



FACULTAT
**DE CIÈNCIES
I TECNOLOGIA**

UVIC | UVIC·UCC

Degree Final Project

*Improved alternative for detecting SMC
tape in TSLF cables using LBP
descriptors*

Aleix López Llauradó

Degree in Mechatronics Engineering

Tutor: Dr. Moisès Serra Serra

Vic, June 2019



DEGREE FINAL PROJECT SUMMARY
DEGREE IN BIOTECHNOLOGY

Title: *Improved alternative for detecting SMC tape in TSLF cables using LBP descriptors*

Key words: *Computer Vision, LBP Descriptor, Texture Filtering, HSV Color Calibration*

Author: Aleix López Llauradó

Tutor/s: Dr. Moisès Serra Serra i Joan A. Pou

Date: Juny de 2019

During the manufacture of the TSLF cable, in one of the last phases of production, the cable must be covered with a sealing semiconducting tape (*SMC Tape*), the application of this tape, can cause defects in the manufacturing, currently there is no system of detection for this defect.

This thesis aims to study, develop and test a feasible detection system that can improve the current quality control and reduce the cost associated if the defect occur.

Before getting the final working algorithm that better fits the needs of this system, research and test was done in order to find a reliable and simple solution, firstly all the tests were developed with MATLAB thanks to its huge package of pre-build functions, once the final algorithm was defined, the first iteration of the code was translated to Python, a more versatile, light and fast programming language.

The results show, a good performance over each iteration. The algorithm perform a good overall detection but it is not robust enough to be implemented in the industry sector. This system is a starting point to improve the current system, although a new software revision is under evaluation (*June, 2019*). The current system, has a in-build database, which saves al the photo frames, the big future step is to train an artificial intelligence to improve even more its performance and increase the varaivility of the detections.

Dedication

I would like to dedicate my work to all of those who trusted in me, believing that I was/am capable to do grater things and achieve every milestone I set.

Thanks father, mother, sister, grandmothers, grandfathers (*although they are no longer with us*), uncles, aunts, and Free (*my beloved dog*), you taught me to never surrender, even if the wind blows against my will, to keep fighting for what I truly believe, to pursue my dreams, sacrificed your own time for me, and even more.

Thanks to all the professors I had during primary school, high school, baccalaureate and University, the list of names is long and probably I will leave someone behind, thanks to all the time and patience you invested on me, Specially Moisès Serra Serra, my advisor for this thesis and professor, thanks for the advices given, and all the time invested in me and the project.

Finally, thanks Joan A. Pou and Manuel Mora, for giving me the opportunity to start this project, support me and teach me during all this time.

This milestone is thanks to all of you, thanks for keep me on track and shape my future as an engineer.

Sincerely,
Aleix López Llauradó

Table of Contents

Contents

1	Introduction	1
1.1	Introduction	1
2	Objectives and Specifications	3
2.1	Objectives	3
2.1.1	Intern objectives	3
2.1.2	Project objectives	3
2.1.3	Secondary objectives	3
2.2	Project Specifications	3
3	Resources	4
3.1	Programming languages	4
3.1.1	MATLAB	4
3.1.2	Python	4
3.2	Hardware	5
3.2.1	Computing System	5
3.2.2	Image Acquisition	6
3.2.3	Sub-Systems and Components	7
4	Theoretical Basis	8
4.1	Algorithms	8
4.1.1	Filter Masks	8
4.1.2	LBP: Local Binary Patterns	10
4.1.3	Rellevant sub-algorithms	12
4.2	Sockets	13
5	Project Development	14
5.1	Hardware	15
5.1.1	Chassis	15
5.1.2	Light source	15
5.2	Software	16
5.2.1	Main SBC	16
5.2.2	SBC camera system	18
6	Methodology	20
6.1	Evolution	20
6.2	Revisions	21

7	Discussion	22
7.1	Results	22
7.2	Conclusions	26
7.2.1	Overall	26
7.2.2	Opinion	26
7.2.3	Next steps	27

List of Tables

1	Table with Raspberry Pi 3 B+ Specifications [1] [2]	5
2	Table with Camera module V2 Specifications [3]	6
3	Table with Raspberry Pi Zero W Specifications [2]	7
4	Table with Results	24

List of Figures

1	TSLF Samples	1
2	Color spaces comparison	8
3	3x3 pixel neighborhood [4]	10
4	LBP encoding procedure [4]	10
5	Dilation: a 3x3 square structuring element [5]	12
6	Erosion: a 3x3 square structuring element [6]	12
7	Socket connection protocol diagram	13
8	Complete system diagram	14
9	Illustration of a possible design	15
10	Main SBC pseudo-code	16
11	Camera System pseudo-code	18
12	Types of labels for the evaluation	22
13	Insulation exposure due a bad SMC Tape overlap	23

List of Acronyms

Acronym	
SMC	Semi-Conductive Material
OS	Operative System
SBC	Simple Board Computer
Gb	GigaByte, the equivalent of 1.000.000.000 bytes.
SDRAM	Synchronous Dynamic Random-Access Memory
AEC	Automatic Exposure Control
AEW	Automatic White Balance
ISP	Image Signal Processing
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet protocol
HSV	Color Space (<i>Hue, Saturation, Value</i>)
LBP	Local binary pattern
TP	True positive
TN	True negative
FP	False positive
FN	False negative
Acc	Accuracy
Pr	Precision
Sn	Sensibility
Sp	Specificity

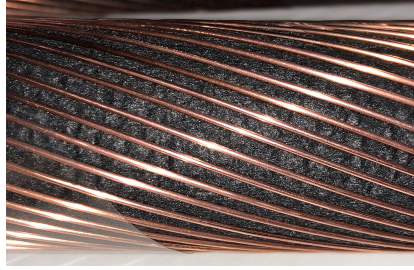
1 Introduction

1.1 Introduction

This thesis aims to study, develop and test a new control system fro a production line which is able to detect an incorrect overlap of a SMC tape. As the correct market fails to offer a trustable and affordable solution in order to address issues in the good placement of the SMC tape.

the methodology of this thesis has its bases in a high speed computer vision system. Because it's currently impossible to fix an incorrect overlap, the system does not need to be real-time , therefore it can perform a discrete analysis.

At the present time, during the production of the cable, the operator has to check manually the condition of the tape, nevertheless as it is a in-line control, it's hard to keep control of the whole production.



(a) TSLF sample



(b) TSLF sample with bad overlap

Figure 1: TSLF Samples

In addition, due tot he numerous parts involved in this process, it is hard for the operator to identify clearly, the correct overlap of the SMC tape.

This proposed system, aims to reduce the participation of the operator in the quality control, resulting in a more optimized process, reducing stress, reducing the probability of failure and a gain of time for other tasks to be done.

The system will be a set of cameras, which will analyze the surface of the cable in a discrete manner.

June 5, 2019

Outline

Chapter 2 Explain all the Objectives to be accomplished during this project and the specifications which need to achieve.

Chapter 3 A brief introduction to every part that will be used, software and hardware explained.

Chapter 4 Explanation of the mathematical bases applied in this project in order to accomplish a correct detection.

Chapter 5 Detailed hardware and software parts, and results.

Chapter 6 How this project was developed and explaining each iteration.

Chapter 7 Closure of this thesis with an overall opinion and self-criticism.

2 Objectives and Specifications

2.1 Objectives

2.1.1 Intern objectives

1. To study the current feature detection algorithms.
2. To study the software possibilities which can be applied.
3. To study the hardware possibilities which can be applied to this project.

2.1.2 Project objectives

1. To develop, build and test a system able to detect the overlapping of the SMC tape.

2.1.3 Secondary objectives

1. To build a system Hardware-ready for the implementation of future detection algorithms.
2. To evaluate the efficiency of the algorithm during production.
3. To explore a low cost but high efficiency detection algorithm.
4. To communicate the system with the PLC of the production machine.

2.2 Project Specifications

For this project there were a set of specifications to achieve.

1. To take a photo at least every 5 minutes.
2. To alert the operator if the system detects an incorrect SMC overlap.
3. To visualize the detection during the fabrication process.

3 Resources

3.1 Programming languages

This section will discuss and explain the current software that has been used along the project.

3.1.1 MATLAB

MATLAB It is a suitable programming architecture for prototyping and early testing purposes due to the tools provided and the big arrange of solutions coming with the software, also MATLAB important pre-build functions and all this capabilities are offered within the base program.

MATLAB is also a mathematical software designed and optimized for matrices operation, because images and videos are a combination of different matrices, this characteristics makes MATLAB indeed suitable for the development of the detection system. Also MATLAB has a big community behind, and a huge documentation page, where it can be found from the description of a function and how to use it, to full-examples in every industry segment, from face-recognition, to signal processing, thanks to this, all the research and development was done easily and effectively.

Unfortunately MATLAB has a major flaw, because it is a high complex program, the executing time of an algorithm is longer compared against other programming languages, such as Python or C/C++, for this reason MATLAB was only used for developing and testing purposes during the project development.

3.1.2 Python

Python, because it is a interpreted language it can be executed in all operative systems this characteristic allows to program and test the algorithms in one operative system (*Windows OS*) so later on it can be implemented in Linux (*Raspbian OS*).

Python's large standard library, commonly cited as one of its greatest strengths [7], provides tools suited to do many tasks, for this case, Python has a computer vision library (*OpenCV*), sockets library and a dedicated library for mathematics operations which optimizes this language for matrix operations (*Numpy*).

Python, thanks to this features was the second step during the development of this project, and as it will be explained later, it is the programming language that has been used to test and build all the current project (*Chapter 6*).

Python was a programming language, initially thought to perform the first test with the final algorithm, but for reasons explained in the following chapters, was decided to stay with Python because it's current efficiency.

3.2 Hardware

3.2.1 Computing System

Initially, it was decided to use a personal computer with windows OS built-in, but as said before, because the algorithm was write in Python, it is suitable to build a Linux OS environment, for this reason and due the low computational cost has the current program it was decided to use a Raspberry Pi 3 B+. It is simple SBC quad-core processor with 1 Gb of SDRAM, with this hardware is enough to process all the data. Supposing that in a further future other algorithms may be implemented in this system, it can be implemented an expansion computing module, which allows the current raspberry to increase a total of 4 extra cores processors running at 1.2GHz and 1 Gb of SDRAM [8].

Table 1: Table with Raspberry Pi 3 B+ Specifications [1] [2]

SOC	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
CPU	1.4GHz 64-bit quad-core ARM Cortex-A53 CPU
Storage	MicroSD
RAM	1GB LPDDR2 SDRAM
Wireless	Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz) and Bluetooth 4.2
Ethernet	Gigabit Ethernet over USB 2.0 (max 300 Mbps). Power-over-Ethernet support (with separate PoE HAT). Improved PXE network and USB mass-storage booting.
Thermal Management	Yes
Video	Yes – VideoCore IV 3D. Full-size HDMI
Audio	3.5 mm jack
USB	2.0 4 ports
GPIO	40-pin
Power	5V/2.5A DC power input
OS support	Linux and Unix
Dimensions	85.60 mm x 56.5 mm x 17 mm
Weight	45 g
Price	\$ 35

3.2.2 Image Acquisition

For the image acquisition sensor it has been used a Raspberry Pi Camera V2, it has a Sony IMX219 8-megapixel sensor. This camera is suitable as it is classified as board level module, meaning that it has a small size and it is easy to fit in a new system design.

Another advantage from this sensor, it is that all the parameters of the camera (*AEC*, *AWB*...) can be modified via ISP, this feature is helpful in order to process the image before passing it to the algorithm.[9]

Table 2: Table with Camera module V2 Specifications [3]

Image Sensor	Sony IMX 219 PQ CMOS image sensor in a fixed-focus module.
Resolution	8-megapixel
Still picture resolution	3280 x 2464 pixels
Max image transfer rate	1080p: 30fps (encode and decode)
Connection	15-pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2).
Image control functions	Automatic exposure control, Automatic white balance, Automatic band filter, Automatic 50/60Hz luminance detection, Automatic black level calibration
Temp. Range	Operating: -20°C to 60°C
Lens size	1/4"
Dimensions	23.86 mm x 25 mm x 9 mm
Weight	3 g
Price	\$ 25

3.2.3 Sub-Systems and Components

In order to communicate the camera module with the computing system, a Raspberry Pi Zero W has been chosen in order to send the data, This SBC has less computational power than the raspberry Pi 3 B+, but is enough to run the camera module and to communicate with the main computing system.

This model has been chosen because it has wireless capabilities, in other words it has a Wi-Fi card being able to communicate both systems via TCP/IP sockets (*Chapter 4, Section 2*).[10]

Table 3: Table with Raspberry Pi Zero W Specifications [2]

SOC	Broadcom BCM2835 32-bit architecture SoC
CPU	1GHz 32-bit ARM1176JZF-S CPU
Storage	MicroSD
RAM	512Mb LPDDR2 SDRAM
Wireless	802.11b/g/n single band 2.4 GHz wireless, Bluetooth 4.1 BLE
Ethernet	No
Thermal Management	Yes
Video	Yes – Mini-HDMI, 1080p60
Audio	Yes - Mini-HDMI, stereo audio through PWM on GPIO
USB	1 Micro-USB (direct from BCM2835 chip)
GPIO	17-pin
Power	5V/0.5A DC power input
OS support	Linux and Unix
Dimensions	65 mm x 30 mm x 5 mm
Weight	9 g
Price	\$ 10

4 Theoretical Basis

4.1 Algorithms

This block aims to analyze the algorithm used for this project and explain other sub-algorithms which have a relevant functions during all the computational analysis.

4.1.1 Filter Masks

Color subtraction, is an algorithm that exclude all colors but the one selected, using a comparison procedure, because of this reason it is a simple task that can be performed in a SBC.

Normaly a change in the color space is needed in order to easily perform this task, the current and most optimal procedure is to work with the HSV/HSB color space eq.(1) [11]. By comparing both color spaces (*RGB and HSV*), it may be seen that in RGB color space, in order to isolate a color it is needed a combination with the three sub-spaces R, G and B (*Assuming it is not a Red, Green or Blue*).

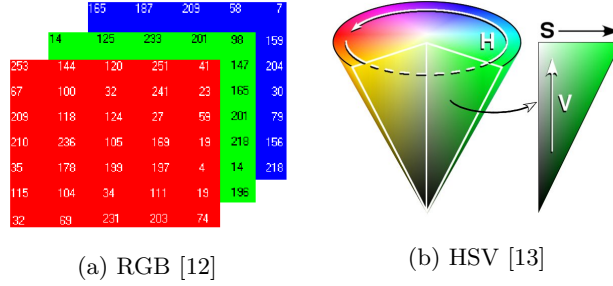


Figure 2: Color spaces comparison

$$\begin{aligned}
 V &= \max(R, G, B) \\
 S &= \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0, & \text{otherwise} \end{cases} \\
 H &= \begin{cases} \frac{60(G-B)}{V - \min(R, G, B)} & \text{if } V = R \\ \frac{120 + 60(G-B)}{V - \min(R, G, B)} & \text{if } V = G \\ \frac{240 + 60(G-B)}{V - \min(R, G, B)} & \text{if } V = B \end{cases}
 \end{aligned} \tag{1}$$

The generated mask, is obtained comparing the current parameter values with the current pixel value of the image, see eq.(2).

$$\begin{aligned}
M_{Hij} &= H_{ij} \geq Ch1m \quad \& \quad H_{ij} \leq Ch1M \\
M_{Sij} &= S_{ij} \geq Ch2m \quad \& \quad S_{ij} \leq Ch2M \\
M_{Vij} &= V_{ij} \geq Ch3m \quad \& \quad V_{ij} \leq Ch3M
\end{aligned} \tag{2}$$

Where, M_{Hij} is a pixel value in a determinate position of the mask image, $Chnm$ is the minimum parameter value for a specific layer (H, S, V) , and $ChnM$ is the maximum value for a specific layer (H, S, V) .
Where n in $Ch(n)m$ and $Ch(n)M$, $n = \{1, 2, 3\}$

4.1.2 LBP: Local Binary Patterns

Local binary patterns (*LBP*) is a visual descriptor used in computer vision for a classification purposes. LBP is a particular case of a Texture Spectrum model proposed in 1990. [14]

Its a very simple and yet very efficient operator, which labels every pixel of an image by thresholding each neighborhood pixel and consider the result as a binary number.

The original LBP operator, was described as a 3x3 neighborhood (Fig. 3), that compares each pixel within its eight neighbors by subtraction with the center pixel. The results of this comparison is encoded as 0 if the values are negative, and encoded as 1 if higher or equal to 0 (Fig. 4). After the encoding, this number is converted to decimal and replaces the center pixel value, see eq.(3,4).

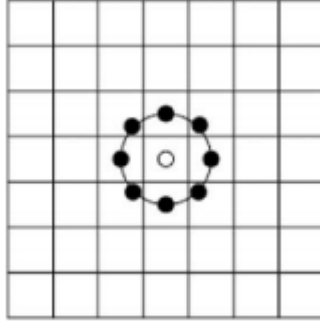


Figure 3: 3x3 pixel neighborhood [4]

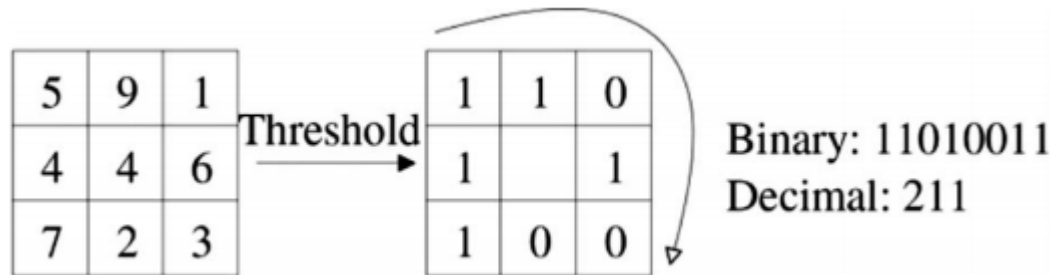


Figure 4: LBP encoding procedure [4]

Formally, given a pixel at (x_c, y_c) , the resulting LBP can be expressed in decimal form as follows:

$$LBP_{P,R(x_C,y_C)} = \sum_{P=0}^{P-1} s(i_P - i_c)2^P \quad (3)$$

where i_c and i_P are, respectively, gray-level values of the central pixel and P surrounding pixels in the circle neighborhood with a radius R , and function $s(x)$ is defined as: [4]

$$s = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (4)$$

This basic LBP operator is invariant to monotonic gray-scale transformations, the operator preserve the order of the pixel intensity in the local neighborhood. Histograms from regions that has been operated can be exploited as a texture descriptor.

4.1.3 Relevant sub-algorithms

Dilation (represented as \oplus), is a basic operation in mathematical morphology. It was first developed for binary images, expanded to gray-scale images and finally to complete lattices (*color images or set of images*). Mathematical morphology see a binary image as a subset of a Euclidean space R^d or the integer grid Z^d , for some dimension d , see eq.(5). [15]

$$A \oplus B = \bigcup_{b \in B} A_b \quad (5)$$

Dilation operator produces a new binary image $g = f \oplus s$ with ones in all locations (x, y) of a structuring element origin at which that structuring element hits the input image f , i.e. $g(x, y) = 1$ if s hits f and 0 otherwise, repeating for all pixel coordinates (x, y) . [16]

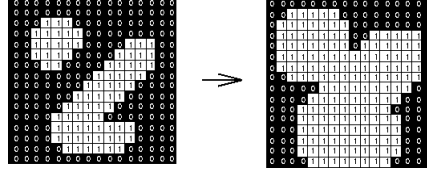


Figure 5: Dilation: a 3x3 square structuring element [5]

Erosion (represented as \ominus), is the other fundamental operation in morphological mathematics, this two types of operations define all others operators based on morphological transformations. As mentioned before, it was defined for binary images, later adapted for gray-scale images, and currently used in complete lattices. [15]

$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad (6)$$

Where B_z is the translation of B by the vector z .

The dilation of an image f by a structuring element s produces a new binary image $g = f \ominus s$ with ones in all locations (x, y) of a structuring element origin at which the structuring element s hits the the input image f , i.e. $g(x, y) = 1$ if s hits f and 0 otherwise, repeating for all pixel coordinates (x, y) . [16]

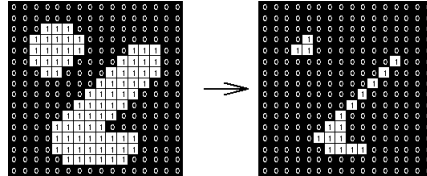


Figure 6: Erosion: a 3x3 square structuring element [6]

4.2 Sockets

Socket is an abstract concept, which permits two different programs to communicate with each other, transmitting arranged data. Distinctions between a socket (*internal representation*), socket descriptor (*abstract identifier*) and socket address (*public address*) are not correctly distinguished in everyday usage, in this project it has been implemented a TCP/IP socket, in order to be able to establish a binding point between two or more nodes (*endpoints connections*), the internet socket is characterized with:

- Local socket address, consisting of the local IP address and a port number.
- Additionally, a protocol can be defined, TCP, UDP or raw IP.

In the following figure is represented the protocol to establish connection between a server and a client (*both endpoint nodes*).

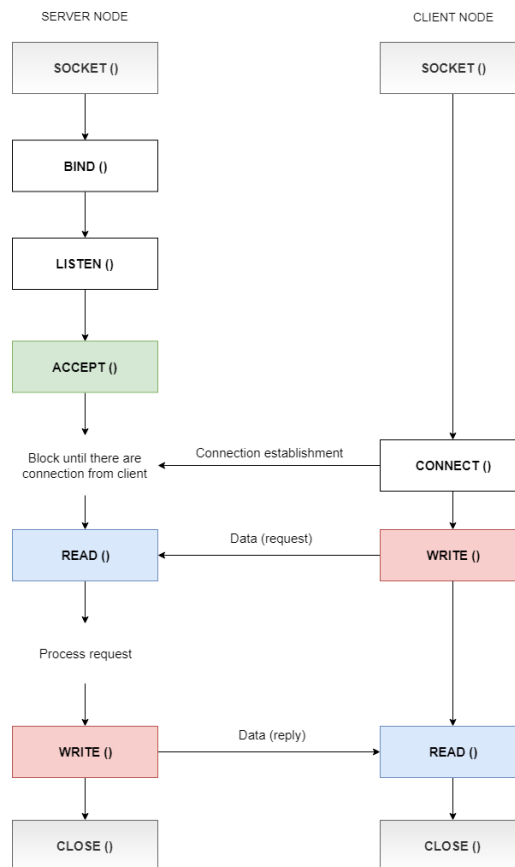


Figure 7: Socket connection protocol diagram

5 Project Development

This part of the thesis aims to explain, taking into consideration to General Cable property, the final design of the system, the code used to perform the detection and the interface that shows the current analysis.

Before going deep to both software and hardware parts, the following figure shows the complete system structure.

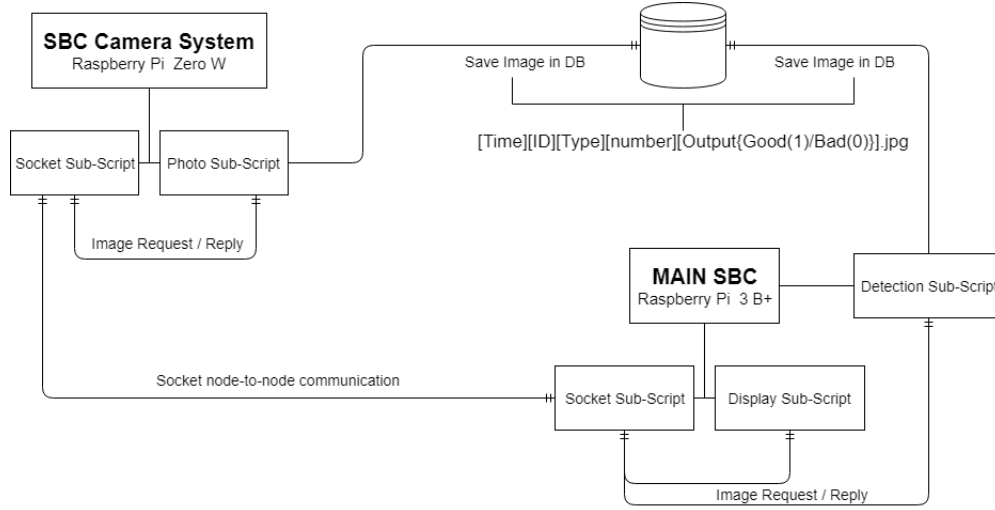


Figure 8: Complete system diagram

Note, that the interconnections with blocks, which are marked with a double orthogonal line at the end, means that communicate and pass information with one to another, if only one end is marked, denotes a single-way communication, and if either ends are not marked, defines a subgroup part from either **SBC Camera System** or **Main SBC**.

5.1 Hardware

5.1.1 Chassis

For this project, was necessary to design a chassis to fit all the electronic components, because the system must cover all the cable surface, the design has to manage all the cables, also because it has several amount of cameras along the perimeter in order to cover the cable surface, the chassis for the camera system has to be modular and easy to replace in case of any part has a malfunction or broke due to fatigue or other causes.

For the Display and the main SBC, there is no need to design a custom chassis, because it does not have any special consideration.

5.1.2 Light source

The light is a crucial factor for this project, and mostly all computer vision projects. This system needs two types of light, A global illumination and a high beam light source, in the following section this need will be explained (*Chapter 5, Section 2*).

For the global illumination light, it has been designed a halo with an arrangement of LED's that covers and offers a complete coverage and homogeneous light source on top the cable surface, in addition, in order to remove unwanted shadows, another halo has been placed in front. The overall illumination systems, has a front and rear halo light, both of them with a light diffuser in order to stabilize even more the light source.

The high beam light, is obtained, coupling a pair of high intensity LED's to the sides of the camera, by integrating a diffuser into the high beam LED's, we can obtain a homogeneous high beam of light that will not generate any high reflection into the cable.

The following figures show the final camera mount with the set of lights and the current chassis, these are just illustrations, and design might vary.

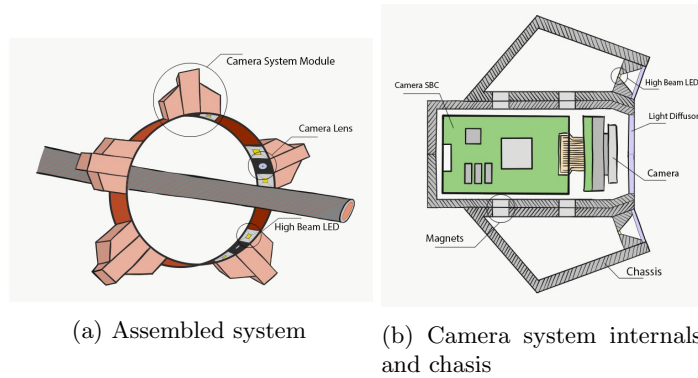


Figure 9: Illustration of a possible design

5.2 Software

This part aims to explain through software diagrams, all the parts and scripts involved in the system, because this project was developed with the collaboration of the entity General cable, the source code will not be published.

5.2.1 Main SBC

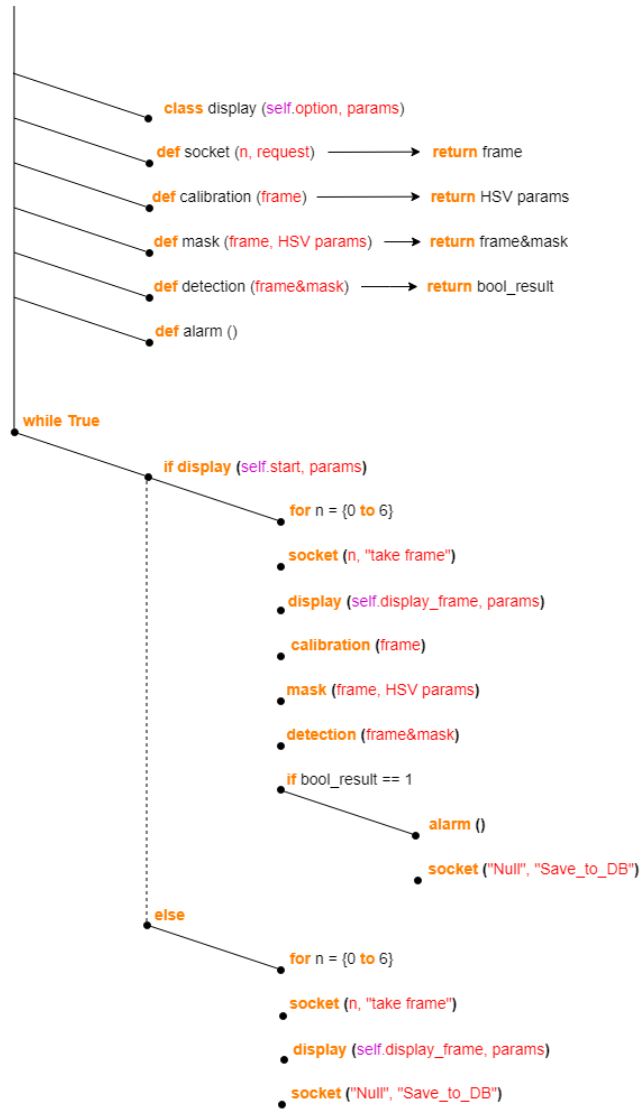


Figure 10: Main SBC pseudo-code

As it can be seen in this chart (*Fig.9*), the main system module has a set of main functions, that communicate with each other in order to obtain the final detection, in the following paragraphs, each function will be explained.

Display, This object class, it's the main part of the system, it interconnects all the functions inside the system and communicate to each function correctly. This class has two sub-functions inside, start function and update function.

If the start function is selected, it initialize a inner timer, and every 30 seconds it performs one analysis of the current frame, during each analysis the class forces the update function and updates the current frame shown on the screen and also save the current frame in the database.

Socket, This function generate a TCP/IP socket connection with the camera modules, because the system has a total of 6 cameras, this function discriminates each camera every time it has to return a frame from each camera, also depending the request, it returns a camera frame or save the current frame in the database.

Calibration, This function is key during the whole analysis, because it returns the proper parameters in order to post-process the frame in order to evaluate the LBP algorithm.

This current function evaluates each frame and iterates some hyper-parameters (*blur, Hue, Saturation, Value, scale*) and for each iteration it generates a internal score, based on a true binary mask, both, calibrated frame mask and the true binary mask are overlapped and the function checks the pixel similarity between both frames. The parameters that return this function are those that has the highest similarity with the true binary mask.

Once this function has been called once, it saves the highest score, and for the following calibrations it checks those hyper-parameters first, if the final score are lower than expected, the whole calibration is performed again.

Mask, with the current frame and the HSV parameters returned from the calibration function it generates a binary masks, and with the current frame it performs a pixel comparison. The frame returned is the final frame operated (*frame@mask*).

Detection, with the frame post-processed the system performs the LBP algorithm, returning a Boolean data type variable (*1 or 0*).

Once the LBP algorithm is done, the system try to find possible contours areas(*For more information, check references: [17], [18], [19]*), if these areas are higher than a pre-established threshold, the Boolean data type variable get assigned as 1 (*True*), otherwise, stays as False (*0*).

Alarm, this function is a software interrupt, that sends a signal via GPIO connection to an alarm in order to alert the worker that a bad overlap has been detected.

5.2.2 SBC camera system

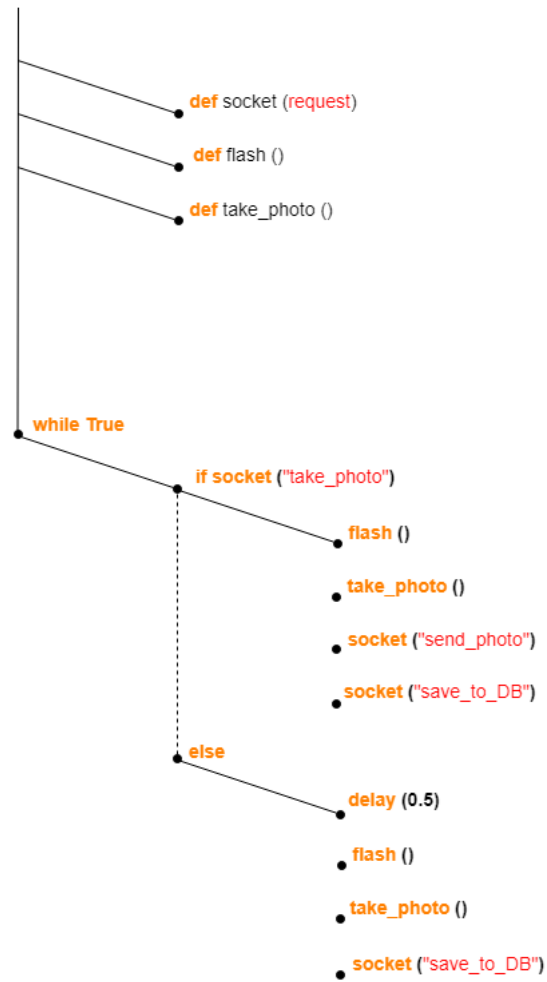


Figure 11: Camera System pseudo-code

As it can be seen in this chart (*Fig.10*), the camera system module has a set of main functions, that communicate with each other in order to take a correct photo and send it or save it in the database, in the following paragraphs, each function will be explained.

Socket, this function is the client side of the socket connection. this function once it is called, it establishes connection with the server (*Main SBC*), also this function parses the photo frame in order to send it correctly to the server.

Flash, as the function name suggest, with this function the system starts a set of LED's, the high beam ones in order to, take the photo, this function it has been implemented in order to increase the life of the high beam LED's, and to not disturb the worker continuously with a high luminosity.

Take_photo, this function connects with the camera and takes the photo, also this function establishes the photo format along with some parameters in order to make the photo with the right exposure, shutter speed, brightness...

6 Methodology

6.1 Evolution

All the current project and its older iteration has been firstly done with MATLAB on account of all the tools that already has build-in, for this reason it was easy to iterate, test various algorithms and select the algorithm that better fits the needs for this project. Once the early stage of prototyping and testing was terminated and the final algorithm was found, with MATLAB, the whole algorithm was made by hand, evading as many pre-build functions as possible in order to reduce execution time, this step was crucial in order to correctly translate the code to Python and also to make the system as fast as possible. In this stage once the algorithms were functional, an upper layer of the system was designed in order to fuse all the system algorithms and provide a final interface to interact with the final customer.

The initial idea was, instead of using Python language, to use C/C++ programming language, a more faster executing code, thanks to its low level programming capabilities, but it was not estimated because with Python it was faster than expected, because as it will be discussed later, the algorithms were designed as low mathematical level as possible, trying to evade pre-build functions in order to decrease execution time.

Another reason that were studied later is the enhancement of functionality, with Python, increase performance, test and add new algorithms are tasks much easier than C/C++, also if at some point some values has to be changed in the current code because the environment is different or any other reason, with Python because it is more readable than C/C++ it will be easier for the person in charge of the maintenance of the system. As it was said before, the hardware of this current system is capable to run other algorithms and be able to detect other important features for this or other types of cables.

6.2 Revisions

Once the initial algorithm was implemented in Python, a several additions to the system as well as some improvements to the actual algorithm were made in order to improve the overall system, the initial improvements were hardware focused, and the last two improvements were related to software.

Revision 1, the first iteration of the system, just had one LED halo in front of the camera, this was the only system to provide luminosity to the picture, and the camera was forced to increase the values of brightness and exposure during the ISP step, these lead the final frame to be noisy. By adding another halo to the rear part, the light was more homogeneous and the camera parameters did not generate noise in order to process the image.

Revision 2, after adding the rear LED's halo, the image gained overall quality, but because of the lateral light, depending the diameter of the copper wire, shadows can form on the surface of the cable, for this reason, a pair of high beam LED's were attached to the camera in order to eliminate this possibility.

Revision 3, this revision added an automatic calibration with the HSV values, because the copper wire, does not maintain the exact same color properties during the whole fabrication, some frames failed during the correct mask creation, to compensate this, the first automatic calibration, iterates all the HSV values in order to obtain the best mask each analysis.

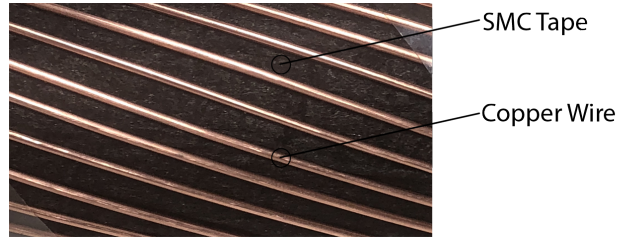
Revision 4, this last iteration, and the actual software that has the current system, the only addition made, was to iterate not just the HSV values, but also the blur factor, this addition improved a bit the overall result of the analysis as it will be seen in the discussion chapter (*Chapter 7*).

7 Discussion

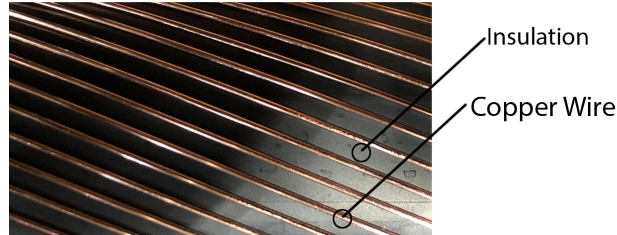
7.1 Results

Note: the current results shown here, are from the latest version of the software, from the 15/05/2019, to 02/06/2019, this values may vary during time, but the expected results, in further revisions, are to obtain a better results in time efficiency and detection performance.

During this last software evaluation, a total of 1527 samples were evaluated. In order to evaluate correctly each frame, every single photo has been labeled manually as correct / incorrect, where correct stands for a SMC tape detection and incorrect, stands for a correct insulation detection. These labels were set due to the fabrication, during the evaluation, it were fabricated other types of cable, TSLF included, for this reason in order to increase the amount of data-set for the evaluation was decided to evaluate other types of cable, in the following figures it can be observed what was labeled as correct or incorrect.



(a) Frame labeled as correct



(b) Frame labeled as not correct

Figure 12: Types of labels for the evaluation

The following picture, is a clear example about what the system must detect, in this case, this sample has been modified for representation purposes.

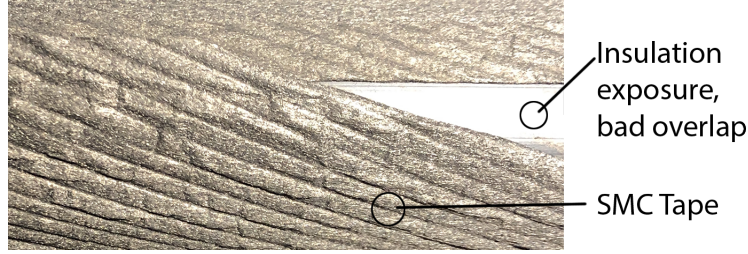


Figure 13: Insulation exposure due a bad SMC Tape overlap

In order to evaluate the system, four parameters where chosen to define how good it performed (*Accuracy*, *Precision*, *Sensitivity*, *Specificity*).

- TP: True Positives, frames labeled as incorrect.
- TN: True Negatives, frames labeled as correct.
- FP: False Positives, frames labeled by the system as correct, but are incorrect.
- FN: False Negatives, frames labeled by the system as incorrect, but are correct.

Accuracy, is a parameter that give the ratio between the correct labeled frames and the whole pool of frames, see eq.(7).

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Precision, is a parameter which gives the ratio between the correct frames and the frames that the system classified as correct, whatever are true positives or false positives, see eq.(8).

$$Pr = \frac{TP}{TP + FP} \quad (8)$$

Sensitivity, or **Recall**, is the capacity of the system to label the frame as correct, when it is really correct, see eq.(9).

$$Sn = \frac{TP}{TP + FN} \quad (9)$$

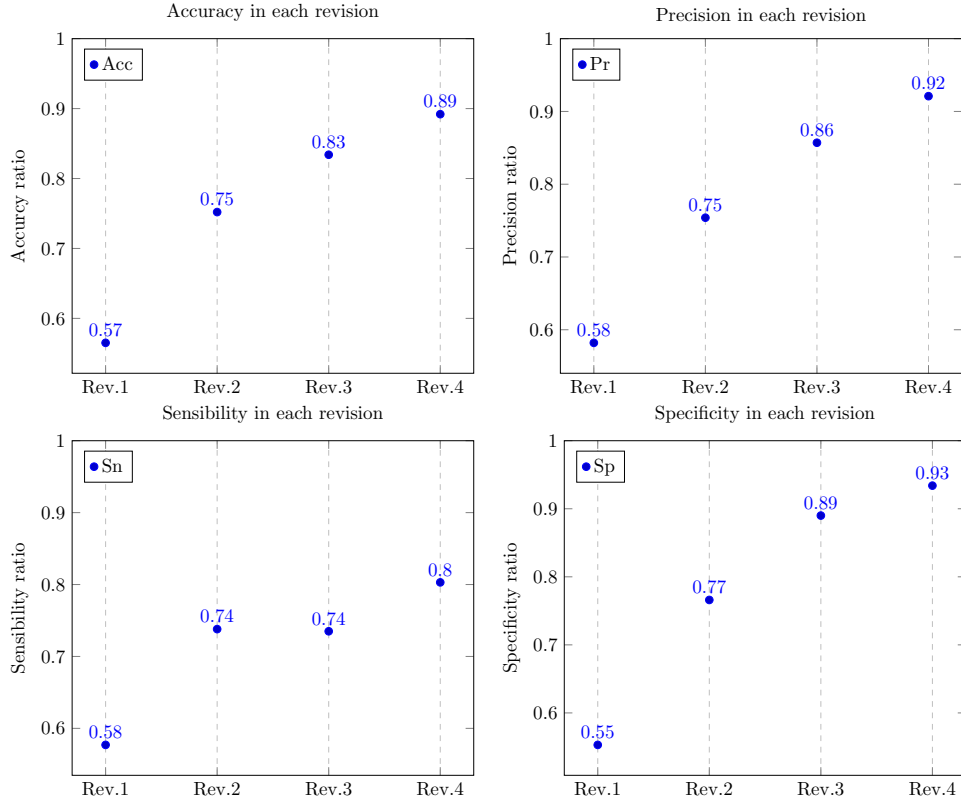
Specificity, is the capacity of the system to label a frame as incorrect, when in reality is an incorrect frame, , see eq.(10).

$$Sp = \frac{TN}{TN + FP} \quad (10)$$

The next table, show the different values for each parameters during the various revision of the system. An appreciable improvement is worth to mention.

Table 4: Table with Results

Revision		Analyzed Frames				Parameters			
Rev	Samples	TP	TN	FP	FN	Acc	Pr	Sn	Sp
1	200	57 (100)	56 (100)	41 (0)	46 (0)	0,565	0,582	0.577	0,553
2	250	98 (125)	90 (125)	32 (0)	30 (0)	0,752	0,754	0.738	0,766
3	326	186 (218)	86 (108)	31 (0)	23 (0)	0,834	0,857	0.735	0,890
4	425	283 (305)	98 (120)	24 (0)	20 (0)	0,896	0,921	0.803	0,934



This plots show graphically the evolution of the system during all the iterations mentioned earlier, during all this iterations, the big steps in performance can be observed in the first two iterations, when the light source was added, this addition has increased the overall performance of the system, later the two software updates, also improved the performance but in a smaller scale.

Is worth mentioning that the sensibility, the parameter which takes into account the true positives within the total ones, is the one with the lowest ratio, meaning that, only 80% of the samples labeled are positives, leaving a 20% of samples as negatives, this means, that the system only detects 8/10 of incorrect frames, the other 2 frames which are detected, contain SMC tape when in reality they do not.

7.2 Conclusions

7.2.1 Overall

As seen in the subsection before, the system performs well in the detection, it can be concluded that the system is able to detect a bad overlap, all three project specification were accomplished without any problem, as well as all the intern objectives have been met, as it has been seen in chapters 4, 5, 6 and 7 (7.1).

For the secondary objectives defined, not every objective has been achieved, the overall system is hardware-ready to implement other detection algorithms, it has been proved that the current system is efficient during production. For the third point, it is hard to describe if it's a low cost and high efficiency algorithm, simply because, this is the only algorithm tested, and it is unknown if there is any other algorithm that is faster and more efficient. Finally it was impossible to communicate with the PLC of the production machine, mainly because the lack of time, and the fact, that the current PLC will be changed by another version.

It has been shown a new approach to detect a bad overlapping of SMC tape in TSLF, with computer vision, yet the results are considered not reliable enough to implement an autonomous machine that relies on itself, nevertheless the system will be installed in the production-line along with the visual inspection of the worker.

7.2.2 Opinion

During the developments of this system, there has been some changes of big steps to take in order to advance until this point. As my university advisor said in the early stages of the project, when a computer vision project has to be started, it is highly recommended that select first all the hardware needed, or at least the vision sensor. Because of this first step, the sensor has to be good enough in order to match the specifications established, because of that the sensor might be over-fitted for the final needs and might be better than needed. Is worth mentioning that this project was started developing first all the software, and later, the hardware parts were selected, this come with a big drawback, the software has to be versatile and robust, because it has to match with any type of possible sensor, and also the software part will be modified once the hardware is selected in order to adapt it as better as possible, nevertheless, once the software works, it acts as a filter for hardware selection.

7.2.3 Next steps

As said in the sections above, the current system prove to be a new approach to detect the correct overlap of SMC tape, unfortunately, the system, is not robust enough to be fully autonomous.

Because my stay in the enterprise (*General Cable*) it was decided, during this time after the Bachelor Thesis was submitted, to keep improving the system and updating the software to achieve better performance, the next steps for the current system, are two, first of all, release a new iteration that is currently in process, this new iteration will normalize the frame depending of the automatic brightness adjusted by the camera, with this new version, all images will be normalized regardless of the automatic adjustment of the camera.

Finally, thanks to the implementation of the database, where all the raw frames are labeled accordingly to cable type, ID, error detected, percentage of area detected and date, an artificial intelligence will be trained in order to understand and adapt to different types of cable, insulation and materials, this last step, will not be in a near future, because it is needed a large amount of samples, and the design, training and evaluation process can be long.

References

- [1] Vivek Gite. Raspberry pi 3 model b+ released: Complete specs and pricing. <https://www.cyberciti.biz>, Mar 2018. Accessed on 03-04-2019.
- [2] Raspberry Pi. The raspberry pi zero w adds wi-fi and bluetooth to the zero. <https://www.lifehacker.com>, Feb 2017. Accessed on 03-04-2019.
- [3] Raspberry Pi. *Camera module V2*, April 2016. Datasheet.
- [4] Mohsen Ardabilian Yunhong Wang Liming Chen Di Huang, Caifeng Shan. Local binary patterns and its application to facial image analysis: A survey. *IEEE Transactions*, 2011.
- [5] Philip Shilane. Mosaic. www.cs.princeton.edu/~pshilane/class/mosaic/. Accessed on 05-04-2019.
- [6] Wolfram. Morphology. <http://documents.wolfram.com/applications/digitalimage/UsersGuide/Morphology/ImageProcessing6.3.html>. Accessed on 05-04-2019.
- [7] Przemyslaw Piotrowski. Build a rapid web development environment for python server pages and oracle. <https://www.oracle.com>, - 2012. Accessed on 03-04-2019.
- [8] Raspberry Pi. Raspberry pi 3 model b+. <https://www.raspberrypi.org>, Feb 2012. Accessed on 04-01-2019.
- [9] Raspberry Pi. Camera module v2. <https://www.raspberrypi.org>, Apr 2016. Accessed on 05-01-2019.
- [10] Thorin Klosowski. Raspberry pi zero w+. <https://www.raspberrypi.org>, Feb 2017. Accessed on 05-01-2019.
- [11] Itseez Intel Corporation, Willow Garage. OpenCV: Color conversions. <https://docs.opencv.org>, Oct 2008. Accessed on 20-10-2018.
- [12] Jane Courtney. A three-dimensional rgb matrix. https://www.researchgate.net/publication/267210444_Application_of_Digital_Image_Processing_to_Marker-free_Analysis_of_Human_Gait/figures?lo=1, Jan 2001. Accessed on 05-04-2019.
- [13] De Wapcaplet. Hsv cone. <https://commons.wikimedia.org/w/index.php?curid=308191>, Sept 2005. Accessed on 05-04-2019.
- [14] DC. He and L. Wang. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions*, 1990.
- [15] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., 1983.

- [16] The University of Auckland. Morphological image processing. <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm>, Sept 2018. Accessed on 05-04-2019.
- [17] OpenCv. cv::moments class reference. https://docs.opencv.org/3.4/d8/d23/classcv_1_1Moments.html. Accessed on 23-10-2018.
- [18] OpenCV. Contour features. https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html. Accessed on 23-10-2018.
- [19] OpenCv. Structural analysis and shape descriptors. https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=cvmatchshapes#moments. Accessed on 23-10-2018.