h} at levels $\tilde{h} + h_t$ and $\tilde{h} - h_t$. All this tolerance zone should be covered horizontally by the set of scans. By reading the figure 2.9, the process is as follows:

1. Compute P_1 , the intersect of the tolerance surface $\tilde{h} + h_t$ vertically and the beginning of the measurable area horizontally.

2. Compute the intersection of a laser beam passing through P_1 which would intersect the inferior tolerance surface at level $\tilde{h} - h_t$.
3. Vertically project the point crossing the laser beam and the inferior tolerance surface on the superior tolerance surface to get P_2 .
4. Repeat the process from the point P_2 since the step 2 while $P_n < 2.D - (D - d) = D + d$.
5. Scan the body with the angles that make the laser beam passing through the points P_i found.

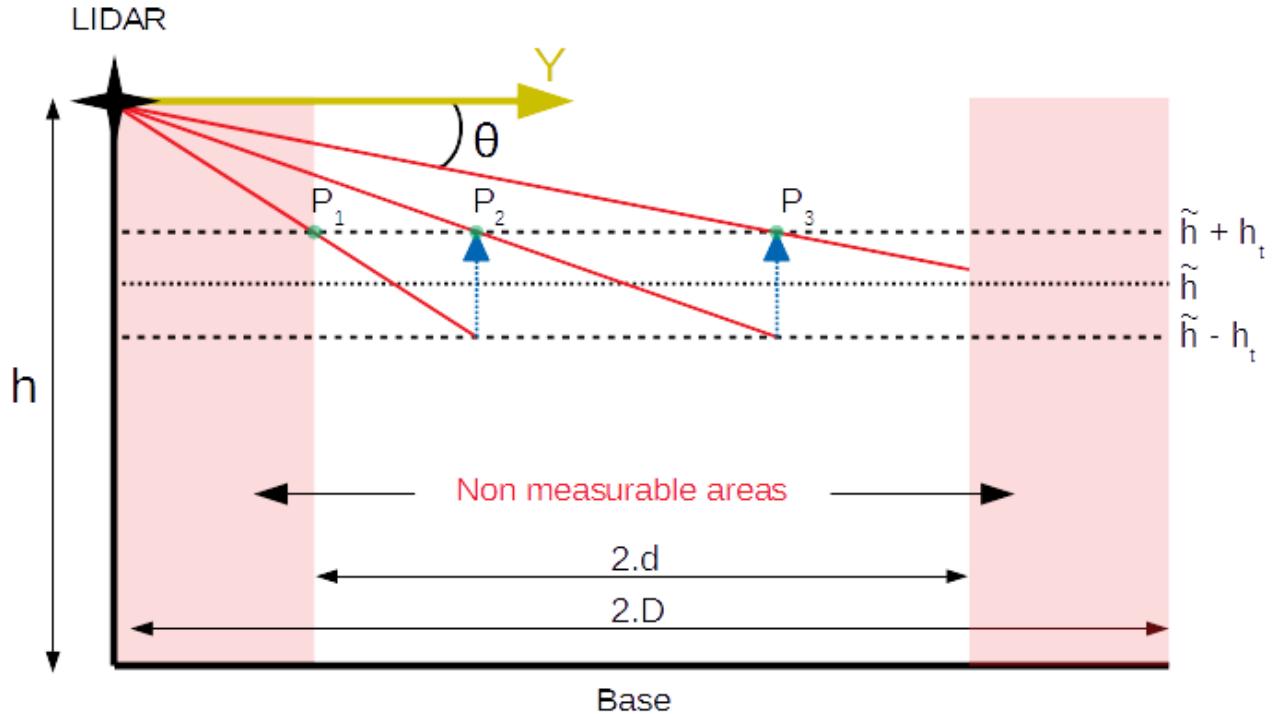


Figure 2.9: Side view of the system with scanning procedure explanation

2.3.3 Running a scan

The 3 LIDARs have to perform their scan all in the same time. However, only one thread is running at a time. It means that the program gathers the measurements one by one on each LIDAR. Moreover, the servomotors are all moving at the same time to avoid too much distortions if the users moves during the scan. Once the Raspi has yielded the measurements, it saves them into text files containing the angle of rotation of the lidar α , the angle of tilting of the servomotors θ and the distance from the measured object m , giving a file containing:

$$\begin{bmatrix} \alpha_j & \theta_j & m_j \end{bmatrix}_i$$

with i number of LIDARs and j number of measurements. Another file contains the scan's information (for now, only the height \tilde{h}).

2.4 Raw data to 3D point cloud

Once the scan is over, 3 text files and 1 info file have been generated. The goal now is to locate every measurement in the 3D space. Each measurement is a 3D vector, having the LIDAR's

position as origin. Let's find the orientation of the vector for each measurement of the LIDAR i . m is the norm of the vector and is the distance measured by the LIDAR. In the LIDAR's frame as shown in the figure 2.6 and figure 2.7, let's assume the vector $v_{0,j} = [0 \ m_j \ 0]$, which mean a vector having the LIDAR as origin and pointing forward. Now we want to rotate it with the rotation matrix given in the Slabaugh's article [6]. The elemental rotation matrices around the axis X and Z can be expressed as follows:

$$R_X(\theta_j) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_j) & -\sin(\theta_j) \\ 0 & \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \quad (2.5)$$

$$R_Z(\alpha_j) = \begin{bmatrix} \cos(\alpha_j) & -\sin(\alpha_j) & 0 \\ \sin(\alpha_j) & \cos(\alpha_j) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Moreover, we need to get the final vector in the system's frame. To transform the LIDAR's frame into the system's frame, we will rotate the vector around Z-axis by $\beta_i + 90^\circ$ (to match the X and Y axis), then rotate it around the X-axis by an angle θ_j and rotate it again around the Z-axis by the value of the measurement angle α_j . Thus, the final rotation matrix can be written by combining (2.5) and (2.6):

$$R_{final}(\beta_i, \theta_j, \alpha_j) = R_Z(\beta_i + 90^\circ) \cdot R_X(\theta_j) \cdot R_Z(\alpha_j) = \begin{bmatrix} c_1c_3 - c_2s_1s_3 & -c_1s_3 - c_2c_3s_1 & s_1s_2 \\ c_3s_1 + c_1c_2s_3 & c_1c_2c_3 - s_1s_3 & -c_1s_2 \\ s_2s_3 & c_3s_2 & c_2 \end{bmatrix} \quad (2.7)$$

For readability purpose, c means \cos and s means \sin , while 1 correspond to $\beta_i + 90^\circ$, 2 to θ_j and 3 to α_j . For instance, $c2$ means $\cos(\theta_j)$. We obtain:

$$v_j^T = R_{final}(\beta_i, \theta_j, \alpha_j) \cdot v_{0,j}^T = m_j \cdot \begin{bmatrix} -c_1s_3 - c_2c_3s_1 \\ c_1c_2c_3 - s_1s_3 \\ c_3s_2 \end{bmatrix} \quad (2.8)$$

So, v_j is the vector in the system frame having the system origin as starting point. A simple shift can be applied to it depending on the position of the LIDAR. The point P_j is the 3D measurement j made by the LIDAR i :

$$P_j = v_{0,j}^T + \begin{bmatrix} D.\cos(\beta_i) \\ D.\sin(\beta_i) \\ h \end{bmatrix} \quad (2.9)$$

It could be enough in theory but practically, the system looks more like presented on figure 2.10. When the LIDAR is tilted, its position changes.

Let's add to our equation (2.9) a position correction with \hat{d} the distance between the center of rotation of the servomotor and the LIDAR.

$$P_j = v_{0,j}^T + \begin{bmatrix} D.\cos(\beta_i) \\ D.\sin(\beta_i) \\ h \end{bmatrix} + \begin{bmatrix} \hat{d}.\cos(\beta_i).(1 - \cos(\theta_j)) \\ \hat{d}.\sin(\beta_i).(1 - \cos(\theta_j)) \\ \hat{d}.\sin(\theta_j) \end{bmatrix} \quad (2.10)$$

The figure 2.11 shows an example of scanned 3D cloud. Once such a point cloud has been obtained, we keep only the point located at the heights $[\tilde{h} - h_t, \tilde{h} + h_t]$ as described in the figure 2.9 and near the center of the base. It gives us the contour of a slice of the user's waist.

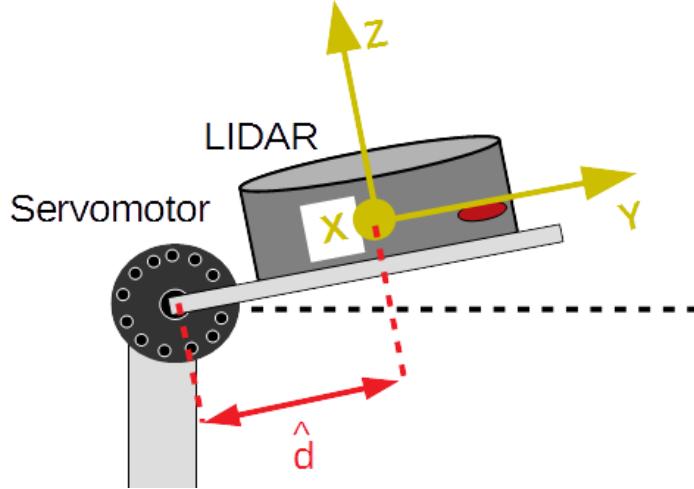


Figure 2.10: Mechanism of the LIDAR being tilted

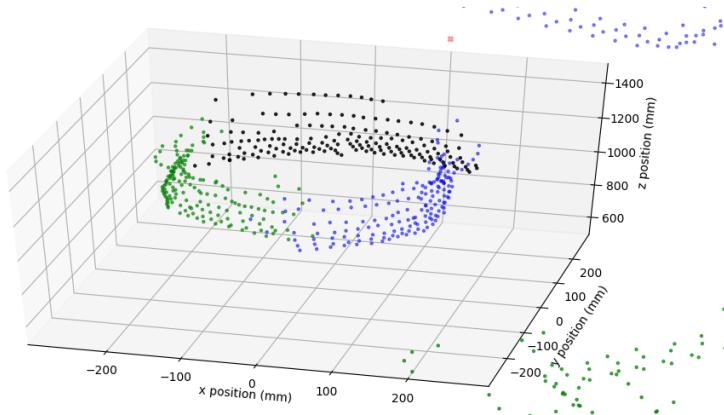


Figure 2.11: Example of a 3D point cloud
Each color corresponds to a different LIDAR

2.5 Detection of outlying points

Removing aberrant points in the cloud is an important feature if we want to implement a curve reconstruction lying in those points. I have tested several solutions that will be described in this section.

2.5.1 Outlier detection via minimum spanning tree

The first algorithm tested is a detection using a Minimum Spanning Tree (MST) which is based on clustering the data according to their local density. It comes from the work of Tang et al.'s paper [7]. A MST is an undirected graph that links all the vertices together with the lowest sum of the edges' weight. Here, we use the Euclidian distance to weight the edges as it seems to be more coherent since we work in a 2D space. The Python library *scipy* provides an optimized algorithm to solve the MST of a graph. However, computing the MST of a graph where all the vertices are connected to each other is quite hard when there is more than a few points as the number of edges has a complexity of O^2 . Thus, a graph where only k closest neighbours of each points are connected together is given to the MST algorithm. Then, by following the

process given in the Tang's paper [7], we rank the edges by distance, compute the mean and standard deviation, and cut the edges that are too far from the others according to a normal distribution (a threshold of 95% can be chosen for instance). Finally, we can keep the clusters that are made of a minimum number of points (10 for instance). In our problem, we want only one cluster, thus, we only keep the biggest one. The figure 2.12a shows an example of a 2D point cloud that is supposed to represent the shape of the user's waist, where we can clearly see that 3 outliers need to be removed on the top left corner. The MST is drawn the figure 2.12b. We can notice that with a threshold of 95%, the only edge that is considered too long (in red) leads the point cloud to these outliers. By removing this edge, we get two clusters and by keeping only the biggest one, the 3 outliers are then removed.

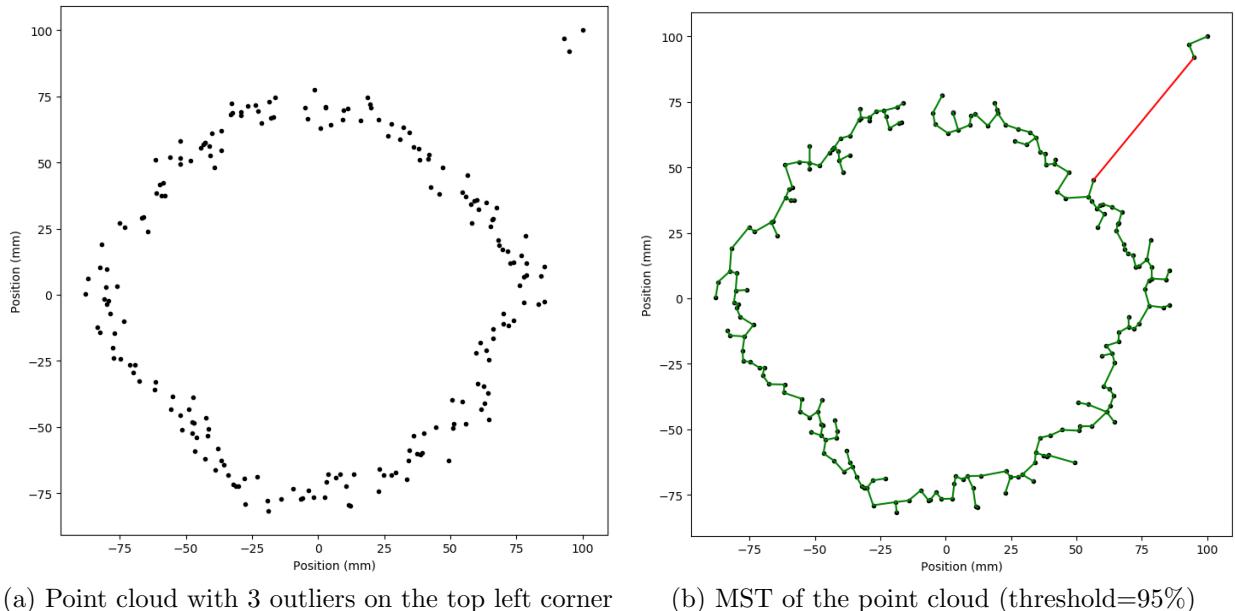


Figure 2.12: Example of a detection via MST on a point cloud

In green are shown the edges of the MST short enough and in red those considered as too long compared to the others.

This method is quite powerful, but, in the case the cloud's density is not homogeneous everywhere (problem described in the Tang's paper [7]), false positives can be detected. This problem is shown with the example in figure 2.13 where a point cloud has various point's density (low in the bottom right side of the shape). In that case, two point that clearly lie in the body's shape will be removed. This can cause issues as those 2 points are needed to describe this side of the shape (there are the only one in the local area) and it can lead to a loss of information. We thus need to increase the specificity of our detection (or reduce the sensibility) by increasing the threshold value. Then the false positive rate will be decreased, while increasing the false negative rate. Thus I implemented another method shown in the section 2.5.2 to solve this issue.

2.5.2 Outlier detection via Gabriel's graph

As we saw in the section 2.5.1, the outlier detection via MST requires a well optimized threshold value. A Sotoodeh's paper [8] describes a new outlier detection that is basically made of two steps: a rough filtering with a MST as in the previous method, followed by a fine filtering made with a Gabriel's graph. A Garbriel's graph is an undirected graph where a circle having as diameter the edge is drawn for each edge of the graph. The edges having their circle that

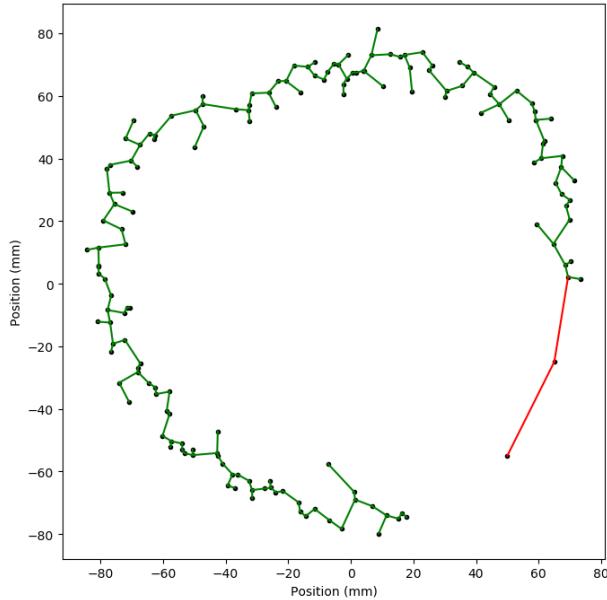


Figure 2.13: Example of a detection via MST on a point cloud with not homogeneous density

contains another point are removed from the graph. The figure 2.14 explains how to transform a graph to a Gabriel's graph.

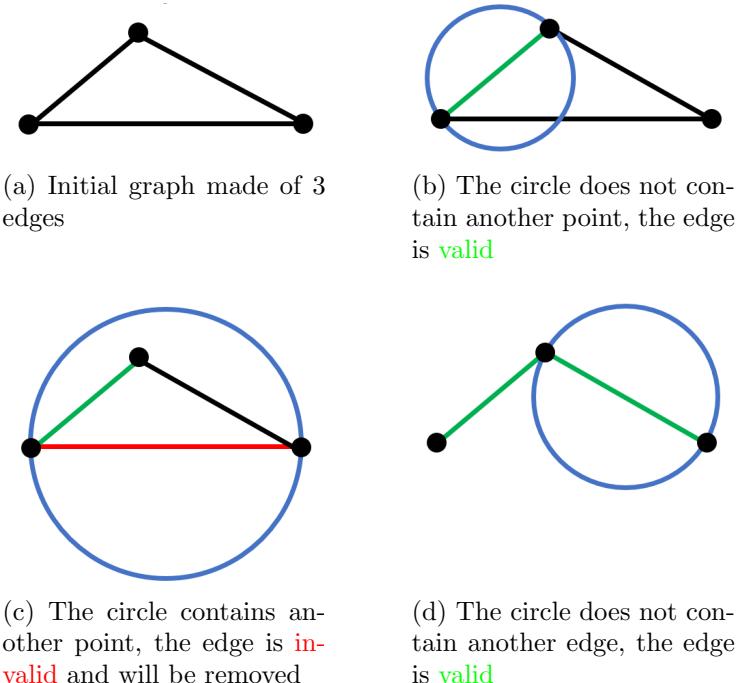


Figure 2.14: Example of process used to get a Gabriel's graph

According to Matula et al.. [9]: *"The Gabriel graph on a vertex set V is a subgraph of the Delaunay triangulation for V . Furthermore, the edge AB of the Delaunay triangulation is an edge of the Gabriel graph iff the straight line joining A to B intersects the boundary line segment common to the Thiessen polygons for A and B at a point other than the endpoints of that boundary line segment"*. I used this definition to implement the Gabriel's graph transformation as the Voronoi diagram and Delaunay triangulation can be found easily with Python libraries. Finally, we obtain a graph that looks like the one shown in figure 2.15. Then, the length of each edges is compared to the others in the same way as in the method with MST described

in section 2.5.1. However, this method could not solve the issue brought by the method with MST as the specificity/sensitivity is not satisfying enough (too many false positives and false negatives).

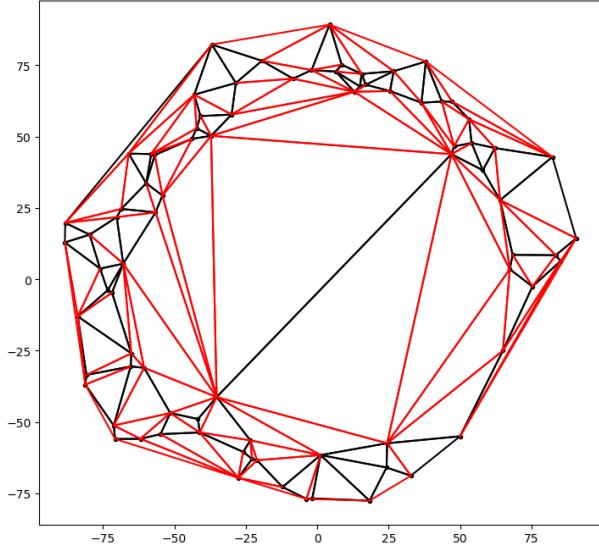


Figure 2.15: Example of a Gabriel’s graph

This is a Delaunay triangulation where the **red** edges are invalid as shown in figure 2.14c and the **black** edge are the Gabriel’s graph.

As these two previous methods for outliers’ detection were not working as expected, I finally implemented the detection in another way during the curve reconstruction. Thus, this new method is described in the section 3.5.4.

2.6 Source code

The technical report and the source codes of the project can be found as a git repository *Github* at the following URLs: www.github.com/Vincema/lidar_waist_scan and www.github.com/Vincema/navel_height_prediction.

Chapter 3

Automatic health checkup system - Reconstructing a curve and computing the waist circumference

Once the set of data in a 2D point cloud is recovered (or generated via the point cloud generator presented in section 3.1), which is supposed to describe the shape of the user's waist, we need to compute a curve reconstruction in order to approximate the real body shape. This section shows the methods I have found and tried in order to make this possible. Finally, it will explain how the waist circumference, in a real physical quantity, has been computed.

3.1 Point cloud generator

In order to test the algorithm and while the system was being built, a generator of point cloud has been implemented. It consists into drawing a circle with a few points, and randomizing their position around a perfect circle. Then, points are generated all around the curve following a normal law. The figure 3.1 shows different shapes generated for testing purpose. Various parameters like the standard deviation errors of the data points and shape points and the number of edges are tested in those examples.

3.2 Research and choice of curve reconstruction methods

At the beginning of my internship, I made a state of the art of the algorithm developed in order to reconstruct a curve from a set of data points. This topic is widely covered as it is used in computer vision, to feed artificial intelligence algorithms, to perform topographic reconstructions or for industrial purpose by detecting defects on manufactured products, etc.

First, we consider our data as an **unorganized** set of points. In this case, "organized" means that the relative position of the point to each other in the considered shape is known. Thus, two types of solutions can be implemented:

- Using a regression algorithm that does not need to be fed by an ordered set of points and which are quite complex
- Ordering the points by ourselves as we know what kind of object will be scanned, and performing a regression using a typical regression algorithm

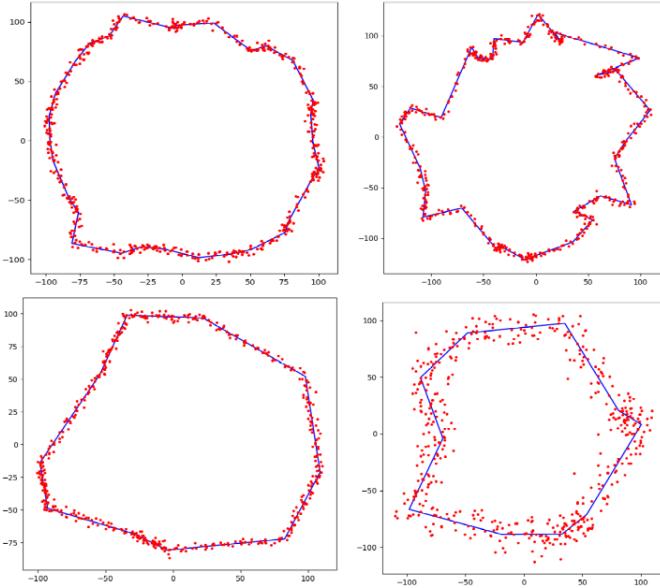


Figure 3.1: Examples of generated testing shapes
The raw shape in **blue** and the data points in **red**

The initial idea from my tutor was to use the LOcally weighted Scatterplot Smoother (LOESS) regression algorithm on a set of points. This method, firstly developed by William S. Cleveland [10], is used to perform a non parametric regression on a large (and dense) set of points. It basically consists into performing a parametric regression of low order for each region of the set of data, based on the local area only, and to concatenate the set of all the linear approximations into one final curve. I tried it and you can find the details, implementation and results in the section 3.3.

A lot of research paper are dedicated to the Active Contour Model or "Snake algorithm", often used for medical image analysis purpose, which consists to minimize the energy of a spline curve, influenced by constraints. This method is mostly used for computer vision on images, but some papers like the Hong Pong et al.'s one [11], shows good results for contour reconstruction on a point cloud. However, this method can be computationally expensive and seems not to be adapted for our embedded microcontroller (a Raspberry Pi).

Else, I have found some documentation on a curve reconstruction via B-spline. Different approach are possible. I have found first a method described by Lin et al. [12] that determines the order of a point cloud with a sequence joining method before fitting a B-spline. I implemented it and the results can be found in the section 3.4. Moreover, another method that Wang et al. suggest in their paper [13], is to fit a set of B-spline curve (parametric) directly into an unorganized point cloud using a specific process called Curvature Based Squared Distance Minimization. Please find the details of the implementation and the detailed results in the section 3.5.

Finally, a paper written by K. Lee [14] suggest an algorithm that orders the data point by thinning them. Then, a simple moving least square smoother would be enough to estimate the shape of the object. This method has not been tested for reasons described in the section 3.4.2.

3.3 Curve reconstruction by LOESS regression

The first curve reconstruction that I used was the LOESS regression. Its concept, implementation and results are presented below.

3.3.1 Concept

The LOESS or LOWESS regression was introduced by William S. Cleveland in a 1979's paper [10]. The principle is to perform a series of low order parametric regressions (typically linear or 2nd order) at several locations of a 1D point cloud, taking into account the local surrounding points to weight the regression coefficients. A condition to the LOESS regression is to have a many and dense data set.

3.3.2 Implementation

Ordering the data points

The first task is to order the data, by converting our parametric 2 axis space into a linear representation. To do such a thing, I considered the human body shape as quite circular and symmetric about the two axis. A circle fitting could thus be a good first approximation of the shape we want to reconstruct. The least square fitting method as described in the Dumbach et al. 's paper [15] is used, computing the radius and position with the equations in the appendix C. Once the position of the circle's center is computed, the points are ordered according to their angle that we can get with the equation:

$$\phi_i = \text{atan2}\left(\frac{y_i - y_0}{x_i - x_0}\right) \quad (3.1)$$

with x_i and y_i the coordinates of the point i and x_0 and y_0 the coordinates of the center of the fitted circle. The figure 3.2a shows how the circle is fitted to a point cloud. Then the distance from the origin of the circle is computed for each point and we obtain the graph of ordered points in figure 3.2b.

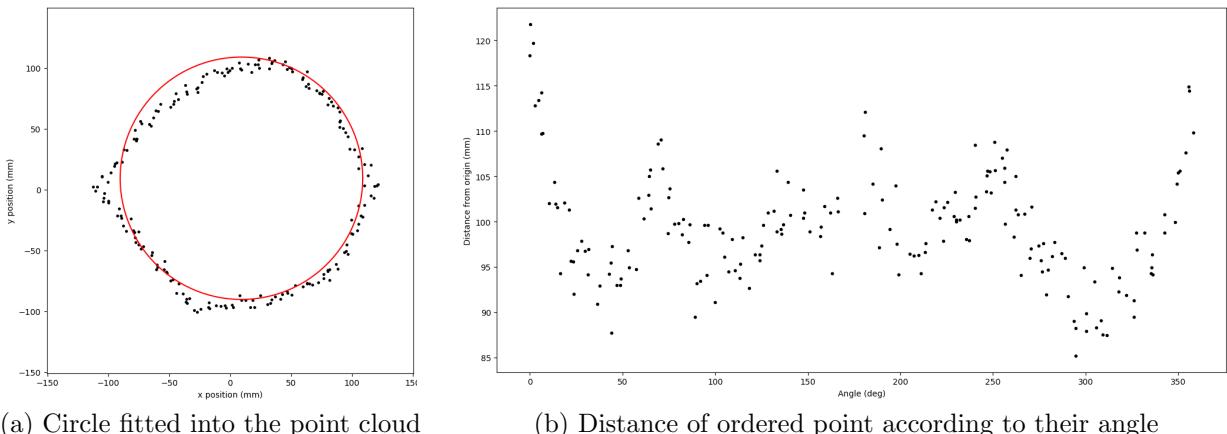


Figure 3.2: Transformation from 2D parametric space to linear representation

Implementing the regression

Let x_i be the point on which we want to perform the local regression. We want to find $\hat{\beta}_0(x_i)$ and $\hat{\beta}_1(x_i)$ the values of the coefficient of the simple linear regression β_0 and β_1 that minimize:

$$\sum_{k=1}^n (\delta_k w_k(x_i)(y_k - \beta_0 - \beta_1 x_k)) \quad (3.2)$$

with n the number of points, y_k the value of the point x_k and $\delta_k w_k(x_i)$ the value of the robustness factor (δ_k) times the weight value (w_k) of the point x_i at the point (x_k, y_k) . Any weight function following several properties described in the article [10] can be chosen. The common function used, and given as example in the reference article, is the tri-cube weight function:

$$\begin{aligned} W(x) &= (1 - |x|^3)^3, & \text{for } |x| < 1 \\ W(x) &= 0, & \text{for } |x| \geq 1 \end{aligned} \quad (3.3)$$

Then, we can write the weight value using the weight function W :

$$w_k(x_i) = W\left(\frac{|x_k - x_i|}{f \cdot h_i}\right) \quad (3.4)$$

with h_i the maximum distance on abscissas between two points (here 360°) to normalize the value among $[0, 1]$. f is the smooth factor and is a very important coefficient as it allows to choose how far from the point x_i , the points x_k are still considered. $f \in]0, 1]$, 1 take into account the whole set of data, 0.5 consider only the points closer than 50% away (180°) from the current point x_i compared to the whole scale to perform the regression. We also have to take into account the cycling of the shape as it is closed. Moreover, we define the bi-square function as follows:

$$\begin{aligned} B(x) &= (1 - x^2)^2, & \text{for } |x| < 1 \\ B(x) &= 0, & \text{for } |x| \geq 1 \end{aligned} \quad (3.5)$$

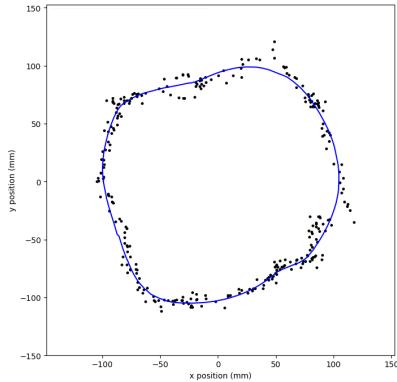
Now, let e_i be the value of the residuals of the current fitted value, $e_i = y_i - \hat{y}_i$ with \hat{y}_i the fitted value of the regression x_i . Furthermore, let's define s as the median of $|e_i|$. It allows us to finally compute the robustness factor with the equation:

$$\delta_k = B\left(\frac{e_k}{6s}\right) \quad (3.6)$$

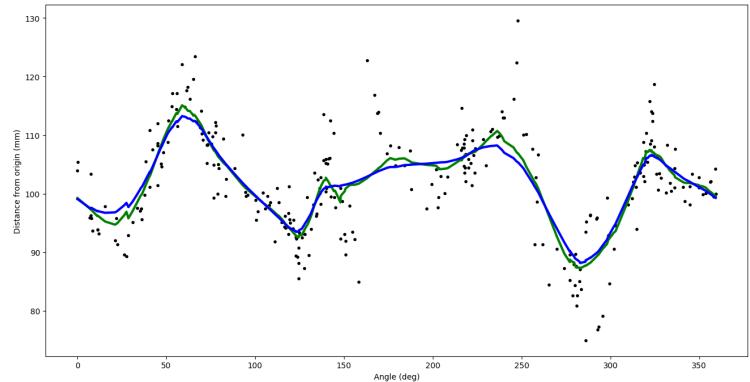
Note that on the first loop, the value of \hat{y}_i is still unknown, thus, we set $\delta_k = 1$. We can now compute \hat{y}_i for each point of the set, and iterating again a defined amount of time (for us, only twice).

3.3.3 Results

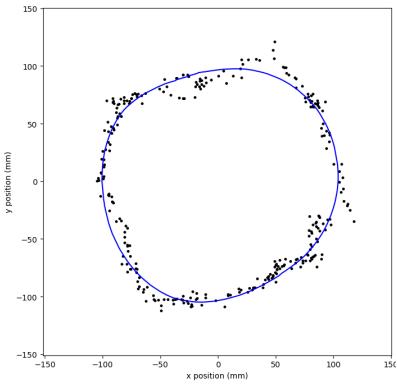
The results are shown in figure 3.3 for different values of the smooth factor f . The greater, the smooth factor, the smoother the curve. The result is quite correct when the smooth factor is low as we clearly notice that the shape is well reconstructed with a smooth curve and robust to the outliers.



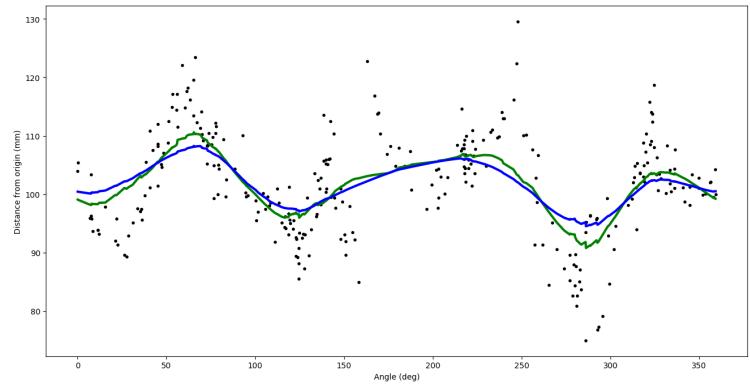
(a) LOESS regression on the "distances" graph for $f = 0.1$



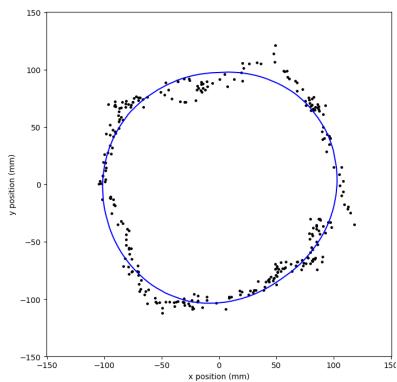
(b) Reconstructed shape for $f = 0.1$



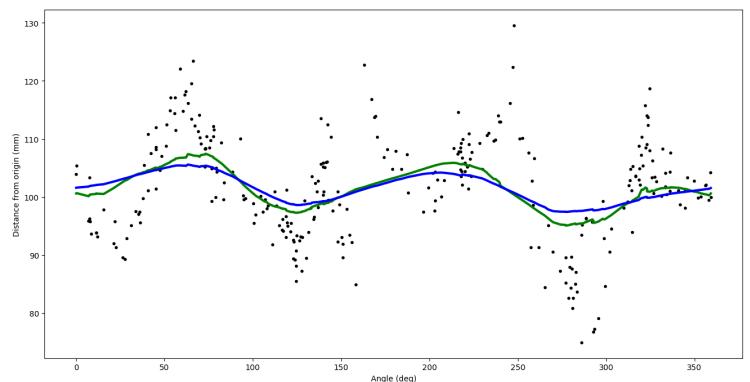
(c) LOESS regression on the "distances" graph for $f = 0.2$



(d) Reconstructed shape for $f = 0.2$



(e) LOESS regression on the "distances" graph for $f = 0.3$



(f) Reconstructed shape for $f = 0.3$

Figure 3.3: Example of a LOESS curve for different values of smooth factor f .
The LOESS curve at the first iteration (without robust factor) is displayed in green while the curve at the second iteration is in blue color.

3.3.4 Discussion

Choosing the smooth factor

As we could notice with the figure 3.3, the lower the smooth factor, the better the curve reconstruction. However, lowering it too much could make the regression too sensitive to outlier. Cleveland gives in his paper [10], a method to adapt the value of f for each set of data:

1. Setting $\delta_k = 1$ for each point k
2. The local regression of each x_i is performed following the equation (3.2) and **without including the value y_i of the point x_i itself**
3. Finding f_0 , the value that minimizes $\sum_{k=1}^n \delta_k (y_k - \hat{y}_k)^2$. To find this value, I simply compute the regression for several values of f
4. Setting $f = f_0$ and repeating the steps 2 and 3

It appears that this solution takes time to be computed and is not very useful as we can set a low value to f manually to keep a correct approximation.

Issues with the method

Even if it shows good results at the first look, problems appear when the data become less homogeneously dense everywhere. Indeed, the smooth factor is still the same for every regression of each point. That might cause some issue in the less dense areas as the curve can become noisy because only few points are taken into account. The figure 3.4 is a good example of the kind of issues that come with this method. Indeed, the shape shows sharp angles and shortcuts.

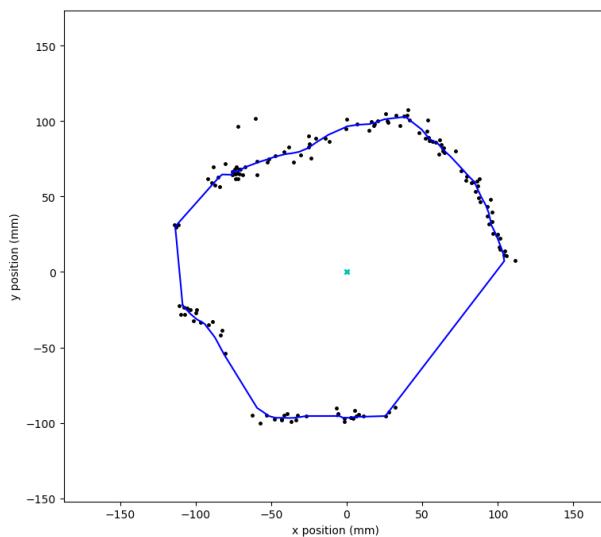


Figure 3.4: Example of a non representative curve of the user's waist shape

3.4 Curve reconstruction by B-spline via sequence joining method

Lin et al. wrote a paper [12] that describes a method that orders a point cloud and fit a B-spline through those points.

3.4.1 Concept

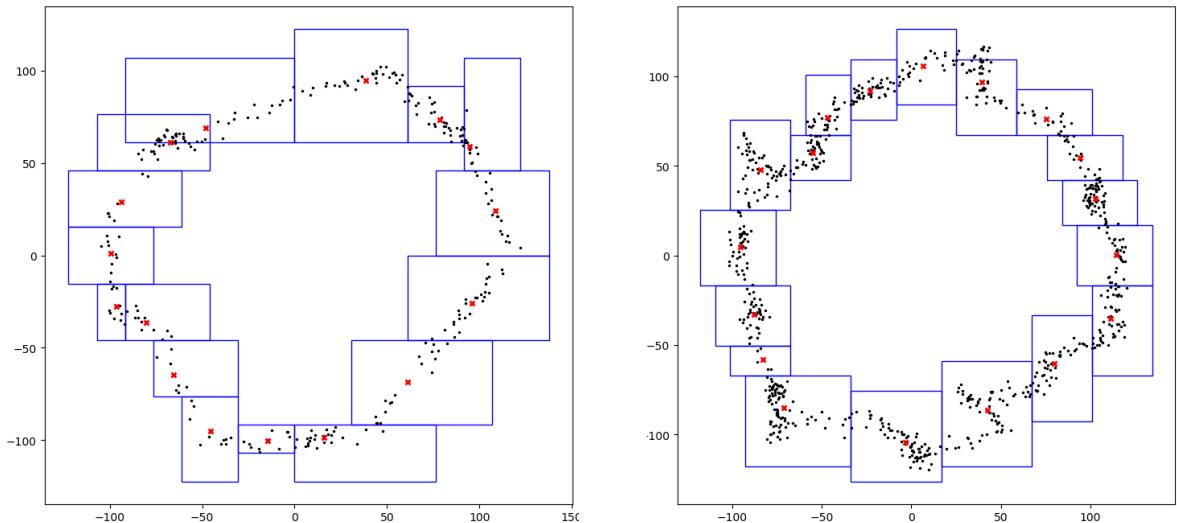
To determine order, the author call the method the "Sequence Joining Method". It consists into considering the 2D space as a grid. The size of each cell of the grid is the value of the average sampling radius that Li determines as follows: let d_p^{max} is the length of the longest edges starting from the vertex p on the Delaunay triangulation of the point cloud, and n is the number of points. Then, the average sampling radius r_{max} is:

$$r_{max} = \frac{\sum_{p=1}^n d_p^{max}}{n} \quad (3.7)$$

Then, following a specific algorithm that I won't describe here, we divide the grid into several subgrids by expanding/compressing them. The barycenter is then computed for each subgrid, as the mean of the x and y coordinates of each points of the subgrid. This barycentric points are then the reference points in which the curve will lie into.

3.4.2 Results

The figure 3.5 shows some examples of the sequence joining method's implementation. The barycentric points in red color will host the curve of the shape. We can notice that the shape will be well smoothed. This results meet our expectation.



(a) Implementation of the sequence joining method (b) Implementation of sequence joining method

Figure 3.5: Two examples of the sequence joining method's implementation

The subgrids are shown in blue and the barycenters of each subgrid are the red crosses.

However, we will validate once again the method with a set of point less favorable, not homogeneously dense and containing less point. This method does not work at all anymore when a part of the shape contains less points than the other part, as shown in the figure 3.6.

Because of this major issue, this algorithm cannot be used, as well as the method described by K. Lee [14].

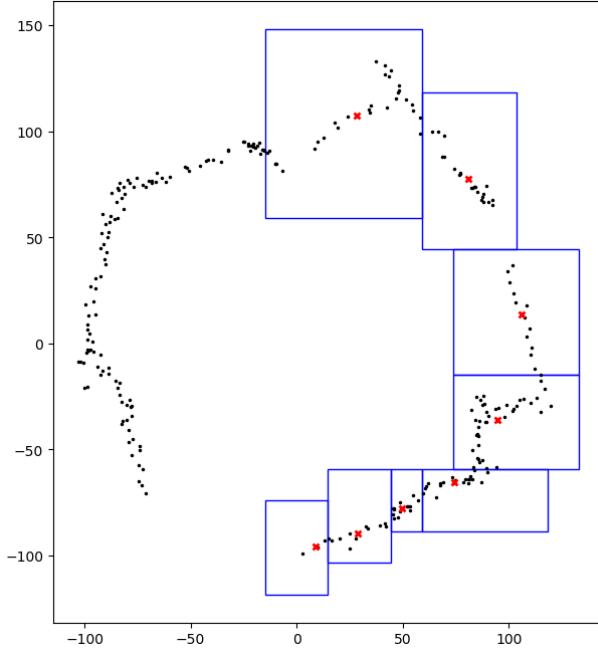


Figure 3.6: Issue with the sequence joining method implementation when the point cloud is not homogeneously dense

3.5 Curve reconstruction by B-spline via Curvature Based Squared Distance Minimization

3.5.1 Concept

The method chosen here is inspired by the Wang et al.'s paper [13] which presents a method called Squared Distance Minimization (SDM) for computing a closed planar B-Spline curve that reconstructs the shape of a point cloud of non-ordered data. A B-spline curve is a specific combination of splines which are themselves, functions defined piecewise by polynomials. The optimization process is a non linear least square problem and will be performed by minimizing an objective function. Contrary to the LOESS based reconstruction described in the section 3.3, this method has been chosen mostly because it is well documented and many papers discuss about it. Thus, some improvements can be made more easily in the future.

3.5.2 Sum up on B-spline

A B-spline curve can be written:

$$P(t) = \sum_{i=1}^m B_{i,n}(t) P_i \quad (3.8)$$

with t the value of the knot, m the number of control points, $B_{i,n}(t)$ the basis element of order n at the knot t and $P_i(t)$ the position of the control point i . A B-spline has an order, which defines how far a node will affect another from. In our case, we will use an order of 3 as it is the most common used and the author of the paper does not provide any further details about this. The B-spline is also set by a knot vector, which gives a value to each knot at each control point. A control point is the coefficient that, to sum up the equation (3.8), represents the position of the B-spline. If we consider a zero order B-spline, then the curve at the knot value t_i will exactly pass through the value of P_i . A B-spline basis element is a recursive function defined by the equation:

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

$$B_{i,n}(t) = \frac{t - t_i}{t_{i+n} - t_i} B_{i,n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} B_{i+1,n-1}(t)$$

3.5.3 Method's steps

We assume that the number of control points and the order of the spline will stay the same. This method consists in translating the control points at every step. The first step requires to find a good initial B-spline curve which fits with the data, since this method is highly dependent to the initial conditions. The fitting B-spline can be written from the equation (3.8):

$$P_c(t) = \sum_{i=1}^m B_i(t) P_{c,i} \quad (3.10)$$

Let P_+ be the updated B-spline after moving the control points as $P_+ = P_c + D$. Then, we will define as a *footpoint* the closest point $P_c(t_k)$ of the curve to the data point X_k . T_k , N_k and ρ are respectively the tangent unit vector, the normal unit vector and the curvature radius of the curve $P_c(t)$ at the point $P_c(t_k)$. d is the signed distance as $|d| = \|P_c(t_k) - X_k\|$ and its sign is chosen following this rule: $d \geq 0$ if X_k is on the same side from the curve of the curve curvature center at the point $P_c(t_k)$, $d < 0$ otherwise. For each iteration of the algorithm, every of the described above characteristics must be computed. Now, let's define the error term named *Squared Distance* as:

$$e_{SD,k}(D) = \begin{cases} \frac{d}{d - \rho} [(P_+(t_k) - X_k)^T T_k]^2 + [(P_+(t_k) - X_k)^T N_k]^2 & \text{if } d < 0 \\ [(P_+(t_k) - X_k)^T N_k]^2 & \text{if } 0 \leq d < \rho \end{cases} \quad (3.11)$$

By computing at each iteration the Squared Distance error term for every data point, we obtain the f_{SD} term that we will be minimized as:

$$f_{SD} = \frac{1}{2} \sum_k e_{SD,k} + \lambda f_s \quad (3.12)$$

Here, f_s is a regularization term to improve the stability and λ is a pondering coefficient. The f_{SD} is quadratic and thus, can be minimized by solving a linear system of equations. Then an incremental change of D to the initial control points is applied. The process can be iterated until a user custom error term is below a threshold or if the incremental change D becomes small enough.

3.5.4 Implementation

This section describes how this method has been implemented for our system.

B-spline computing

Actually, the program is written in Python, which provides libraries for B-spline computing (through *Scipy*). Two are particularly interesting and their characteristics are presented in the table 3.1. In this project, these two functions have been used.

To compute a B-spline, `scipy.interpolate.BSpline` is called twice, in order to generate a B-spline on both x and y axis. Then the values on both axis are evaluated independently. Then,

<code>scipy.interpolate.BSpline</code>	<ul style="list-style-type: none"> • Creates a 1D open B-spline function object from the knots, control points and order • Evaluates the value, the derivative and integral of the B-spline at t • Returns the basis element of the B-spline at t
<code>scipy.interpolate.splprep</code>	<ul style="list-style-type: none"> • Creates a 2D open or closed interpolation B-spline object from an ordered list of points

Table 3.1: Two interesting B-spline computing libraries provided by *Scipy*

to make the B-spline periodic as we want to reconstruct a closed shape, it is necessary to append the n first knot values and coefficients to the B-spline knot values and coefficient. That way, the B-spline will repeat its last values as if it was closed.

Finding the initial B-spline

As said in the section 3.5.3, an initial B-spline must be found before running the algorithm. The chosen implementation here is a circle fitting, exactly the same way we did for the LOESS regression in the section 3.3.2 and with the equations in the appendix C. Now that we have a circle that barely fits with the point cloud, we split it into n (the number of control points of the B-spline) equal sections as shown in the figure 3.7. Let's assume moreover that it creates areas with infinite size as they grow even outside the circle.

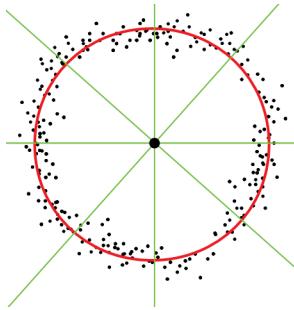


Figure 3.7: How to split the fitting circle

Example with 8 control points, the red circle is the fitted circle with the least square method.

Then, the median of the distance of the points to the center is computed for each area, taking into account the points that lie in the concerned area. We use the median as it is more robust to outliers than the mean. Finally, since we need the value of the first control points, the function `scipy.interpolate.splprep` is called to create an interpolation B-spline that fits the computed points. The control points of the initial B-spline are then extracted from this previous B-spline. In the case an area does not include any points, then we compute the average of the median of the two surrounding zones. If the surrounding zones do not include any points too, a linear interpolation is made between the two closest zone that contain points.

This method brought good results and becomes even more accurate when the number of control points increases. Some examples of the results for several number of control points n are shown in figure 3.8.

Computing the points' parameters

Many different parameters should be computed for each point of each iteration as presented in the section 3.5.3. First, the footpoint, which is the closest point $P_c(t_k)$ of the curve to the data

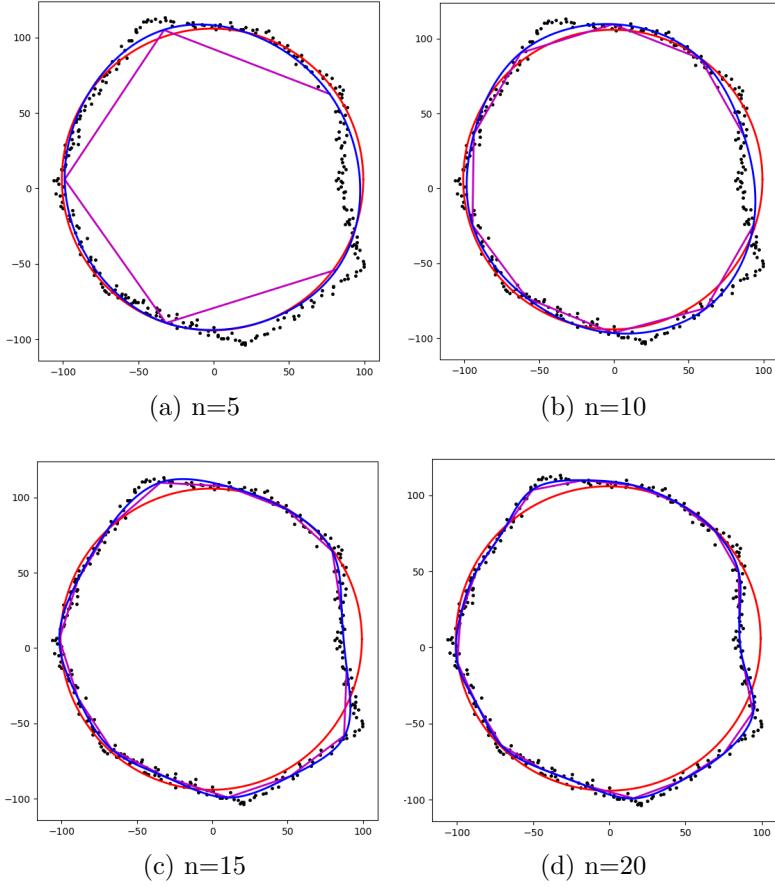


Figure 3.8: Initial B-spline results

In red the fitted circle, in magenta the median of the distance for each area and in blue the initial B-spline found.

point X_k must be computed. Many different methods can be implemented. In our case, the following process is applied:

1. Evaluating the B-spline with a large number of point (here $50n$ points)
2. Computing the distance of each data point X_k to each point of the evaluated curve
3. Finding the minimum distance and returning t_k the knot value of the footpoint $P_c(t_k)$

Then the tangent and normal vector can be recovered by deriving respectively once and twice the B-spline and dividing each x and y parameters by the vector's norm to make it unit. The radius ρ of the curve at t_k is the norm of the normal vector N_k . To find the signed distance d , we compute $K = (X_k - P_c(t_k))^T N_k$. If the parameter of $K = [K_x, K_y]$ with the greatest value is negative, then $d < 0$, else $d \geq 0$.

Minimizing the objective function

The function f_{SD} described in the equation (3.12) must be minimized in order to find the optimal incremental update D . The implementation of the algorithm is inspired from the work of Pekelny [16] and Mhala [17]. As shown in Wang et al. 's paper [13], f_{SD} is positive and quadratic in D_i , which means that it can be minimized solving $\frac{\partial f_{SD}}{\partial D} = 0$. It can be written in the form $Ax = b$ as follow:

$$\begin{bmatrix}
\left(\frac{\partial f_{SD}}{\partial D_{1,x}} \right)_{1,x} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{1,x}} \right)_{n,x} & \left(\frac{\partial f_{SD}}{\partial D_{1,x}} \right)_{1,y} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{1,x}} \right)_{n,y} \\
& & \vdots & & & \\
\left(\frac{\partial f_{SD}}{\partial D_{n,x}} \right)_{1,x} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{n,x}} \right)_{n,x} & \left(\frac{\partial f_{SD}}{\partial D_{n,x}} \right)_{1,y} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{n,x}} \right)_{n,y} \\
\left(\frac{\partial f_{SD}}{\partial D_{1,y}} \right)_{1,x} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{1,y}} \right)_{n,x} & \left(\frac{\partial f_{SD}}{\partial D_{1,y}} \right)_{1,y} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{1,y}} \right)_{n,y} \\
& & \vdots & & & \\
\left(\frac{\partial f_{SD}}{\partial D_{n,y}} \right)_{1,x} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{n,y}} \right)_{n,x} & \left(\frac{\partial f_{SD}}{\partial D_{n,y}} \right)_{1,y} & \cdots & \left(\frac{\partial f_{SD}}{\partial D_{n,y}} \right)_{n,y}
\end{bmatrix}_{2n,2n} = \begin{bmatrix} D_{1,x} \\ \vdots \\ D_{n,x} \\ D_{1,y} \\ \vdots \\ D_{n,y} \end{bmatrix}_{1,2n} \begin{bmatrix} b_{1,x} \\ \vdots \\ b_{n,x} \\ b_{1,y} \\ \vdots \\ b_{n,y} \end{bmatrix}_{1,2n} \quad (3.13)$$

The left matrix is A , the middle one is x and the right one is b , with n the number of control points, D the incremental update and b the constant term of the derivative. Each term $\left(\frac{\partial f_{SD}}{\partial D_{i,x/y}} \right)_{j,x/y}$ must be computed, which is the coefficient of $D_{j,x/y}$ when f_{SD} is derived by $D_{i,x/y}$. With the equation (3.12), we see that f_{SD} is a combination of the error term e_{SD} and the regularization term f_s . First, we do not consider the regularization term, thus:

$$\frac{\partial f_{SD}}{\partial D_{i,x/y}} = \frac{1}{2} \sum_k \frac{\partial e_{SD,k}}{\partial D_{i,x/y}} \quad (3.14)$$

When $0 \leq d < \rho$, with the formula of e_{SD} defined in the equation (3.11):

$$\begin{aligned}
e_{SD,k} &= [(P_+(t_k)) - X_k]^T N_k]^2 \\
\frac{\partial e_{SD,k}}{\partial D_{i,x}} &= \frac{\partial [(P_+(t_k)) - X_k]^T N_k]^2}{\partial D_{i,x}} \\
&= 2\beta_i(t_k) N_{k,x} [(P_+(t_k)) - X_k]^T N_k \\
&= 2\beta_i(t_k) N_{k,x} \left[\left(\sum_{j=1}^n \beta_j(t_k) D_j \right)^T N_k \right] + 2\beta_i(t_k) N_{k,x} [(P(t_k)) - X_k]^T N_k \\
&= 2\beta_i(t_k) N_{k,x}^2 \sum_{j=1}^n \beta_j(t_k) D_{j,x} + 2\beta_i(t_k) N_{k,x} N_{k,y} \sum_{j=1}^n \beta_j(t_k) D_{j,y} \\
&\quad + 2\beta_i(t_k) N_{k,x} [(P(t_k)) - X_k]^T N_k
\end{aligned} \quad (3.15)$$

In the same way as the equation (3.15), we can find $\frac{\partial e_{SD,k}}{\partial D_{i,y}}$ when $0 \leq d < \rho$:

$$\begin{aligned}
\frac{\partial e_{SD,k}}{\partial D_{i,y}} &= 2\beta_i(t_k) N_{k,y}^2 \sum_{j=1}^n \beta_j(t_k) D_{j,y} + 2\beta_i(t_k) N_{k,x} N_{k,y} \sum_{j=1}^n \beta_j(t_k) D_{j,x} \\
&\quad + 2\beta_i(t_k) N_{k,y} [(P(t_k)) - X_k]^T N_k
\end{aligned} \quad (3.16)$$

Now, when $d < 0$, the error term is very similar as it is the sum of the error term when $0 \leq d$ and itself when N is replaced by T and multiplied by $\frac{d}{d-\rho}$. Thus, when $d < 0$, we have:

$$\begin{aligned}
\frac{\partial e_{SD,k}}{\partial D_{i,x}} &= 2\beta_i(t_k)N_{k,x}^2 \sum_{j=1}^n \beta_j(t_k)D_{j,x} + 2\beta_i(t_k)N_{k,x}N_{k,y} \sum_{j=1}^n \beta_j(t_k)D_{j,y} \\
&\quad + 2\beta_i(t_k)N_{k,x}[(P(t_k) - X_k)^T N_k] \\
&\quad + 2\beta_i(t_k)\frac{d}{d-\rho}T_{k,x}^2 \sum_{j=1}^n \beta_j(t_k)D_{j,x} + 2\beta_i(t_k)\frac{d}{d-\rho}T_{k,x}T_{k,y} \sum_{j=1}^n \beta_j(t_k)D_{j,y} \\
&\quad + 2\beta_i(t_k)\frac{d}{d-\rho}T_{k,x}[(P(t_k) - X_k)^T T_k] \\
\frac{\partial e_{SD,k}}{\partial D_{i,y}} &= 2\beta_i(t_k)N_{k,y}^2 \sum_{j=1}^n \beta_j(t_k)D_{j,y} + 2\beta_i(t_k)N_{k,x}N_{k,y} \sum_{j=1}^n \beta_j(t_k)D_{j,x} \\
&\quad + 2\beta_i(t_k)N_{k,y}[(P(t_k) - X_k)^T N_k] \\
&\quad + 2\beta_i(t_k)\frac{d}{d-\rho}T_{k,y}^2 \sum_{j=1}^n \beta_j(t_k)D_{j,y} + 2\beta_i(t_k)\frac{d}{d-\rho}T_{k,x}T_{k,y} \sum_{j=1}^n \beta_j(t_k)D_{j,x} \\
&\quad + 2\beta_i(t_k)\frac{d}{d-\rho}T_{k,y}[(P(t_k) - X_k)^T T_k]
\end{aligned} \tag{3.17}$$

From the equations (3.15), (3.16) and (3.17), we can extract the coefficients as you can find in the following tables:

$d < 0$	Derivative by $D_{i,x}$	Derivative by $D_{i,y}$
Coefficient $D_{j,x}$	$2\beta_i(t_k)\beta_j(t_k)N_{k,x}^2$ $+2\frac{d}{d-\rho}\beta_i(t_k)\beta_j(t_k)T_{k,x}^2$	$2\beta_i(t_k)\beta_j(t_k)N_{k,x}N_{k,y}$ $+2\frac{d}{d-\rho}\beta_i(t_k)\beta_j(t_k)T_{k,x}T_{k,y}$
Coefficient $D_{j,y}$	$2\beta_i(t_k)\beta_j(t_k)N_{k,x}N_{k,y}$ $+2\frac{d}{d-\rho}\beta_i(t_k)\beta_j(t_k)T_{k,x}T_{k,y}$	$2\beta_i(t_k)\beta_j(t_k)N_{k,y}^2$ $+2\frac{d}{d-\rho}\beta_i(t_k)\beta_j(t_k)T_{k,y}^2$
Constant b_j	$2\beta_i(t_k)N_{k,x}[(P(t_k) - X_k)^T N_k]$ $+2\frac{d}{d-\rho}\beta_i(t_k)T_{k,x}[(P(t_k) - X_k)^T T_k]$	$2\beta_i(t_k)N_{k,y}[(P(t_k) - X_k)^T N_k]$ $+2\frac{d}{d-\rho}\beta_i(t_k)T_{k,y}[(P(t_k) - X_k)^T T_k]$

Table 3.2: Coefficients $D_{j,x/y}$ of derivative by $D_{i,x/y}$ when $d < 0$

$0 \leq d < \rho$	Derivative by $D_{i,x}$	Derivative by $D_{i,y}$
Coefficient $D_{j,x}$	$2\beta_i(t_k)\beta_j(t_k)N_{k,x}^2$	$2\beta_i(t_k)\beta_j(t_k)N_{k,x}N_{k,y}$
Coefficient $D_{j,y}$	$2\beta_i(t_k)\beta_j(t_k)N_{k,x}N_{k,y}$	$2\beta_i(t_k)\beta_j(t_k)N_{k,y}^2$
Constant b_j	$2\beta_i(t_k)N_{k,x}[(P(t_k) - X_k)^T N_k]$	$2\beta_i(t_k)N_{k,y}[(P(t_k) - X_k)^T N_k]$

Table 3.3: Coefficients $D_{j,x/y}$ of derivative by $D_{i,x/y}$ when $0 \leq d < \rho$

It is now possible to fill a matrix for each point with those coefficients, then summing them all.

We can now consider the regularization term. It is used to smooth the curve. The Wang et al.'s paper [13] gives two regularization functions that can be used:

$$\begin{aligned} F_1 &= \int \left\| P'(t) \right\|^2 dt \\ F_2 &= \int \left\| P''(t) \right\|^2 dt \end{aligned} \quad (3.18)$$

However, these functions are computationally very expensive. A simpler regularization term could be to align each control point with its closest neighbors. This equals to minimize the distance between each control points and the middle m_i of the segment which links its two closest neighbors.

$$f_s = \sum_{i=0}^n \|m_i - P_+\|^2 \quad (3.19)$$

The norm is squared in f_s in order to make it quadratic and to minimize it easily. We need to compute $\min f_s$, which can be performed by solving $\frac{\partial f_s}{\partial D} = 0$. Moreover, the middle of the segment which links P_{i-1} and P_{i+1} is computed as: $m_i = \frac{P_{i+1} + P_{i-1}}{2}$.

$$\begin{aligned} f_s &= \sum_{i=0}^n \|m_i - (P_i + D_i)\|^2 \\ &= \sum_{i=0}^n \left([(m_{i,x} - (P_{i,x} + D_{i,x}))^2 + (m_{i,y} - (P_{i,y} + D_{i,y}))^2 \right) \end{aligned} \quad (3.20)$$

$$\frac{\partial f_s}{\partial D_{i,x}} = -2m_{i,x} + 2P_{i,x} + 2D_{i,x}$$

$$\frac{\partial f_s}{\partial D_{i,y}} = -2m_{i,y} + 2P_{i,y} + 2D_{i,y}$$

We can now add the coefficients found in the equation (3.20) to the matrices A and b as follows:

$$A_{reg}[i][i] = 2\lambda, \quad A_{reg}[i+n][i+n] = 2\lambda$$

$$b_{reg}[i] = 2\lambda(P_{i,x} - m_{i,x}), \quad b_{reg}[i+n] = 2\lambda(P_{i,y} - m_{i,y})$$

The influence of the λ value will be discussed later in the section 3.5.5. It is now possible to solve the system $(A + A_{reg}).x = (b + b_{reg})$ to get D .

Outliers removal

A single false data point can produce an high change on the curve. Thus, at each iteration, we compute the standard deviation σ_d of the distances d between the points and their footpoint on the curve. Then, if $d_k > 6\sigma_d$, the data point k is not considered anymore. The result are discussed in the section 3.5.5.

Instability due to the lack of points

In some cases, no or too few data points have their footpoint on the curve between two control points: $\exists i \text{ as } \neg\exists t_k \in [t_i, t_{i+1}]$. The matrix A can thus become very small in some lines and bring very high values for D . To counter this issue, we sum each line of the matrix A and compare it to a threshold. If the sum is lower than the threshold, the control point is moved manually, half way to the middle of the segment connecting its valid neighbours. The method is shown in figure 3.9 from the Pekelny's report [16].

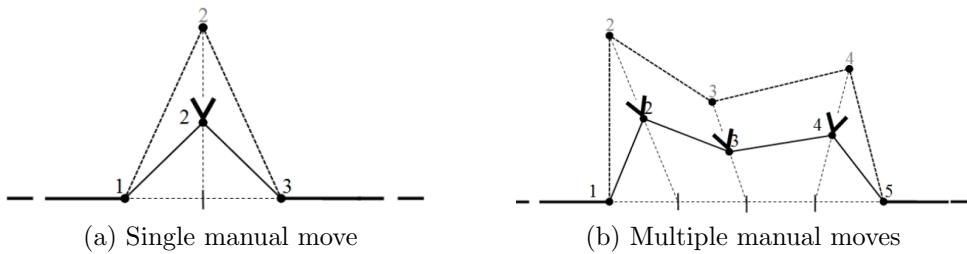


Figure 3.9: Method for moving manually the control points

Then considering the j^{th} line of the A matrix which was too small, we now set the coefficients as follows:

$$\begin{aligned} A[j][j] &= 1, \quad A[j][\neq j] = 0, \quad A[\neq j][j] = 0 \\ b[j] &= 0 \end{aligned}$$

Stop conditions

For each iteration (or loop) l of the algorithm, an error term ϵ_l is computed as follows:

$$\epsilon_l = \frac{1}{nb_of_points} \sum_k d_k^2 \quad (3.21)$$

Then, this error term is compared to a threshold value and to the previous value of ϵ_l with $\epsilon_0 = +\infty$. If ϵ_l is lower than the threshold, the current B-spline is optimized and the algorithm stops. If $\epsilon_l > \epsilon_{l-1}$, the B-spline at the iteration $l-1$ is kept and the algorithm stops. Finally, if $l = l_{max}$, the algorithm stops and the last B-spline is saved. For a greater processing speed, the gradient of ϵ_l can be computed in order to stop the algorithm if the variation becomes too small.

3.5.5 Results

This section will describe the results of this method and some improvements that can be done.

Testing

All along this section, we will try to fit a point cloud and varying:

- The ratio $\frac{n}{p}$ with n the number of control points and p the number of point in the cloud
- The weighting factor λ of the regularization term
- The sharpness and density of the point cloud

First the figure 3.10 shows what the result looks like. It has reached its minimum error in 1 iteration.

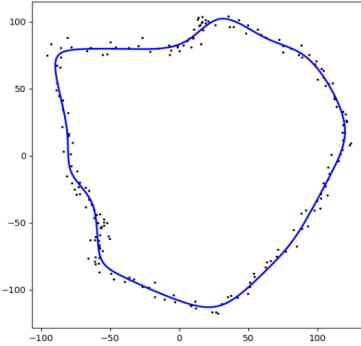


Figure 3.10: B-spline fitting in a basic point cloud ($n=20$, $p=200$, $\lambda=0.1$)

Influence of the number of control points

The figure 3.11 displays several curve reconstruction processes for various number of control points.

The number of control points describes the roughness of the shape. In our case, we want a high degree of precision on the body but if the data points become too few or too noisy, we will face an over-fitting problem as on the curve 3.11e. However, choosing too few control points won't be enough to fit the contour of the human waist, as on the curve 3.11a that is barely a round shape. 15 control points seem to be great for our purpose.

Influence of the regularization coefficient's value

Tests with several regularization coefficient's value are shown in figure 3.12.

As we expected, the higher the λ 's value, the more the curve will tend to minimize its circumference, i.e. becoming a circle.

Influence of the sharpness of the point cloud

The figure 3.13 shows the curve fitting for low, high and both point density:

The results are quite satisfying as the curve always fit with the data. When the density is quite low as on the figure 3.13b, a few number of control points is required, contrary to the figure 3.13c where we notice that the curve is lacking of control nodes to fit perfectly with the data. Moreover, as we did for the LOESS regression (figure 3.4) and for the sequence joining method (figure 3.6), we need to test this method with a non homogeneously dense data cloud with holes in the shape. The figure 3.14 clearly shows that the SDM shows better results than the method with LOESS regression.

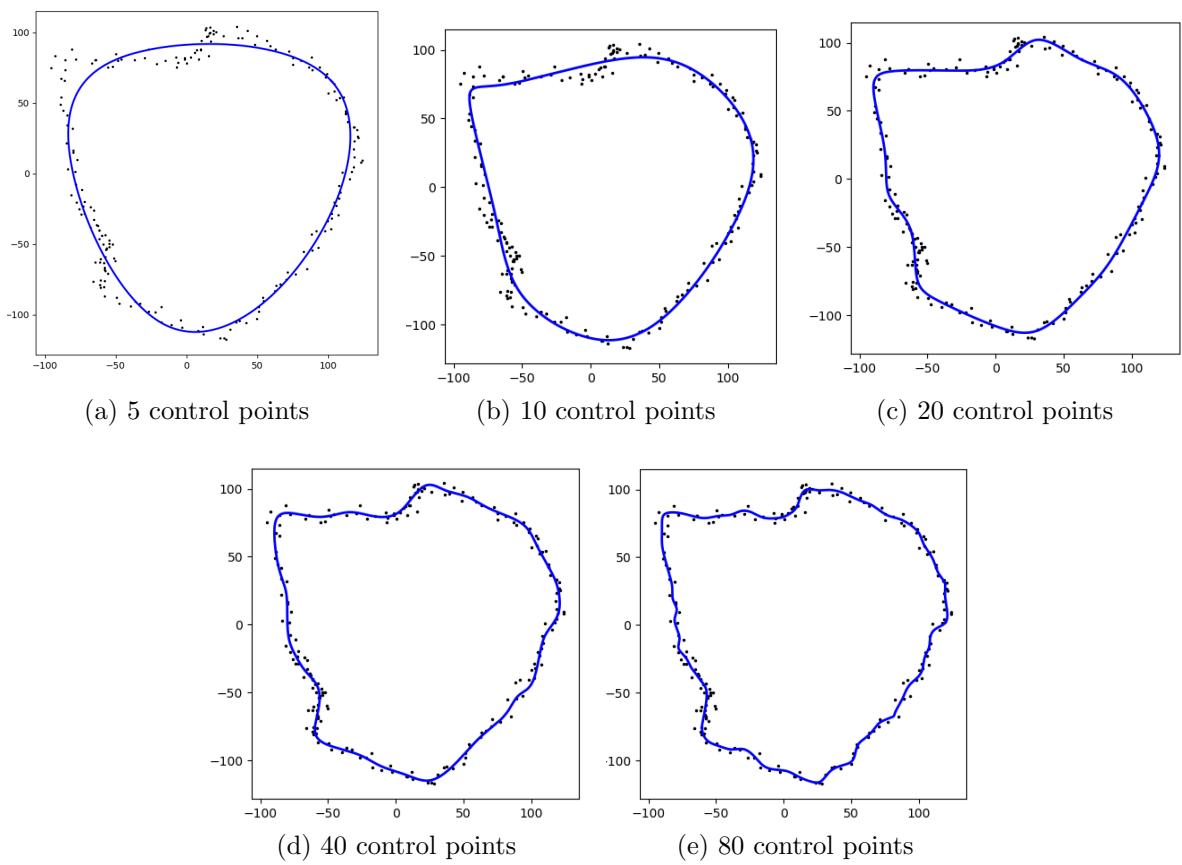


Figure 3.11: Influence of the number of control points on the curve ($p=200$, $\lambda=0.1$)

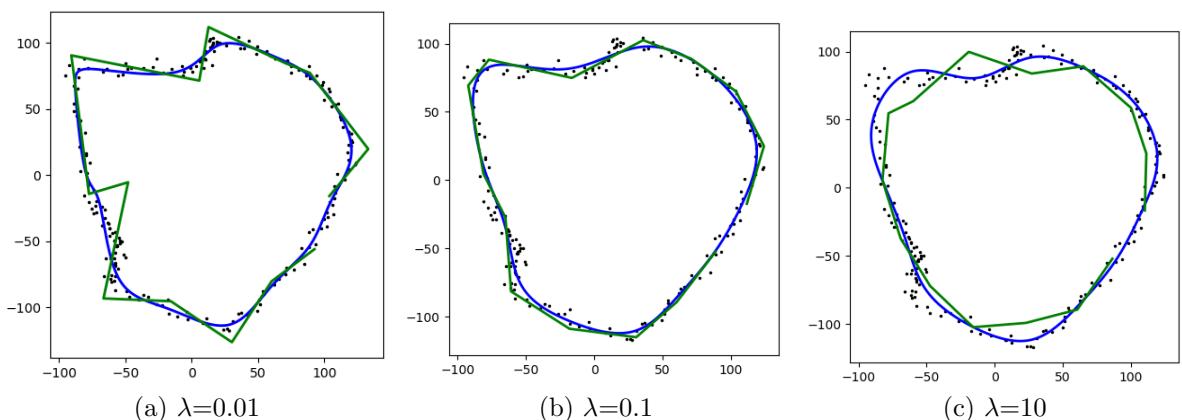


Figure 3.12: Influence of regularization coefficient's value on the curve ($n=15$, $p=200$)
The control points are shown in **green** while the curve is in **blue**

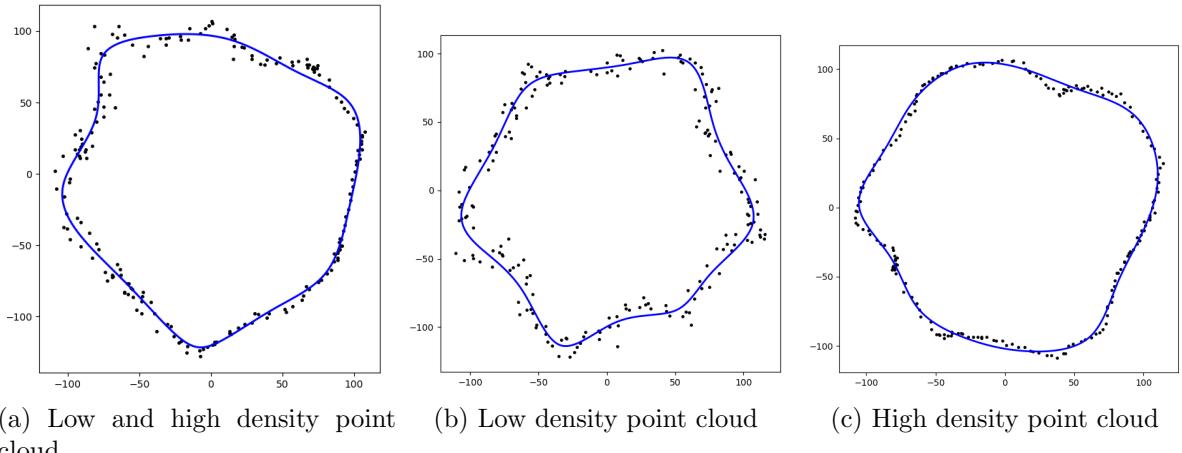


Figure 3.13: Influence of point cloud's sharpness on the curve ($n=15$, $p=200$, $\lambda=0.1$)

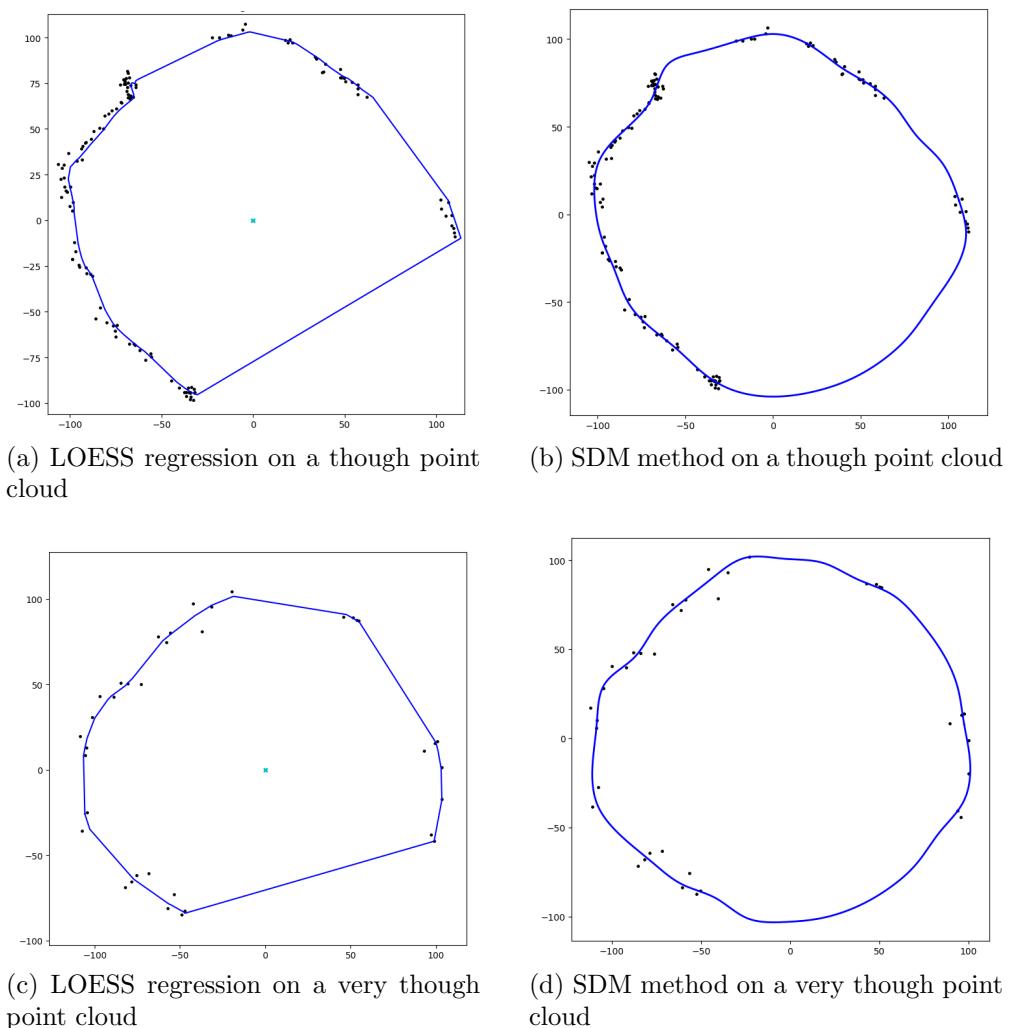


Figure 3.14: Comparison of the LOESS and SDM method when the point cloud is not homogeneously dense and with few points

Outlier detection

The figure 3.15 shows goods results when the outliers are few enough.

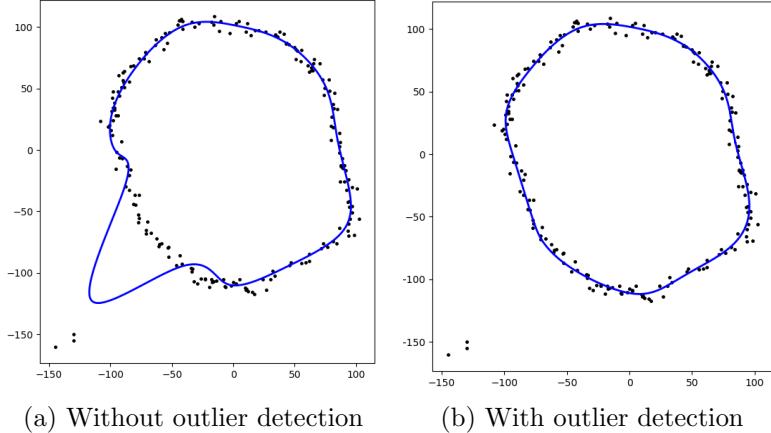


Figure 3.15: Test of the outlier detection ($n=15$, $p=200$, $\lambda=0.1$)

Possible improvements

Several issues occurred while testing this solution. First, if the set of data is becoming large or if the desired number of control points is too high, the algorithm can take a lot of time to run. A solution could be to run the program on a faster microprocessor or to run the algorithm with multiple threads. It appears also that sometime, the number of control points is not well adapted to the shape (as seen in the section 3.5.5). A paper from Yang et al. [18] presents a method to increase or decrease the number of control points at each iteration until the desired error becomes lower than the objective. Moreover, it could be a great idea to implement an improved outlier detection as the currently implemented one is not robust to an high number of outlying points. Finally, a better regularization function should be integrated, bringing probably better results. Indeed, the current one only take in account the control points' position instead of the B-spline curve.

3.6 Computation of the waist circumference

Once the shape of the user's waist is reconstructed by the previously described methods, the waist circumference can be measured easily by splitting the curve in small discrete intervals and summing the euclidean distance between all the points.

Chapter 4

Automatic health checkup system - Validation

Validating such a system is quite though as it requires a large set of test patients with different stature, skin color, waist shape in different measuring environment conditions. Thus, I could not validate it properly via statistical analysis, comparing the measured and real waist circumferences. Moreover, this project is divided into several different independent parts and each one need to be validated (navel's height estimation, scan, shape reconstruction). However, I have demonstrated the capability of such a system to perform measurement and reconstruct a shape, only the accuracy of the system need to be measured.

To illustrate the results with real data, I have anyway performed two tests in real condition. The figure 4.1 is a measurement of my own waist wearing clothes, while the figure 4.2 show the system and the test over the waist shape of a dummy.

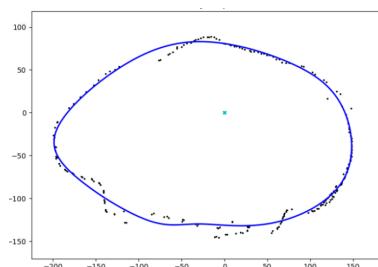


Figure 4.1: Measurement and shape reconstruction of my own waist

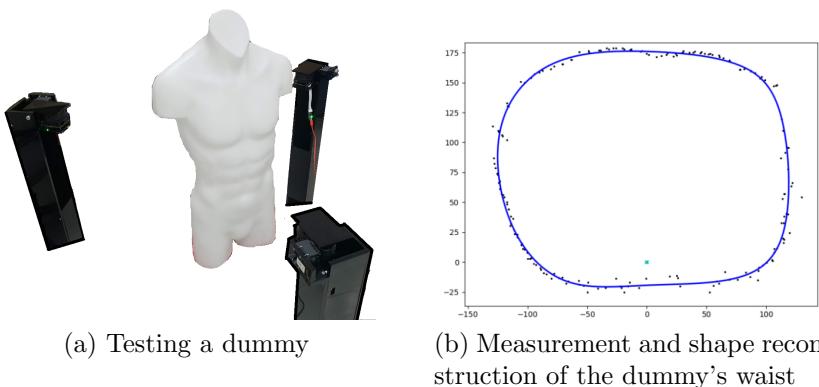


Figure 4.2: Test in real conditions with a dummy

Chapter 5

Orientation measurement system

The following chapter describes the work I have done on the project presented in 1.2, which aims to recover, process and display the orientation of an Inertial Measurement Unit based system for medical purpose.

5.1 System's architecture

The system is divided into 3 main sub-parts as shown in the figure 5.1:

- The 10 DOF IMU which is a *MPU9250* board.
- A micro-controller, which communicates with the *MPU9250* through an I2C protocol and recovers the raw data (external forces in $m.s^{-2}$, angular rate in $m.s^{-1}$, magnetic field in μT and sensor's temperature in $^{\circ}C$). An Arduino nano is used for this purpose.
- A C#.NET software that yields the data from the micro-controller. It filters them, convert them into a quaternion representation and displays the results to an user-friendly interface.

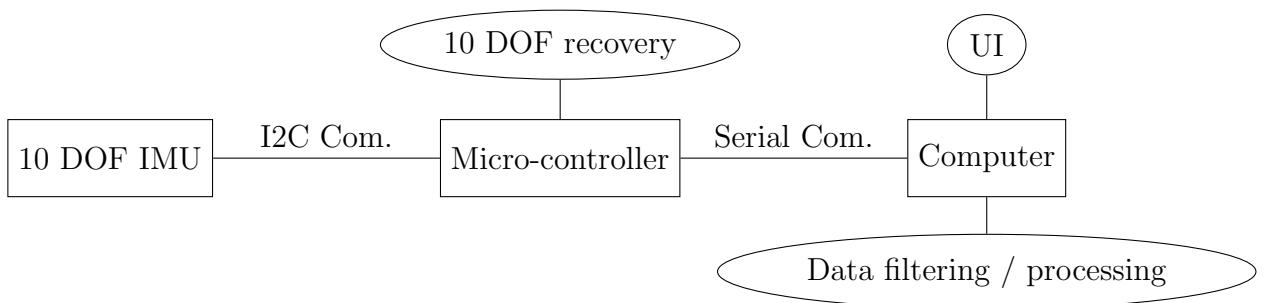


Figure 5.1: Basic architecture of the developed system

5.2 Raw data recovery

Only 4 wires are required to recover the data from the *MPU9250*:

- POWER: 3.3V/GND
- I2C Com.: SCL/SDA

The code downloaded on the Arduino [19] is open source, shared by *FaBo* and distributed under Apache license 2.0 (non-restrictive). Once recovered, the raw values are transmitted through an high speed serial communication to the filtering and processing software.

5.3 Software's structure

I was free to choose the programming language and environment. I have chosen C#.NET, under Visual Studio, as I had a few of experience into it. I programmed it entirely to make it able to manage the serial communication, to process the orientation and to display it through a 3D view and a readable angle. More features have been implemented and are presented in the section 5.6. An overview of the software is shown in figure 5.2.

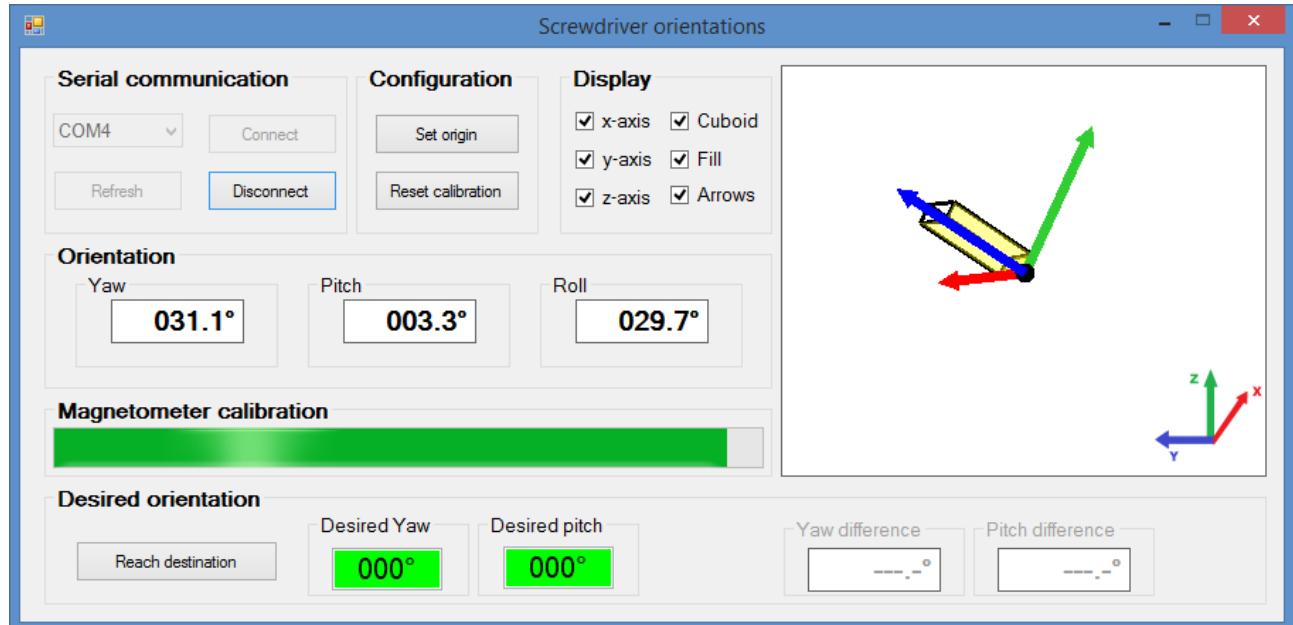


Figure 5.2: Overview of the software

Basically, the program is made of 3 threads:

- The serial thread that is triggered on data received. It parses the data from the serial and calibrates the received magnetometer values on-the-fly (as presented in the section 5.4).
- The filter thread that implements a Mahony filter in order to compute the quaternions from the raw data. It is triggered on a timer interrupt with a period of 2ms.
- The display thread is also triggered on a timer with a period of 20ms. It refreshes the display of the Euler angles and the 3D view.

5.4 Magnetometer values' calibration

Some corrections need to be done on the recovered magnetometer data. This section describes why they need to be calibrated and how I made it.

5.4.1 Origins of the magnetometer biases

Contrary to the gravity, the Earth magnetic field is not homogeneous everywhere on the planet. Thus, the magnetometer cannot use it as reference to self calibrate. Moreover, the surrounding magnetic noise due to wires electronic equipment add some more uncertainty on the measurement.

5.4.2 Proposed solution

A proposed solution described by Renaudin et al. [20] is to complete a full rotation of the IMU for each axis and to keep the minimum and maximum measured values on each axis. This solution has been implemented following the algorithm 2 that you can find in the appendix D. It loops for each data measurement, which means that the calibration is performed on-the-fly. The more the user rotates the IMU, the better the calibration. However, this process partially solve only hard and soft iron distortions (refer the glossary page 7).

5.4.3 Results

This solution has been tested and seems to be powerful enough for our specific application. The figure 5.3 shows the difference between the raw magnetometer measurements and the calibrated ones. Each axis' unit is μT for the raw data points and $a_i \cdot \mu T + b_i$ for the calibrated data, with a_i the coefficient after calibration, b_i the offset and $i \in \{x, y, z\}$. We notice that the calibrated measurements are quite well circularized, normalized and centered as they lie inside the black circle in the 2D plot. In the 3D plot, we want the measurement to fit in the black sphere.

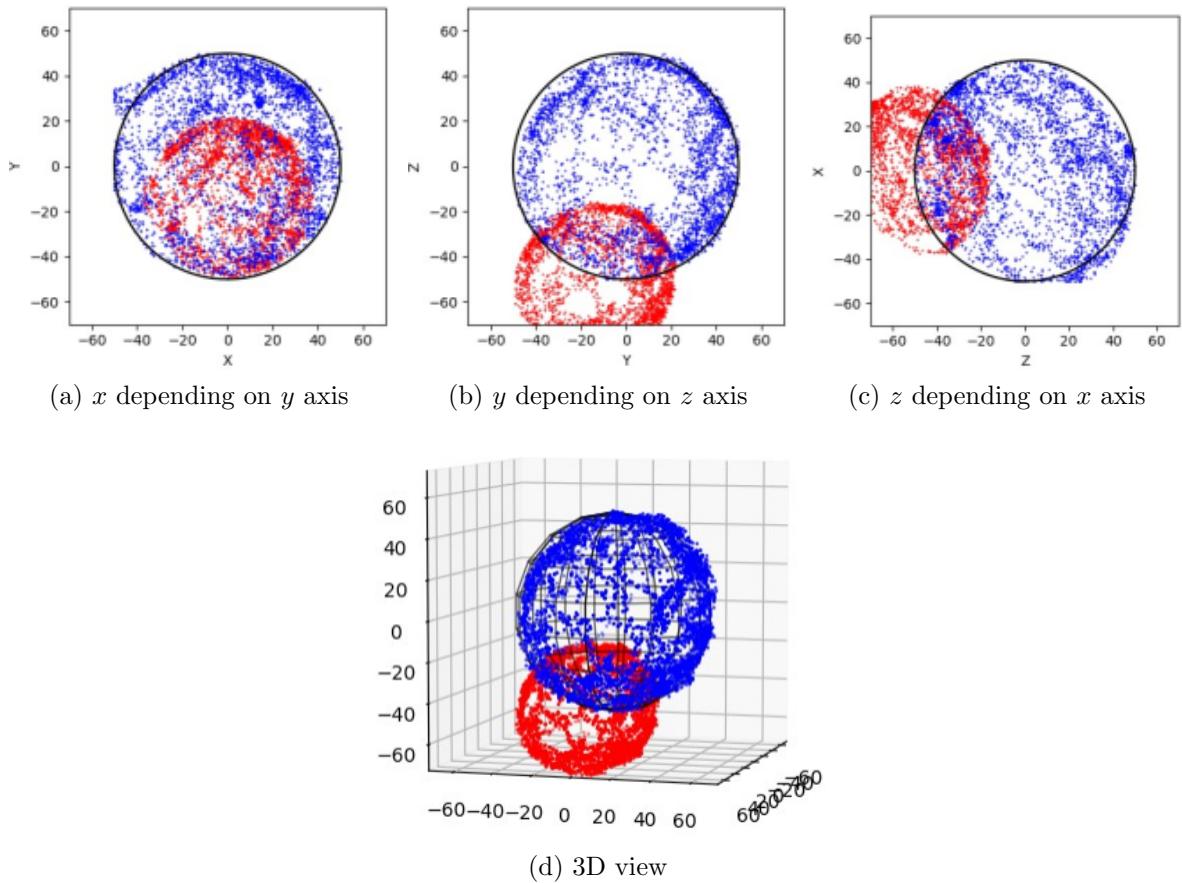


Figure 5.3: Results of the magnetometer calibration
The raw data are shown in red and the calibrated data in blue.

However, this solution is the simplest to implement and is far to be perfect as it cannot compensate some of the sensor biases like sensor non-orthogonality, zero bias error, scale factor error, misalignment error, noise... As the system is intended to be used in such places like hospitals, which contain many magnetic disturbances due to the heavy equipment and medical machines, finding a powerful calibration process should become an important feature to implement. Many research papers are dedicated to 3 axis magnetometer calibration. Vasconcelos et al. describe

in their study [21] a more powerful calibration using a geometrical approach, without needing the attitude of the IMU. Another research from Kok et al. [22] uses the gyroscope and accelerometer to perform a magnetometer calibration. Another problems with this solution is the on-the-fly calibration. Indeed, in case the magnetic field become more intense even for a short amount of time, the maximum or minimum value on an axis could take a too high value indefinitely. I have chosen to keep the on-the-fly calibration as it more user convenient for prototyping and testing.

5.5 Orientation processing

This section describes the computation steps required in order to get the IMU's orientation from the sensor's data.

5.5.1 Mahony's filter

The system uses an open source implementation of a Mahony's filter [23] which is a non-linear complementary filter, based on the work of Mahony [24]. It output the orientation in quaternion form from 9 DOF data. The filter needs to be fed with few parameters: the raw data in a correct order, a proportional and integral gain. Regarding to the gains, they have been chosen arbitrary in order to have a good response time and stability, manually. Moreover, the order of the raw data have to be provided with attention and regarding to the IMU used. In the case of the *MPU9250*, the datasheet need to be carefully read as the magnetometer's axis of sensitivity are placed in a non-intuitive way.

5.5.2 Usage of the quaternion representation

Quaternions are a powerful representation of a rotation in a 3D space as, among other advantages, require less computation for basic operations than the Euler angles, and are gimbal lock-free (refer the glossary page 7).

Quaternions' rotation

Let $v = (v_1, v_2, v_3)$ be the 3D vector we want to rotate with the quaternion $q = (q_1, q_2, q_3, q_4)$. The equation (5.1) gives the method to rotate such a vector. It can be useful if we want to offset the measured orientation by a know starting orientation.

$$v_R = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (5.1)$$

In the same way, we can rotate a quaternion by another quaternion representation by multiplying them term-by-term together.

Quaternion to Euler's representation

Contrary to the Euler angles, quaternions lie into a 4 dimensional space. Thus, it might be useful to display the Euler angles to the user as they are more understandable, and are required for our specific application (screws implantation with pre-defined angles). Working with the proper Euler angles (not the Tait-Bryan ones), we should use the angles $[\Phi \Theta \Psi]$ but for more convenience, let's use in the same order: $[yaw \ pitch \ roll]$. The equation (5.2) gives the computation to switch a quaternion into a Euler form.

$$\begin{bmatrix} yaw \\ pitch \\ roll \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \text{asin}(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix} \quad (5.2)$$

Note that degenerating the quaternion representation (4D space) into an Euler form (3D space) makes the system sensitive again to the gimbal lock effect and the inverse operation cannot be performed without losing information.

5.6 Software's features

This section explores the software's features implemented.

5.6.1 Serial communication

The user can manage the serial communication by connecting/disconnecting the session, choosing the com port among the available ones.

5.6.2 Orientation processing

The program process the orientation as described in the section 5.5. Moreover, the surgeon has to datum the sensor in the way he needs to use it. First, the user places the screwdriver with the IMU mounted on it, on the body and rotate it like he would do with the free-hand technique, which means parallel to the axis determined on the x-ray pictures. Then, by pressing a button, the software set this position as origin: $q_0 \leftarrow q$ with q the current measured quaternion. Now, the final orientation is computed as follows:

$$\tilde{q} = q \cdot q_0^{-1} \quad (5.3)$$

Printing the axis on the sensor could be a good idea to help the surgeon to move the sensor, as shown on figure 5.4.

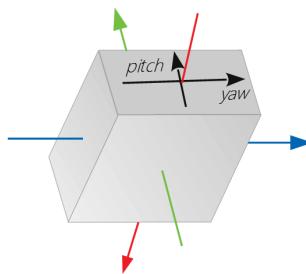


Figure 5.4: Example of the IMU's hull on which the axes orientation is displayed

5.6.3 Magnetometer's calibration

The procedure used to calibrate the magnetometer is presented in the section 5.4. The user can reset the calibration of the magnetometer. A progress bar also displays the quality of the calibration by computing the absolute difference of the expected norm of the magnetic field's vector and the measured norm after calibration. The progress bar is smoothed with a moving average filter.

5.6.4 Displaying the orientation

First, the raw Euler angles are displayed after computation with the equation (5.2). A 3D graphical representation is also displayed and the user can manage some display options. No 3D graphical library has been used, indeed, a 2D graphic is printed and the equation (5.1) is applied on each object in order to rotate the system and emulate a 3D view. This representation is not very useful for the surgeon but gives an idea of how the system works. Finally, the surgeon enters the difference in degrees from the origin that he chose from the x-ray picture and, as the figure 5.5 shows, a polar chart is displayed in order to help the surgeon to reach the objective orientation. Then, he moves the sensor by the yaw and pitch axis (the roll does not matter) to match the position of the red cross and blue circle. Note that, due to the gimbal lock effect, the difference of angles between the origin and the desired orientation must be small enough. This effect cannot be avoided as it is intrinsic to the Euler angles, which is the only representation comprehensible by humans.

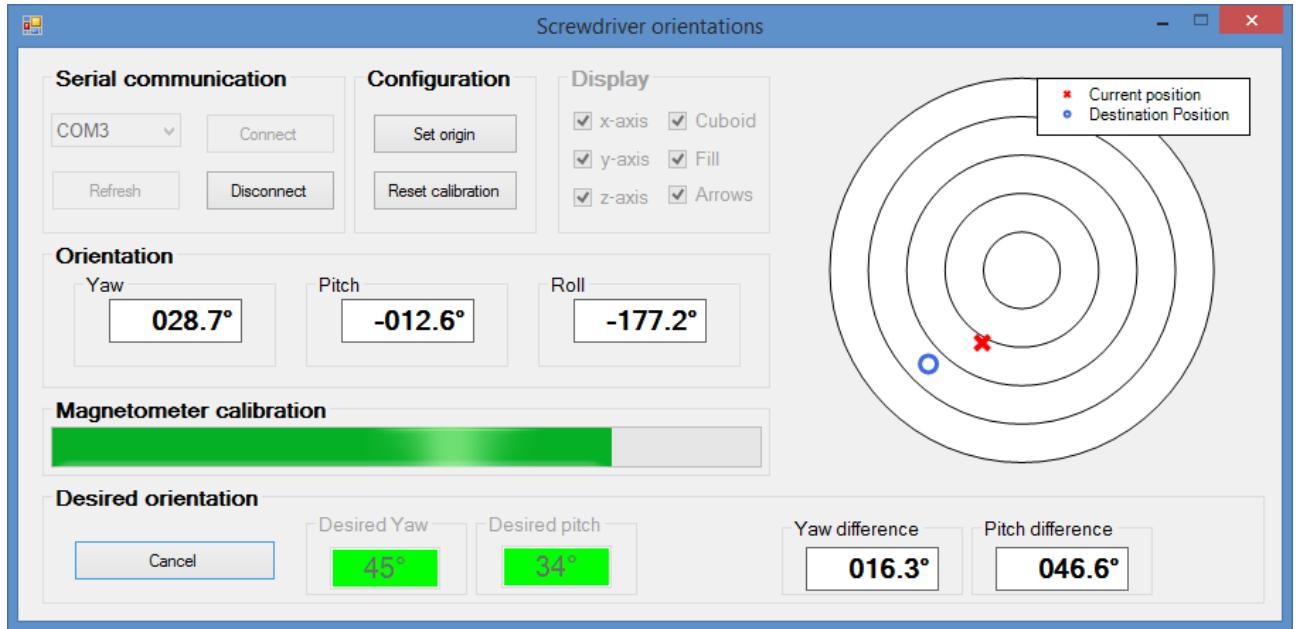


Figure 5.5: IMU's assistant for screw implantation orientation finding

5.7 Project's conclusion

This project could meet the specifications and a functional software has been delivered. Moreover, a technical report has been written to transmit skills. This cheap orientation system can now be tested by a surgeon and, after his feedback, it can be made more user-friendly. Moreover, it can be easily tuned for other applications as it provides and displays the relative orientation of an IMU. The sensor and micro-processor can also be changed as they are independent to each other. However, it has not been tested accurately. Many improvements are possible: in the first hand, the user-interface must be redesigned and the 3D graphical view could look more realistic with a dedicated 3D graphical library. In a second hand, the main issue with the computed orientation is its lack of accuracy in some positions. It is due mainly to the distortion of the magnetic field and could be sharply improved by using more powerful calibration techniques as discussed in the section 5.4.3. Finally, the raw data filtering can be improved and some papers, as Valenti et al. [25]'s one, describe a more accurate processing. Moreover, a comparison of different attitude algorithms can be found in the Cavallo et al. [26] article.

5.8 Source code

The technical report and the source codes of the project can be found as a git repository *Github* at the following URL: www.github.com/Vincema/MPU9250Interface.

Chapter 6

Conclusions & perspectives

In this chapter, I will develop the conclusions I could extract from the work I have done during this internship, in both technical and personal points of view.

6.1 Technical contributions

I could reach the goals given to me in both two projects I worked onto it. Indeed, I prove the ability for such an automatic waist circumference measurement system to work properly (even if the accuracy could not be validated); and the orientation measurement system also can perform accurate attitude processing. My work was almost entirely based on previous researches and the all the relevant sources are cited in this report.

However, my prototype of the automatic waist circumference measurement system was meant to lead to the drafting of a research paper, and my work as it is now is not suitable for such a purpose as it does not implement further details about the process and its accuracy. It behaves more like a black box, thus I further work will be needed to be done to implement a process to study the behavior and the details of each part of the process. On the other hand, the orientation measurement system has its own graphical user interface and can easily be tested, modified and improved. As I said in the section 5.7, improvements (particularly on the magnetometer's calibration) can be made and are going to be necessary for such a system to work in a though environment like hospitals with strict rules and norms.

6.2 Improvements that could have been made on my work

All along my internship, I could notice many mistakes I have made on my personal organization and working methods. Firstly, I missed to report on a regular basis the state of my work to my tutor, who could give me many useful information and advice. Moreover, I should have written more small reports for each part, that could help me to step back and take more relevant decisions. I also went too fast reading the researches paper and technical documentation, without studying in details the discussions and feedback of the author, but also of the other researchers and engineers, as the author of the researches paper tend to minimize the issues brought by their method or system. Finally, I have learned how to use Python in a deeper and proper way all along the internship, thus, the code has changed many time in its structure and architecture, that could make it very hard to be understood.

6.3 Personal feedback on my internship

In many ways, this work taught me many technical skills that I have developed in this very report, but it was also the opportunity for me to learn better and proper working methods. Among them, I discovered the power of the scientific methods, as I read and studied a large amount of research papers, and I also realized that a paper itself is not enough to represents its fields of application and its limits. As I was free to work in the way I wanted, with the methods I thought would be helpful, I could understand that taking a step back and reflecting for a while was important as it could avoid to make basic mistakes or to reach dead ends. In addition to this, I used many tools such as git repositories, development environments, script editors, configurations and networking monitoring softwares... which have made this internship very interesting in terms of diversity of disciplines and knowledge.

On a very personal point, working abroad was for me the opportunity to improve my English, to discover a culture that is very different from my native's one, and I believe that it will help me for business and personal human relations, while giving me a different and maybe more innovative approach for my career in engineering. It also gave me the chance to broaden my business network. Moreover, this internship did not make my professional project more clear even though it made me more confident into my ability to adapt myself to an environment that I am not used to.

Appendix A

NECTEC's presentation and board

The National Electronics and Computer Technology Center (NECTEC) is a Thai governmental organization established in 1986 which aims to be a "research organization collaborating with alliances for achieving practical works / excellence that contributes to the economic and social impacts of the country and region". It runs research and development activities to promote and support the electronic computing, telecommunications, and information technologies. Its headquarter is based in Pathum Thani, a province of the northern surrounding area of Bangkok.

NECTEC is under the supervision of the National Science and Technology Development Agency (NSTDA) which belongs to the Thai Ministry of Science and Technology. This institution clusters many laboratories and employs more than 600 people as described in the figureA.1.

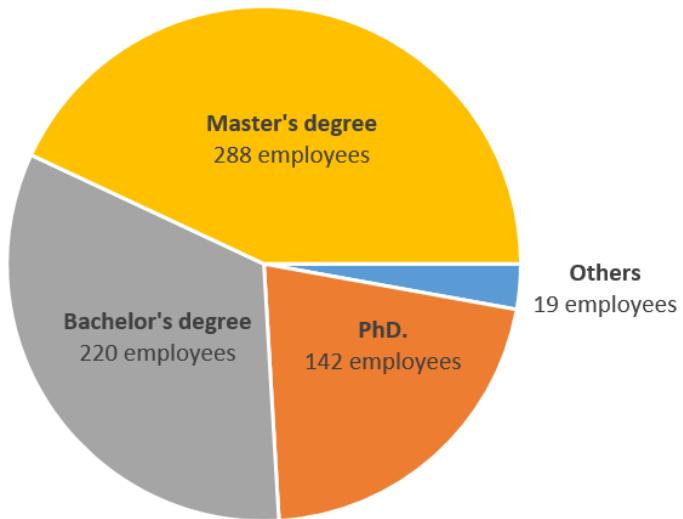
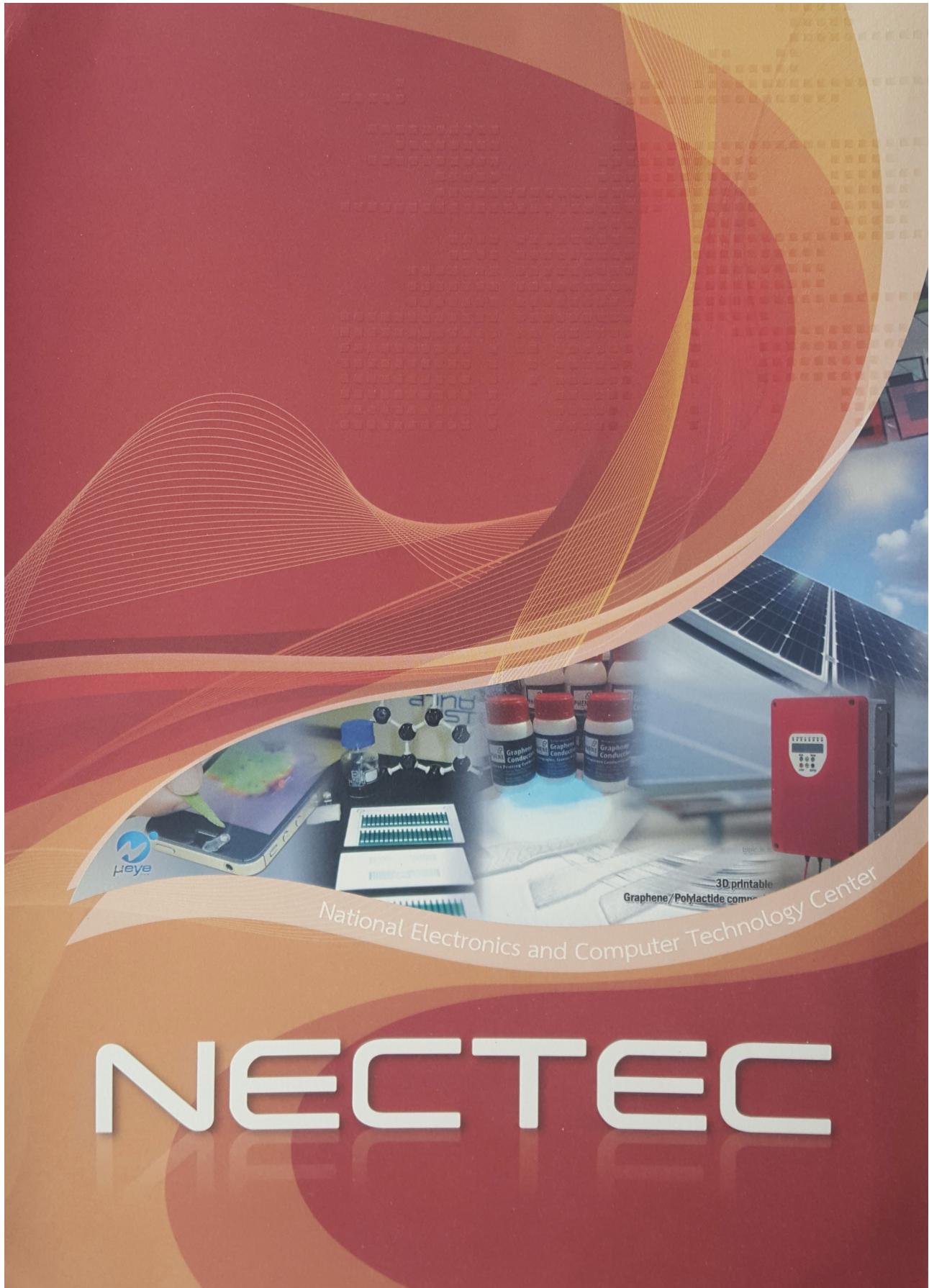


Figure A.1: Distribution of the NSTDA's staff

I ran my internship in the Biomedical Signal Processing Laboratory (BSPL), a NECTEC's laboratory. It focuses its work on information technologies and computer science combined with innovative solutions for medical or education purpose.

Please find furthermore (6 next pages) the NECTEC's presentation board.



The National Electronics and Computer Technology Center (NECTEC)

NECTEC in a Nutshell

Background

The National Electronics and Computer Technology Center (NECTEC) was established on 16 September 1986 under the Ministry of Science, Technology and Energy (the former name of the Ministry of Science and Technology). On 30 December 1991, following the enactment of the Science and Technology Development Act of 1991, NECTEC was transformed into a national technology center under the National Science and Technology Development Agency, Ministry of Science and Technology. At present, NECTEC is a statutory government organisation with its main responsibilities of undertaking, supporting and promoting the research and development of electronics and computer technologies. NECTEC also provides a linkage between research communities and industries through the established industrial clusters and programmes.

Vision

"Being a research organization collaborating with alliances for achieving excellence in science and technology that contributes to the economic and social impacts of the country and region."

Mission

- Research, development, design and engineering
- Technology transfer to industries and communities
- Human resource development
- Policy research and industrial intelligence and knowledge infrastructure

NECTEC Timeline



R&D for Economy and Society

NECTEC has focused on research and development to respond to 7 industrial target groups in accordance with the National Economic and Social Development Plan as following;



I. Food and Agriculture

- Electronics and Computer Innovation for Agriculture (ECIA)



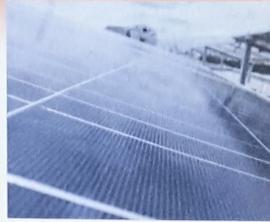
II. Manufacturing Industry

- High efficiency motor and drives
- Eco-inverter
- Printed electronics
- Real white Light Emitting Diodes (LEDs)



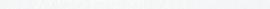
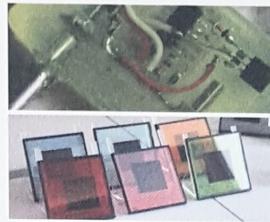
III. Service Industry

- Semi-automatic translation center
- Open services platform
- Opinion mining



IV. Resource, Energy and Environment

- Power monitoring system
- Renewable energy source and energy storage systems
- Building energy management system



V. Community, Rural Area and Disadvantaged

- Self-learning service platform
- Applications for STEM
- Sensor of thinking process,
- Solar cell and electronic devices



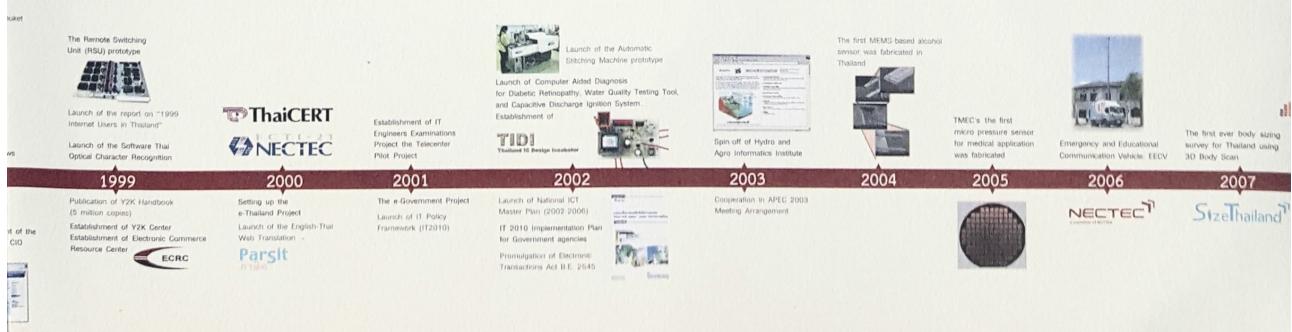
VI. Health and Medical Sector

- Data integration and health analytics
- Accessible and applied technology for people with disabilities and elderly
- Integrated medical device development ecosystem



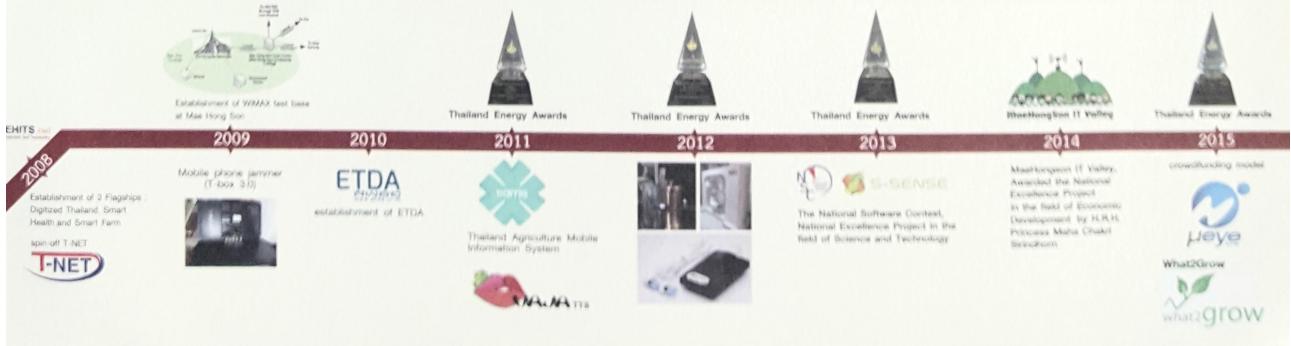
VII. National Security

- Homeland security
- Digital forensics
- Cybersecurity





Research Units	Laboratories
1. Intelligent Informatics Research Unit (INIRU)	<ul style="list-style-type: none"> - Speech and Audio Technology Laboratory - Image Technology Laboratory - Language and Semantic Technology Laboratory
2. Information Communication and Computing Research Unit (ICCRU)	<ul style="list-style-type: none"> - Large Scale Simulation Research Laboratory - Internet Innovation Laboratory - Intelligent Transportation Systems Laboratory - Knowledge Elicitation and Archiving Laboratory - Computational Process Analytics Research Laboratory
3. Biomedical Electronics and Systems Research Unit (BESRU)	<ul style="list-style-type: none"> - Rehabilitation and Assistive Technologies Laboratory - Biomedical Signal Processing Laboratory - X-Ray CT and Medical Imaging Laboratory - Healthcare Systems and Data Analytics Laboratory - Health and Lifestyle Monitoring Laboratory
4. Wireless Information Security and Eco-Electronics Research Unit (WISRU)	<ul style="list-style-type: none"> - RF Electronics Laboratory - Wireless Network and Protocol Laboratory - Cybersecurity Laboratory
5. Thai Microelectronics Center (TMEC)	<ul style="list-style-type: none"> - Microelectronics Research and Development Division - Engineering Division
6. Innovation and Engineering Research Unit (IENRU)	<ul style="list-style-type: none"> - Industrial Prototype and Product Development Laboratory - Standard and Testing Development Laboratory - Intelligence Service Analytics and Automation
7. Advanced Automation and Electronics Research Unit (AAERU)	<ul style="list-style-type: none"> - Embedded System Technology Laboratory - Machine Vision Laboratory - Intelligent SCADA Laboratory - Advanced Automation System Laboratory - Machine and Drive Design Laboratory
8. Intelligent Devices and Systems Research Unit (IDSRU)	<ul style="list-style-type: none"> - Photonics Technology Laboratory - Nano-Electronics and MEMS Laboratory - Solar Energy Technology Laboratory - Optical Thin-Film Laboratory
9. Institute of Technology for Persons with Disabilities and Elderlies (ITDE)	<ul style="list-style-type: none"> - Accessible Innovation and Universal Design Laboratory
10. Thai Organic and Printed Electronics Innovation Center (TOPIC)	



International Collaborations

NECTEC bilaterally and multilaterally collaborates with partners across the world through science, technology and innovation (STI) excellence projects/activities aimed at creating economic and social impacts and sustainable development.

Partners in Europe



Partners in Asia



Bilateral Cooperations



National Institute of Information and Communications Technology (NICT)



Department of Information Technology (reorganization), Ministry of Science and Technology, Lao PDR



Japan Advanced Institute of Science and Technology (JAIST)



Department of Information Technology (DIT), Ministry of Information and Communications, Bhutan



Japan Aerospace Exploration Agency (JAXA)



University Sains Malaysia (USM)



National Institute of Advanced Industrial Science and Technology
AIST



National Information Technology Park (NIPTP)



The University of Electro Communications (UEC)



School of Information Technology National University of Mongolia (NUM)



Chinese Academy of Sciences (CAS)



Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre



Beijing Jiaotong University (BJTU)



Université de Bourgogne, France



Fraunhofer-Gesellschaft (FhG), Germany

i-CREATE



Singapore Therapeutic, Assistive & Rehabilitative



Technologies (START) Centre, Singapore Korea International Exhibition Center (KINTEX) was



Global Ingogenesis
GyeongGi-Do
GyeongGi-Do

CONNECT2SEA



Advanced Science and Technology (ASTO), Department of Science and Technology Logo nectec



University (IT-VNU)



Badan Penelitian Dan Pengembangan Teknologi (BPPT)



Athens Technology Center (ATC)



Foundation for Research and Technology Hellas (FORTH)
Centre National de la Recherche



CNRS Scientifique (CNRS)



University Sainte Maloja (USM)
Information Technology Institute, Vietnam National

tu

technische universität
dortmund



WLCG
Worldwide LHC Computing Grid

ASEAN IVO



NIPICT



National University of Singapore



Ministry of Science, Technology and Environment



Asian Technology University (ATU)



UniNet



Vietnam National University



BPPT



NICI



ASSOCIATION OF SOUTHEAST ASIAN NATIONS

Spin-off Units

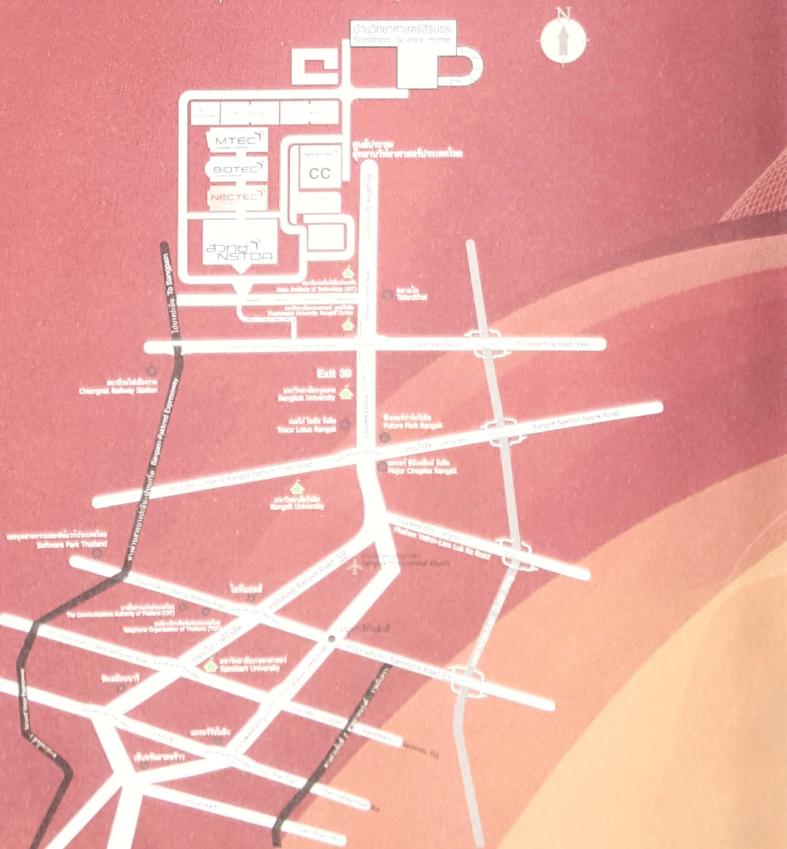
NECTEC has deployed the spin-off policy aimed to support some units/projects, that have the potential to grow and generate their own income autonomously to be spun off from NECTEC/NSTDA. These potential units/projects become establishing themselves as autonomous organisations or private companies. NECTEC's spin-off units are categorised into 2 groups as follows:

I. Autonomous organisations under the supervision of the Royal Thai Government

1. Software Park Thailand <http://www.swpark.or.th/>
2. Government Information Technology Services (GITS) <http://www.gits.net.th/>
3. Electrical and Electronic Products Testing Center (PTEC) <http://ptec.nectec.or.th/>
4. Hydro and Agro Informatics Institute (HAI) <http://www.haii.or.th/>

II. Private companies

1. Electric Vehicle (Thailand) Co., Ltd. <http://www.evthai.com/>
2. Internet Thailand Co., Ltd. <http://www.inet.co.th/>
3. Truehits <http://truehits.net/>
4. T-Net Co., Ltd. <http://www.tnetsecurity.com/>



NECTEC
a member of NSTDA

National Electronics and Computer Technology Center
112 Thailand Science Park, Phahonyothin Road,
Klong 1, Klong Luang, Pathumthani 12120, Thailand

Tel. +66 (0) 2-564-6900
Fax. +66 (0) 2-564-6901-3
Web site : <http://www.nectec.or.th/>

Appendix B

Scanning procedure algorithm

Algorithm 1 Pseudo code of the scanning procedure

Input: $h, \tilde{h}, h_t \in \mathbb{R}^+$
Output: θ_{arr} // Array representing the pattern of angles

```
1:  $\theta_{arr} \leftarrow []$ 
2: if  $h \leq \tilde{h} + h_t$  and  $h \geq \tilde{h} - h_t$  then
3:   return [0]
4: end if
5: if  $h \geq \tilde{h} + h_t$  then
6:    $h_1 \leftarrow \tilde{h} + h_t - h, h_2 \leftarrow \tilde{h} - h_t - h$ 
7: else
8:    $h_1 \leftarrow \tilde{h} - h_t - h, h_2 \leftarrow \tilde{h} + h_t - h$ 
9: end if
10:  $d_1, d_2 \leftarrow D - d$ 
11: while  $d_2 < d + D$  do
12:    $new\_theta \leftarrow Atan2(h_1, d_1)$ 
13:    $\theta_{arr}.append(new\_theta)$ 
14:    $d_2 \leftarrow \frac{h_2}{Tan(new\_theta)}$ 
15: end while
16: return  $\theta_{arr}$ 
```

Appendix C

Equations for circle fitting

$$\begin{aligned}
 A &= n \sum_{j=1}^n x_j^2 - \left(\sum_{j=1}^n x_j \right)^2 \\
 B &= n \sum_{j=1}^n x_j y_j - \left(\sum_{j=1}^n x_j \right) \left(\sum_{j=1}^n y_j \right) \\
 C &= n \sum_{j=1}^n y_j^2 - \left(\sum_{j=1}^n y_j \right)^2 \\
 D &= 0.5 \left\{ n \sum_{j=1}^n x_j y_j^2 \left(\sum_{j=1}^n x_j \right) \left(\sum_{j=1}^n y_j^2 \right) + n \sum_{j=1}^n x_j^3 - \left(\sum_{j=1}^n x_j \right) \left(\sum_{j=1}^n x_j^2 \right) \right\} \\
 E &= 0.5 \left\{ n \sum_{j=1}^n y_j x_j^2 \left(\sum_{j=1}^n y_j \right) \left(\sum_{j=1}^n x_j^2 \right) + n \sum_{j=1}^n y_j^3 - \left(\sum_{j=1}^n y_j \right) \left(\sum_{j=1}^n y_j^2 \right) \right\} \\
 a_M &= \frac{DC - BE}{AC - B^2} \\
 b_M &= \frac{AE - BD}{AC - B^2} \\
 r_M &= \frac{1}{n} \sum_{j=1}^n \sqrt{(x_j - a_M)^2 + (y_j - b_M)^2}
 \end{aligned}$$

with n the number of points, x_j and y_j the coordinates of the j-th point, a_M and b_M the position of the fitted circle's center and r_M its radius.

Appendix D

Magnetometer calibration algorithm

Algorithm 2 Pseudo code of the magnetometer calibration

```
1:  $min, max \in \mathbb{R}^3$ 
2:  $min \leftarrow \{+\infty, +\infty, +\infty\}$ 
3:  $max \leftarrow \{-\infty, -\infty, -\infty\}$ 
4: while  $True$  do
5:   if  $is\_raw\_data\_received$  then
6:      $mag \leftarrow read\_magnetometer()$ 
7:     for  $i \leftarrow 1; i < 3; i \leftarrow i + 1$  do
8:       if  $mag[i] > max[i]$  then
9:          $max[i] \leftarrow mag[i]$ 
10:      else if  $mag[i] < min[i]$  then
11:         $min[i] \leftarrow mag[i]$ 
12:      end if
13:       $offset \leftarrow \frac{max[i] + min[i]}{2}$ 
14:       $calibrated\_mag[i] \leftarrow \frac{mag[i] - offset}{max[i] - min[i]}$ 
15:    end for
16:  end if
17: end while
```

Bibliography

- [1] P. Treleaven and J. Wells, “3d body scanning and healthcare applications,” *Computer*, vol. 40, pp. 28–34, jul 2007.
- [2] Y. Jang, H. W. Noh, I. B. Lee, Y. Song, W. I. Jang, and S. Lee, “Development of an integrated obesity management waist belt system composed of calorie tracking and waist circumference measuring module for long term monitoring,” *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2011, pp. 2172–2175, 2011.
- [3] G. F. Jost, J. Walti, L. Mariani, S. Schaeren, and P. Cattin, “Inertial Measurement Unit-Assisted Implantation of Thoracic, Lumbar, and Sacral Pedicle Screws Improves Precision of a Freehand Technique,” *World Neurosurgery*, vol. 103, pp. 11 – 18, 2017.
- [4] United States Consumer Product Safety Commission, “Database: Anthropometry of infants, children and youths to age 18 for product safety design.” www.humanics-es.com/recc-children.htm, 1977.
- [5] US Army, “Database: Anthropometric survey of us army personnel (ANSUR II).” www.openlab.psu.edu/ansur2/, 2012.
- [6] G. G. Slabaugh, “Computing euler angles from a rotation matrix,” Jan. 1999.
- [7] X. Tang, W. Huang, X. Li, S. Li, and Y. Liu, “Outlier Detection via Minimum Spanning Tree,” *PACIS 2016 Proceedings*, June 2016.
- [8] S. Sotoodeh, “Hierarchical clustered outlier detection in laser scanner point clouds,” *IAPRS*, vol. 36, 01 2007.
- [9] D. W. Matula and R. R. Sokal, “Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane,” *Geographical Analysis*, vol. 12, pp. 205–222, jul 1980.
- [10] W. S. Cleveland, “Robust locally weighted regression and smoothing scatterplots,” *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979.
- [11] H. P. Ho, Y. Chen, H. Liu, and P. Shi, “Level set active contours on unstructured point cloud,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 655 – 662 vol. 2, 07 2005.
- [12] H. Lin, W. Chen, and G. Wang, “Curve reconstruction based on an interval b-spline curve,” *The Visual Computer*, vol. 21, pp. 418–427, jul 20005.
- [13] W. Wang, H. Pottmann, and Y. Liu, “Fitting B-spline Curves to Point Clouds by Curvature-based Squared Distance Minimization,” *ACM Trans. Graph.*, vol. 25, pp. 214–238, Apr. 2006.

- [14] I.-K. Lee, “Curve reconstruction from unorganized points,” *Computer Aided Geometric Design*, vol. 17, no. 2, pp. 161 – 177, 2000.
- [15] D. Umbach and K. N. Jones, “A few methods for fitting circles to data,” *IEEE Transactions on Instrumentation and Measurement*, vol. 52, pp. 1881–1885, Dec. 2003.
- [16] Y. Pekelny, “Implementation of an algorithm of fitting b-spline curve to cloud points.” www.cs.technion.ac.il/~cs234326/projects/BspFitting_website/index.html, Oct. 2005.
- [17] V. V. Mahla, “Fitting b-splines curves to coarse data cloud using modified squared distance minimization method.” uta-ir.tdl.org/uta-ir/bitstream/handle/10106/26161/MHALA-THESIS-2016.pdf?sequence=1, Aug. 2016.
- [18] H. Yang, W. Wang, and J. Sun, “Control point adjustment for b-spline curve approximation,” *Computer-Aided Design*, vol. 36, pp. 639 – 652, June 2004.
- [19] FaBo, “Fabo 9 axis mpu9250 library.” github.com/FaBoPlatform/FaBo9AXIS-MPU9250-Library.
- [20] V. Renaudin, M. H. Afzal, and G. Lachapelle, “Complete triaxis magnetometer calibration in the magnetic domain,” *Journal of Sensors*, vol. 2010, p. 10, 2010.
- [21] J. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, “A geometric approach to strapdown magnetometer calibration in sensor frame,” *IFAC Proceedings Volumes*, vol. 41, no. 1, pp. 172 – 177, 2008. 2nd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles.
- [22] M. Kok and T. B. Schon, “Magnetometer calibration using inertial sensors,” *IEEE Sensors Journal*, vol. 16, pp. 5679–5689, July 2016. arXiv: 1601.05257.
- [23] x-io Technologies, “Open source imu and ahrs algorithms.” x-io.co.uk/open-source-imu-and-ahrs-algorithms.
- [24] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear Complementary Filters on the Special Orthogonal Group,” *IEEE Transactions on Automatic Control*, vol. 53, pp. 1203–1217, June 2008.
- [25] R. G. Valenti, I. Dryanovski, and J. Xiao, “Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs,” *Sensors (Basel, Switzerland)*, vol. 15, pp. 19302–19330, Aug. 2015.
- [26] A. Cavallo, A. Cirillo, P. Cirillo, G. D. Maria, P. Falco, C. Natale, and S. Pirozzi, “Experimental Comparison of Sensor Fusion Algorithms for Attitude Estimation,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7585 – 7591, 2014.

List of Figures

1.1	Weight scale for BMI measurement in Thailand	9
1.2	Schematic of the solution made of an array of distance sensors	11
1.3	Schematic of the solution made of set of 3 LIDARs	12
1.4	Illustration of the LIDARs tilting for patients with different statures	12
1.5	Prototype of a pole with a LIDAR mounted on a servomotor	13
1.6	Flow chart of my internship's tasks on the automatic health checkup project . .	14
1.7	Common screwdrivers for surgery purpose	15
2.1	Basic architecture of the developed system	16
2.2	Electronic circuit diagram for the servomotors' control	17
2.3	Simulations of the user's navel detection	18
2.4	Patient navel's height depending on their stature, weight and age	19
2.5	Distribution of the estimated navel height's error	20
2.6	Top view of the system	21
2.7	Side view of the system	22
2.8	Cylinder cut from an higher source	22
2.9	Side view of the system with scanning procedure explanation	23
2.10	Mechanism of the LIDAR being tilted	25
2.11	Example of 3D a point cloud	25
2.12	Example of a detection via MST on a point cloud	26
2.13	Example of a detection via MST on a point cloud with not homogeneous density	27
2.14	Example of process used to get a Gabriel's graph	27
2.15	Example of a Gabriel's graph	28
3.1	Examples of generated testing shapes	30
3.2	Transformation from 2D parametric space to linear representation	31
3.3	Example of a LOESS curve for different values of smooth factor f	33
3.4	Example of a non representative curve of the user's waist shape	34
3.5	Two examples of the sequence joining method's implementation	35
3.6	Issue with the sequence joining method implementation when the point cloud is not homogeneously dense	36
3.7	Method to split the cloud in n equal section	38
3.8	Initial B-spline results	39
3.9	Method for moving manually the control points	43
3.10	B-spline fitting in a basic point cloud	44
3.11	Influence of the number of control points on the curve	45
3.12	Influence of regularization coefficient's value on the curve	45
3.13	Influence of point cloud's sharpness on the curve	46
3.14	Comparison of the LOESS and SDM method	46
3.15	Test of the outlier detection	47
4.1	Measurement and shape reconstruction of my own waist	48

4.2	Test in real conditions with a dummy	48
5.1	Basic architecture of the developed system	49
5.2	Overview of the software	50
5.3	Results of the magnetometer calibration	51
5.4	Example of the IMU's hull on which the axes orientation is displayed	53
5.5	IMU's assistant for screw implantation orientation finding	54
A.1	Distribution of the NSTDA's staff	58

List of Tables

2.1	Correlation of various anthropomorphic data among each others	20
2.2	Error of the estimated navel's height	21
3.1	Two interesting B-spline computing libraries provided by <i>Scipy</i>	38
3.2	Coefficients $D_{j,x/y}$ of derivative by $D_{i,x/y}$ when $d < 0$	41
3.3	Coefficients $D_{j,x/y}$ of derivative by $D_{i,x/y}$ when $0 \leq d < \rho$	42