Master of Science in Aerospace Engineering
Major: Aircraft Systems and Control

Vision based object detection and avoidance

Authors:
Student 1: Álvaro LÓPEZ GUTIÉRREZ
Student 2: Andrés GONZÁLEZ MORENO

Tutors:
Tutor 1 (ISAE): Regine LECONTE
Tutor 2 (Sterblue): Vincent LECUBRIER
Tutor 3 (ISAE): Jean-Baptiste CHAUDRON

Goal of the project: This project aims to research into the possible improvement in the autonomous Unmanned Aerial Vehicles performance. Particularly, the analysis will be focused on data obtained with an stereo Camera. The fields of study will be obstacle avoidance, object recognition and tracking, which currently are not totally developed taking into account the source of data. Lastly, the study will be performed with the goal of high efficiency so the Graphic Processing Units will be a key element all along the development of the project.

1 Project Issues

Since the very beginning in which automatic aerial devices and vehicles were created, their applications have unavoidably spread to almost any field of use. That is why, the Unmanned Automatic Systems, normally abbreviated as UAS, increasingly require more and more devices in order to improve their implementation and performance. Thanks to the aforementioned development, the performance of these devices is nowadays object of study as the final goal is always to make this vehicles a hundred per cent automatic and reliable.

Laying aside the technical development and focusing on the economical one, many enterprises have taken advantage of the current situation in the automatic vehicles field and have started several ambitious projects. One of this enterprises is Sterblue, a French start-up which aims to carry out high performance inspections in power lines and wind turbines, not only offshore but also onshore. To do so, automatic systems are used as not only do the quality but also the efficiency of the services are noticeably improved. For example, the current price of an inspection offshore would be around 800 €, with the added problem that only two or three inspections per day would be possible. In contrast, the objective of this enterprise after have fully developed the entire automatic aerial system of inspection, the price could be reduced to 400 € and the number of inspections per day could be increased up to 16.

In particular, this project focuses on the study of a visual detection system, that is, the use of cameras in order to detect the surroundings of the vehicle. With the information received, some algorithms should be developed so that the vehicle is able to act automatically in consequence. One of the best developments that this start-up has taken into account is the use of General Purpose Graphic Processing Units, commonly known as GPGPU systems. These processing units are considered to have a much higher performance in vectorial computing and graphical analysis than the CPUs, or Computer Processing Units, the ones that have been used until some years ago. In the next chapters we will go deeper in these topic as it is one of the most important.

Finally, this report aims to explain the basics of the project. A study about the state-of-art has been carried out to know well the current situation in the field of study. Besides, we will present the main hardware and software of which we will make use in order to achieve the final targets as well as the first experiments and tests that have been done. Right after this, there is a list regarding the main sources of information and bibliography. And last but not least, we will present the planning of the project, including the expected milestones and deadlines so that all the goals are successfully satisfied.

1.1 Objectives of the project

The main objectives of this project, according to the necessities of Sterblue and mentioned in priority order are:

- To develop the relevant algorithms which allow the vehicle to compute the distance to the obstacles and to avoid them so that the most suitable trajectory is implemented. As it was mentioned, the main sensor of the drone will be visual.

- To come up with an appropriate system and to develop the corresponding algorithms to carry out an automatic precision landing with an UAV.

- To implement the algorithms which let the automatic vehicles follow specific patrons so that the main trajectory to follow is based on these particular patrons. To do so, of course we should take into account the previous steps and also implement the obstacle avoidance objectives.

- To be able to create algorithms of object recognition in order to identify specific complex objects and shapes.

1.2 Resources

In this chapter, we will explain which are the main tools that we have available in order to do the necessary tests and finally achieve the objectives.

1.2.1 Cameras

As it was already explained the main sensors of the systems are stereoscopic cameras. The two main models are:

- *Intel RealSense R200* from Intel. This camera is equipped with two infrared cameras so that it is possible to analyze the depth of the image, that is, the distance from the camera to the objects detected. According to the manufacturer, this camera has a high resolution up to 1080p. In terms of the internal architecture, the processing in this camera is carried out by a Computer Processing Unit.

- *Zed Stereo Camera* from Stereo Labs. As well as the previous camera, this device allows the user to detect the depth and the distance in the images captured. However, the main difference with respect to the Intel RealSense is that it processes the images in a Graphic Unit, which theoretically improves the performance and the speed in data processing.

Apart from the image processing units, there exist more difference between both cameras. Firstly, the range of the Intel RealSense reaches the upper limit in around 5 meters whereas the Stereo Camera can capture images at a distance of 20 meters. Secondly, the stereo camera has a better orientation system and positional tracking sensors thus the 3D mapping is much easier as the points detected are positioned very fast. In short, the Stereo Camera is a better option to use in the final vehicle.

Some tests are thought to be carried out in a near future to definitely prove which one will be the best.

1.2.2 Processing Unit

Another key element in the present project is the Graphic Processing Unit. It will be demonstrated that the performance of a Graphic Processing Unit is needed due to the large amount of data that the devices will be handling. The model of embedded computer we are going to use is Jetson TK1 released by Nvidia, figure 1. This supercomputer features a quad-core processor (ARM Cortex A-15 CPU) and a GPU unit with 1 processor of 192 CUDA cores in addition to excellent graphics and deep learning characteristics. With such features, it is a cutting-edge computer for on board devices and real-time data analysis. this supercomputer may be substituted by Jetson TX1 in the future in order to reach even better performance (256 CUDA cores), but this is not definitve.

Figure 1: Jetson TK1 Embedded Development Kit from NVIDIA.

But, what are the computing tools which can be used in order to get to the top of the possibilities of such high performance devices? First of all, we have to consider the parallel computing features. Nowadays there exist a really large variety of libraries and software that lets the users manage to work in parallel with more than one core. A simple example is when making a program in Linux terminal: we can choose the number of cores working and the higher the number the better the performance. Another of these libraries is CUDA [8]. CUDA (Compute Unified Device Architecture) is a software created by Nvidia which allows the user to configure their own parallel computing space. In this way, large operations and processes that would require a lot of time if they were done with an only CPU, can be done much faster and with a better performance by using more than one core.

This software is really beneficial for the purpose of the project as the user can deal with shared memory and the memory positions are totally accessible. Moreover, in image processing the most common operations are those related to matrix computation. Multiplications among matrices and vectors as well as other operations within the subsets of a matrix can be much easily carried out by making use of specific libraries. The one that we decided to use is BLAS (Basic Linear Algebra Subprograms). BLAS is present in worldwide known mathematics programs like MatLab. It has three levels, which contain lots of functions to perform operations either between vectors, matrix and vector or between matrices. Its way of processing is supposed to be very efficient; nevertheless it is no useful for our purpose unless it's used in a GPU environment.

Fortunately, as Nvidia provides the users with a lot of libraries and programs in order to be able to work with their devices thus we can use the library called CUBLAS [7]. Obviously, its name comes from CUDA and BLAS and it corresponds precisely to that: the main goal is to perform the computation that BLAS is able to but by using a multicore device. Besides, it is quite autonomous as the user does not have to configure the multicore environment; it is configured by the library itself.

In order to put of all of this into practise, we did several tests to check out how these tools work and which is the best way of using them. In the following graph, figure 3, we show the results obtained after computing the matrix product between n*n size matrices with both BLAS and CUBLAS. The main difference between these two libraries is that CUBLAS configures the computation in order to perform it with severals CPUs instead of just one, like BLAS does.

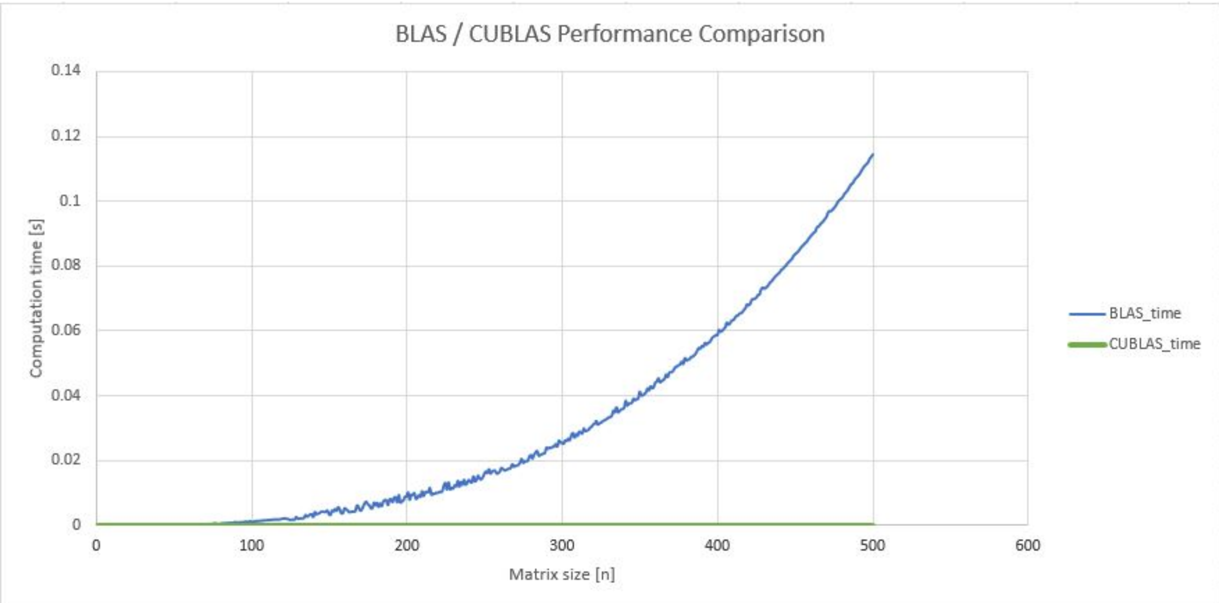


Figure 2: Comparison between BLAS and CUBLAS performances.

As we can see, for sizes which are not large, it seems that the performance is similar but there is a point (n*n = 25*25) in which the performance of CUBLAS shoots up and exceeds more than enough BLAS performance. In conclusion, when dealing with high quality images and videos, we will not be dealing with 25*25 matrices but much larger; therefore, we need devices computing in multicore computers or GPUs. More tests have been performed to check the efficiency of the Jetson Card.

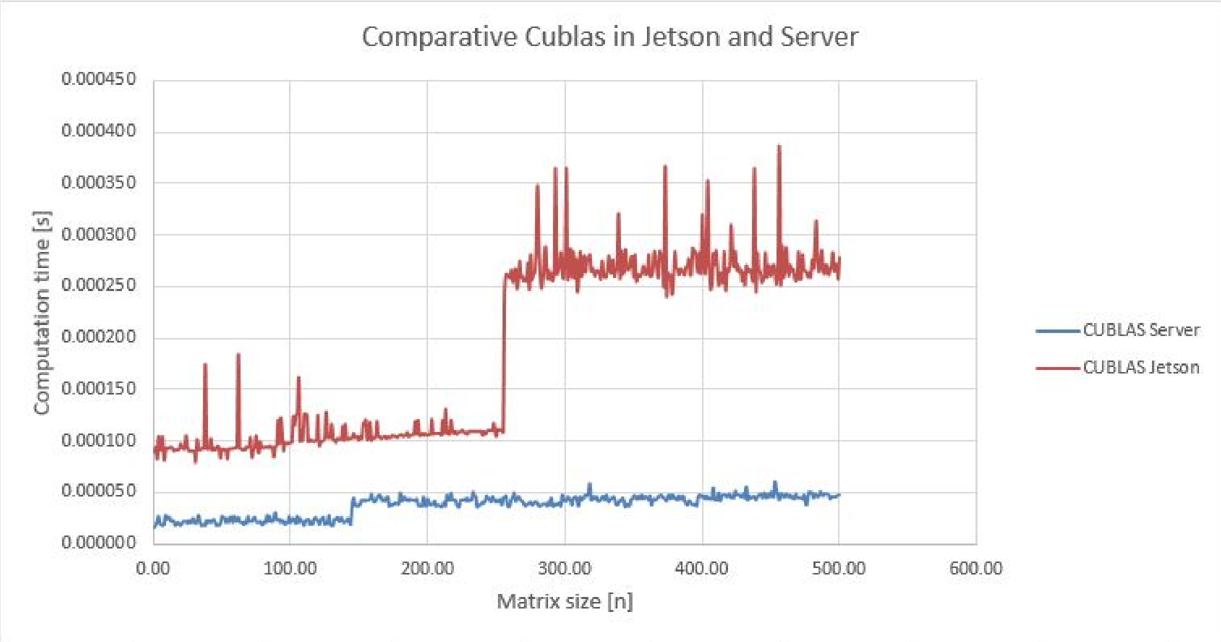


Figure 3: Comparison between the GPU of ISAE's server and the Jetson TK1 performances.

The comparison of performance has been done between Jetson TK1 and the processor Tesla C2050, which is the one available on the Server computers. The main differences between the processing units in the aforementioned examples are displayed in this table:

	Jetson TK1 GK20A	Server Tesla C2050
Total amount of global memory	1892 Mb	2687 Mb
Number of multiprocessors	1	14
CUDA cores (per processor / total)	1 / 192	32 / 448
GPU clock rate	852 MHz	1147 MHz
Memory clock rate	924 MHz	1500 MHz
Memory bus width	64 bit	384 bit
Integrated GPU sharing host memory?	YES	NO

Table 1: Comparison between the GPU of ISAE's server and the Jetson TK1 performances.

1.2.3 Image processing

After analyzing the devices we are going to use and the reasons why, we had to think about the algorithms that are most suitable to achieve the aforementioned objectives. Of course, this algorithms bring along various libraries and packages from which currently we know only a few but that it's probable that while researching deeper we will find some else.

- **Depth Maps**

  The cameras we will be working with are based on depth detection. This system is truly widespread in terms of visual-detection devices, which from now on we will call Stereo Cameras as it is the name with which they are most commonly known. The pillars of the depth maps algorithms are the triangulation [11] and the block matching or disparity.

  The disparity consists on the comparison between several images, generally two, from several cameras or lenses. Basically, after having taken the two images, the algorithm is configured to pick a certain area of one of the images and compare it with the other image, looking for the less different region of pixels. In the algorithm is also specified the zone of searching because knowing that both images are the same there is no point in comparing all blocks of one image with all blocks of the other, but with a close region.

  The numerical comparison is performed through the sum of absolute differences or SAD method. The first step of this method is to convert the image in a unique scale-of-values matrix, thus a black and white image. Afterwards, the values of the selected region in the first image are subtracted from the values of all the regions of the second image with which we are doing the comparison. Finally, in every result matrix, all the absolute values will be added up so that we will have an only value per comparison. Then, it is obvious to deduct that the lowest value belongs to the comparison of two equal pixel regions. We have to remark that this is the base algorithm. Of course in the Stereo cameras, this algorithm is slightly modified in order to overcome possible drawbacks such as noise or excessive light exposure.

  With this explanation, the only aspect missing is how to get definitely the depth. For a point in the space, we will have two images of it, one per camera. Because the distance between the projectors is known and we know exactly where that point is in both images captured (that is, the disparity) it is possible to know the third coordinate, the depth; thanks to a process of triangulation.

  The depth maps are a key element for the aim of this project. By knowing at with distance the points around our device are, it could be possible not only to develop and perform the optimum trajectories to avoid the obstacles but create complete 3D models whose applications could extend even out of the targets of the present project.

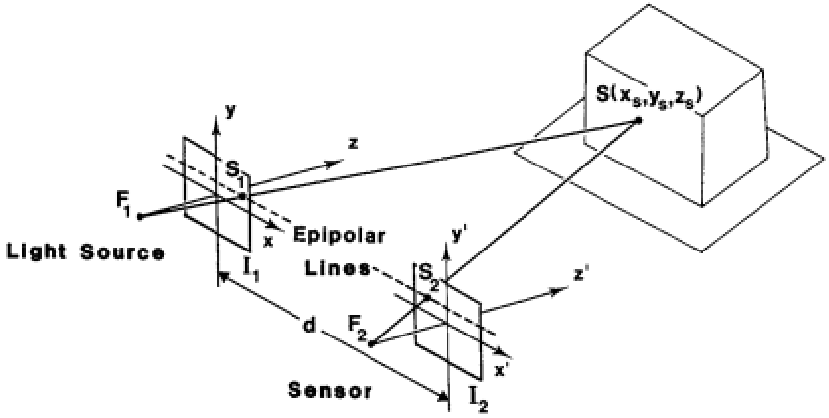


Figure 4: Scheme of triangulation process with two focal points. Source:

- **Point Clouds**

  Despite the fact that point clouds was not a tool of which we thought at the beginning, it has turned out that it could be really applicable and convenient for this project. As its own name reflects, the basis is the macro-analysis of images through the micro-analysis of each individual point. There exist a lot of libraries able to implement cutting-edge algorithms which can extend from simple noise removal or color saturation variations to object and patron detection, 3D models building or bidimensional and tridimensional reconstruction. There are several programs and libraries to do so, one of the most well-known is PCL (Point Cloud Library).

- **Other algorithms and libraries**

  The aim of this subsection is to complete the previous ones with more available libraries and algorithms that will be tested (if they haven't yet) to see if they are a good option for this project.

  OpenCV seems to be a useful released. It is a library supporting the main programming languages (C, C++, Python) whose characteristics for image processing are quite astonishing. For example, it was particularly designed for real-time applications and that is why the multicore processing is possible with it. Particularly, and quite beneficial for our project, it has already implemented a CUDA compatible library that allows to use all the image processing algorithms with the parallel processing capabilities of CUDA. In its website it is claimed that it can even be used to mines inspection or maps stitching. We will include in the bibliography the manuals that contain all the functions and commands of OpenCV as the possibilities are so many that it would be impossible to make a brief summary in the present report.

  In terms of the algorithms, the possibilities are also really high. However, the process is quite similar in all of them and it follows a similar scheme that the one of depth mapping. This scheme consists of applying firstly some filters to make the image clear and better to be analysed. Secondly, some matrix operations are carried out. Depending on the purpose of the algorithm these operations will be different but generally they are based in comparisons among different regions of the matrix that defines the image or video. And last, the results are displayed and the data is stored to possible future use. We, of course, have tried out some algorithms in OpenCV to observe its performance.

One of them is Viola-Jones algorithm. It is founded in the comparison of different areas of a face in order to detect the faces in an image. For example, the zone of the eyes is much darker than the upper cheeks. Similarly, the nose bridge is usually brighter than the eyes. There are more possible comparisons and all of them are inside the program. It is important to underscore that Viola Jones often uses two region comparisons, although the configuration of this areas can be variable (horizontal, vertical, one region inside another etcetera). In such a way, the program detects in an image or video the objects that satisfy all the matrix coincidences. Attaching some images is considered to be opportune in order to clarify the explanation. This particular algorithm has to be trained, that is, having some references or examples of what is a face and what's not [10].

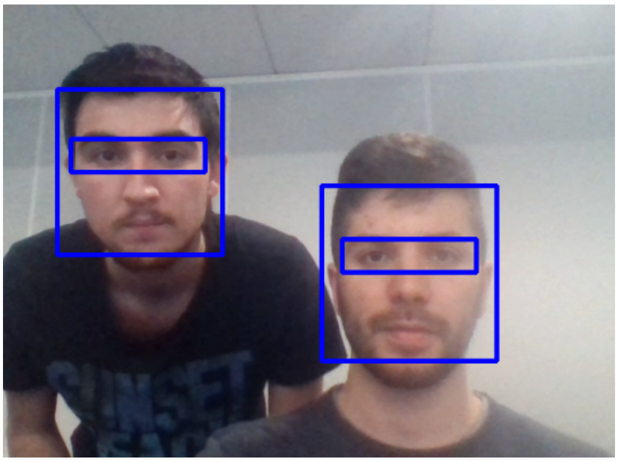


Figure 5: Result of Viola Jones algorithm applied to face Detection

As it was mentioned, other algorithms have been tested. In this case, one to detect the contours of the objects in an image.

Images contour detect

- **Robotics libraries**

  ROS (Robotic Operative System) has been the selected software to mainly help us do the experiments regarding the unmanned vehicles. It allows the user to make a really large quantity of packages and also to create their own. Through these packages, lots of functions can be called and the variety of applications is truly large and advantageous. As we need to do some experiments in robots, we will need to develop the algorithms to make the robot or UAV move in the correct trajectory or avoid the obstacles, for instance. This could be carried out more easily by using a library which at the same time permitted to detect the camera, to perform the appropriate algorithms, to make the vehicle acting consequently and to store the data so as to use it later. This is exactly what ROS could do, or at least, the main usage that we are going to make out of it.

We saw in the previous sections and subsections that this field is in constant development and the tools available to work on it are countless. Nevertheless, we have presented the main means thorough we will intend to develop the present project p to the top of the objectives.

2 State of art

2.1 Stereo Vision for object detection and avoidance

Several research projects have been already conducted in the field of image processing applied to the automation of unmanned vehicles. More precisely, the use of stereo vision algorithms have already been proven to be able to compute the depth map of a given pair of images with a sufficient precision for the detection of common sized objects that may represent a threat for the integrity of the vehicle. This method has been widely used in investigations conducted for both UAVs and cars, as the need for automation in this two fields is increasing every year.

The main challenges in the stereo depth map reconstruction are related to the trade-off present between the quality of the the depth map and the speed of the algorithm.

The quality of the depth map is mainly based o the uncertainty derived from the disparity computed between the pixels of both images, which has to be carefully taken into consideration. Depending on the method used and the characteristics of the environment been analyzed (brightness, repetitive features, occlusions..), the noise present in the depth map can reach unacceptable values for the reliability of the object avoidance system. Usually, a post filtering technique is used to remove the number of outliers and smooth the final image.

The speed of the algorithm can represent a highly restrictive constrain for the object avoidance systems in unmanned vehicles. The reason for this is obvious: as the vehicle operates in real environments, the system must be able to compute the potential object collisions in real time. Due to the size of the elements that are being operated, i.e. large size pixel arrays, the iterative computations can turn into heavy computer processes.

There exist a large number of different methods used to compute the disparity between two stereo images, which every year seem to be more focused into the deep learning field due to the high accuracy shown by the latest developments. However, when it comes to satisfying the image quality and speed trade-off, the block matching methods (simpler but faster than the deep learning based ones), provides more than sufficient results in most of the cases.

For some extra information about depth and disparity maps computation, refer to [4] [5].

2.2 Stereo applied to CUDA

Regarding the specific application of this methods to our project and in order to ease the limiting aspect of the real time operations, CUDA [7] [8] is used as a performance boosting tool. This has already been tested in several applications, with promising results.

2.3 Trajectory computation

2.3.1 3D trajectory optimization

Once the objects have been properly detected, the trajectory currently computed for the vehicle must be redefined in order to avoid the obstacle. For this purpose, the most extended technique consists in the mapping of the surrounding of the vehicle in 3 dimensions and computing an optimized trajectory that avoids the collision with the detected objects.

Various trajectory computation algorithms have to be considered. From [6] , we extracted some surveys. The most remarkable algorithms that have been found are the Random Belief Trees (RRBT), the Kinodynamic Motion Planning by Interior-Exterior Cell Exploration (KPIECE), the Covariant Hamiltonian Optimization and Motion Planning (CHOMP) and the Stochastic Trajectory Optimization for Motion Planning (STOMP).

Again, the selected method must meet the required performances for real time applications and thus, the quality of the feature provided by the algorithm may be decreased to favor its performance.

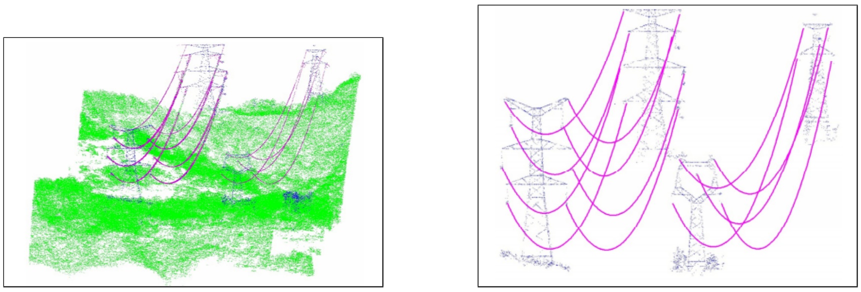
2.3.2 Push broom

A singular but interesting trajectory re-computation technique is the push broom. This method, used in [2], consists in the computation of only a window of the depth map, located at a sufficient distance from the drone to have the necessary time to perform the avoidance maneuver. As the obstacles are only detected are an specific distance and no 3D reconstruction of the surroundings is needed, the computational load of this method is significantly lower. Due to this property, this method may be significantly interesting for high speed applications.

In order to meet the secondary objectives set for this project, more specific methods and algorithms have been investigated. The two secondary objectives for this project and the methods so far identified as potentially helpful are explained in the following section.

2.4 Power lines tracking

The first of the two secondary objectives is providing the UAV with tracking capabilities of power lines in order to perform automated inspections. For this purpose, two main possible research directions have been identified:



(a) Points cloud obtained after applying some filters

(b) Final power line recognition

Figure 6: Some results of applying specific algorithms for cable and power lines detection.

2.4.1 Edge detection based cable identification

This has been previously done in [1], with remarkable results. The main advantage of this technique is, as it only relies on a plain image and not in a stereo one, the computational load of the process will be reduced.

2.4.2 3D based feature match

This technique is based in the matching of a prerecorded feature with its possible candidates in a 3 dimensional space. It shows a clearly higher computational load than a 2D application, but the tracking performances are highly improved, es seen in [9], [12].

From [3] we have extracted some results of the aforementioned methods for tracking high tension cables in power lines. They reflect the good performances and results that Point Cloud algorithms could end providing us with.

2.5 Feature detection based positioning

The second non priority objective is the improvement of the precision of the automatic landing procedure based on the information obtained from image feature recognition algorithms. The idea is to improve the precision in the positioning of the vehicle by means of the recognition of key points in the landing structure. In order to achieve this objective, the Viola-Jones algorithm [10] will be studied. It is a simple but fast algorithm based in the training of cascade classifiers that will match with the desired features.

3 References

In the present section the main references of which either we have made use or we are going to are listed. We would like to remark that this list could be made longer all along the development of the project.

[1] Antich, J. and Ortiz, A.. *Underwater Cable Tracking by Visual Feedback*

[2] Barry, A. J.. *High-Speed Autonomous Obstacle Avoidance with Pushbroom Stereo*

[3] Bo Guo and others. *An Improved Method for Power-Line Reconstruction from Point Cloud Data*

[4] Georgoulas, C. and Kotoulas, L.. *Real-time disparity map computation module*

[5] Merrell, P. and Akbarzadeh, A.. *Real-Time Visibility-Based Fusion of Depth Maps*

[6] Nieuwenhuisen, M. and Behnke S.. *3D Planning and Trajectory Optimization for Real-time Generation of Smooth MAV Trajectories*

[7] Nvidia . *CUBLAS Library, User Guide*

[8] Nvidia . *CUDA C programming Guide*

[9] Sithole, G. and Vosselman, G.. *Automatic Structure Detection in a Point-Cloud of an Urban Landscape*

[9] Viola, P. and Jones, M. J.. *Robust Real-Time Face Detection*

[11] Whal, Friederich M. *Triangulation: A Coded Light Approach for Depth Map Acquisition*

[12] Zhu, L. and Hyyppä, J.. *Fully-Automated Power Line Extraction from Airborne Laser Scanning Point Clouds in Forest Areas*

4 Milestones

Taking into account the official dates and deadlines for the presentations, here there is the expected planning we are willing to follow:

1					
2	State of the art study	23 jours	Mer 08/02/17	Dim 12/03/17	
3	CUDA-BLAS-CUBLAS formation	13 jours	Mer 01/02/17	Dim 19/02/17	
4	Proofs on Jetson TK1	1 jour	Lun 20/02/17	Lun 20/02/17	3
5	Open CV First Approach	9 jours	Mer 22/02/17	Lun 06/03/17	
6	Driver for camera Intel	7 jours	Lun 27/02/17	Mar 07/03/17	
7	Proofs with camera Intel	4 jours	Ven 10/03/17	Mer 15/03/17	5;6
8	Final objective definition	7 jours	Mar 07/02/17	Mer 15/02/17	
9	Definition of mid-term report	1 jour?	Jeu 02/03/17	Jeu 02/03/17	
10	Mid-term Report edition	6 jours	Jeu 16/03/17	Jeu 23/03/17	9;8;7;4;5;2
11	Mid-term report re-edition	7 jours	Ven 24/03/17	Lun 03/04/17	10
12	Mid-term Presentation	1 jour?	Mar 18/04/17	Mar 18/04/17	11
13	ROS formation	30 jours	Mer 01/03/17	Mar 11/04/17	
14	Drivers Camera Stereo ZED	4 jours	Lun 20/03/17	Jeu 23/03/17	
15	Proves with Stereo Caera	5 jours	Ven 24/03/17	Jeu 30/03/17	14;7
16	Development of algorithms for object avoidance	17 jours	Mer 19/04/17	Jeu 11/05/17	15
17	Proofs on Jetson TX1	11 jours	Lun 27/03/17	Lun 10/04/17	
18	Proofs on Rumba Robots	22 jours	Mar 25/04/17	Mer 24/05/17	17
19	Verification	7 jours	Lun 05/06/17	Mar 13/06/17	18
20	Final-term Report Definition	1 jour	Jeu 01/06/17	Jeu 01/06/17	
21	Final-term Report Edition	10 jours	Mer 14/06/17	Mar 27/06/17	20;19
22	**Final-Term Presentacion**	**1 jour**	**Jeu 29/06/17**	**Jeu 29/06/17**	**21**
23	Proofs on drones??	19 jours	Jeu 01/06/17	Mar 27/06/17	

Table 2: Table of the main tasks to complete until June 2016

There are some tasks which are doubtlessly necessary to allow us to start with the next ones. However; there are other tasks, like the formations, which do not need to be fully completed before others. That means that thanks to the experiments, we will better understand the different programs and software so the experiments and the formations will be commonly developed at the same time. In the image 7 the Gant Diagram that shows our planning is represented. There is one task in a different colour: the trials with drones. This is because we cannot ensure that the algorithms and the proofs with the ground robots are totally successful to be tested on UAVs.

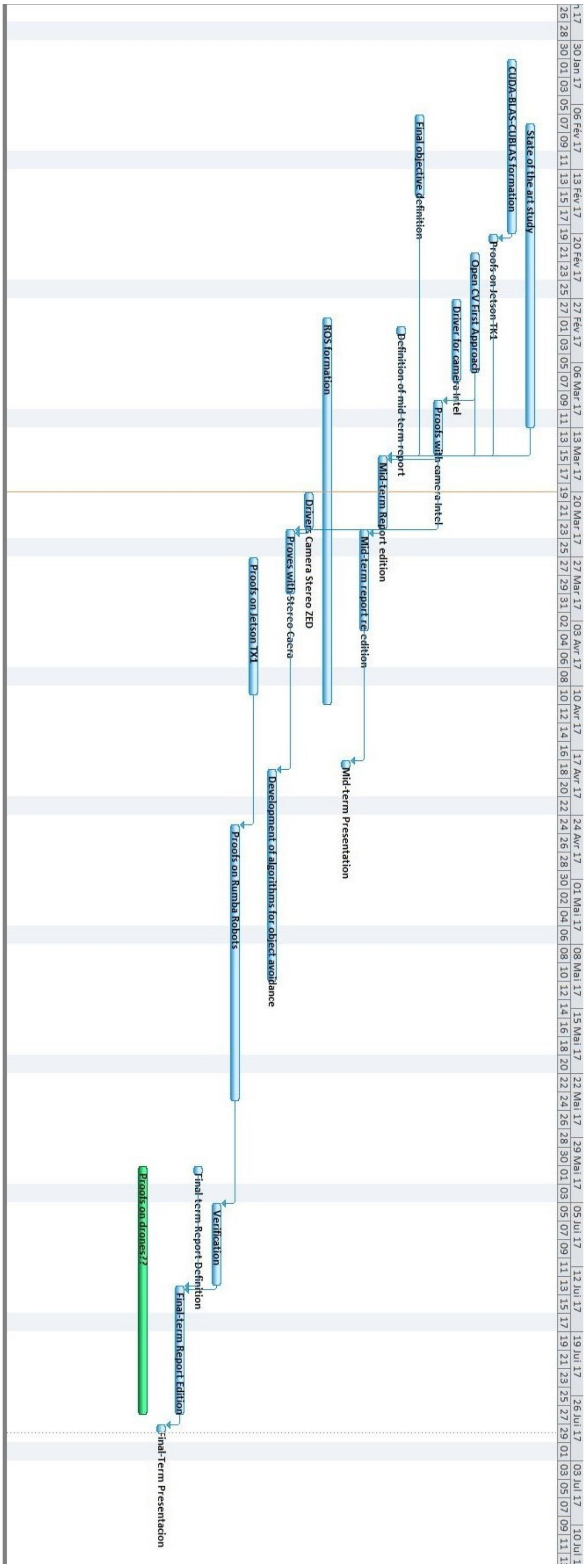


Figure 7: Gant Diagram of second semester 2016-17