Contents

1 Introduction

1.1 Summary

1.2 Field of application

This project relates to an industrial monitoring system. More specifically the main application of the developed device is to monitor, in real time, the position of a point machine.

A point machine is an electric motor driven switch that enables an operator to switch a train from one railroad track to another. The main elements of a switch are the points. Figure 1 shows the points in the normal position (top) and in the reverse position (bottom). When the points are in normal position the train, which goes from left to right in this diagram, continues by the same track. In the other hand, when the points are in reverse position the train changes from one track to another.

Nowadays point machines are typically operated from a remote location. Because their closure is imperative it has a device to inform the operator about its current position. The most common realization of this device consists on two locking bars or detector bars, as we can see in figure 2. The motor is connected to the strecher (or trhow bar) through gears. At the other end of the bar there are two points, or switch rails, attached to it. Each point has a locking bar attached to it. These locking bars go from the points to the engine closing. In the engine side there are two two holding elements, called hammers, which are used to lock the locking bars. Each locking bar has a notch that allows the hammer to lock it in the current position. When the points are in normal position the first hammer locks one bar and when the points are in reverse position the second hammer locks the other bar. Figure 2 shows a real point machine engine. We can see locking hammers are one in front of the other and the locking bars are one above the other. Figure 3 shows a detail of the hammers and the notch in the locking bars. Images are taken at different time instants and at different positions of the point machine. In the image on the left the hammer on the top fits the notch of one locking bar. In the other position, the hammer in the bottom fits the notch of the other bar.

When the operator moves the motor causes the linear and perpendicular movement of the drive bar. In turn, this causes the points to move and change their position. The points drag the locking bars which they are attached to. Inside of the engine housing, the hammer locks the locking bar corresponding to the side where the points are. This produces an electric signal that informs the remote operator that the switch has been successfully completed. When the locking bar does not arrive to its final position the electric circuit remains opened so the position of the switch is unknown and the operator cannot operate the switch. Security rules establish that a train cannot pass trough an intersection where the point machine is in an unknown position. This affects directly the railway traffic in a high demand network like a subway or the suburban train.

Although the position of the point machine is the only information required

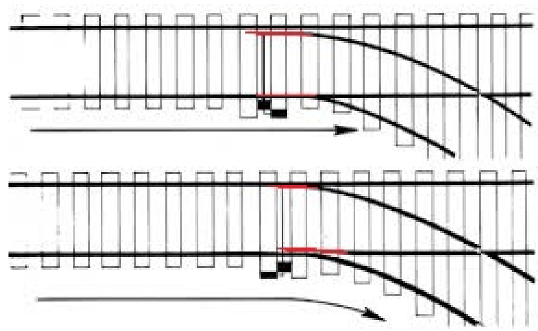

Figure 1: Points in a railroad switch. Normal position in the top and reverse position in the bottom.

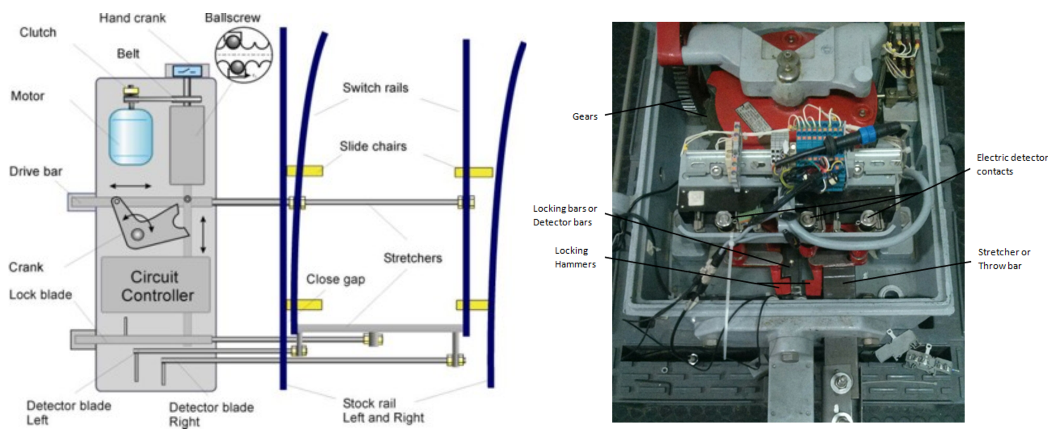

Figure 2: Parts of a point machine. Left: generic diagram. Right: Real point machine engine.

from the point of view of railway safety it is not enough from the point of view of the maintenance of a big number of point machines. There exists devices like the one described in [1] which uses inductive proximity sensors or [2] which uses a transformer with two coils to detect the position of the bar. There exists another device researched by the UPC and developed by Thinking Foward XXI SL that monitors all the signals available in the point machine including the exact position of the lock bars. This device is part of a system that will be further explained in chapter 2.

1.3 Goal

The goal of the present project is to develop a device to monitor the exact position of the lock bars. The task of the monitoring device can be divided in two sub-tasks. The first one consists in detecting the position of the point machine, this is, which lock bar is currently locked. The second one consists on measuring the real gap between the lock bar notch and the lock blade as we can see in figure 3.

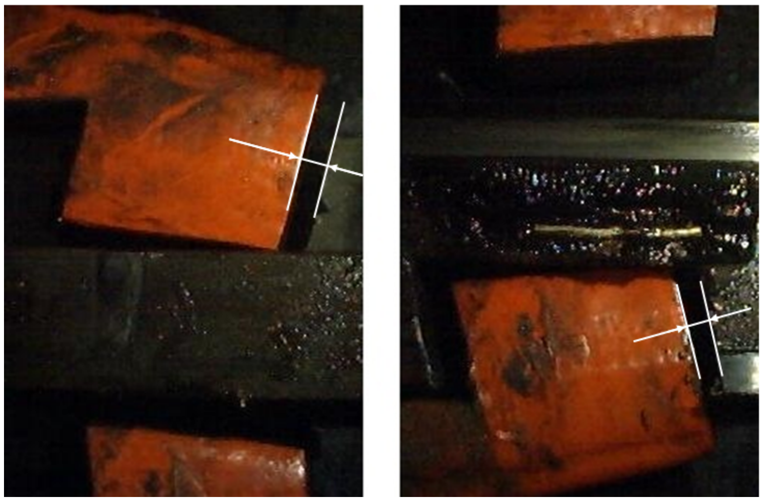

Figure 3: Gap to be measured. Left: Lower lock bar. Right: Upper lock bar.

There are some requisites that this device has to accomplish.

1. Easiness to install. In the railway sector there is very limited period of time to do the maintenance tasks during the night where there is no operation of the service, therefore it is very important that the developed device to be very easy and quick to install.

2. Robustness. Sometimes point machines are in the exterior subject to bad weather, dust and humidity. Although the device will be installed inside of the motor housing it has to be robust to this unfavorable conditions.

3. Precision. It is very important that the developed solution gives exact measurements with high repetitiveness. The gap to be measured in the lock bar is of few millimeters, so the desired precision of the measurements is between 0.1mm and 0.5mm.

4. Reduced execution time. As it will be explained in chapter 4 there are two modes of operation. In normal operation there is a window of 10 seconds to give the measure. In continuous operation the device has to be able to give at least one measure per second.

5. Reduced manufacturing costs. In a railway infraestructure are hundreds of engines to monitor, therefore, the unit price of the monitoring device must be low. The system should reduce the cost of infraestructure maintenance. Potential customers of the system are public or semi-public companies so

the budget is considered at most for four years. All investments must have its pay-back less than this period of time.

We will have these points in mind in order to do the design of the device in chapter 4.

1.4 Planning

Figure 4 shows a summary of the Gantt Chart of the project and figure 5 shows a more detailed planning of the phases and tasks. There are 5 people in the design and development team. This has been taking into account to establish the duration and planning of the tasks.

Projects where both, hardware and software, are involved are very hard to plan because sometimes there are several iterations in the hardware design. At the beginning of this project there is a first prototype with a basic functionality that consists on taking pictures and sending them to the Server. I have been involved in the hardware design, schematic and layout, and also in the firmware development of this prototype. Because of this, the planning of this project is more realistic and the hardware design is shorter than in other projects.

All members of the team are multidisciplinary but mainly the hardware team includes 3 people and the software and firmware team consists on 2 people. My tasks are mainly in the software and firmware development although as I have previously explained I have also been involved in the hardware development of the prototype which will be taken as the starting point of the final device. In order to make it more clear, I have marked in red the tasks where I will be actively involved in figure 5.

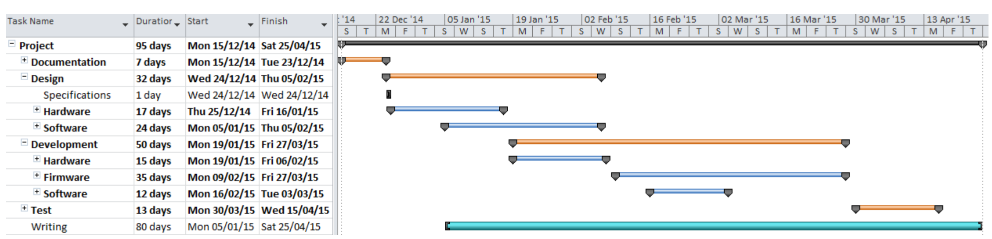

Figure 4: Planning. Gantt Chart Summary.

Task Name	Duration	Start	Finish
Project	**95 days**	**Mon 15/12/14**	**Sat 25/04/15**
Documentation	**7 days**	**Mon 15/12/14**	**Tue 23/12/14**
Prior Art	5 days	Mon 15/12/14	Fri 19/12/14
Fundamentals	3 days	Fri 19/12/14	Tue 23/12/14
Design	**32 days**	**Wed 24/12/14**	**Thu 05/02/15**
Specifications	1 day	Wed 24/12/14	Wed 24/12/14
Hardware	**17 days**	**Thu 25/12/14**	**Fri 16/01/15**
PCB	**15 days**	**Thu 25/12/14**	**Wed 14/01/15**
Schematic	10 days	Thu 25/12/14	Wed 07/01/15
Layout	5 days	Thu 08/01/15	Wed 14/01/15
Mechanics	2 days	Thu 15/01/15	Fri 16/01/15
Software	**24 days**	**Mon 05/01/15**	**Thu 05/02/15**
Communications Protocol	4 days	Mon 05/01/15	Thu 08/01/15
Image processing prototype	20 days	Fri 09/01/15	Thu 05/02/15
Web client integration	1 day	Mon 05/01/15	Mon 05/01/15
Installation program	1 day	Tue 06/01/15	Tue 06/01/15
Development	**50 days**	**Mon 19/01/15**	**Fri 27/03/15**
Hardware	**15 days**	**Mon 19/01/15**	**Fri 06/02/15**
PCB	10 days	Mon 19/01/15	Fri 30/01/15
Housing	15 days	Mon 19/01/15	Fri 06/02/15
Firmware	**35 days**	**Mon 09/02/15**	**Fri 27/03/15**
Image processing library	25 days	Mon 23/02/15	Fri 27/03/15
Image sensor configuration	15 days	Mon 09/02/15	Fri 27/02/15
Communications Protocol	20 days	Mon 02/03/15	Fri 27/03/15
Software	**12 days**	**Mon 16/02/15**	**Tue 03/03/15**
Web client integration	2 days	Mon 02/03/15	Tue 03/03/15
Installation program	5 days	Mon 16/02/15	Fri 20/02/15
Test	**13 days**	**Mon 30/03/15**	**Wed 15/04/15**
Laboratory measurements	3 days	Mon 30/03/15	Wed 01/04/15
First installation	10 days	Thu 02/04/15	Wed 15/04/15
Writing	80 days	Mon 05/01/15	Sat 25/04/15

Figure 5: Planning. Tasks detail. The tasks where I have been actively involved are marked in red.

2 Prior Art

2.1 Computer vision in industrial applications

Computer vision is becoming widely used in industry and many applications have appeared in recent years because it enables the automation of a wide range of processes.

Industrial computer vision applications can be classified in two groups. The first and most extended one consists on using computer vision to visual inspection. We can find many examples in the literature that uses image processing for this purpose. In [3] a method for locating and inspecting integrated circuit chips is described. Another application consists on automatic verification of the quality of printed circuit boards like in [4], fabric quality [6] or industrial plastic components [5].

The second group comprise the control of robots by artificial vision systems. One example of this application is robot guidance to a precise position or trajectory planning like in [7] and obstacle avoidance [8]. In [9] a more sophisticated method of an industrial application with two CCD sensors for spray painting of a general three dimensional surface is described.

In [10] a measurement technique using computer vision is described. We will explain a little more this article because it relates more directly to the task that we want to perform in this project. The goal of the work presented in this article is to measure area of leafs. In order to do so, the fist step consists on obtaining binary images performing a segmentation of color images with an Otsu approach using the hue information. Otsu gives the optimal gray level in the range of [0,255] for image binarization. This will be further explained in chapter 3. Due to some colors in background are close to the color of the leafs noise appears in the binary image. An opening is applied in the binary image to delete this kind of noise. Once they have filtered binary images they look for connected components using a two-scan algorithm. After the labeling step they use geometric characteristics of the a leaf in order to filter residual noise. Finally they count the foreground pixels and multiply this number by a precomputed constant to compute the leaf area. This constant is initialized with a calibration card of a known area. Similarly in this project we will follow more or less the same strategy to determine the exact position of the lock bar as it will be explained in chapter 4.

2.2 Embedded computer vision

Because of image processing require lots of memory resources and processor time a variety of applications are still using ordinary computers to perform these tasks. Some examples are video surveillance applications, car license plate number identification, face recognition applications, etc. Another approach is to use an embedded system for this kind of applications.

Figure 6 shows a diagram of an embedded system. It consists mainly of an image sensor, some signal processing Integrated Circuit (IC) like an Application-

Specific Integrated Circuit (ASIC), Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA), Micro Controller Unit (MCU) or Reduced Instruction Set Computer (RISC) and optionally some external memory, other sensors, communications IC and interfaces. The only part of the different approaches that can be changed is the signal processing unit.

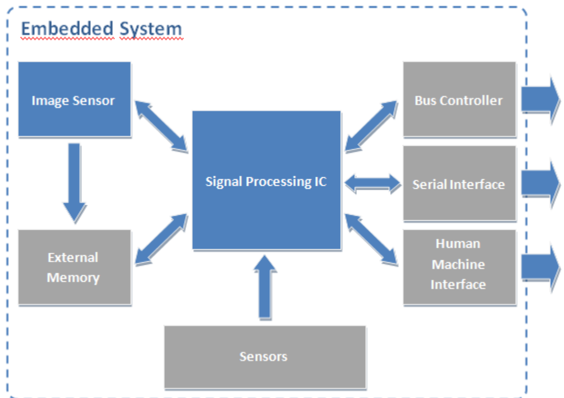

Figure 6: Embedded system for image processing.

The main advantage of this type of systems is that they are smaller than the computer approach so they can be installed in many other places. The drawbacks are the constraints in power consumption, memory, and processing speed. The emergence of more powerful microprocessors and microcontrollers has led to the gradual appearance of embedded systems for image processing.

In [11] an image sensor, a MCU and some laser diodes are used to measure object dimensions. They use an object with known dimension to calibrate the system and translate pixel information extracted from the laser lines in the image to real dimensions of the object. In [12] the authors have developed and embedded system based on a DSP to do fingerprint detection. The fingerprint database is stored in an external SDRAM memory. They also uses a keyboard and a display as the human machine interface. Another application of embedded systems can be found in [13]. This application consists on finding and recognizing car license plate numbers. An ARM processor is the core of this system that also contains an external memory, a keyboard, an LCD and different communications interfaces. In this application there are two different tasks. The first one consists on locating the car plate in the image. If this task is successful they extract the characters and perform a single-character processing to recognize the license number. Finally the recognized license number is shown in an LCD. Finally, in [14] an embedded system has been developed for face detection. In this approach authors use an FPGA as the core of the system. FPGA provides higher computational power but in the other hand they are much more expensive than MCU or DSP.

2.3 Control bar monitoring systems

3 Fundamentals

In this chapter the basics we will expose the theory of the methods we are going to use in the image processing algorithms. The first step of the algorithm consists on image binarization. Then, the binary image is filtered using a morphological filter. Finally the connected regions are found with a two scan algorithm. In addition to these three points camera calibration theory will also be explained.

3.1 Binarization

The first step in most image processing algorithms consists on binarizing the image in order extract objects from the background. In order to perform a good binarization it is very important to select the adequate threshold. We will use the Otsu method [16] to select the threshold. This method is based on the image histogram and it selects the optimum threshold by maximizing the inter-class variance of two classes. Classes are defined as C_0 which contains all pixels with a gray level between $[0, 1, \ldots, k]$ and C_1 which contains all pixels with gray level $[k+1, \ldots, L]$ where L is the maximum gray level of the image and k is the threshold. In order to find the optimum threshold k^*it uses one discriminant criterion measure used in the discriminant analysis

$$\eta = \frac{\sigma_B^2}{\sigma_T^2}$$

where

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0\omega_1(\mu_1 - \mu_0)^2$$

and

$$\sigma_T^2 = \sum_{i=1}^{L}(i - \mu_T)^2 p_i$$

p_i is the probability of a pixel to have gray level i and ω_i and μ_i are the zeroth-order and first-order cumulative moments of classes defined with kth level. σ_B^2 and σ_T^2 are the between class variance and the total variance of levels respectively. Maximizing the discriminant criterion η is equivalent to maximizing σ_B^2 because σ_T^2 does not depend on threshold k. Therefore the optimal threshold k^* is found using a sequential search of the maximum of σ_B^2 for all possible values of $k = [0, \ldots, L)$

$$\sigma_B^2(k^*) = \max_{0 \le k < L} \sigma_B^2(k)$$

3.2 Morphological filtering

3.3 Connected components

3.4 Camera calibration

4 Design

In this chapter we will explain the key aspects of the design. We will start by establishing the system specifications that fulfills the goals presented in secion 1.3. Then we will continue explaining the hardware design which involves not also the electronics but the optical element and the housing of the device. After that a communicatinos challenge will be presented and we will explain a custom communications protocol to solve it. Finally the image processing algorithms will be discussed.

4.1 System specifications (according to the goals)

The device must be designed taking into account the goals stated in 1.3 and the size constraints.

- First of all the device must be easy to install. There are only 3 daily hours to perform the maintenance tasks and maintenance teams are small according to the size of the infraestructure to maintain. The most easy to install the device the better. This should be taken into account in the hardware, specially in the box design, and in the firmware design.

- The device needs to be robust in two senses. The first one relates directly with the hardware design because it has to be installed inside a point machine which is subject to vibration, temperature changes, dust and humidity. The box of the device must resists this unfavourable conditions. The other sense refers to the repeatability and confidence of the measurements and it is related with the software design and implementation. The error rate of the device should be less than 5%, including bad measures and unknown measures when the lock bar is in its correct position.

- The minimum total gap to measure is of 5 milimeters. Maintenance operators want to know where the gap i less than 10% in order to plan a task to correct the position of the bars. The precision of the measurement should be less than 0.5mm and it is desirable to be 0.25mm. This should be taken into account to determine the resolution of the CCD and image processing algorithms.

- Another constraint of the system is the execution time that consists on the image capture plus the image processing times. The image capture time is determined by the hardware (the image sensor, the microcontroller and the memory latency). The image algorithm determines the processing time. As it has been explained in 1.3 the device has to two modes of operation. In normal operation it gives a measure every 10 seconds. In general, the execution time could be up to 10 seconds. In continuous operation the device must provide at leas one measure per second. This has to be considered in the design of the image processing algorithm.

- Power consumption has also be taken into account. The wire length between the control cabin and the point machine location can sometimes be quite long, up to 1Km. The device is powerd by a DC voltage up to 24V with a cable of $1.5mm^2$. The resistivity of the copper is

  $$\rho = R\frac{S}{l} = 1.71 \times 10^{-8}\Omega m$$

  as we can see in [15]. S is the section of the conductor in squared meters and l is the wire length in meters. We can compute the total wire resistence as

  $$R = \rho\frac{l}{S} = 11.86\Omega$$

  If we take, for instance, that the device consumes 150mA, the total drop of potential at the device is

  $$E = I \times 2R = 3,56V$$

  Tipically the same cable is used to power more than one device so the power consumption represents a constraint on the design. This constraint relates not only with the hardware but also with the firmware of the device. If the execution time is so long that the device is always running then the power consumption increases and it probably does not fullfill the power consumption constraint.

- Size constraints. The device has to fit in the interior of the engine housing. There is a protective element which can be used to hang it. Its position is the best for the device because it is just above the region of interest and the vertical distance from the bars to the position of this element is suitable. If we design the device to be hung from this element there is a size constraint. Figure 7 shows this element. First of all, we want to device to be as small as possible because it should not interfer with the visual inspection of the gap. Furthermore this device could be used in another kind of engine and the smaller the better to find a suitable position for it. Secondly the space between the protective element and the cover of the engine is about 10 cm. The cover is not fixed to the engine housing so it has some freedom when an operator opens or closes it. If the device is to close to the cover the likelihood of breaking it increases. So we must design it to be at least 2cm away from the cover.

4.2 Hardware design

4.2.1 Electronics

The most important element of the product is the image sensor and a suitable lens. The key point in the hardware design is that we have to develop a product within the minimum possible time and with the available resources in Thinking

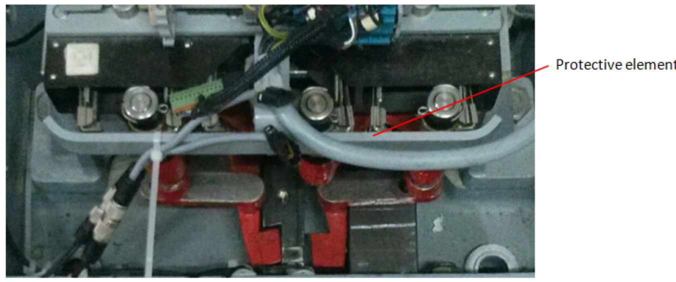

Figure 7: Protective element where the device can be hung.

Forward XXI. In order to simplify the development, the first option to evaluate is to buy a module which includes the optical sensor and the lens. We have rejected this aproach because we have found that manufacturers like SHARP or TOSHIBA are not interested in selling little quantites of their integrated modules and furthermore almost all available modules in the market have a CCTV output that it is not usefull for our application.

The second option we have evaluated is to use an image sensor with a microcontroller. There are some reasons why we are going to develop this device using this option. First of all, the team in Thinking Forward XXI which is going to design and develop the device has some expertise in developing hardware based on microcontrollers. Secondly, microcontrollers are chepear than FPGA or ASICs. And finally we have a first prototype designed with this approach. In addition, the microcontroller approach has been used by Pauli et al in [11] so we know in advance that it is a feasible solution.

Microcontroller The microcontroller must have a CAN interface and a Camera interface. The CAN interface is compulsory because the device has to be installed inside the engines. Usually there are free wires from the control point to the engine which are the only way to establish the communication. The new device has to be integrated in a system which uses these wires to make a CAN bus and send information over it. The best choice is to use the same bus to send the engine position, gap measurements and images. The camera interface is also compulsory to connect the microcontroller with the image sensor. The microcontroller we have selected is an ARM Cortex-M4 from ST. More specifically it is the STM32F407 wich goes up to 168MHz.

Image sensor In order to select the image sensor we have to take into account the resolution we want to achieve. The area we have to analyse is more or less a square of $w = 12cm$ side. The resolution we want to achieve is about $0.25mm$ as we have said in 4. We can compute the minimum resolution of the sensor in pixels to be

$$min\,res = \frac{120}{0.25} = 480px$$

Nor the maximum resolution neither the number of channels are important parameters for this application. Although the maximum resolution should be taken into account because it is related to the processing time, it is not a restrictive paramter in this application because usually image sensors can be configured to the desired resolution. Despite the application only requires one channel images (gray level images), color sensors have the same price as gray sensors.

Another important aspect to take into account is the availability of the sensor. After talking with some sensor manufacturers and distributors we have found that the best choice is Aptina. Although you have to sign a non disclousure agreement (NDA), the sensors can be bought in any component provider like RS or Farnell. After signing the NDA Aptina provides usefull datasheets and developer guides that allows you to configure the sensor.

Taking this into account we have requested information about an Aptina and an Omnivision sensor and finally we have selected the MT9M131C12STC sensor from Aptina because the documentation is much better. It is a 1.3Mpx color sensor and it can be configured through an I2C interface which is also available in the selected microcontroller.

4.2.2 Optical element

We are now focusing in the optical element. To select the lens we have to focus on the geometry of the problem. The device is going to be placed aproximately $D = 15cm$ over the image plane. As we have said previously, the area we have to analyse is more or less a square of $w = 12cm$ side. With 1 we can compute the angle of view α of the lens.

$$\alpha = \arctan\left(\frac{w}{2 \times D}\right) \tag{1}$$

The range of the angle of view can be defined to cover an area from $w = 10cm$ to $w = 20cm$. With this range we have that α must be compressed between 20.48^o and 37.43^o.

The angle of view determines the relation of the image sensor size d and the focal length f of the optical element with the equation 2. Before computing the required focal length we must know the image sensor size. Usually image sensors have different width and height. We have to take the most restrictive size, wich is the smaller one.

$$\alpha = 2 \times \arctan\left(\frac{d}{2 \times f}\right) \tag{2}$$

The size of the selected image sensor is 4.6mm x 3.7mm. So we can compute the minimum and maximum focal length to be $5.46mm$ and $10.24mm$ respectively.

Another parameter we have to take into account is the distance between the image sensor and the optical element. This parameter is also related with the focal lenght of the element with equation 3. This equation relates the focal lenght f, the distance between the optical element and the sensor S_1 and the distance between the lens and the focused object S_2.

$$\frac{1}{f} = \frac{1}{S_1} - \frac{1}{S_2} \tag{3}$$

Due to the limitation in the size of the device we do not want the lens to be far away from the sensor. If we take that the maximum allowed distance between the sensor and the lens is $S_1 = 10mm$ then the maximum focal length will be $f = 10.71mm$. This parameter does not affect to the design because it is outside the range we have computed previously.

4.2.3 Housing

4.3 Custom communications protocol over CAN

The device we are developing have to be placed inside the point machine engine housing. Furthermore it have to be integrated in the current monitoring system explained in section 2.3 of chapter 2. The easiest way to integrate the new device into the system is using the same communication bus to transfer the computed data and the captured images when necessary. Data transmission can be done with a single CAN frame. A CAN frame can have up to 8 bytes of information that are enough to transmit the gap measure and the current point machine position.

In order to send the entire image to the server one frame it is not enough. We must send a large number of CAN frames. CAN specification does not provide neither reliable nor ordered delivery of data frames so we must implement a custom protocol over the CAN layer.

The custom protocol must ensure that all frames arrive to the communications concentration device and that they arrive in the right order. Figure 8 shows the diagram of this protocol. The key aspects of it are the following:

- The first frame of the protocol contains the number of bytes that will be sended.
- Dataframes with image data bytes contain a number indicating the position of the data.
- ACK are used in order to ensure that all frames arrive to the receiver.
- NACK are sended to the transmitter to say that some frames have been lost. The transmitter send again the lost frames.

Figure 8: Custom communication protocol diagram.

4.4 Image processing. Matlab prototypes

The device we are developing has to do two tasks. The first one consists on detemining the current position of the point machine. If this task is completed successfully then the device has to measure the gap between the lock bar notch and the lock blade. In order to do this tasks we will use an algorithm similar to the one used by Kaiyan et al. in [10].

The fist step of the algorithm consists on the image binarization. In order to perform this task we will use the Otsu method which has been explained in the point 3.1. Otsu method gives us the optimum threshold level, all pixels above this level are considered to be the foreground while others are part of the background. After the binarization we apply an opening in order to filter the undisired noise. The most important step in the algorithm consists on extract the connected regions of the image foreground. In order to do so we apply the image labeling method explained in section 3.3. Once we have found the different connected components in the image we can remove some objects with the geometrical information. We are looking for the laser lines in the image. These are horizontal lines. In the side where the lock blade is present the line will be discontinued while in the other side the horizontal line goes from right to left. In both cases lines we are looking for are connected to the left and/or right margins of the image. Applying this prior information we can remove all the objects that are not connected to the side margins of image. Once we have exctracted laser lines we are able to estimate the current position of the point machine. This first task is performed by looking wich line is connected to both left and right margins of the image. If we find and object that fullfill this condition we can detrmine the position of the engine. Once we know the current position we have to measure the existing gap. In order to this this task we will focus on the broken line. We will find the endings of both parts of the line in order to establish the number of pixels of the gap. To transform the number of pixels into a real measure we need some reference. We will use the lock blade as the reference because its size is always the same. In order to do this we have to measure this part of the point

Figure 9: Steps of the image processing algorithm

4.4.1 Binarization

The main step in image binarization consists on finding the optimal threshold to extract objects from the background. In order to perform this step we are using the Otsu's method which has been explained in section 3.1 of chapter 3. Algorithm 1 shows the algorithm steps. The first step consists on computing the image histogram. This is accomplished by initializing an array of length equal to the number of gray levels in the image to 0's. Then for every pixel in the image the counter corresponding to its gray level is increased by 1. At the end of the scanning the counter array contains the number of pixels with each gray level in the image n_i. Then the probability for each gray level is computed as

$$p_i = \frac{n_i}{T},\ i = [0 \dots L]$$

where $T = M \times N$ is the total number of pixels in the image. It is important to note that

$$\sum_{i=0}^{L} p_i = 1 \tag{4}$$

Remember that we are looking for maximizing the intra-class variance of two classes C_0 and C_1. We can compute the first-order momentum of classes as

$$\omega_0(k) = \sum_{i=0}^{k} p_i$$

$$\omega_1(k) = \sum_{i=k+1}^{L} p_i$$

$\omega_1(k)$ cal also be more efficiently computed taking into account 4 as

$$\omega_1(k) = 1 - \omega_0(k) \tag{5}$$

Then at every iteration we only have to add the next p_k to $\omega_0(k-1)$ and compute $\omega_1(k)$ using 5.

We can accelerate the algorithm implementation using two conditions. The first one consists on going to the next iteration while $\omega_0(k)$ is equal to 0. This means that there are not any pixels in C_0 and the optimal threshold cannot be in that level because all pixels would be in C_1. The second one consists on ending the loop when $\omega_1(k)$ is equal to 0. This means that there are not any pixels with a gray level greater than the current one so it is not necessary to continue computing.

4.4.2 Morphological Filtering

4.4.3 Connected components

4.4.4 Position estimation

4.4.5 Gap measure display

Algorithm 1 Otsu's method for optimal histogram threshold search

1. Compute histogram and probabilities of each intensity level
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$, $i = [0, 1]$, and set $\max(\sigma_B^2) = 0$
3. Step through all possible thresholds $k = 1 \dots L$
   (a) Update $\omega_i(k)$ and $\mu_i(k)$ at every threshold level.
   (b) If $\omega_0(k) = 0$ go to the next iteration.
   (c) If $\omega_1(k) = 0$ stop the loop.
   (d) Compute $\sigma_B^2(k) = \omega_0(k)\omega_1(k)(\mu_1(k) - \mu_0(k))^2$
   (e) If $\sigma_B^2(k)$ update $\max(\sigma_B^2)$ and set $k^* = k$
4. Desired threshold k^* corresponds to the maximum $\sigma_B^2(k)$

5 Development

5.1 Hardware constraints

5.2 Image processing library

5.3 Image sensor configuration

5.4 Communications protocol

6 Results

6.1 Accuracy and repetitiveness

6.2 Image vs magnetic sensor

6.3 On site measurements

7 Conclusions

7.1 Goals review

7.2 Future work

References

[1] R. C. Franke. “Railway Switch Machine Point Detection System”. Union Switch & Signal, Inc., assignee. Patent US 6382567 B2. 25 Aug. 1999.

[2] M. A. Hager, M. F. Towey, Jr. “Contactless point detection system for railroad switch”. Alstom Signaling, Inc., assignee. Patent US 6427949 B1. 23 Jan 2001.

[3] M. L. Baird, "SIGHT-I: A Computer Vision System for Automated IC Chip Manufacture". Systems, Man and Cybernetics, IEEE Transactions on . Nov. 1976, pp. 3-7.

[4] J. F. Jarvis. “A Method for Automating the Visual Inspection of Printed Wiring Boards”. Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:PAMI-2 , Issue: 1). Jan. 1980, pp. 77-82.

[5] A. Anzalone, G. Gugliotta, A. Machi’, G. Sardisco. “Automatic Quality Control of Industrial Products for Irrigation”. Image Analysis and Processing, 1999. Proceedings. International Conference on. Sep. 1999, pp. 588-593

[6] Y. Rong, D. He, Y. Lin. “Rapid Detection Method for Fabric Defects Based on Machine Vision”. Computer Application and System Modeling (ICCASM), 2010 International Conference on (Volume:10). Oct. 2010, pp. 662-666.

[7] C.J.Zhao, G.Q. Jiang. “Baseline detection and matching to vision-based navigation of agricultural robot”. Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on. July 2010, pp. 44-48.

[8] F. Kunwar, B. Benhabib. “Rendezvous-Guidance Trajectory Planning for Robotic Dynamic Obstacle Avoidance and Interception” Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on (Volume:36 , Issue: 6). Dec. 2006, pp. 1432-1441.

[9] A. Zaki, M. Eskander. “Spray Painting of a General Three-Dimensional Surface”. Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on (Volume:3). Oct. 2000, pp. 2172-2177.

[10] L. Kaiyan, W. JunHui, C. Jie, S. Huiping. “Measurement of Plant Leaf Area based on Computer Vision”. Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on. Jan. 2014, pp. 401-405.

[11] N. Pauly, N.I. Rafla, “An Automated Embedded Computer Vision System for Object Measurement”. Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on. Aug. 2013, pp. 1108-1111

[12] M. Kamarajui, P.A. Kumar. "DSP based Embedded Fingerprint Recognition System". Hybrid Intelligent Systems (HIS), 2013 13th International Conference on. Dec. 2013, pp. 6-11

[13] K. Zhang, W. Tang, H. Wei, R. Shi. "Study on the Identification System of Car License Plate Based on Imbedded Computer System". Education Technology and Computer (ICETC), 2010 2nd International Conference on (Volume:1). June 2010, pp. 146-149

[14] L. Acasandrei, A.l Barriga. "Embedded Face Detection Implementation". Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the. Sept. 2013, pp. 1-8

[15] http://es.wikipedia.org/wiki/Resistividad

[16] N. Otsu. "A Threshold Selection Method from Gray-Level Histograms". Systems, Man and Cybernetics, IEEE Transactions on (Volume:9). Jan. 1979, pp. 62-66